



DEGREE PROJECT IN ELECTRICAL ENGINEERING 300 CREDITS,
SECOND CYCLE
STOCKHOLM, SWEDEN 2016

Model Predictive Control for Autonomous Driving of a Truck

MIA ISAKSSON PALMQVIST

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING

Acknowledgement

I would like to thank my supervisor Jonas Mårtensson for giving me an interesting thesis topic and for guidance throughout the project. Furthermore, I would like to thank my co-supervisor Pedro Lima for giving me valuable comments on my work and for discussing the project with me.

Abstract

Platooning and cooperative driving can decrease the emissions of greenhouse gases and increase the traffic capacity of the roads. The Grand Cooperative Driving Challenge, GCDC, is a competition that will be held in May 2016 focusing on cooperative driving.

A cornerstone in the cooperative driving is autonomous driving. The main objective of this thesis is to design a Model Predictive Control for a truck so it autonomously can perform the following tasks: follow a straight road, make a lane change and make a turn. Constraints are added to the vehicle states and the control signals. Additionally, a constraint is added to make the vehicle keep a safe distance to preceding vehicles.

A Linear Time-Varying (LTV) MPC for reference tracking is derived. To use the MPC for reference tracking references for all the states and control signals are derived. For the lane change and turn scenario Bezier curves are used to obtain the position references.

The controller is implemented in MATLAB and validated through simulations. For the simulations both a bicycle model of the vehicle and a more complicated four wheel model are used. The latter is implemented in Simulink. The bicycle model is on-line linearised around the references in order to be used as the prediction model for the LTV-MPC.

The simulations show that the controller can make the vehicle perform the above mentioned tasks. With a horizon of 18 the time in average to perform one iteration of the control loop is 0.02 s. The maximum deviation in the lateral direction is 0.10 m and occurs for the turn scenario when the four wheel model is used for the simulations. Simulations are also done with a preceding vehicle. The controller is able to make the vehicle keep a safe distance to the preceding vehicle. If the preceding vehicle is driving slower than the controlled vehicle the controller is able to decrease the velocity of the controlled vehicle.

In addition to the above mentioned, simulations are also done where disturbances and noise, separately, are added. As a disturbance an error in the start position is used. The vehicle can start 1.3 m from the real start position, in the lateral direction, and still find its way back to the trajectory. The noise is added as white noise to the position updates. The controller can deal with noise with a standard deviation up to 0.3 m.

Sammanfattning

Kolonmkörning och kooperativkörning kan minska utsläppen av växthusgaser och öka trafikkapaciteten på vägarna. The Grand Cooperative Driving Challenge, GCDC, är en tävling med fokus på kooperativkörning som kommer att hållas i maj 2016.

En hörnsten i kooperativkörning är autonomkörning. Det huvudsakliga målet med det här examensarbetet är att designa en MPC för en lastbil så att den autonomt kan genomföra följande: följa en rak väg, göra ett filbyte samt göra en sväng. Begränsningar är lagda på fordonets tillstånd och kontrollsignalerna. Utöver det begränsas avståndet till framförvarande fordon.

En linjär tidsvariant (LTV) MPC för referensföljning tas fram. För att använda MPC:n för referensföljning härleds referenser för fordonets tillstånd och kontrollsignaler. För filbytet och svängen används Bezier kurvor för att få fram positionsreferenserna.

Regulatorn implementeras i MATLAB och valideras genom simuleringar. För simuleringarna används både en cykelmodell av fordonet och en mer komplicerad fyrhjuls-modell. Den senare implementeras i Simulink. Cykelmodellen linjäriseras online kring referenserna i syfte att användas som prediktionsmodell för LTV-MPC.

Simuleringarna visar att regulatorn kan få fordonet att genomföra de ovan nämnda uppgifterna. Med en horisont på 18 är tiden det tar att genomföra en iteration av regulator-loopen i genomsnitt 0.02 sekunder. Den maximala avvikelsen i den laterala riktningen är 0.10 meter och uppstår när simuleringar görs för en sväng med fyrhjuls-modellen som fordonsmodell. Simuleringar görs även med ett framförvarande fordon. Regulatorn kan få fordonet att hålla ett säkerhetsavstånd till fordonet framför. Regulatorn kan vidare få fordonet att sänka hastigheten om det framförvarande fordonet kör långsammare.

Utöver de ovan nämnda simuleringarna görs simuleringar där störningar och brus, var för sig, introduceras. Som störning används ett fel i startpositionen. Fordonet kan starta 1.3 meter från den korrekta startpositionen, i den laterala riktningen, och hitta tillbaks till referensbanan. Bruset adderas som vitt brus på positionsuppdateringarna. Regulatorn kan hantera vitt brus med en standardavvikelse på upp till 0.3 meter.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Grand Cooperative Driving Challenge | 1 |
| 1.2 | Objective | 3 |
| 1.2.1 | Scope | 4 |
| 1.3 | State of Art | 4 |
| 1.4 | Method | 5 |
| 1.5 | Outline | 6 |
| 2 | Modelling | 7 |
| 2.1 | Literature Review | 7 |
| 2.2 | Model to use in Model Predictive Control | 8 |
| 2.2.1 | External Forces | 10 |
| 2.3 | Simulation Model | 10 |
| 2.3.1 | Vehicle Powertrain | 11 |
| 2.3.2 | Forces Acting on the Vehicle | 14 |
| 2.3.3 | Slip Ratio | 16 |
| 2.4 | Summary of the two Models | 18 |
| 3 | Control | 21 |
| 3.1 | Literature Review | 21 |
| 3.2 | Model Predictive Control | 22 |
| 3.3 | Linear Model Predictive Control | 23 |
| 3.4 | Model Predictive Control for Reference Tracking | 23 |
| 3.5 | Formulate as Quadratic Programming | 24 |
| 3.5.1 | Constraints | 25 |
| 4 | Reference Generation | 29 |
| 4.1 | Generate the Waypoints | 29 |
| 4.2 | Acceleration and Velocity References | 31 |
| 4.3 | References for Yaw Angle, Yaw Rate and Front Wheel Steering Angle | 32 |
| 5 | Implementation | 35 |
| 5.1 | The Simulation Model | 35 |
| 5.2 | The Control Algorithm | 36 |
| 6 | Result and Discussion | 41 |
| 6.1 | Follow a Trajectory | 41 |
| 6.2 | Handling of Disturbances and Noise | 45 |
| 6.3 | Other Vehicles Involved | 47 |
| 6.4 | Follow a Real Road | 49 |
| 7 | Conclusion | 51 |
| 8 | Future Work | 53 |

Chapter 1

Introduction

The combination of an increased demand for transportation and the need for a decrease in the emissions of greenhouse gases leads to a contradiction. One way to tackle the problem and decrease the traffic emissions while increasing the roads' traffic capacity is vehicle platooning. Platooning implies a convoy of vehicles autonomously following a lead vehicle with small inter-vehicle distances. By doing so the air resistance is decreased, decreasing the fuel consumption. Alam [2014] showed that platooning can decrease the fuel consumption with about 5-8 percent for heavy duty vehicles. Apart from improving the environment the reduced fuel consumption will be economically beneficial, for the drivers in general and for the transportation companies in particular.

Furthermore, vehicle platooning can increase the traffic capacity since more vehicles can be at the roads at the same time. With an increasing population this can be one tool to reduce traffic congestion.

The advantages of cooperative driving have made it an active research area since the 1980s. One of the early projects was PROHETHEUS (Programme for a European Traffic with Highest Efficiency and Unprecedented Safety), active between 1987-1995. It focused at on-board driver assistance, inter-vehicle communication system and vehicle-roadside communication system [Williams, 1988]. Another project focusing at cooperation and communication between vehicles and the road is PATH (The California Partners for Advanced Transit and Highways). PATH dates back to 1986 and is still active, focusing on intelligent transportation systems [PATH, 2015].

One of the first projects to focus on truck platooning was the European project Chaffour. The project managed to implement platooning with two trucks without any requirements on the road infrastructure [Fritz, 1999].

Even if research about platooning has been ongoing since the 1980s it was not until 2009 that the first test of heavy vehicle platooning in real traffic was conducted. It was done as a part of the KONVOI project [IKA, 2015]. This was followed by the first test of mixed platoons, both trucks and passenger cars, in 2012 conducted by the SARTE (Safe Road Trains for the Environment) project [SARTE, 2012].

Scania has in collaboration with KTH and VTI also performed tests with heavy vehicle platooning in real traffic. The tests were conducted at the free-way between Södertälje and Helsingborg with platoons consisting of 2 – 5 trucks. Between June 2011 and December 2013, 43 different trucks drove in total 70000 hours in platoon [Johansson, 2014].

To stimulate the research in the area of cooperative and autonomous driving competitions have been held. The two major competitions are DARPA (Defence Advanced Research Projects Agency) Urban Challenge and Grand Cooperative Driving Challenge. The former, have been held three times, 2004, 2005 and 2007 [DARPA, 2016]. The Grand Cooperative Driving Challenge has been held once, 2011, but a new competition will be held in May 2016.

1.1 Grand Cooperative Driving Challenge

The Grand Cooperative Driving Challenge, GCDC, is a competition coordinated by the Netherlands Organization for Applied Scientific Research, TNO, in collaboration with Technische Universiteit Eindhoven, Viktoria Swedish ICT and Institute for Applied Automotive Research (IDIADA). It aims to accelerate the implementation of autonomous and cooperative driving, so it can be used in real

traffic situations. The first competition was held in the Netherlands, 2011, focusing on platooning and longitudinal control. In May 2016 the second competition will be held, this time additionally focusing on lateral control [i GAME GCDC, 2015].

In the competition two different scenarios will be included:

1. a highway scenario and

2. an intersection scenario.

The highway scenario is demonstrated in Figure 1.1. As can be seen the scenario starts with two platoons driving side by side. At the same time the platoons will receive a message about a construction site ahead of them, blocking one of the lanes. The aim is then for the platoon in the blocked lane to merge into the platoon in the free lane. To facilitate the merging the vehicles in the free lane should increase the inter-vehicle distance if needed. However, the length of the platoon in the free lane is used as an evaluation parameter, hence the inter-vehicle distance should increase as little as possible.

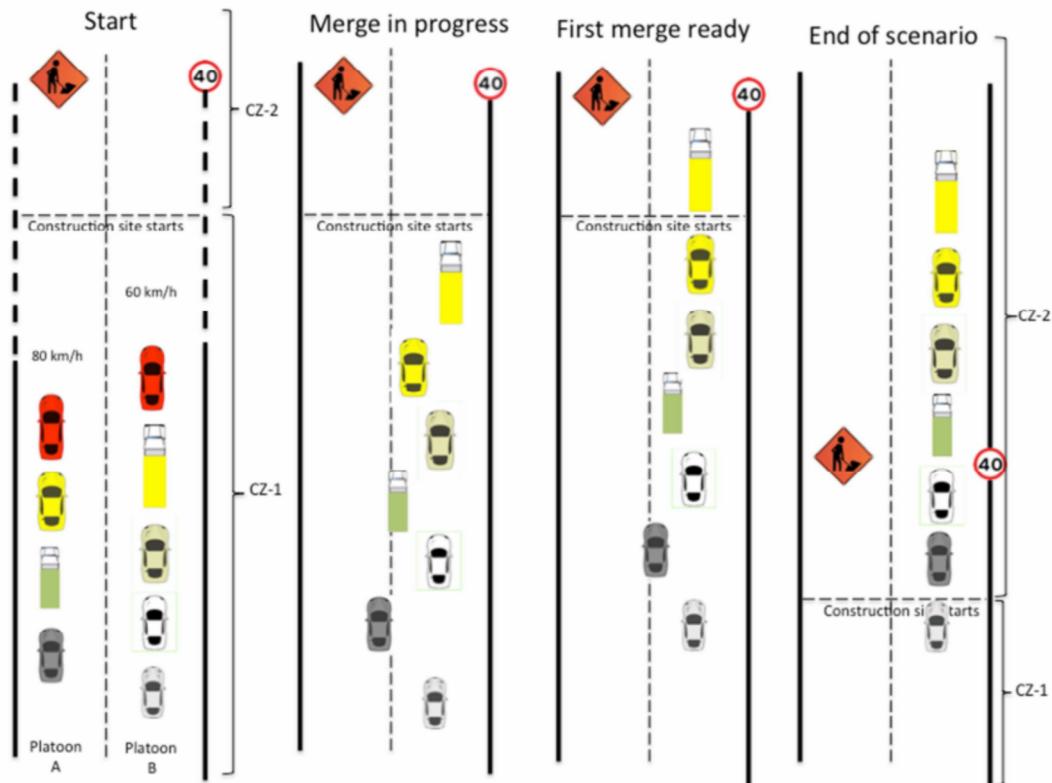


Figure 1.1: A step-by-step description of the highway scenario in the GCDC [Didoff, 2014]

The intersection scenario will not consider platoons but individual vehicles, the scenario is demonstrated in Figure 1.2. The scenario includes three vehicles, the yellow vehicles in the figure. These will reach the intersection area, defined as CZ in the figure, at the same time and with the same speed. The vehicles then have to negotiate about a suitable velocity in order to let the turning vehicle, V1 in figure, has a free passage through the intersection. It is desirable that none of the vehicles come to a full stop.

In the two scenarios above the participants will be judged both on individual and group performance. The individual performance includes if the vehicle keeps its velocity below the speed limit and that it, in every scenario, keeps a safe distance, specified beforehand, to the preceding vehicle. The group performance is about how the vehicles cooperate to achieve the aim of each scenario. Two

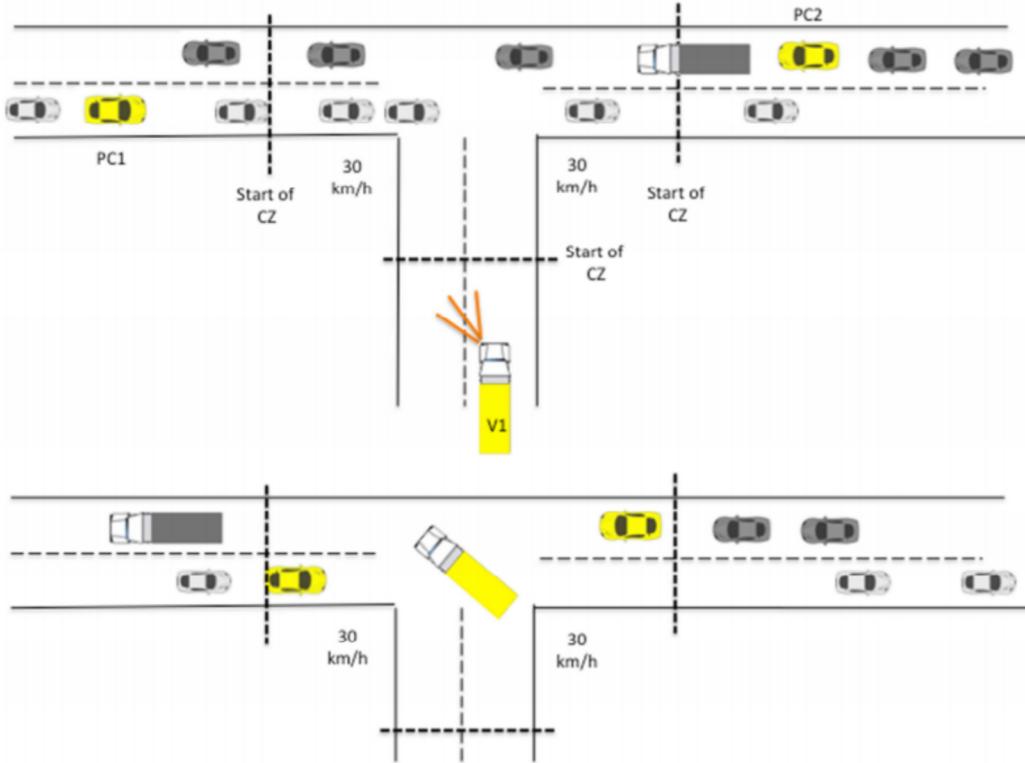


Figure 1.2: A step-by-step description of the intersection scenario in the GCDC [Didoff, 2014]

group evaluation parameters for the two scenarios are already mentioned, the length of the platoon in the free lane for scenario 1 and that none of the vehicles should come to a standstill for scenario 2.

In addition to the previous scenarios an emergency vehicle scenario will be demonstrated but not as a competitive element. For this scenario all the vehicles will be manually driven. The emergency vehicle will approach a dense traffic situation from behind and communicate to the vehicles that it is coming and where it would like to pass. The last vehicles in the platoon will receive the message first and act accordingly to the request from the emergency vehicle. A vehicle cannot react before it receives the message [Didoff, 2014].

1.2 Objective

The objective of this thesis is to design the control for a truck participating in the GCDC. The controller should include both longitudinal and lateral control and enable a vehicle to autonomously perform the following tasks,

- follow a straight road,
- make a lane change and
- do a left turn.

While performing the mentioned tasks the vehicle has to fulfil a number of specifications. First, it has to keep its velocity below the speed limit and keep the distance to preceding vehicles. This is important from a safety perspective. Furthermore, there are comfort aspects to address. The acceleration or deceleration both in the longitudinal and lateral direction should not be too high since that decreases the comfort for the driver. Finally, the vehicle should not deviate too much from the position references.

1.2.1 Scope

In Figure 1.3 the whole control system for the GCDC is presented and the focus of this thesis is highlighted with a red box. Hence, as can be conducted from the figure, this thesis will exclusively focus on the control agents. We will assume that we are given controller settings from a supervisory layer and that the acceleration and front wheel angle references, that are the output from our control agents, are transformed to actuator signals by the low-level controllers.

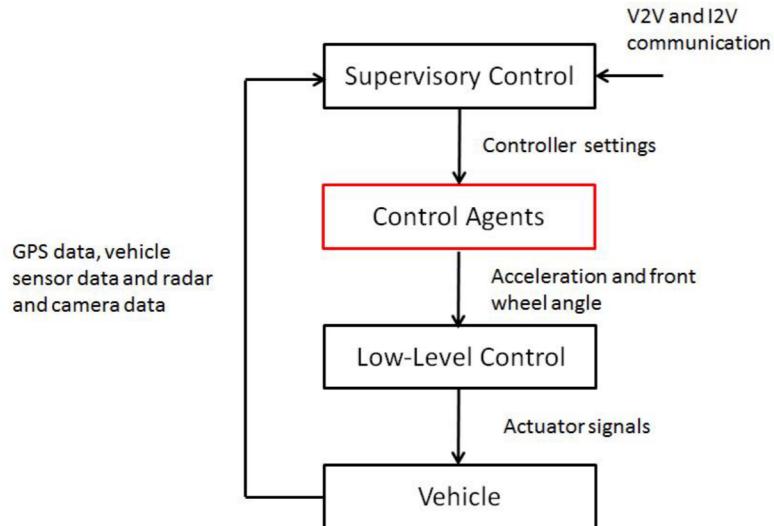


Figure 1.3: The control system for GCDC. The red box is the focus of this thesis

The supervisory layer of the control system will be designed simultaneously with this thesis by another project. It will be assumed that this supervisory layer has access to processed data and is able to receive messages from the other vehicles, V2V communication, and from the infrastructure, I2V communication. We will assume the supervisory layer gives the control agents information of when to perform the different tasks. Furthermore, it is assumed the supervisory layer gives commands to perform different tasks when it is possible to perform them. For example, ask the vehicle to change lane so the safety distances to the other vehicles are not violated during the change.

1.3 State of Art

There is a lot of ongoing research in the area of autonomous driving and numerous active safety systems, e.g. adaptive cruise control, are already implemented in vehicles [Turri et al., 2013]. The big challenge now is to design fully autonomous vehicles able to drive in urban traffic.

For autonomous driving of a vehicle both the longitudinal and lateral dynamics need to be controlled. In most vehicle models these dynamics are coupled, i.e. a change in one state might influence other states as well. One way to handle this, commonly used, is decoupled control. The decoupled control tries to isolate the system so that one input just affects one output. By doing so separate controllers can be used for the different states, e.g. one for the longitudinal control and one for the lateral. This technique was, for example, implemented in Ying et al. [2014] where the lateral control was done with a PID controller and a sliding mode controller was used for the longitudinal control. According to Ying et al. [2014] the sliding mode control is robust and the control laws have low complexity.

In Kianfar et al. [2014] another way of implementing decoupled control was demonstrated. For the implementation three different controllers were used, one in the frequency domain to deal with

the string stability and two Model Predictive Controllers, MPC, to deal with the longitudinal and lateral control respectively. String stability is often defined in the frequency domain why a frequency domain controller is suitable. However, a frequency domain controller is not suitable for control of the vehicle dynamics since the constraints then have to be translated from time to frequency. The MPC, on the other hand, works in the time domain.

In Turri et al. [2013] one MPC was used to control both the lateral and longitudinal dynamics of a vehicle, i.e. coupled control. According to Turri et al. [2013] an advantage of the MPC is the ability to deal with constraints, both on the states and the control signals. Furthermore, it is able to work with non-linear and time varying systems. Since the dynamics of a vehicle is highly non-linear and several constraints are present, MPC is a suitable control for autonomous driving.

As the name indicates, MPC uses a model to predict the future dynamics of the system. These predictions are used in a cost function that together with some constraints formulates an open-loop optimization problem. The optimization problem is solved for each sampling time and gives an optimal control sequence. Only the first elements of the control sequence are applied to the system before the optimization is done again. This way the control law is updated which makes it suitable for problems where it is difficult or even impossible to calculate the control law beforehand [Mayne et al., 2000].

1.4 Method

To fulfil the objective we decided to use Model Predictive Control, MPC, due to its ability to deal with constraints and non-linear systems.

To design the MPC a model of the vehicle to use as the prediction model will be derived. Additionally a more complex model will be designed to use for validation of the controller, through simulations. In the simulations, other vehicles will be included to make it as similar as possible to the real situation.

In Figure 1.4 a schematic picture of the different parts included in the thesis are presented. Where z represents the state and u the control signal. The control signal consists of the acceleration, a , and the steering angle of the front wheels, δ . The information about other vehicles could be obtained both by measurements, e.g. radar, and V2V communication.

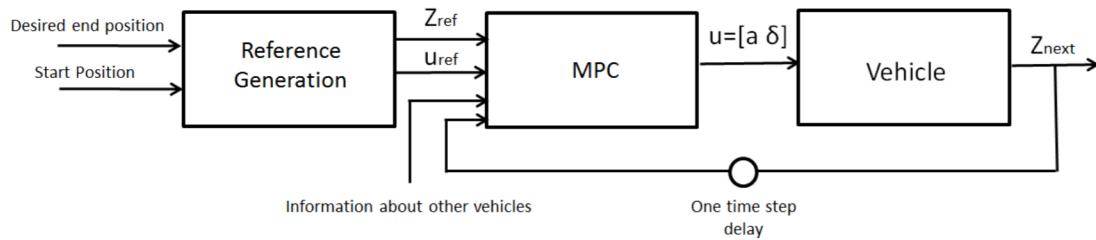


Figure 1.4: The different parts included in the thesis

Below follow a list of the sub goals included in the thesis to fulfil the main objective of autonomous driving of a truck,

- derive two models, one for the MPC and one more complicated to use for the simulations,
- find a way to generate suitable trajectories for the truck,
- design an MPC and
- evaluate the controller through simulations.

1.5 Outline

The thesis is divided into two parts: Theoretical Background (ch. 2, 3, 4) and Implementation (ch. 5, 6, 7, 8).

In Chapter 2 the vehicle is modelled both for control and simulation purposes. In Chapter 3 the control of the vehicle is addressed. The background to MPC is given and the MPC approach chose to use for this thesis is presented. In Chapter 4 the reference generation is presented. Chapter 5 describes the implementation and give numerical values both to the model and the control parameters. In Chapter 6 the results are presented and discussed. In Chapter 7 the conclusions are given. Finally, in Chapter 8, thoughts about how the implementation could be further developed are given.

Chapter 2

Modelling

A cornerstone in Model Predictive Control, MPC, is the model of the system. It is used to predict the future states of the systems, hence it needs to capture the most important dynamics of the vehicle. However, the computational burden cannot be too high. In Section 2.1 we present vehicle models with different computational burden and accuracy.

For this thesis we used a dynamic bicycle model of the vehicle, derived in Section 2.2, as the prediction model.

Additionally, since the MPC will be evaluated through simulations, a simulation model is needed. The simulation model needs to well describe the behaviour of a real truck. A four-wheel model have been derived and implemented in Simulink. The derivation is described in Section 2.3.

In the last section of the chapter, Section 2.4, we summarize the most important parts of the two models.

2.1 Literature Review

To find a suitable prediction model of the system is a crucial part for the MPC. The aim is to find a model describing the behaviour of the vehicle well enough without being too computational expensive. If the model is too computational expensive the time it takes to solve the optimization problem will be too long, since the prediction model is included in the optimization problem. The time to perform the optimization should be less than the sample time of the system.

In the industry it is common to use impulse or step response models. These models require a lot of parameters to be identified but less a priori information is needed than for state-space models. The state space models, on the other hand, are suitable for multi-variable processes [Camacho and Bordons, 2007]. Due to this we decided to use a state space representation of the vehicle, and below we present some different models.

One often used model is the so called bicycle model. It includes the lateral, longitudinal and yaw motion of the vehicle but simplifies the model by just consider two wheels, the four wheels are modelled as one front and one rear wheel at the center axis. Often a dynamic bicycle model is used, as demonstrated in Keviczky et al. [2006] and Falcone et al. [2008]. Since the dynamic bicycle model describes the system by the forces acting on it, the tire forces have to be considered. These forces are highly non-linear and the existing models are often semi-empirical. In Keviczky et al. [2006] and Falcone et al. [2008] the Pacejka Model was used to model the tire forces. The use of non-linear models for the tire forces increases the computational burden.

The modelling of the tire forces can be avoided by the use of a kinematic bicycle model, since the kinematic model uses the motion of the vehicle, instead of the forces acting on it, to describe its behaviour. In Kong et al. [2015] a comparison was done between the dynamic and kinematic bicycle model, using a linear approximation of the tire forces for the dynamic model. It showed that the two models performed, more or less, equally well, when the kinematic model was discretized at 200 ms and the dynamic at 100 ms. In addition, the kinematic model was implemented with MPC. It was less computational expensive and the vehicle performed well in low velocities and stop-and-go scenarios. Nevertheless, it was concluded that the dynamic model was more appropriate at higher speeds.

An extension of the dynamic bicycle model, used in for example Turri et al. [2013], is the extended bicycle model. It uses four wheels and is thereby more similar to a real vehicle. The use of four wheels makes it possible to account for the load transfers, both due to longitudinal and lateral acceleration. Nevertheless, due to the computational burden the load transfer is calculated assuming no aerodynamic forces acting at the vehicle. Furthermore the rotational inertia of the wheels is neglected and it is assumed that the forces at the four wheels are evenly distributed.

2.2 Model to use in Model Predictive Control

In order to fulfil the requirement of accuracy and at the same time not be too computational expensive a dynamic bicycle model has been used to model the vehicle for the MPC. By using the position in the global coordinate system, X, Y , the yaw angle, ψ , the vehicle velocity in vehicle's coordinate system, v_x, v_y , and the yaw rate, $\dot{\psi}$ as the states, see Figure 2.1, the most important parts of the lateral and longitudinal dynamics were captured.

The derivation of the vehicle model is inspired by F. Borrelli [2005] and Rajamani [2012].

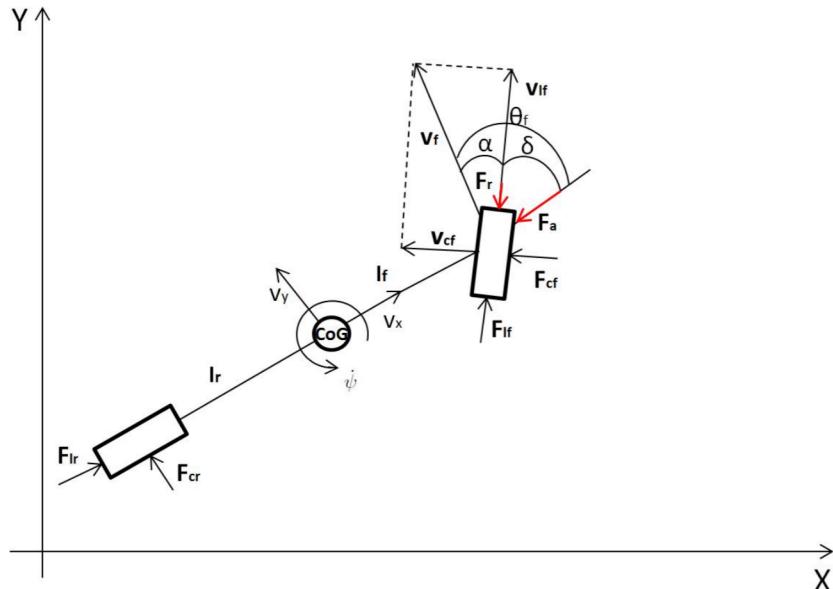


Figure 2.1: A vehicle represented by the bicycle model

The simplification made in the Bicycle model is to model the two front wheels and the two rear wheels as one front and one rear wheel, see Figure 2.1. v_{lf}, v_{cf} are the longitudinal and lateral velocity vectors for the front wheel and v_f is the resultant velocity vector for the front wheel. Furthermore, $\dot{\psi}$ represents the yaw rate and δ the front wheel angle. The forces F_l and F_c are the longitudinal and lateral (cornering) wheel forces. The subscripts r and f stand for rear and front, respectively.

According to Newtons second law, $F = ma$, the following equations hold,

$$m\dot{v}_x = mv_y\dot{\psi} + 2F_{xf} + 2F_{xr} \quad (2.1a)$$

$$m\dot{v}_y = -mv_x\dot{\psi} + 2F_{yf} + 2F_{yr} \quad (2.1b)$$

$$I\ddot{\psi} = 2l_f F_{yf} - 2l_r F_{yr} \quad (2.1c)$$

where, m is the vehicle mass, I the vehicle inertia around the z-axis and F_x and F_y the longitudinal and lateral forces acting at the CoG of the vehicle. These forces arise from the wheels and are multiplied by two to compensate for the use of one wheel instead of two. Furthermore, the yaw rate is given by,

$$\dot{\psi} = \frac{v_x}{l_f + l_r} \tan(\delta) \quad (2.2)$$

and the change of position in the fixed frame is calculated as,

$$\dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \quad (2.3a)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (2.3b)$$

where l_f and l_r can be seen in Figure 2.1 and are the distances from the front wheel and rear wheel, respectively, to the center of gravity.

The forces acting on the CoG, F_x and F_y , can be expressed in terms of the longitudinal and lateral tire forces, F_l and F_c ,

$$F_x = F_l \cos(\delta) - F_c \sin(\delta) \quad (2.4a)$$

$$F_y = F_l \sin(\delta) + F_c \cos(\delta) \quad (2.4b)$$

where F_l and F_c can be expressed as,

$$F_f = f(\alpha, \mu, s, F_z) \quad (2.5a)$$

$$F_c = f(\alpha, \mu, s, F_z) \quad (2.5b)$$

where α is the angle between the direction of the wheel velocity and the direction of the wheel itself, see Figure 2.1, called the slip angle. μ is the friction coefficient for the road, s is the difference between the wheel ground point velocity and the equivalent rotational velocity called the slip ratio and F_z is the vertical load acting on the wheels of the vehicle. The tire forces are highly non-linear but by assuming small slip angles the forces can be approximated with linear equations.

The lateral tire forces are, for small slip angles, proportional to the slip angle [Rajamani, 2012] according to,

$$F_{lf} = C_f \alpha_f \quad (2.6a)$$

$$F_{lr} = C_r \alpha_r \quad (2.6b)$$

where the constant $C_{f/r}$ is a tire stiffness parameter and the slip angles are expressed as,

$$\alpha_f = \delta - \theta_f \quad (2.7a)$$

$$\alpha_r = -\theta_r \quad (2.7b)$$

where θ is the angle the resultant velocity vector of the front and rear wheel respectively makes with the longitudinal axis of the vehicle. In Figure 2.1 θ_f is demonstrated. θ can be calculated according to,

$$\theta_f = \arctan \left[\frac{v_y + l_f \dot{\psi}}{v_x} \right] \quad (2.8a)$$

$$\theta_r = \arctan \left[\frac{v_y - l_r \dot{\psi}}{v_x} \right] \quad (2.8b)$$

The longitudinal forces acting on the wheels are, if still assuming small slip angles, proportional to the slip ratio. The slip ratio describes the difference between the wheel ground point velocity and the equivalent rotational velocity. By assuming these two velocities are equal the longitudinal slip equals zero, hence, the longitudinal tire forces are assumed to be zero.

2.2.1 External Forces

A vehicle is, apart from the wheel forces, affected by external forces. It will be assumed that the vehicle travels on flat roads, hence no gravitational force will affect the truck. However, it will be affected both by air drag, F_a and rolling resistance, F_r , see Figure 2.1. The rolling resistance can, according to Alam [2014] be expressed as,

$$F_r = D_r m g \cos(\alpha) \quad (2.9)$$

where D_r is the roll resistance coefficient, m is the vehicle mass and g the gravitational acceleration ($= 9.81 \text{ m/s}^2$). Since the roads are assumed to be flat the road grade, α , is zero. Hence the rolling resistance, for flat roads, is given by,

$$F_r = D_r m g \quad (2.10)$$

The air drag can, according to Rajamani [2012], be expressed as;

$$F_a = \frac{1}{2} C_D A_a \rho_a (v + v_{wind})^2 \quad (2.11)$$

where C_D is the air drag coefficient, A_a is the max cross-sectional area of the vehicle and ρ_a is the air density. We only consider the wind acting in the longitudinal direction and opposite to the heading of the vehicle. Wind acting on the vehicle in other directions are not modelled and considered as disturbances. The wind velocity v_{wind} is often unknown and small in comparison to the vehicle velocity and therefore put to zero, given the following equation for the air drag,

$$F_a = \frac{1}{2} C_D A_a \rho_a (v)^2 \quad (2.12)$$

To conclude, the states of the vehicle can be expressed as;

$$\dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \quad (2.13a)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (2.13b)$$

$$\dot{\psi} = \frac{v_x}{l_f + l_r} \tan(\delta) \quad (2.13c)$$

$$m \dot{v}_x = F_x + m v_y \dot{\psi} - 2 F_{cf} \sin(\delta) - F_a - F_r \quad (2.13d)$$

$$m \dot{v}_y = -m v_x \dot{\psi} + 2 (F_{cf} \cos(\delta) + F_{cr}) \quad (2.13e)$$

$$I \ddot{\psi} = 2 (l_f F_{cf} \cos(\delta) - l_r F_{cr}) \quad (2.13f)$$

where, the wheel angle, δ , is one of the control inputs to the system and F_x is the driving force of the vehicle, dependent on the other control input to the system, namely, the acceleration, a .

2.3 Simulation Model

The purpose of the simulation model is to mimic the real behaviour of a vehicle as good as possible in order to use it for validation of the controller. We used the same states as for the Bicycle model but included the vehicle power-train and the interaction between the road and the vehicle to the model. Furthermore, we modelled all four wheels instead of two. In the Bicycle model we used a driving force, F_x , acting directly on the vehicle as an input. This is not how it is done in a real vehicle and for the simulation model a cruise controller was modelled. The simulation model was implemented in Simulink.

The derivation of the vehicle model for the simulations is inspired by Attia et al. [2014] and Kiencke and Nielsen [2005]. The parameters used for the engine are partly received from a data sheet, Nordström [2013], given as press info from Scania and partly through communication with Scania.

2.3.1 Vehicle Powertrain

The powertrain is giving the vehicle its driving force. In Figure 2.2 the most important components of the powertrain are presented. The angular speed, ω , and torque, T , from the engine is transformed through the powertrain. The input to the engine is a throttle value (throttle position), t_p , coming from the cruise controller, see Figure 2.3.

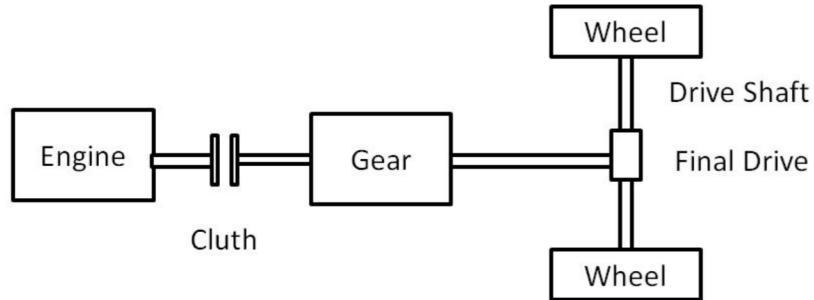


Figure 2.2: The main components of the vehicle powertrain

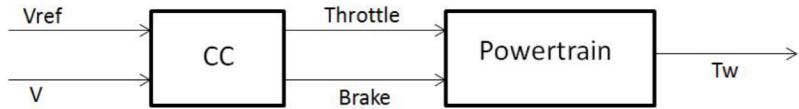


Figure 2.3: The inputs and outputs from the cruise controller and the powertrain

The cruise controller, CC, is modelled by a PI-controller acting at the difference between the reference velocity and the actual velocity coming from the chassis model of the vehicle. The output, the throttle and brake positions, t_p and b_p , takes values between 0 and 1 where 1 means full throttle or brake and 0 means none. If one of the parameters has a non-zero value the other is zero. The throttle value is used as an input to the engine whereas the brake value directly affects the drive shaft. In Figure 2.4 the step response for a step in the velocity reference from 5 m/s to 10 m/s is shown. As can be seen in the figure it takes about 20 s to reach the new velocity, this give an average acceleration of 0.25 m/s^2 . Furthermore, since the acceleration is dependent on the difference between the reference velocity and the actual velocity of the vehicle the velocity increases faster in the beginning when the difference is bigger. This means that a bigger step in the reference velocity gives rise to a bigger acceleration, which is how we like a cruise controller to behave. If the velocity should be increased by a small amount it could be done by adding a small acceleration but if the velocity should increase with a big amount the acceleration need to be bigger, otherwise it will take too long time to reach the new velocity. However, the acceleration should stay below a maximum value to ensure comfort for the driver. In order for a vehicle to not exceed the speed limits it is important that the CC does not have an overshoot and to ensure driver comfort the step response should not oscillate. Both of these requirements are fulfilled for the CC in Figure 2.4.

The throttle value, coming to the engine, is used to decide the torque produced by the engine. The net torque produced, $T_e(\omega_e, \gamma)$, is a function of the angular velocity of the engine, ω_e , and the

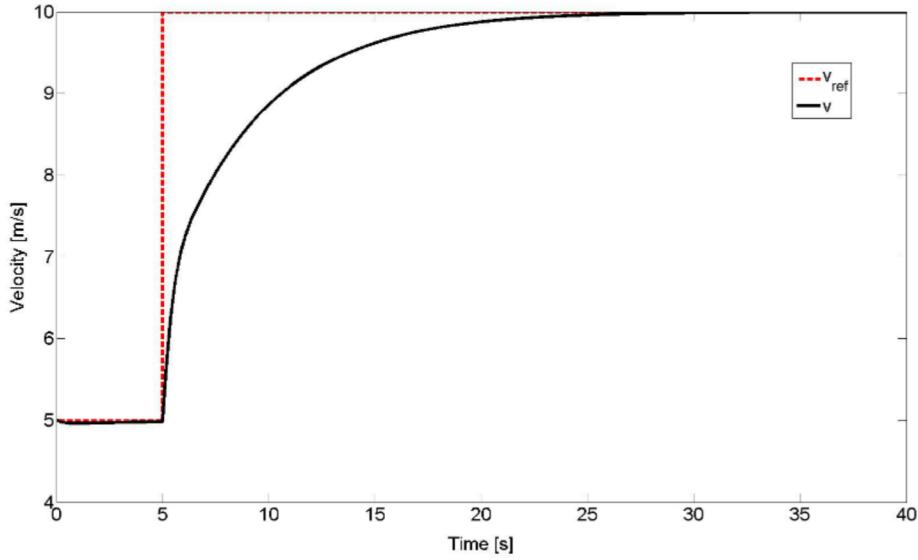


Figure 2.4: The step response for a step in the reference velocity from 5 m/s to 10 m/s

amount of injected fuel, γ . To model this we used a lookup table to find the maximum available torque for different engine speeds. A graph demonstrating the values can be found in Figure 2.5. The output torque from the engine is then given by,

$$T_e = T_{emax}(\omega_e)t_p - T_f \quad (2.14)$$

where t_p is the throttle value and T_f the friction in the engine, modelled as,

$$T_f = 50 + 0.021\omega_e \quad (2.15)$$

If the clutch is assumed to be stiff the engine torque and angular speed will go unchanged through the clutch, i.e. the angular speed and the torque into the gear, ω_G and T_G will be the same as the angular speed and torque coming from the engine,

$$T_G = T_e \quad (2.16a)$$

$$\omega_G = \omega_e \quad (2.16b)$$

Next the torque and angular speed is transformed through the gear to the final drive as,

$$T_f = \gamma_G \eta_G T_G \quad (2.17a)$$

$$\omega_f = \frac{1}{\gamma_G} \omega_G \quad (2.17b)$$

where η_G is the efficiency of the gear and γ_G is the gear ratio. For this thesis we assumed the gear efficiency to be one and the gear changes to happen instantly. In a real vehicle the gear shifting takes some time, e.g. 1 s, and during this time no torque is transferred from the engine to the wheels. By assuming instantly gear shift this drop in torque will not be considered. However, we decided to assume this due to the time scope of the thesis and due to the few known parameters for the powertrain.

The gear ratio is different for different gears as seen in table 2.1 where the gear ratios for a truck is shown. The gear is up shifted if the engine angular speed is above a specific value and down shifted if it is less than a specific value, in order to make the engine work at a desired engine velocity. For this thesis the upper angular speed was 1400 rpm and the lower 1000 rpm. In Figure 2.6 the changes

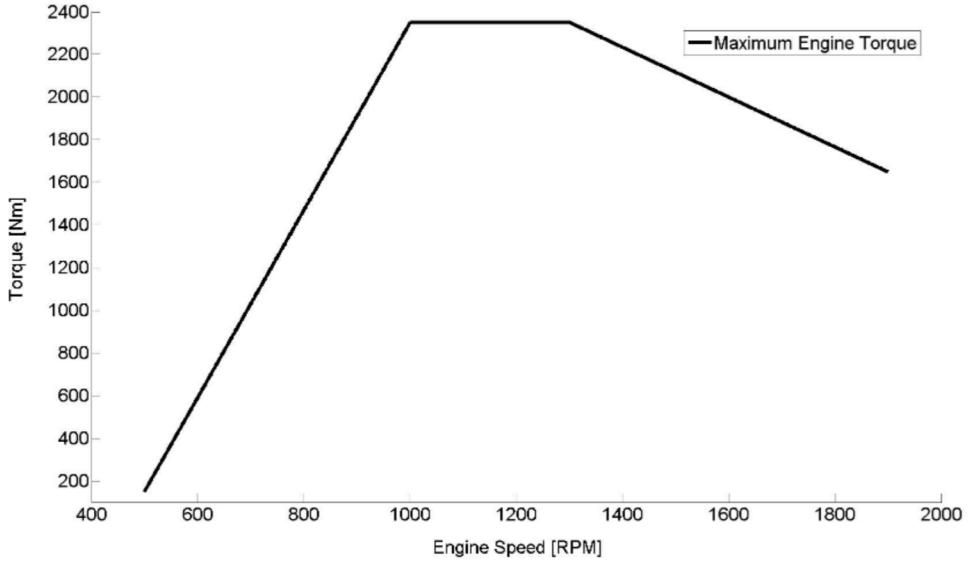


Figure 2.5: The engine torque as a function of the angular velocity of the engine

| Gear | Gear ratio |
|------|------------|
| 1 | 11.38 |
| 2 | 9.23 |
| 3 | 7.17 |
| 4 | 5.81 |
| 5 | 4.62 |
| 6 | 3.75 |
| 7 | 3.03 |
| 8 | 2.46 |
| 9 | 1.91 |
| 10 | 1.55 |
| 11 | 1.23 |
| 12 | 1 |

Table 2.1: The gear ratios of the different gears

in the gear and engine speed for a step in the reference velocity is demonstrated. As can be seen the gear is up-shifted when the angular speed of the engine reaches 1400 rpm. It can also be seen in the figure that the engine speed increases faster for lower gears, this was expected since the gear ratio is higher for lower gears. Finally, we can see that the angular speed of the engine is constant when the vehicle velocity is constant.

After the gear the torque and angular velocity is transformed through the final drive to the wheels. The final drive has a fixed conversion ratio, γ_f and an efficiency parameter, η_f . In this thesis we used $\gamma_f = 2.59$ and, as with the gears, we assumed the efficiency to one. The angular speed, ω_w , and the torque, T_w , of the wheels is then given by,

$$T_w = \gamma_f \eta_f T_f - T_B \quad (2.18a)$$

$$\omega_w = \frac{1}{\gamma_f} \omega_f \quad (2.18b)$$

where T_B is the applied braking force,

$$T_B = T_{Bmax} b_p \quad (2.19)$$

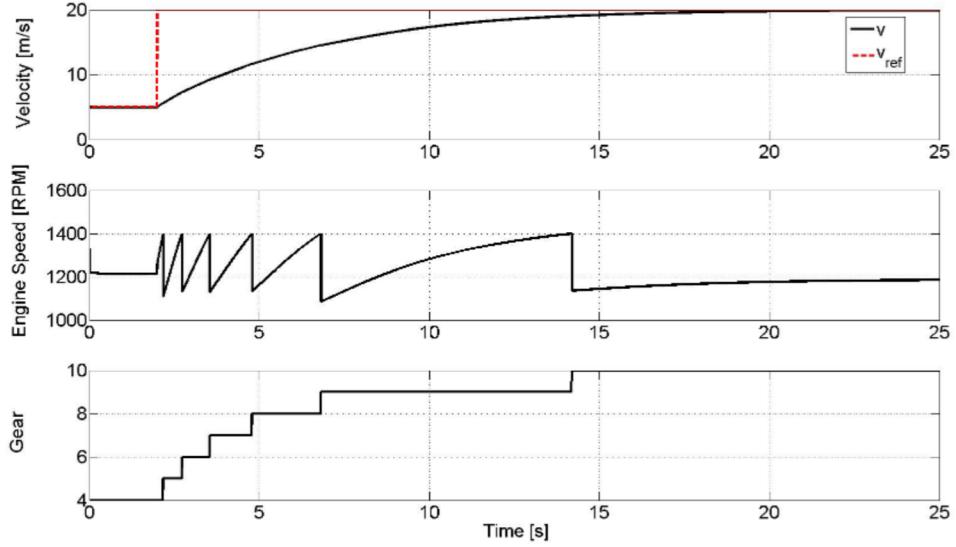


Figure 2.6: The gear changes, vehicle velocity and engine speed for a step in the reference velocity from 5 m/s to 20 m/s

where b_p is the brake parameter from the cruise controller and T_{Bmax} is the maximum available brake torque, for the implementation we used $T_{Bmax} = 84$ kNm.

2.3.2 Forces Acting on the Vehicle

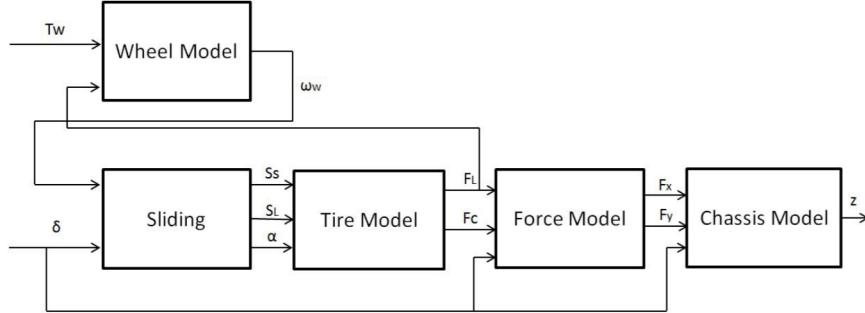


Figure 2.7: The structure of the Four Wheel Simulink model

Apart from the torque coming from the powertrain, the vehicle is affected by the interaction between the wheels and the road. In the Bicycle model linear tire forces and small slip angles were assumed. In the simulation model the modelled wheel forces are non-linear and dependent on the slip ratio of the wheels. In Figure 2.7 the structure of the Four Wheel model is demonstrated. As can be seen, the wheel torque generated from the powertrain works as an input together with the control parameter δ (the wheel angle).

The model of the chassis is similar to the Bicycle model with the difference that the simulation model includes four wheels instead of two. The equations are,

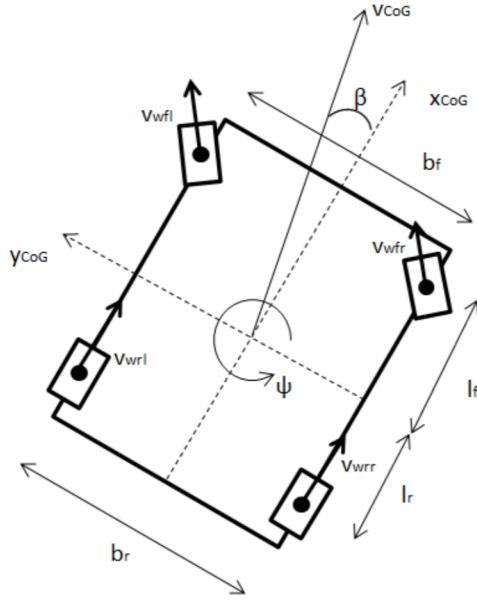


Figure 2.8: A vehicle represented by the Four Wheel model

$$\dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \quad (2.20a)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (2.20b)$$

$$\dot{\psi} = \frac{v_x}{l_f + l_r} \tan(\delta) \quad (2.20c)$$

$$m\dot{v}_x = mv_y \dot{\psi} + F_{xfl} + F_{xfr} + F_{xrl} + F_{xrr} - F_a - F_r \quad (2.20d)$$

$$m\dot{v}_y = -mv_x \dot{\psi} + F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr} \quad (2.20e)$$

$$I\ddot{\psi} = l_f(F_{yfl} + F_{yfr}) - l_r(F_{yrl} + F_{yrr}) + b_r(-F_{xfl} + F_{xfr} - F_{xrl} + F_{xrr}) \quad (2.20f)$$

where F_x and F_y are the wheel forces in the longitudinal and lateral direction, transformed to the center of gravity, CoG, of the vehicle. The subscripts after x and y indicates the different wheels according to, fl =front left, fr =front right, rl =rear left and rr =rear right. The parameter b_r is the distance between the rear wheels and l_f and l_r are, as before, the distance from the front and rear wheels to the CoG, respectively. See Figure 2.8 for a graphical explanation.

The geometry of the vehicle is used to transform the forces acting on the wheels, F_l and F_c , to forces acting on the center of gravity, F_x and F_y .

$$F_{xfl/r} = F_{ll/r} \cos(\delta) - F_{cl/r} \sin(\delta) \quad (2.21a)$$

$$F_{xrl/r} = F_{ll/r} \quad (2.21b)$$

$$F_{yfl/r} = F_{ll/r} \sin(\delta) + F_{cl/r} \cos(\delta) \quad (2.21c)$$

$$F_{yrl/r} = F_{cl/r} \quad (2.21d)$$

The wheel forces, F_l and F_c , describing Buckhardt's model are given by [Kiencke and Nielsen, 2005],

$$F_l = \mu \left(\frac{s_L}{s_{Res}} \cos(\alpha) + \frac{s_s}{s_{Res}} \sin(\alpha) \right) F_z \quad (2.22a)$$

$$F_c = \mu \left(\frac{s_s}{s_{Res}} \cos(\alpha) - \frac{s_L}{s_{Res}} \sin(\alpha) \right) F_z \quad (2.22b)$$

where α is the slip angle, defined in equation (2.7) (page 9), and demonstrated in Figure 2.10. s_L , s_s and s_{Res} are slip ratios and are explained in the next section, Section 2.3.3. In the same section the friction coefficient, μ , is defined. F_z is the vertical load distributed at the front and rear wheels. According to F. Borrelli [2005] it can be expressed as,

$$F_{zf} = \frac{mgl_r}{2(l_f + l_r)} \quad (2.23a)$$

$$F_{zr} = \frac{mgl_f}{2(l_f + l_r)} \quad (2.23b)$$

2.3.3 Slip Ratio

The longitudinal slip of a vehicle is the difference between the wheel ground point velocity, v_w , and the equivalent rotational velocity, $v_R = \omega_w R$, where R is the wheel radius and ω_w the angular speed of the wheel. Both the v_R and v_w are demonstrated in Figure 2.10. Furthermore, the longitudinal slip ratio of a vehicle is defined in the direction of v_w as [Kiencke and Nielsen, 2005],

$$s_L = \frac{v_R \cos(\alpha) - v_w}{\max(v_w, v_R \cos(\alpha))} \quad (2.24)$$

The slip ratio takes values between -1 and 1 , where a negative value means $v_R < v_w$, the wheel is skidding, whereas a positive value means $v_R > v_w$, the wheel is spinning.

In Figure 2.9 the lateral and longitudinal forces as a function of the longitudinal slip ratio are shown. As can be seen in the figure the longitudinal wheel force is, for a small longitudinal slip, proportional to the slip as was assumed for the Bicycle model.

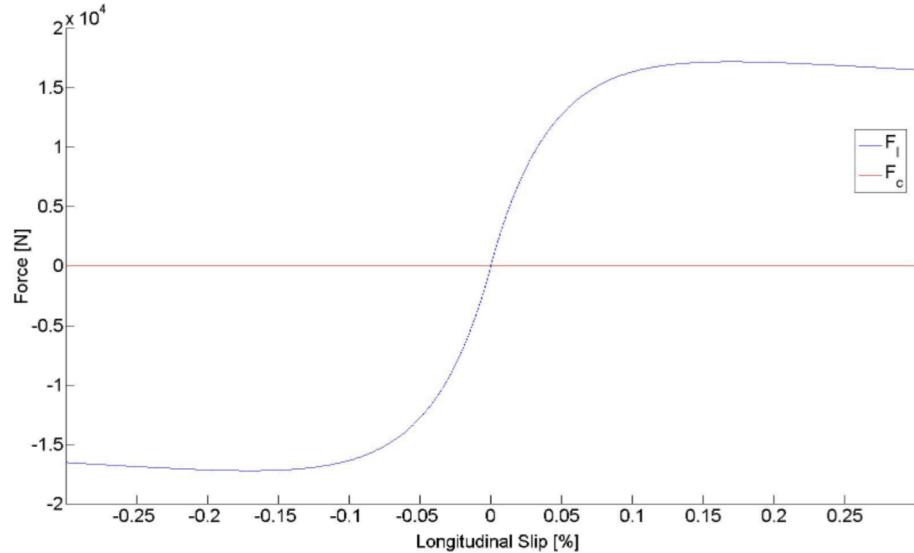


Figure 2.9: The longitudinal and lateral wheel forces as a function of the longitudinal slip ratio

The angular speed of the wheels, ω_w are given by the wheel dynamics as,

$$I_w \dot{\omega}_{wfl} = -F_{lfl}R + T_w \quad (2.25a)$$

$$I_w \dot{\omega}_{wfr} = -F_{lfr}R + T_w \quad (2.25b)$$

$$I_w \dot{\omega}_{wrl} = -F_{lrl}R + T_w \quad (2.25c)$$

$$I_w \dot{\omega}_{wrr} = -F_{lrr}R + T_w \quad (2.25d)$$

According to Kiencke and Nielsen [2005] the wheel ground point velocity, v_w in equation 2.24, can be computed as;

$$v_{wfl} = v - \dot{\psi} \left(\frac{b_f}{2} - l_f \beta \right) \quad (2.26a)$$

$$v_{wfr} = v + \dot{\psi} \left(\frac{b_f}{2} + l_f \beta \right) \quad (2.26b)$$

$$v_{wrl} = v - \dot{\psi} \left(\frac{b_r}{2} + l_r \beta \right) \quad (2.26c)$$

$$v_{wrr} = v + \dot{\psi} \left(\frac{b_r}{2} - l_r \beta \right) \quad (2.26d)$$

where β is the the vehicle body side slip angle, defined as the angle between the x-axis of the vehicle and the velocity vector, see Figure 2.8. β can be calculated as,

$$\beta = \arctan \left(\frac{l_r \tan(\delta)}{l_f + l_r} \right) \quad (2.27)$$

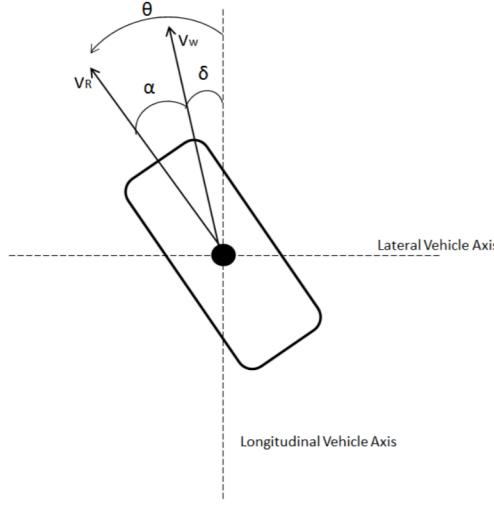


Figure 2.10: A demonstration of the different wheel angles

The lateral slip occurs when the wheels are turning, i.e. $\alpha \neq 0$. It is defined in right angles to v_w and dependent on the side slip angle. Conditional upon the sign of the longitudinal slip the lateral slip is calculated as,

$$s_s = \begin{cases} \frac{v_R \sin(\alpha)}{v_w} & \text{if } s_L < 0 \\ \tan(\alpha) & \text{if } s_L \geq 0 \end{cases}$$

When the longitudinal and lateral slip ratios have been calculated the resulting slip can easily be obtained by,

$$s_{Res} = \sqrt{s_L^2 + s_s^2} \quad (2.28)$$

Finally the resulting slip is used to calculate the friction coefficient, according to Burckhardt's model,

$$\mu = c_1(1 - e^{-c_2 s_{Res}}) - c_3 s_{Res} \quad (2.29)$$

The parameters, c_1 , c_2 and c_3 are related to the road conditions. If the vehicle is driving at dry asphalt they take the following values; $c_1 = 1.2801$, $c_2 = 23.99$ and $c_3 = 0.52$ [Kiencke and Nielsen, 2005].

2.4 Summary of the two Models

For both the Bicycle model and the Four Wheel model the same vehicle states and control signals were used, namely,

$$z = [X, Y, \psi, v_x, v_y, \dot{\psi}] \quad (2.30a)$$

$$u = [a, \delta] \quad (2.30b)$$

where, X, Y is the position in the global coordinate system, ψ the yaw angle, v_x, v_y the vehicle velocity in vehicle's coordinate system, and $\dot{\psi}$ the yaw rate. a is the acceleration and δ the wheel angle.

| Abbreviation | Force |
|--------------|-------------------------------------|
| F_{lf} | Longitudinal tire force front wheel |
| F_{lr} | Longitudinal tire force rear wheel |
| F_{cf} | Lateral tire force front wheel |
| F_{cr} | Lateral tire force rear wheel |
| F_a | Air drag force |
| F_r | Rolling resistance force |
| F_x^* | Driving force |

Table 2.2: The forces used in the vehicle models. * only used in the Bicycle model

The Bicycle model is given by,

$$\dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \quad (2.31a)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (2.31b)$$

$$\dot{\psi} = \frac{v_x}{l_f + l_r} \tan(\delta) \quad (2.31c)$$

$$m\dot{v}_x = F_x + m v_y \dot{\psi} - 2 F_{cf} \sin(\delta) - F_a - F_r \quad (2.31d)$$

$$m\dot{v}_y = -m v_x \dot{\psi} + 2(F_{cf} \cos(\delta) + F_{cr}) \quad (2.31e)$$

$$I\ddot{\psi} = 2(l_f F_{cf} \cos(\delta) - l_r F_{cr}) \quad (2.31f)$$

and the Four Wheel model it is given by,

$$\dot{X} = v_x \cos(\psi) - v_y \sin(\psi) \quad (2.32a)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (2.32b)$$

$$\dot{\psi} = \frac{v_x}{l_f + l_r} \tan(\delta) \quad (2.32c)$$

$$m\dot{v}_x = m v_y \dot{\psi} + F_{ll} \cos(\delta) - F_{cl} \sin(\delta) + F_{lr} \cos(\delta) - F_{cr} \sin(\delta) + F_{ll} + F_{lr} - F_a - F_r \quad (2.32d)$$

$$m\dot{v}_y = -m v_x \dot{\psi} + F_{ll} \sin(\delta) + F_{cl} \cos(\delta) + F_{lr} \sin(\delta) + F_{cr} \cos(\delta) + F_{cl} + F_{cr} \quad (2.32e)$$

$$I\ddot{\psi} = l_f (F_{ll} \sin(\delta) + F_{cl} \cos(\delta) + F_{lr} \sin(\delta) + F_{cr} \cos(\delta)) - l_r (F_{cl} + F_{cr}) + b_r (-F_{ll} \cos(\delta) + F_{cl} \sin(\delta) + F_{lr} \cos(\delta) - F_{cr} \sin(\delta) - F_{ll} + F_{lr}) \quad (2.32f)$$

where the input forces are wheel forces and presented in Table 2.2. The last subscripts, l and r , in the Four Wheel model stands for left and right wheel, respectively. In the Bicycle model the tire forces were modelled as linear and in the Four Wheel model they were modelled according to Buckhardt's model.

Chapter 3

Control

The aim of this thesis is to design a controller that allows autonomous driving of a truck. We decided to use Model Predictive Control, due to its abilities to work with constraints both on the states and the control signals. This is crucial for the control of a vehicle since it is constrained not only by mechanics of the vehicle but also by the environment. For example, a vehicle should not exceed the speed limits or drive too close to other vehicles.

This chapter starts with a literature review in the area of autonomous driving and platooning. Then the basis of MPC is addressed together with the derivation of the Linear Time-Varying MPC, LTV-MPC, we used to control the vehicle.

3.1 Literature Review

To manually drive close to another vehicle is hazardous due to the slow reaction time of humans. Hence, to enable platooning the vehicle needs longitudinal control. The first type of longitudinal control was the cruise controller, CC, patented by Ralph R. Teetor in 1950 [Teetor, 1950]. An extension to the CC is the adaptive cruise controller, ACC, patented in 1995, [Labuhn and Chundrlik Jr., 1995], and commercially implemented in some passenger cars and trucks. ACC uses the distance to the preceding vehicle and adapts the speed according to the distance. If there is no preceding vehicle it works as a CC. Even though it seems similar to platooning there is a big difference, ACC adapts the speed in order to keep a safe distance to the preceding vehicle whereas the aim of platooning is to keep a small distance. Furthermore, ACC will amplify disturbances acting on one vehicle to the following ones, i.e. string instability ¹. In order to ensure the string stability vehicle to vehicle, V2V, communication is required [Wang et al., 2015].

Cooperative Adaptive Cruise Control, CACC, is a prolongation of ACC using the V2V communication. V2V Communication can give information about, e.g. position and accelerations of other vehicles. In Wang et al. [2015] and Kianfar et al. [2012] CACC was achieved with two types of controllers, namely linear control and Model Predictive Control. In Mårtensson et al. [2012] CACC was achieved by a linear–quadratic regulator, LQR. Both Kianfar et al. [2012] and Mårtensson et al. [2012] describe the design of a vehicle participating in GCDC 2011.

This far only the longitudinal control of vehicles has been addressed. In order to achieve fully automated platoons the lateral control has to be considered as well. In platooning the main focus of the lateral control might be to keep the vehicle in the lane. However, it could also include to autonomously do lane changes or obstacle avoidance. Lane Keeping Assistance and Lane Centring Assistance are lateral controllers already commercially implemented in several vehicle models, e.g. some Volvo models [Volvo Car Corporation, 2015]. In 2014 Tesla released an autopilot to the Tesla S model that made the vehicle stay in the lane and enabled the driver to press a button in order for the vehicle to autonomously change lane [Tesla Motors Team, 2014].

Several studies on how to implement MPC for autonomous driving have been done. One prominent researcher in the area is F. Borrelli, he has co-written many articles regarding the use of MPC for autonomous driving. Numerous of them are used for this thesis, namely Falcone et al. [2007a,b, 2008]; Keviczky et al. [2006]; Kong et al. [2015]; Turri et al. [2013].

¹A platoon is said to be string unstable if oscillations do not attenuate as they propagate in the platoon

In Falcone et al. [2007a] a comparison was done between the execution time of non-linear MPC and linear MPC. Even though MPC is able to deal with non-linear problems and constraints it is very computational expensive to solve a constrained non-linear optimization problem. Falcone et al. [2007a] studied a scenario of a double lane change, where the computational time for the linear MPC was 0.03 seconds whereas it took between 0.15 to 1.30 seconds for the non-linear one, depending on the velocity of the vehicle. Hence, the latter is difficult to implement in vehicles not driving at low velocities. However, linear MPC cannot be used to a non-linear system, unless the system is linearised.

The linearisation can be done off-line, Winstead [2005], or on-line, Falcone et al. [2007b]. The former can be used if the vehicle is moving at a constant velocity, i.e. in light traffic so it can drive close to the speed limit. The latter, on the other hand, can be used when the vehicle drives in different velocities, hence it is more general but at the expense of an increase in the computational burden.

After linearisation, linear MPC can be used. The linear MPC can be written on quadratic form with linear constraints, i.e. be formulated as a Quadratic Programming, QP, problem that can be solved with numerical methods. In addition, since it is convex, solving the problem will give the optimal solution. Nevertheless, the linearised model is not as accurate as the non-linear one; there is a trade-off between the computational complexity and the accuracy of the model.

3.2 Model Predictive Control

In Model Predictive Control, MPC, the control problem is formulated as an optimization problem. So, basically it is solving an optimal control problem with the difference that the optimizations are done on-line in contrast to determining an off-line feedback policy [Mayne et al., 2000].

How the MPC problem is formulated may differ regarding the model and the cost function but there are four different steps in common for all methods [Glad and Ljung],

1. start at time t and predict a number of future output signals for the system,
2. build the cost function based on the future output and control signals and optimize with respect to the control signal,
3. input the control signal to the system,
4. wait until next time step and start over with step 1.

To calculate the future outputs of the system a prediction model is used. In this thesis the predictions were done with a state space model, derived in Chapter 2.

If we have a discrete model and are controlling the system towards the origin the MPC problem can be expressed as,

$$\begin{aligned} & \underset{u}{\text{minimize}} \quad \sum_{i=1}^H J(z_i, u_i) \\ & \text{subject to} \quad z_{i+1} = f(z_i, u_i) \\ & \quad C(z_i, u_i) \leq M \end{aligned} \tag{3.1}$$

where the function J is called the cost function and can be both linear and non-linear. However, it is often chosen to be quadratic in order to reduce the complexity of the problem. The function C expresses the constraints on the system and the function f represents the prediction model, these can, as well, be both linear and non-linear.

The sum is taken to a value H , called the horizon. The horizon decides the number of predictions and future control signals to take into consideration at each time step. In equation (3.1) the same horizon is used for the control signal and the state predictions. Nevertheless, it is common to use two different horizons, one prediction horizon, H_p , and one control horizon, H_c . The control horizon is often shorter than the prediction horizon in order to reduce the size of the optimization problem. If $H_c < H_p$ the control signals $u_{H_c+1} \dots u_{H_p-1}$ need to be assigned an implicit value, commonly this is done by assigning them the same value as the last control signal u_{H_c} [Glad and Ljung].

Solving the optimization problem gives rise to H number of optimal control signals, however only the elements of the first control signal are used as an input to the system before the optimization problem is solved again for the next time step.

3.3 Linear Model Predictive Control

Since the optimization problem need to be solved each sampling time it is computational expensive. In order to reduce the complexity the constraints and the prediction model are often chosen linear, additionally to the quadratic cost function mentioned beforehand. If this is done the optimization problem can be formulated as,

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & z_{H_p}^T Q_f z_{H_p} + \sum_{i=1}^{H-1} z_i^T Q z_i + u_i^T R u_i \\ \text{subject to} \quad & z_{i+1} = A z_i + B u_i \\ & C z_i + D u_i \leq b \end{aligned} \quad (3.2)$$

where Q_f , Q and R are the terminal state weight matrix, state weight matrix and control weight matrix, respectively and $Q_f \geq 0, Q \geq 0, R \geq 0$. The optimization problem is subject to both the prediction model given by, $z_{i+1} = A z_i + B u_i$, and the constraints given by, $C z_i + D u_i \leq b$.

In order to formulate the problem like (3.2) the prediction model has to be linear. If the system is non-linear the prediction model need to be linearised. Since the Bicycle model we derived in Chapter 2 is non-linear we linearised the system at each sampling time around the reference points, i.e. on-line linearisation. If the derivative of the state is given by $\dot{z} = f(z, u)$, the linear prediction model is given by,

$$z_{i+1} = A_i z_i + B_i u_i \quad (3.3)$$

where, $A_i = \frac{\partial f(z, u)}{\partial z} \big|_{z=z_{ri}, u=u_{ri}}$ and $B_i = \frac{\partial f(z, u)}{\partial u} \big|_{z=z_{ri}, u=u_{ri}}$ and z_{ri} and u_{ri} are the references for the control and state signals. Up to this point we have controlled the system towards the origin, hence, $z_{ri} = u_{ri} = 0$ but in the next section we will demonstrate how MPC can be used for tracking varying references.

3.4 Model Predictive Control for Reference Tracking

The MPC formulations in (3.1) and (3.2) are controlling the system to the origin. Nevertheless, MPC can be used to control a system to a non-zero reference value. To do this the MPC is formulated as in (3.2) but instead of minimizing z and u the errors between the actual state and control signal and their references are minimized. To do this two new variables are defined, namely, \tilde{z} and \tilde{u} , where $\tilde{z} = z - z_{ref}$ and $\tilde{u} = u - u_{ref}$ and the prediction model is linearised around the non-zero references. The minimization problem is then formulated as,

$$\begin{aligned} \underset{\tilde{u}}{\text{minimize}} \quad & \tilde{z}_H^T Q_f \tilde{z}_H + \sum_{i=1}^{H-1} \tilde{z}_i^T Q \tilde{z}_i + \tilde{u}_i^T R \tilde{u}_i \\ \text{subject to} \quad & \tilde{z}_{i+1} = A \tilde{z}_i + B \tilde{u}_i \\ & \tilde{z}_i = z_i - z_{ref} \\ & \tilde{u}_i = u_i - u_{ref} \\ & C \tilde{z}_i + D \tilde{u}_i \leq b \end{aligned} \quad (3.4)$$

This optimization problem minimizes the cost function with respect to \tilde{u} hence the optimal control signal, u^* , is given from, $u^* = \tilde{u} + u_{ref}$.

In the optimization problem (3.4) the references are constant over time but it can be reformulated to be used for time variant references, so called Linear Time-Varying (LTV) MPC. To formulate the MPC as a LTV-MPC we reformulate, z_{ref} and u_{ref} to vectors, $z_{ref} = [z_{ref1}, \dots, z_{refH_p}]$ and $u_{ref} = [u_{ref1}, \dots, u_{refH_c}]$. This means the vehicle tracks a new reference for each iteration. Furthermore, the prediction model is linearised around the reference trajectories instead of a constant reference value. The optimization problem is then transformed to,

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & \tilde{z}_H^T Q_f \tilde{z}_H + \sum_{i=1}^{H-1} \tilde{z}_i^T Q \tilde{z}_i + \tilde{u}_i^T R \tilde{u}_i \\
\text{subject to} \quad & \tilde{z}_{i+1} = A \tilde{z}_i + B \tilde{u}_i \\
& \tilde{z}_i = z_i - z_{refi} \\
& \tilde{u}_i = u_i - u_{refi} \\
& C z_i + D u_i \leq b
\end{aligned} \tag{3.5}$$

To be able to solve this optimization problem the references for all the future states and the control signals have to be known. One way to generate these references, used in this thesis, is described in Chapter 4.

3.5 Formulate as Quadratic Programming

There are different ways to solve the optimization problem formulated in (3.5). In this thesis the problem was formulated as a quadratic programming, QP, problem and solved in MATLAB. The reformulation of the problem to QP form is inspired by Kuhne et al. [2005].

The general QP problem has the form,

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & \frac{1}{2} u^T H u + f^T u \\
\text{such that} \quad & D u \leq b \\
& D_{eq} u = beq \\
& lb \leq u \leq ub
\end{aligned} \tag{3.6}$$

In this thesis only inequality constraints were used and the problem was formulated for reference tracking. Hence the optimization problem was formulated as,

$$\begin{aligned}
\underset{\tilde{u}}{\text{minimize}} \quad & \frac{1}{2} \tilde{u}^T H \tilde{u} + f^T \tilde{u} \\
\text{such that} \quad & D \tilde{u} \leq b
\end{aligned} \tag{3.7}$$

In order to rewrite problem (3.5) to this form the summation, in the cost function, has to be removed. This can be done by introducing two new vectors,

$$\bar{z}(i+1) = \begin{bmatrix} \tilde{z}_{i+1|i} \\ \tilde{z}_{i+2|i} \\ \vdots \\ \tilde{z}_{i+H|i} \end{bmatrix} \tag{3.8}$$

$$\bar{u}(i) = \begin{bmatrix} \tilde{u}_{i|i} \\ \tilde{u}_{i+1|i} \\ \vdots \\ \tilde{u}_{i+H-1|i} \end{bmatrix} \tag{3.9}$$

where the notation $i+1|i$ indicates the estimated value at instant $i+1$ predicted at instant i . By using these two vectors the cost function in (3.5) can be written,

$$J = \bar{z}^T \bar{Q} \bar{z} + \bar{u}^T \bar{R} \bar{u} \tag{3.10}$$

where $\bar{Q} = \text{diag}(Q, Q, \dots, Q_f)$ and $\bar{R} = \text{diag}(R, R, \dots, R)$.

To describe how the states and control signals evaluate during the horizon two matrices are introduced,

$$\bar{A}(i) = \begin{bmatrix} A_{i|i} \\ A_{i+1|i} A_{i|i} \\ \vdots \\ \alpha(i, 2, 0) \\ \alpha(i, 1, 0) \end{bmatrix} \quad (3.11)$$

$$\bar{B}(i) = \begin{bmatrix} B_{i|i} & 0 & \dots & 0 \\ A_{i+1|i} B_{i|i} & B_{i+1|i} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(i, 2, 1) B_{i|i} & \alpha(i, 2, 2) B_{i+1|i} & \dots & 0 \\ \alpha(i, 1, 1) B_{i|i} & \alpha(i, 1, 2) B_{i+1|i} & \dots & B_{i+H-1|i} \end{bmatrix} \quad (3.12)$$

where,

$$\alpha(i, j, l) = \prod_{k=H-j}^l A_{i+k|i} \quad (3.13)$$

By using the vectors (3.8) and (3.9) and the matrices (3.11) and (3.12) the linearised prediction model, equation (3.3), can be expressed as,

$$\bar{z}_{i+1} = \bar{A}_i \bar{z}_{i|i} + \bar{B}_i \bar{u}_i \quad (3.14)$$

Then equation (3.14) can be used to reformulate the cost function in equation (3.10) to the form of the cost function in the QP minimization problem, (3.7),

$$J = \frac{1}{2} \bar{u}^T H_i \bar{u} + f_i^T \bar{u} \quad (3.15)$$

with,

$$\begin{aligned} H_i &= 2(\bar{B}_{i|i}^T \bar{Q} \bar{B}_{i|i} + \bar{R}) \\ f_i &= 2\bar{B}_{i|i}^T \bar{Q} \bar{A}_{i|i} \bar{z}_{i|i} \\ d_i &= \bar{A}_{i|i} \bar{z}_{i|i} \bar{Q} \bar{A}_{i|i} \bar{z}_{i|i} \end{aligned}$$

where we receive H , f and d by collecting the quadratic \bar{u} terms, the linear \bar{u} terms and the terms not depending on \bar{u} . The latter are not included in the cost function since the minimization is done with respect to \bar{u} , hence terms not depending at \bar{u} will not influence the result.

3.5.1 Constraints

Since we have rewritten our problem as a QP problem the constraints have to be formulated as $D\bar{u} \leq b$. We chose to constrain both the control signal and the states. The control signals, the acceleration and the front wheel angle, were constrained according to,

$$u_{min} \leq u_i \leq u_{max} \quad (3.16)$$

According to Kuhne et al. [2005], this can be written on the form $D\bar{u} \leq b$ by using

$$D_1 = \begin{bmatrix} I \\ -I \end{bmatrix} \quad (3.17)$$

and,

$$b_1 = \begin{bmatrix} u_{max} - u_{refi} \\ u_{min} + u_{refi} \end{bmatrix} \quad (3.18)$$

where I is the identity matrix and u_{ref} the reference value of u . Furthermore we constrained the change in the acceleration, longitudinal jerk, and the change rate of the front wheel angle,

$$|u_i - u_{i-1}| \leq \Delta u_{max} \Leftrightarrow -\Delta u_{max} \leq u_i - u_{i-1} \leq \Delta u_{max} \quad (3.19)$$

where Δu_{max} is a vector containing the values for the maximum allowed change in acceleration and the maximum allowed change rate of the front wheel angle. By introducing the following matrices the constraints will be at the desired form,

$$D_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3.20)$$

and,

$$b_2 = \begin{bmatrix} \Delta u_{max} + u_{i-1}^* - u_{ref1} \\ \Delta u_{max} - u_{i-1}^* + u_{ref1} \end{bmatrix} \quad (3.21)$$

Where u_{i-1}^* is the optimal control signal given to the system in the previous time step and u_{ref1} are the first elements in the reference vector for the control signals.

Additionally we added constraints to the states. To do this the constraints on the state, z , have to be expressed in terms of \tilde{u} . First we constrained the state error, \tilde{z} ,

$$|\tilde{z}| \leq \Delta_z \quad (3.22)$$

Where Δ_z is a column vector containing the maximum allowed error for each state. By using equation (3.14) we can formulate the problem as,

$$|\bar{A}\tilde{z} + \bar{B}\tilde{u}| \leq \Delta_z \Leftrightarrow -\Delta_z \leq \bar{A}\tilde{z} + \bar{B}\tilde{u} \leq \Delta_z \quad (3.23)$$

This can be written as,

$$\bar{B}\tilde{u} \leq \Delta_z - \bar{A}\tilde{z} - \bar{B}u_{ref} \quad (3.24a)$$

$$-\bar{B}\tilde{u} \leq \Delta_z + \bar{A}\tilde{z} + \bar{B}u_{ref} \quad (3.24b)$$

Which can be formulated as,

$$D_3\tilde{u} \leq b_3 \quad (3.25)$$

if,

$$D_3 = \begin{bmatrix} \bar{B} \\ -\bar{B} \end{bmatrix} \quad (3.26)$$

and,

$$b_3 = \begin{bmatrix} \Delta_z - \bar{A}\tilde{z} - \bar{B}u_{ref} \\ \Delta_z + \bar{A}\tilde{z} + \bar{B}u_{ref} \end{bmatrix} \quad (3.27)$$

Next constraints were added to make the vehicle keep a safe distance to preceding vehicles. In Figure 3.1 we demonstrate the forbidden area for preceding vehicles.

As can be seen in the figure we only need to consider the longitudinal distance to preceding vehicles if they are close in the lateral direction. The lateral distance is safe, i.e. the vehicle does not need to keep a distance in the longitudinal direction, if the following holds,

$$\text{Safe Y-distance if : } \begin{cases} y_P - \frac{1}{2}W_P > y_v + \frac{1}{2}W_v + y_{safe} & \text{if } y_P > y_v \\ y_P + \frac{1}{2}W_P < y_v - \frac{1}{2}W_v - y_{safe} & \text{if } y_P < y_v \end{cases}$$

where W is the width of the respective vehicle and y_{safe} is a constant added in order for the vehicle to keep a specified distance to the preceding vehicle when passing, i.e. not touch the side of the other vehicle, see Figure 3.1.

If the two vehicles are in the same lane or if the distance in the y -direction is too small the trailing vehicle has to keep a distance in the longitudinal, x , direction. We want the following to hold,

$$x_V - x_P \leq -\Delta_s \quad (3.28)$$

where x_v is the position of our vehicle and x_P the position of the preceding vehicle. Δ_s is the longitudinal safe distance between the two vehicles defined in the specifications for GCDC as,

$$d_{safe} = h_{safe}v + r_{safe} \quad (3.29)$$

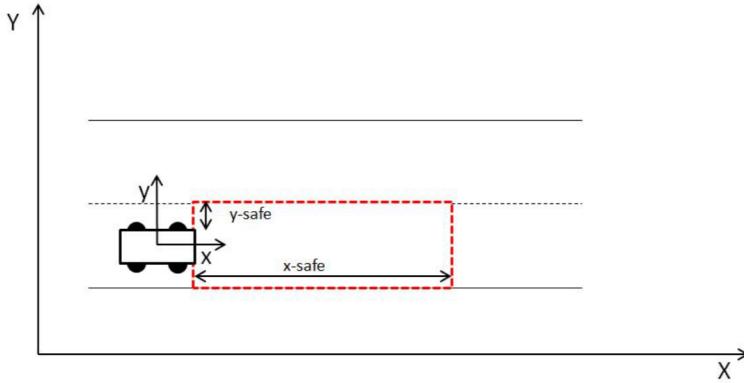


Figure 3.1: The area marked in red is forbidden are for preceding vehicles

where r_{safe} is the stand-still safety distance, v the host vehicle speed and h_{safe} the headway time, defined as the time it will take for the host vehicle to reach the preceding vehicle. This safety distance cannot be violated in any of the GCDC scenarios [Didoff, 2014].

Since we used a coordinate system defined from the CoG of our vehicle the vehicles positions in the global coordinate system, (X, Y) , had to be transformed to coordinates in the vehicle's coordinate system, (x, y) , see Figure 3.1. This was done by using a transformation matrix, T , defined by Kiencke and Nielsen [2005] as,

$$T = \begin{bmatrix} \cos(-\psi) & -\sin(-\psi) \\ \sin(-\psi) & \cos(-\psi) \end{bmatrix} \quad (3.30)$$

where ψ is the heading of our vehicle.

To express equation (3.28) at the form $D\tilde{u} \leq b$ we once again used equation (3.14) and a new matrix called M to only get the x -state from the state matrix z . This gave us the following expression,

$$M\bar{B}\tilde{u} \leq -\Delta_s + x_P - M\bar{A}\tilde{z} - Mz_{ref} - M\bar{B}u_{ref} \quad (3.31)$$

Hence,

$$D_4 = [M\bar{B}] \quad (3.32)$$

and,

$$b_4 = [-\Delta_s + x_P - M\bar{A}\tilde{z} - Mz_{ref} - M\bar{B}u_{ref}] \quad (3.33)$$

Note that the constraint on the distance to preceding vehicles only shall be active when the preceding vehicle is too close in the y -direction.

Finally all the constraints were added together by using the following matrices,

$$D = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{bmatrix} \quad (3.34)$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (3.35)$$

For numerical values of the constraints see the Implementation Chapter, Chapter 5, Section 5.2.

The constraints presented above are so called hard constraints meaning that if one of the constraints cannot be fulfilled the problem will be infeasible and no solution to the problem will be found [Camacho and Bordons, 2007].

However, sometimes we would like to allow the violation of some of the constraints instead of ending up with an infeasible problem. For example, if you are driving a car and suddenly see an obstacle in front of you it is preferable to decelerate fast, even if it violates the constraints regarding

the acceleration and longitudinal jerk, than to collide with the obstacle. To make the system operate like this we can use soft constraints instead of hard.

Soft constraints are obtained by adding a variable, ϵ , to the cost function and to the inequality constraints, $D\ddot{u} \leq b + \epsilon$. This makes it possible to temporarily relax the constraints by putting $\epsilon > 0$. But since ϵ is included to the cost function such violations of the constraints will be penalized [Camacho and Bordons, 2007].

For this thesis we used soft constraints in all cases apart from the constraint on the safety distance to other vehicles. Since the other constraints are to ensure comfort they may be violated if needed but if the distance constraint is violated it may lead to collisions which is unacceptable.

Chapter 4

Reference Generation

In order to use the MPC for trajectory tracking the references for all the states, $z = [X, Y, \psi, v_x, v_y, \dot{\psi}]$ and for all the control signals, $u = [a, \delta]$ need to be generated. To generate these references we have chosen to use trajectory tracking. Trajectory tracking uses time and space references in opposite to the path following where only space references are used. We decided to use trajectory tracking since we wanted the vehicle velocity to be incorporated already in the waypoints.

The X and Y references represent the vehicle position in the global frame. The goal of the control is to reach a specific position, (X, Y) , at a specific time, depending on the velocity of the vehicle. We will refer to these positions as waypoints. Depending on the waypoints the other references, both for the states and the control signals, can be calculated.

4.1 Generate the Waypoints

In order to generate the waypoints we assumed to have a map with coordinates. The global coordinates, (X, Y) , are fixed in the map whereas the vehicle's coordinates, (x, y) , varies with the heading of the vehicle, see Figure 4.1.

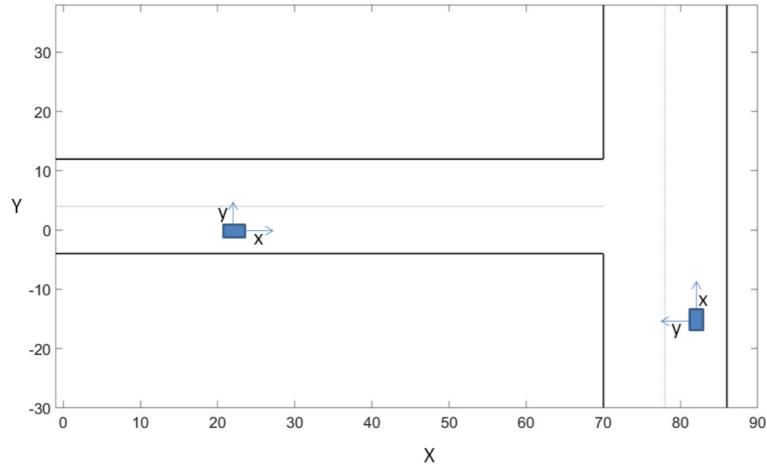


Figure 4.1: A map of a T-intersection demonstrating the global coordinate system and the vehicle's coordinate system

The generation of the waypoints was done in two steps; first coordinates for the path the vehicle should follow were found and second these points were spaced accordingly to the velocity of the vehicle.

If the vehicle is supposed to follow an existing road on the map we can receive the coordinates for the road by looking at the map. Nevertheless, for this thesis we also studied the scenarios when the vehicle should make a lane change or do a turn. For these two scenarios the path the vehicle should

follow need to be generated. To generate it we have used quadratic Bezier curves. These curves are constructed by three control points, P_0 , P_1 , and P_2 according to [Chen et al., 2013],

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \quad (4.1)$$

where t takes values between 0 and 1.

A useful property of the Bezier Curve is that it always passes through the point defined as the start point, P_0 and the point defined as the end point, P_2 . This is suitable, since the vehicle should start from its start position and end in the point specified as the end position.

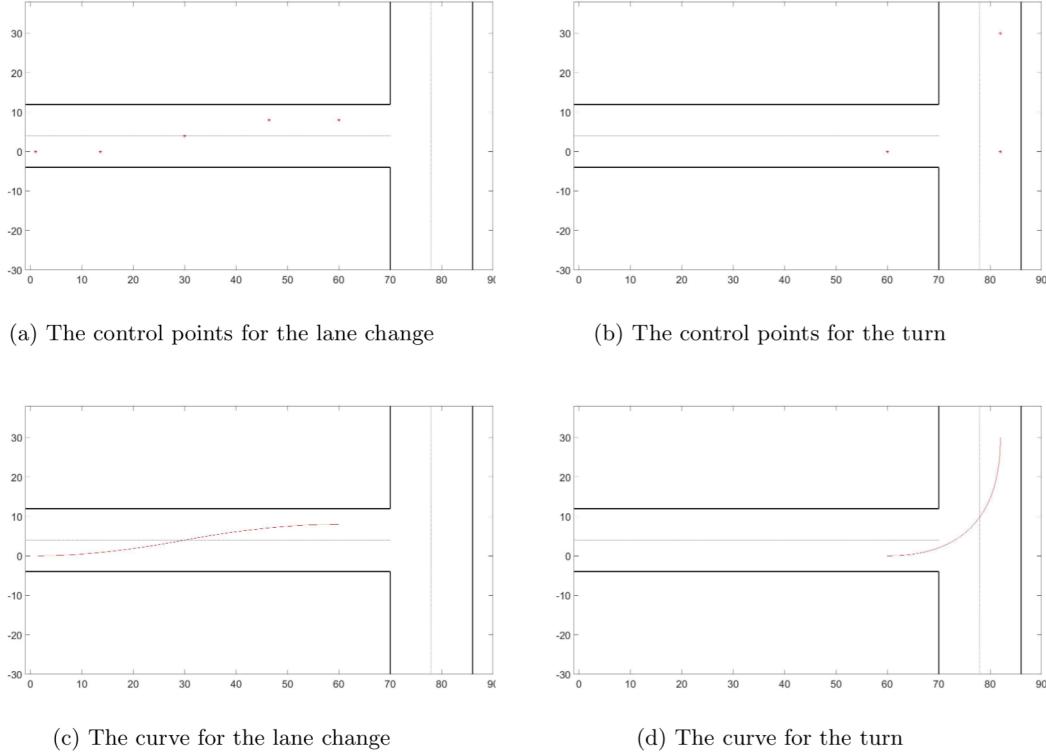


Figure 4.2: Plots of the control points and corresponding Bezier curves for a lane change and a turn, respectively

By the use of several Bezier curves we were able to construct paths for both the lane change and the turn scenario. For the lane change we used two quadratic Bezier curves, described by the points P_0 , P_1 and P_2 and Q_0 , Q_1 , and Q_2 , respectively. According to Chen et al. [2013] three specifications have to be fulfilled to make the resulting curve smooth,

- Positional Continuity, the end point of the first curve must be equal to the start point of the second curve, $P_2 = Q_0$,
- Slope Continuity, the coincident point, $P_2 = Q_0$, has to be collinear with two neighbouring control points, and
- Curvature Continuity, five of the six points have to be coplanar.

In order to fulfil these specifications the points were chosen according to,

$$P_0 = (X_{start}, Y_{start}), P_1 = (X_{start} + cX_a, Y_{start}) \text{ and } P_2 = (X_{start} + X_a, Y_{middle})$$

$$Q_0 = (X_{start} + X_a, Y_{middle}), Q_1 = (X_{start} + 2X_a - cX_a, Y_{end}) \text{ and } Q_2 = (X_{start} + 2X_a, Y_{end})$$

The points P_2 and Q_0 represent when the vehicle has travelled half the way between the start and end position. Hence,

$$Y_{middle} = \frac{|Y_{start} - Y_{end}|}{2} \quad (4.2)$$

X_a is the distance the vehicle has travelled after half the lane change time. According to Chen et al. [2013] a lane change takes approximately six seconds. The variable c is a tuning parameter, $c < 1$, used to tune the behaviour of the curve.

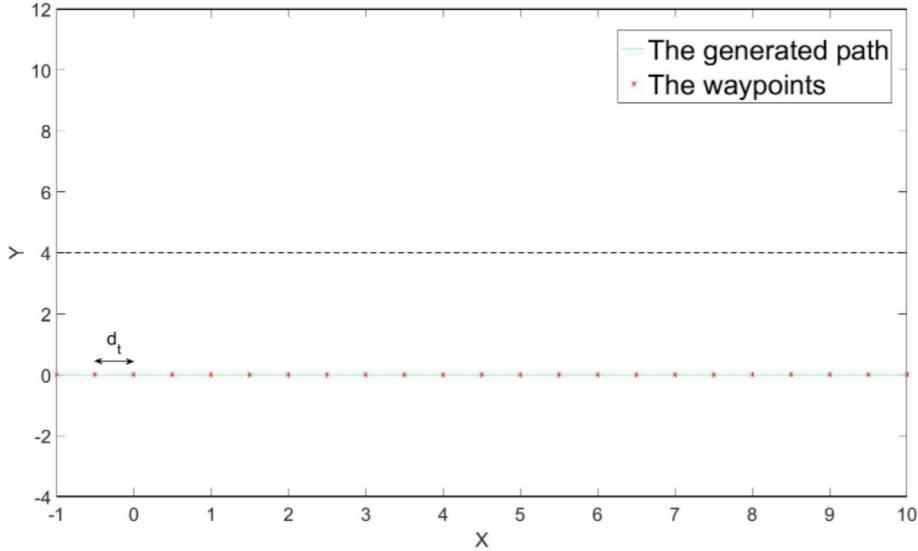


Figure 4.3: The generated path for a straight road and the corresponding waypoints

To find a suitable path for the vehicle to follow when it should make a turn one Bezier curve is sufficient. The first control point then represented the coordinates for the vehicle position right before it should make the turn and the last control point was the desired end position of the vehicle. The middle point was tuned until the curve had a desirable behaviour.

In Figure 4.2 the control points and the corresponding curves can be seen for the lane change and the turn.

The paths we have derived for all three scenarios, follow a road, make a lane change and do a turn are discrete and the distance between the points is Euclidean. However, the points are separated with a small distance that does not correspond to the vehicle velocity. We would like to increase the distance, so every point is separated a distance, d_t , corresponding to the distance the vehicle travels for one sampling time, T_s . The travelled distance, d_t , is given by,

$$d_{ti} = v_{refi} T_s \quad (4.3)$$

where v_{refi} is equal to the speed reference at instant i , derived in the next section. Hence, we take the points separated with a distance, d_t and use as our waypoints, (X_{refi}, Y_{refi}) , see Figure 4.3.

4.2 Acceleration and Velocity References

In most driving scenarios the velocity reference in the longitudinal direction of the vehicle, v_x , is given by the speed limit for the road and, since the speed should be constant, the acceleration reference is zero. In both the road-follow and lane change scenario we assumed this, hence both the velocity references and the acceleration references were constant. Additionally, the references for the velocity in the lateral direction of the vehicle, v_y , were zero for all the scenarios.

Nevertheless, when the vehicle shall make a turn it has to slow down to take the curve. Furthermore, we like the vehicle to decelerate from the speed before the curve, v_{max} to the suitable speed for the curve, v_{curve} without exceeding the maximum deceleration, a_{min} or the maximum longitudinal jerk, J_{max} .

If the maximum longitudinal jerk not should be exceeded the number of time samples it takes to

reach the maximum deceleration is given by,

$$T_a = \frac{|a_{min} - a_{start}|}{J_{max}} \quad (4.4)$$

where a_{start} is the acceleration the vehicle has at start, usually zero.

When the maximum deceleration is reached the velocity has changed from v_{start} to a lower velocity, called v_{new} . The difference in velocity is given by,

$$v_{diff} = |v_{start} - v_{new}| \quad (4.5)$$

Next we let the vehicle decelerate with maximum deceleration until it reaches the velocity $v_{curve} + v_{diff}$. The number of time samples it takes to reach this velocity is given by,

$$T_v = \frac{|(v_{start} - v_{diff}) - (v_{curve} + v_{diff})|}{a_{min}} \Big/ T_s \quad (4.6)$$

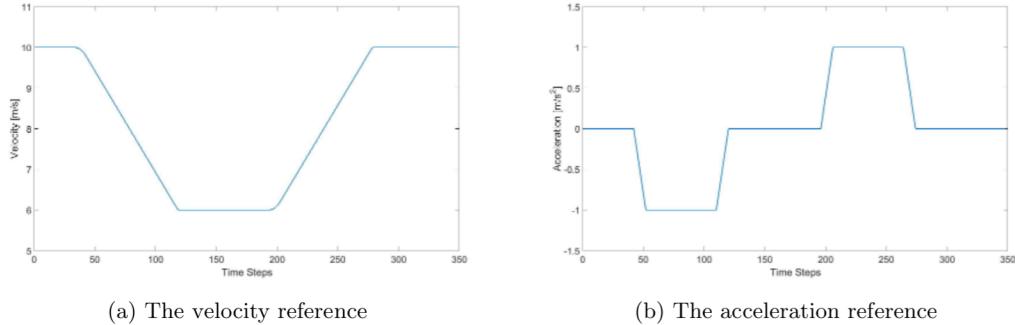


Figure 4.4: The velocity and acceleration reference for a vehicle driving through a curve

The deceleration will then go back to zero, which takes T_a time samples and lower the velocity with v_{diff} m/s according to (4.4) and (4.5).

Hence, by starting to decelerate $2T_a + T_v$ time steps from the start of the curve the velocity will be reached in time without exceeding the maximum deceleration or longitudinal jerk. The same approach is used when the vehicle shall accelerate up to the original velocity after the curve.

The positions can be translated from time to distance by using the information about the velocity for each time step. In Figure 4.4 the acceleration and velocity references for a vehicle driving through a curve is demonstrated.

4.3 References for Yaw Angle, Yaw Rate and Front Wheel Steering Angle

Finally the references for the yaw angle, ψ , yaw rate, $\dot{\psi}$ and the front wheel angle, δ , need to be calculated. The yaw angle reference, ψ_{refi} , is calculated by using the waypoints according to,

$$\psi_{refi} = \arctan\left(\frac{Y_{refi+1} - Y_{refi}}{X_{refi+1} - X_{refi}}\right) \quad (4.7)$$

This can then be used to calculate the references for the yaw rate, $\dot{\psi}_{refi}$,

$$\dot{\psi}_{refi} = \frac{\psi_{refi+1} - \psi_{refi}}{T_s} \quad (4.8)$$

To calculate the references for the wheel angle, δ , the prediction model from Chapter 2 is used. It states that the yaw rate is given by,

$$\dot{\psi} = \frac{v_x}{l_f + l_r} \tan(\delta) \quad (4.9)$$

By reformulate this equation an expression for δ_{ref} is given,

$$\delta_{ref} = \arctan\left(\frac{(l_f + l_r)(\psi_{refi+1} - \psi_{refi})}{v_{refi}T_s}\right) \quad (4.10)$$

All these references are then used in the MPC to generate an optimal control signal for the system.

Chapter 5

Implementation

In the following chapter the implementation is presented. In the previous chapters the chosen methods for the implementation and the theory behind them are presented. Thus, this chapter will not focus on these parts but instead present numerical values used for the simulations.

5.1 The Simulation Model

For the simulations the Bicycle model and the more complicated Four Wheel model, both derived in Chapter 2, were used. Since the Four Wheel model is more complicated it is supposed to better describe the real behaviour of a vehicle and therefore the simulations are mainly done with this model. Nevertheless, some simulations were done with the bicycle model in order to compare the behaviour of the two models. The prediction model, used for the MPC, is based on the bicycle model. Thus, the simulations done with the bicycle model is expected to be more similar to the generated references.

| Parameter | Abbreviation | Value |
|------------------------------------|--------------|---|
| Vehicle mass | m | 17130 kg |
| Inertia around z-axis | I | 93343 kgm ² |
| Wheel inertia* | I_w | 20 kgm ² |
| Wheel radius* | R | 0.5 m |
| Cornering stiffness front wheels** | C_f | 151550 N/rad |
| Cornering stiffness rear wheels** | C_r | 52020 N/rad |
| Distance from front wheels to CoG | l_f | 2.4739 m |
| Distance from rear wheels to CoG | l_r | 3.7761 m |
| Distance between front wheels | b_f | 1.245 m |
| Distance between rear wheels | b_r | 1.245 m |
| Cross sectional area | A_a | 10.26 m ² |
| Roll resistance coefficient | D_r | 1.5×10^{-3} |
| Air drag coefficient | C_D | 0.56 |
| Air density | ρ_a | 1.29 kg/m ³ |
| Road friction parameters * | - | $c_1 = 1.2801, c_2 = 23.99, c_3 = 0.52$ |
| Sampling time Four Wheel model* | T_s | 0.001 s |
| Sampling time Bicycle model** | T_s | 0.05 s |

Table 5.1: The parameters used in the models of the vehicles.

*Only needed in the Four Wheel model. ** Only needed in the Bicycle model

In table 5.1 we present the parameters used for the two models. Both the Bicycle model and the Four Wheel model were given acceleration, a , and front wheel angle, δ , as input and gave the next state, z_{i+T_s} , as output. In the Bicycle model the acceleration was transformed to a driving force according to,

$$F_{xi} = ma_i + F_{ri} + F_{ai} \quad (5.1)$$

before added to the model. Where m is the vehicle mass and F_r and F_a are the rolling resistance and air drag forces, both defined in Chapter 2. Since we assumed to receive measurements for the vehicle velocity the assumed air drag and rolling resistance could be calculated. Nevertheless, if the assumed vehicle mass or F_r and F_a have values different from the real parameters it will lead to an error in the velocity and, hence, in the longitudinal position. Since the difference between the states and the state references are included in the cost function the controller will compensate for these errors by increasing or decreasing the acceleration.

The vehicle states are updated with the Bicycle model according to,

$$X_{i+T_s} = X_i + (v_{xi} \cos(\psi_i) - v_{yi} \sin(\psi_i)) T_s \quad (5.2a)$$

$$Y_{i+T_s} = Y_i + (v_{xi} \sin(\psi_i) + v_{yi} \cos(\psi_i)) T_s \quad (5.2b)$$

$$\psi_{i+T_s} = \psi_i + \left(\frac{v_{xi}}{l_f + l_r} \tan(\delta_i) \right) T_s \quad (5.2c)$$

$$v_{xi+T_s} = v_{xi} + \left(F_{xi} + mv_{yi} \dot{\psi}_i - 2F_{cfi} \sin(\delta_i) - F_{ai} - F_{ri} \right) \frac{T_s}{m} \quad (5.2d)$$

$$v_{yi+T_s} = v_{yi} + \left(-mv_{xi} \dot{\psi}_i + 2(F_{cfi} \cos(\delta_i) + F_{cri}) \right) \frac{T_s}{m} \quad (5.2e)$$

$$\dot{\psi}_{i+T_s} = \dot{\psi}_i + \left(2(l_f F_{cfi} \cos(\delta_i) - l_r F_{cri}) \right) \frac{T_s}{I} \quad (5.2f)$$

where T_s represents the sampling time and is given in table 5.1.

In the Four Wheel model the acceleration reference was transformed to a velocity reference in order to be used as an input to the cruise controller. This was done by using the following approximation,

$$a_i \approx \frac{v_{i+T_s} - v_i}{T_s} \rightarrow v_{i+T_s} \approx v_i + a_i T_s \quad (5.3)$$

However, the Cruise Controller in the Four Wheel model makes the model respond to a change in the acceleration slower than predicted by the prediction model. Hence, by using v_{i+T_s} as the input to the cruise controller we will not reach the desired velocity in one sample time. To deal with this we instead used v_{i+T_R} as the input, where T_R is a time sample ahead of T_s compensating for the slower response of the CC. For this thesis we used $T_R = 500$, which corresponds to a sample 0.5 s ahead, since we used $T_s = 0.001$ s for the Simulink model.

The reason the cruise controller was not included in the vehicle model was because we wanted to study if a more simple prediction model was able to control a more complicated system. It is desirable to have a simple prediction model to decrease the computational expenses.

5.2 The Control Algorithm

As mentioned in Chapter 1, three blocks can describe the parts needed to design and validate a MPC for autonomous driving, i.e. reference generation, MPC and vehicle, see Figure 1.4 (page 5). The MPC block can be separated into smaller blocks demonstrating what is done for each iteration, see Figure 5.1. The linearisation block is explained in Chapter 3, page 23 and the formulation of the

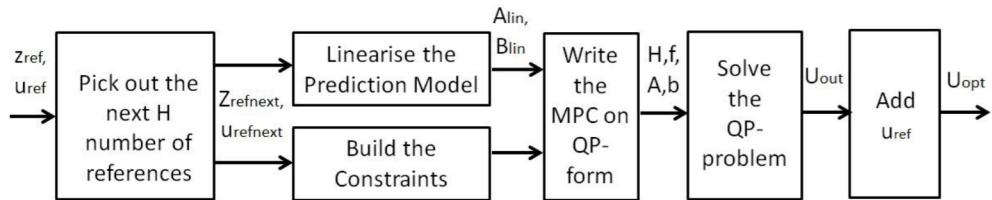


Figure 5.1: The different parts included in the MPC

MPC algorithm on QP-form is explained in the same chapter, Section 3.5. For convenience we rewrite the optimization problem below,

$$\begin{aligned} \underset{\tilde{u}}{\text{minimize}} \quad & \frac{1}{2} \tilde{u}^T H \tilde{u} + f^T \tilde{u} \\ \text{such that} \quad & D \tilde{u} \leq b \end{aligned} \quad (5.4)$$

The QP problem was solved with the built in MATLAB function quadprog, see MathWorks [2016].

In table 5.2 numerical values for the constraints are given and in table 5.3 the specifications for the vehicle to participate in the GCDC are stated.

| Parameter | Value |
|--------------------------------------|--|
| Max acceleration | 1 m/s ² |
| Min acceleration | -1 m/s ² |
| Max front wheel angle | 0.7 rad |
| Min front wheel angle | -0.7 rad |
| Max longitudinal jerk | 20 m/s ³ |
| Max change rate in front wheel angle | 3 rad/s |
| Min distance to preceding vehicle | 20 m |
| Max state error | [0.2 m, 0.2 m, 0.1 rad, 5 m/s, 5 m/s, 5 rad/s] |

Table 5.2: Numerical values for the constraints used in the MPC

Additionally, as explained in Section 3.5.1, soft constraints were added to some of the constraints. In order to do that the H and f matrices, in (5.4), have to change, according to,

$$H = \begin{bmatrix} H_{old} & Z \\ Z & I \end{bmatrix} \quad (5.5a)$$

$$f = \begin{bmatrix} f_{old} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.5b)$$

where H_{old} and f_{old} are the same as the H and f matrices used before the soft constraints were added. The size of the identity matrix, I , and the number of zeros in matrix f corresponds to the number of soft constraints added. Z , is a matrix of zeros used to keep the H matrix quadratic.

Furthermore, the soft constraints were added to the D matrix by adding columns to the matrix corresponding to the number of soft constraints. The extra columns had elements with a value of one where a soft constraint should be added and zero otherwise.

In the MPC problem, formulated in Section 3.4, the weight matrices, Q , Q_f and R are used to increase or decrease the importance of the error in the states or control signals. It may seem simple to just put weight to the different signals accordingly to which of them that are most important. But since the system is coupled it is not that simple and an increase in one state or control weight affects the other states as well.

| Parameter | Specification | Comment |
|--|---------------------|----------------------------|
| Velocity range | [0, 90] km/h | |
| Maximal lateral acceleration | 2 m/s ² | |
| Maximal longitudinal acceleration/deceleration | ±2 m/s ² | |
| Longitudinal control accuracy | 0.5 m/s | |
| Lateral control accuracy | 0.2 m | |
| Minimum lateral acceleration | 2 m/s ² | For manual driving |
| Minimum longitudinal deceleration | 6 m/s ² | For manual driving |
| Maximal longitudinal deceleration | 4 m/s ² | Only if an emergency occur |

Table 5.3: Specifications to be fulfilled in order to participate in GCDC

For our controller the important references to follow were the references in X and Y and the longitudinal velocity, v_x . Thus, we started with higher weights on these three states. Nevertheless, we still wanted the other signals to stay inside the constrained area and since we were using soft constraints the constraints may be violated if too much weight was added to any of the states or control signals. Hence, while checking so the signals were in the constrained area we tuned the weights to decrease the error in the position. By having a small error in the position a small error in the velocity will follow since the waypoints are spaced accordingly to the velocity. For all the simulations, where not stated otherwise, the matrices presented in table 5.4 were used.

| | | |
|--|--|--------------------------|
| $Q = \text{diag}[25, 25, 5, 15, 1, 5]$ | $Q_f = \text{diag}[50, 50, 10, 30, 2, 10]$ | $R = \text{diag}[1, 10]$ |
|--|--|--------------------------|

Table 5.4: The weight matrices used for the simulations

Finally, we chose a sample time and a horizon to use for the MPC. In order to do this we started with deciding a sample time. The control signal will be updated every sample time so we do not want the vehicle to drive too long distances before it is updated. On the other hand, a low sample time will increase the computational expenses. With a sample time of 0.05 s and a vehicle driving at 10 m/s the control signals would be updated every 0.5 m. If the speed is increased the vehicle will travel longer distances between the updates e.g. if the velocity would increase to 25 m/s the updates would come every 1.25 m instead. This mean the sample time may be needed to decrease if the vehicle is driving faster. After testing the controller with a sample time of 0.05 s we found it suitable for our simulations scenarios where the vehicle speed always was below 10 m/s.

Next we chose a horizon. The horizon tells the controller how many future time steps to take into consideration. Since we know the sample time and the vehicle velocity we can translate the number of steps to a distance. If the vehicle is driving at 10 m/s and the horizon is 10 the last predicted state will be 5 meters ahead and with a horizon of 20 it will be increased to 10 meters. We tried the controller with horizons in this area and concluded that increasing the horizon to more than 18 did not enhance the behaviour of the controller substantially.

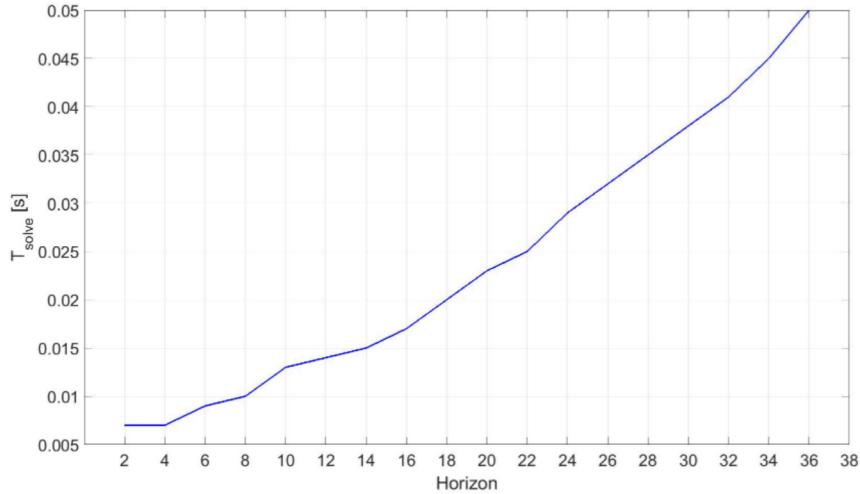


Figure 5.2: The average time for one iteration of the control loop as a function of the horizon

Furthermore, we studied the time it took to compute the control algorithm, T_{solve} , with different horizons. We defined T_{solve} as the time it took in mean to perform one iteration of the control loop, thus the time it took to update the state was included. Due to the way we implemented the Four Wheel model it had to be initialized each iteration and additionally it updated the states rather slow. This was time consuming and led to a high T_{solve} . We assumed that the state update would be faster in a real vehicle and used the Bicycle model in order to receive values for T_{solve} . In Figure 5.2, T_{solve}

is presented for different horizons. The figure is generated by doing 203 iterations with each horizon and take the average time of the last 200 iterations. The first 3 iterations were not included because due to MATLAB features the first iterations of a loop takes longer time.

As can be seen in the figure one iteration with a horizon of 18 took 0.02 s. We wanted the optimal control signal to be obtained in less than one sampling time, $T_s = 0.05$ s, and since $0.02 < 0.05$ we can use that horizon. With this horizon, the time for one iteration is below the sampling time with a margin of 0.03 seconds. Hence, the time to update the state could be increased some amount, compared to the Bicycle model, without resulting in a too high computational time. Furthermore, the sampling time could be decreased if needed, e.g. if the vehicle should drive at higher speeds.

Chapter 6

Result and Discussion

In this chapter the results from the simulations are presented and discussed. For validation purposes the two parameters Δ_x and Δ_y were used. The former gives information about how much the vehicle in average deviates in the longitudinal direction and the latter how much the vehicle in average deviates in the lateral direction, defined from the center of gravity of the vehicle.

For all the simulations, where not stated otherwise, the Four Wheel model of the vehicle was used.

6.1 Follow a Trajectory

The first simulations aimed to see how well the vehicle was able to follow a trajectory with no other vehicles involved. In the GCDC, the vehicle is supposed to be able to drive straight, change lane and make a turn in a T-intersection. Hence, trajectories for these different scenarios were generated, see Figure 6.1. The drive straight and lane change scenario had a constant velocity reference at 10 m/s and for the turn scenario the velocity reference started at 10 m/s and was then decreased to 6 m/s in the curve, according to Chapter 4. For all the scenarios a horizon of 18 and the weight matrices presented in table 5.4 were used.

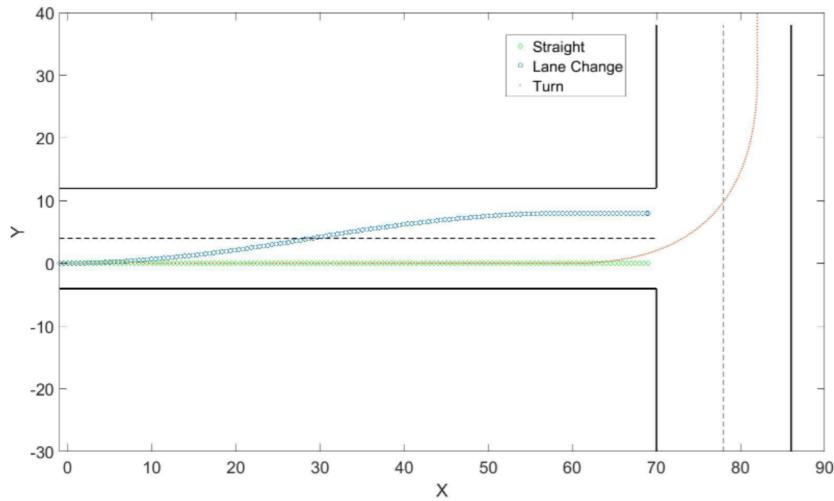


Figure 6.1: The three different trajectories used for the simulations

In Figure 6.2 the lateral deviation for all the scenarios is presented. The time steps where the lane change and turn started and ended are highlighted with arrows. It can be seen that the deviation was zero for the straight road. For the lane change and the turn the deviation was zero until the vehicle started to turn. When the vehicle was driving straight again it went back to zero after some

time. Nevertheless, the deviation is kept below 0.2 m as was specified by GCDC as an upper limit for the lateral deviation, see Table 5.3.

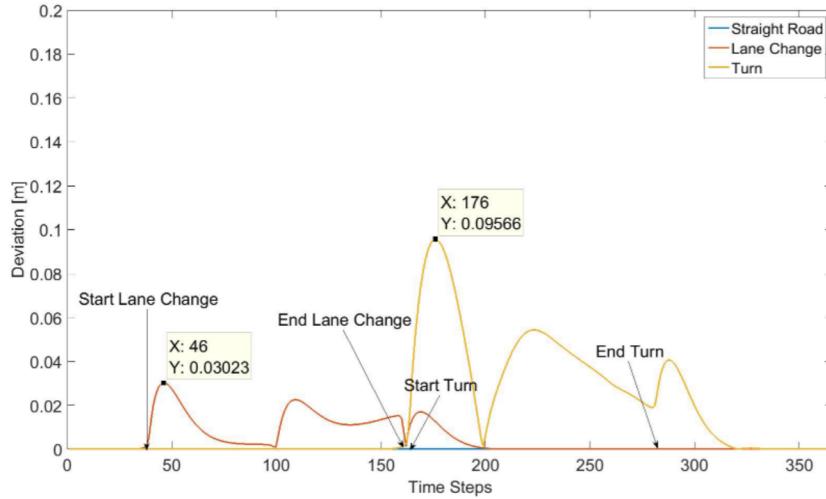


Figure 6.2

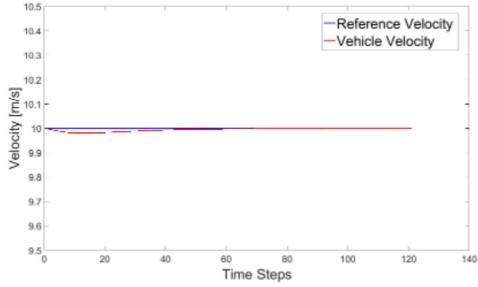
| Scenario | Δ_x | Δ_y | Δ_{xmax} | Δ_{ymax} |
|-------------|------------|------------|-----------------|-----------------|
| Straight | 0.032 m | 0.000 m | 0.035 m | 0.000 m |
| Lane Change | 0.033 m | 0.002 m | 0.035 m | 0.030 m |
| Turn | 0.058 m | 0.005 m | 0.181 m | 0.096 m |

Table 6.1: The mean and max deviation from the trajectories

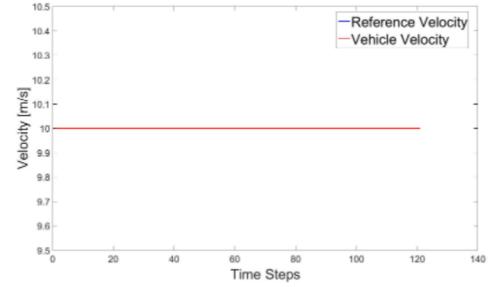
In Table 6.1 the mean and max deviation from the position references (X_{ref}, Y_{ref}), both in the lateral and longitudinal direction are demonstrated. When the average deviation for the lane change and the curve were calculated the straight segments before and after the lane change and turn, respectively, were not included. From the table we can conclude that the longitudinal deviation is bigger than the lateral deviation for all the scenarios. This has two main reasons. The first reason has to do with the generation of the waypoints. We chose waypoints to be separated by a distance corresponding to the travelled distance during one time step. However, since the path we generated the waypoints from was discrete there may not be a point exactly at that position. This lead to a deviation in the longitudinal direction if the vehicle is following the velocity well. The second reason is the tuning of the matrices. Since the specifications from GCDC were on the lateral deviation and the velocity deviation we kept these deviations low at the cost of an increased longitudinal deviation.

In the specifications from GCDC, see Table 5.3, the deviation from the velocity reference was constrained to below 0.5 m/s. To see if this was fulfilled we looked at the velocity during the scenarios. We will start by demonstrating how well the velocity and acceleration references were followed for the simulations with the straight road. In Figure 6.3 the acceleration and velocity for both the Bicycle model and the Four Wheel model, while driving straight, is demonstrated. It should be noted that the acceleration is not the actual acceleration of the vehicle but the acceleration output from the MPC. For the Bicycle model the vehicle followed the velocity well and the acceleration output from the MPC was zero as the acceleration reference and the actual acceleration of the vehicle (it had constant speed). Nevertheless, if we look at the acceleration output from the MPC for the Four Wheel model we can see that it had a value above zero even though the actual acceleration of the vehicle is zero, it had constant speed.

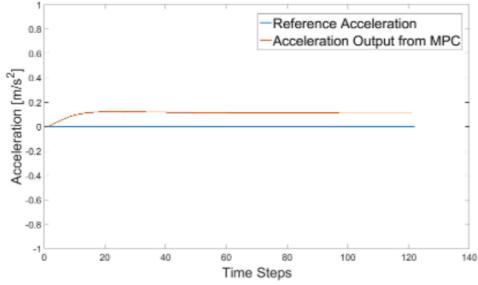
The reason to this behaviour is the slowness in the cruise controller. If the Four Wheel model was



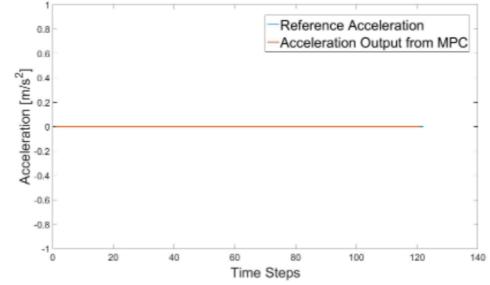
(a) Velocity, Four Wheel model



(b) Velocity, Bicycle model



(c) Acceleration, Four Wheel model



(d) Acceleration, Bicycle model

Figure 6.3: Demonstrating the difference between the velocity and the velocity references and the difference between the acceleration reference and output acceleration from the MPC for the two models

initialized with a reference velocity that was the same as the start velocity the velocity first decreased a little due to the environmental forces. The cruise controller compensated for this decrease and after a while the velocity was constant at the reference velocity. However, the time it took for the cruise controller to compensate for the environmental forces was longer than the sampling time used for the control loop, $T_s = 0.05$ s, and since the model was initialized at each iteration the reference velocity never would have been reached if the acceleration was zero. Hence, to compensate for the longer time it took for the cruise controller to react the acceleration reference from the MPC needed to be above zero.

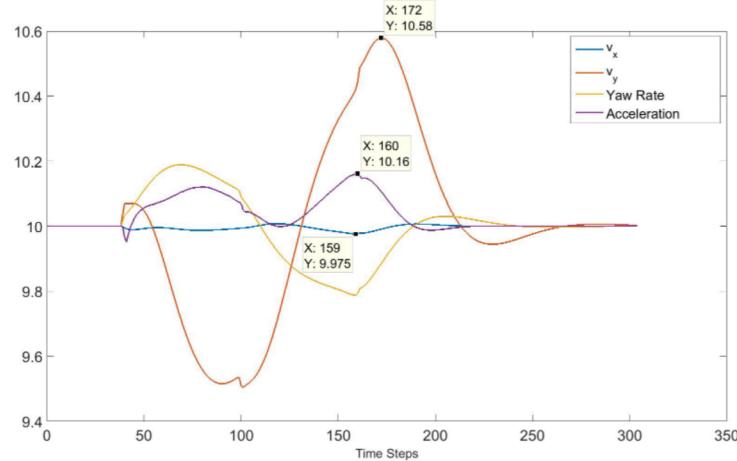


Figure 6.4: v_x , v_y , the yaw rate and the acceleration for a lane change. The yaw rate, acceleration and v_y is moved from their center around 0 to a center around 10

Next we looked at the velocity and the acceleration for the lane change. The vehicle had two velocity states, one in the longitudinal direction, v_x , and one in the lateral direction, v_y . The references for v_y were zero for all the scenarios. Nevertheless, it was seen that v_y had values different from zero when the vehicle was turning. In Figure 6.4 the velocities v_x and v_y are demonstrated together with the yaw rate and the acceleration when the vehicle was doing a lane change. Since the acceleration output from the MPC was above zero for constant speed when the Four Wheel model was used, as discussed above, we used the Bicycle model to generate the figure since it demonstrated the situation better. However, the same reasoning holds for the Four Wheel model. Furthermore, a , v_y and the yaw rate, $\dot{\psi}$, are moved so they have their center around 10 instead of 0 in order to easier compare the signals.

To understand what happened in Figure 6.4 we looked at how the velocities in the longitudinal and lateral direction were updated, equation 5.2d and 5.2e. We can see that if $mv_x\dot{\psi} \neq 2(F_{cf}\cos(\delta) + F_{cr})$ the velocity v_y will increase or decrease. This is what happened when the vehicle turned. Furthermore, we can see, equation 5.2d, that a velocity in the lateral direction of the vehicle adds a term, $mv_y\dot{\psi}$ to the update of v_x . By looking at Figure 6.4 we can see that $\dot{\psi}$ is negative when v_y is positive and vice versa. Thus, the term added to the update of v_x is always negative and the acceleration is positive in order to compensate for the decrease in v_x due to v_y . Nevertheless, the influence of the velocity in the y direction was small and the acceleration did not increase with more than 0.16 m/s^2 and the deviation in v_x had a maximum value of 0.025 m/s which is below the maximum deviation allowed at 0.5 m/s .

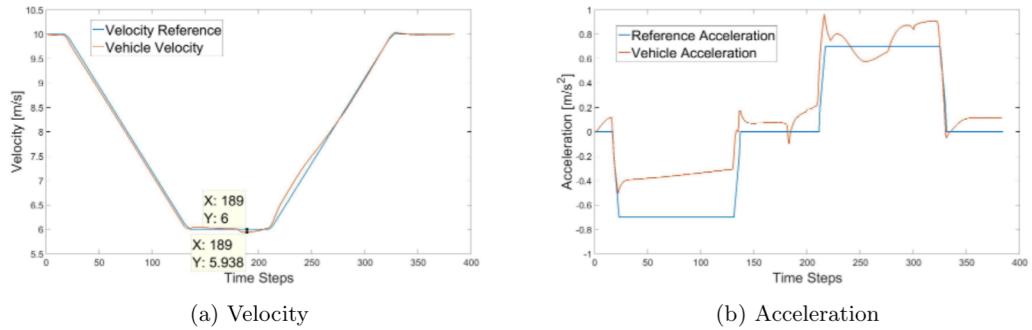
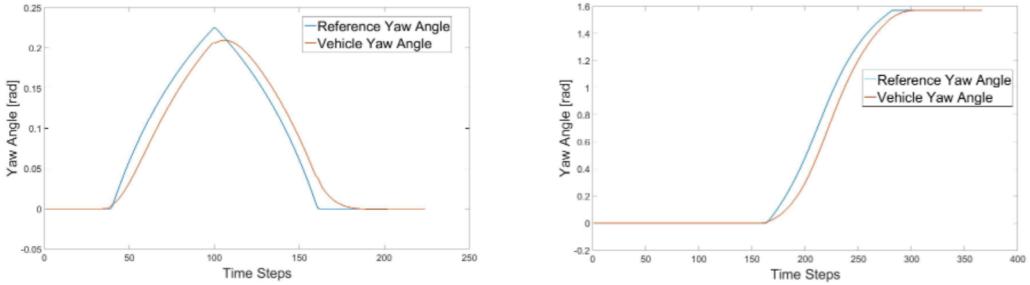


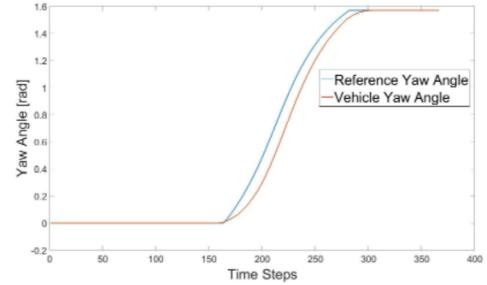
Figure 6.5: The velocity of the vehicle and the output acceleration from the MPC when the vehicle made a turn

We have now demonstrated that the velocity deviation is below 0.5 m/s for the first two scenarios. The velocity and acceleration reference for the third scenario, the lane change, is demonstrated in Figure 6.5. This figure is from simulations with the Four Wheel model and we can see that the cruise controller affect the output acceleration from the MPC as explained beforehand. We decided to put the acceleration reference maximum (minimum) to 0.7 m/s^2 (-0.7 m/s^2) so it could increase or decrease some amount, if needed, without violating the constraint on the maximum output acceleration from the MPC at 1 m/s^2 (-1 m/s^2). As in the previous scenarios the deviation in velocity is below 0.5 m/s with a maximum deviation of 0.062 m/s .

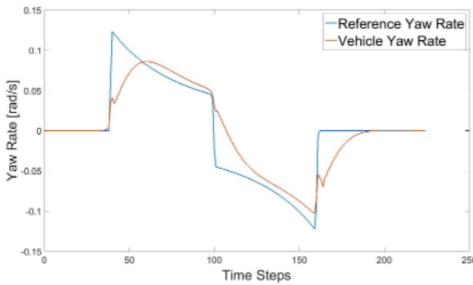
Finally, we will demonstrate how the vehicle followed the references on the yaw angle, yaw rate and wheel angle. For the straight scenario the references for yaw angle, yaw rate and wheel angle equals zero and the controller made the vehicle follow these references well. For the two other scenarios the yaw angles and yaw rates are presented in Figure 6.6 and the front wheel angle for both scenarios is presented in Figure 6.7. Both figures are generated by simulations with the Four Wheel model. As can be seen in the figures the signals deviated from the references. This was expected since the references were generated by specifying the road the vehicle should follow and the velocity it should have. The other references were calculated in order to make the vehicle follow the road and the velocity references. Nevertheless, these calculated references were estimates and the vehicle will react differently than predicted by the model. Hence, in order to follow the waypoints and the velocity these references could not be followed exactly.



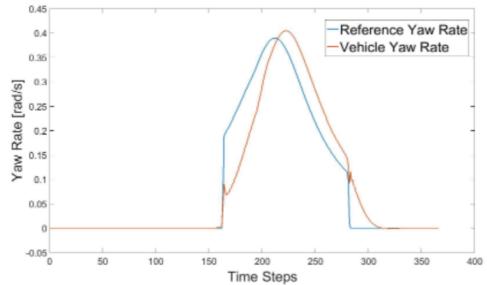
(a) The yaw angle for the lane change



(b) The yaw angle for the turn

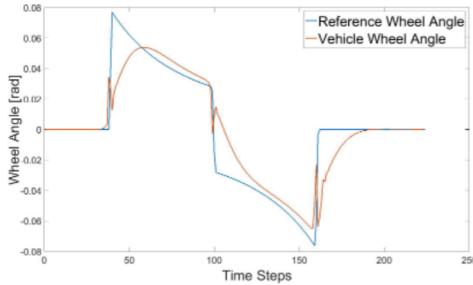


(c) The yaw rate for the lane change

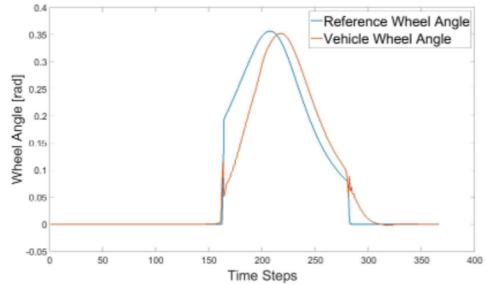


(d) The raw rate for the turn

Figure 6.6: The references and the vehicle yaw angle and yaw rate for both the lane change and the turn



(a) The front wheel angle for the lane change



(b) The front wheel angle for the turn

Figure 6.7: The references and vehicle front wheel angle, δ , for both the lane change and the turn

6.2 Handling of Disturbances and Noise

In the previous section we assumed exact measurements and no disturbances. If the controller should be implemented in a real vehicle this assumptions will not hold. Due to this we decided to introduce disturbances and noise to the controller in order to see how it could deal with it.

To simulate a disturbance we let the vehicle start away from the start position in the lateral direction. The controller behaviour for this scenario is applicable also to scenarios where the vehicle deviates from the trajectory during the drive, for example due to a strong side-wind.

The straight trajectory with a velocity of 10 m/s and a horizon of 18 were used for the simulations. To define how well the controller behaved we looked at the time it took to reach the correct position in the lateral direction. Furthermore, we looked at the behaviour of the controller while going back to the correct position, i.e. the path it took back to the trajectory. It was concluded that in order to find its way back to the trajectory the error could be maximum 1.3 m.

In order to see how the state weight matrices influenced the behaviour of the controller the weights

| Q | Q_f | R | Error in start position | Time to trajectory |
|-----------------------------|-----------------------------|-------------|-------------------------|--------------------|
| diag[25, 25, 1, 15, 1, 1] | diag[50, 50, 2, 30, 2, 2] | diag[1, 10] | 1.0 m | 4.15 s |
| diag[25, 25, 1, 15, 1, 1] | diag[50, 50, 2, 30, 2, 2] | diag[1, 10] | 1.2 m | 4.10 s |
| diag[25, 25, 1, 15, 1, 1] | diag[50, 50, 2, 30, 2, 2] | diag[1, 10] | 1.3 m | 3.80 s |
| diag[25, 25, 5, 15, 1, 5] | diag[50, 50, 10, 30, 2, 10] | diag[1, 10] | 1.0 m | 4.20 s |
| diag[25, 25, 5, 15, 1, 5] | diag[50, 50, 10, 30, 2, 10] | diag[1, 10] | 1.2 m | 4.35 s |
| diag[25, 25, 5, 15, 1, 5] | diag[50, 50, 10, 30, 2, 10] | diag[1, 10] | 1.3 m | 4.20 s |
| diag[25, 25, 20, 15, 1, 20] | diag[50, 50, 40, 30, 2, 40] | diag[1, 10] | 1.0 m | 4.40 s |
| diag[25, 25, 20, 15, 1, 20] | diag[50, 50, 40, 30, 2, 40] | diag[1, 10] | 1.2 m | 4.45 s |
| diag[25, 25, 20, 15, 1, 20] | diag[50, 50, 40, 30, 2, 40] | diag[1, 10] | 1.3 m | 4.50 s |

Table 6.2: The matrices used to test the influence of the weight when the vehicle is started away from the start position. The time to trajectory is the time it took for the vehicle to reach the correct lateral position

on ψ and $\dot{\psi}$ were varied, state 3 and 6, respectively. In table 6.2 the different weight matrices are presented together with the time it took to reach the correct lateral position when the vehicle was started at different distances from the start position. In Figure 6.8 the path the vehicle took back to the trajectory when it started 1.3 m away from the correct start position and with the different values of ψ and $\dot{\psi}$ is presented. For the three cases in Figure 6.8 the diagonal elements of the control weight matrix, R , equalled [1, 10] and the terminal state weight matrix, Q_f had values twice as big as the state weight matrix, Q .

It can be concluded that when the weight on ψ and $\dot{\psi}$ was decreased the vehicle reached the reference trajectory faster but to the cost of an increased error in the heading and yaw rate.

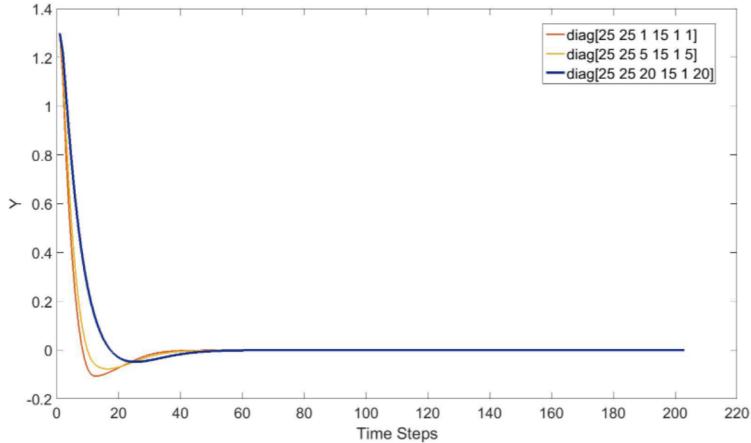


Figure 6.8: The path the vehicle followed back to the trajectory after been started 1.3 m from the correct start position, in the lateral direction

Next, the controller was tested with added noise. We decided to use white noise with different standard deviations and add it to the position updates in order to simulate that the MPC received position measurements with noise. We chose white noise since it is uncorrelated. As for the simulations with a disturbance we used the straight trajectory and a velocity of 10 m/s. The horizon was 18 and the matrices as presented in Table 5.4.

In Table 6.3 the average deviation in the lateral and longitudinal direction with the added noise is

presented. The deviation is received by taking the average of the deviation from five runs with noise with the specified standard deviation. As can be seen the lateral deviation was small as long as the

| Standard deviation | Δ_x | Δ_y |
|--------------------|------------|------------|
| 0.1 | 0.093 m | 0.024 m |
| 0.2 | 0.188 m | 0.047 m |
| 0.3 | 0.277 m | 0.065 m |
| 0.4 | 11.012 m | 2.81 m |

Table 6.3: Deviation from the trajectory when white noise, with different standard deviation, was added to the position measurements

standard deviation was less or equal to 0.3 m. For a standard deviation of 0.4 m the deviations from the trajectory were much bigger. In Figure 6.9 the lateral deviation is demonstrated for one run with each of the standard deviations. It can be seen that the maximum value for the lateral deviation for noise with a standard deviation of 0.4 meters is 1.85 m. For this scenario the controller was able to control the vehicle back to the trajectory but for some cases with the same standard deviation the trajectory was lost. This clearly leads to big deviations in the lateral direction and hence the average lateral deviation for noise with a standard deviation of 0.4 m was big.

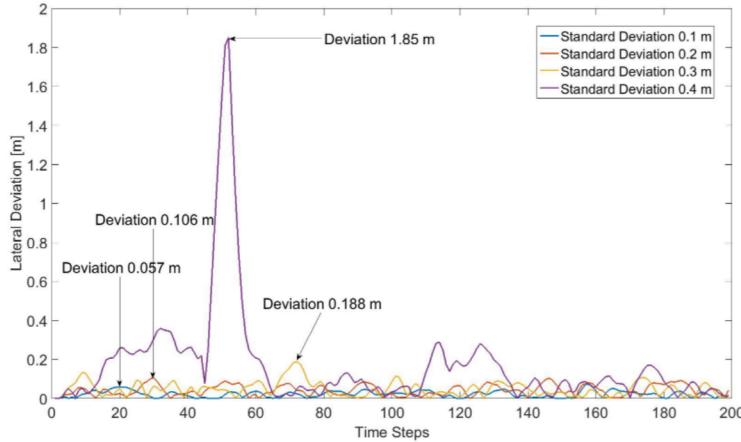
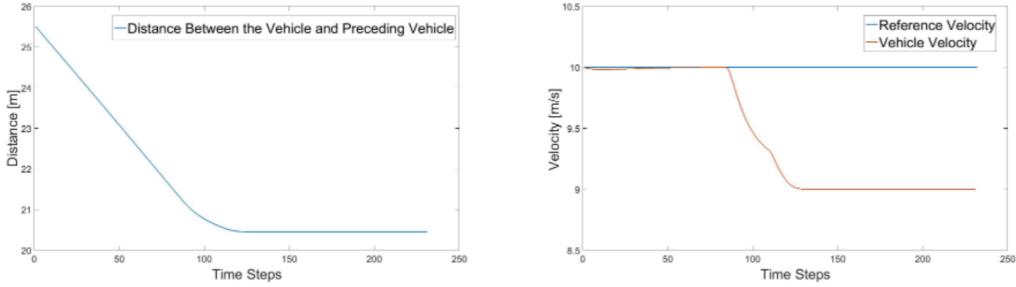


Figure 6.9: The lateral deviation of the vehicle when noise with different standard deviations was added

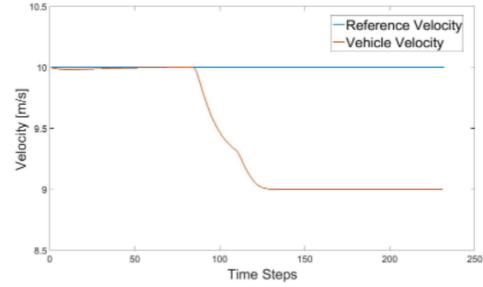
6.3 Other Vehicles Involved

Next, simulations were done to see if the vehicle could keep the safety distance to preceding vehicles. In order to validate this the vehicle was simulated together with another vehicle. The movement of the other vehicle was described by a kinematic model. Two different scenarios were used, in both the horizon was 18 and the weight matrices according to Table 5.4.

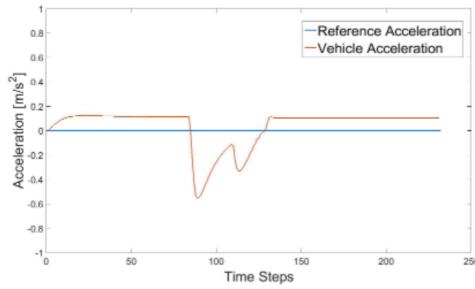
In the first test scenario the controlled vehicle approached a preceding vehicle driving at a lower velocity. In Figure 6.10 we demonstrate the distance to the preceding vehicle, the velocity of the controlled vehicle and output acceleration reference from the MPC. As can be seen in the figure the controlled vehicle started at a velocity of 10 m/s but started to decelerate in order to keep the distance to the preceding vehicle driving at a velocity of 9 m/s. It could also be concluded from the figure that the reference velocity for the vehicle stayed at 10 m/s. This was the expected behaviour since the references were generated off-line and without considering other vehicles. Nevertheless, it gives rise to



(a) The distance between the two vehicles

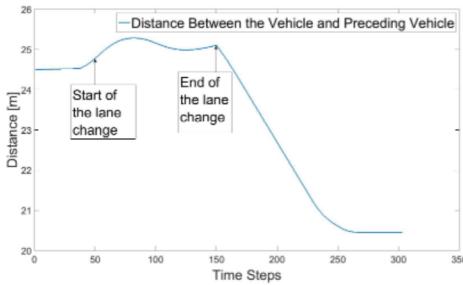


(b) The velocity of the controlled vehicle

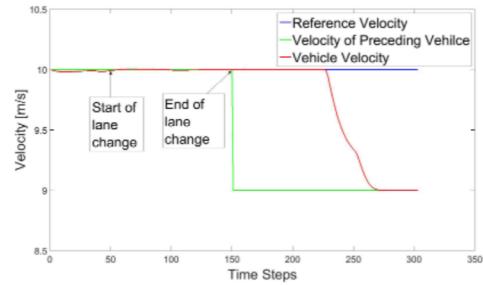


(c) The output acceleration reference from the MPC

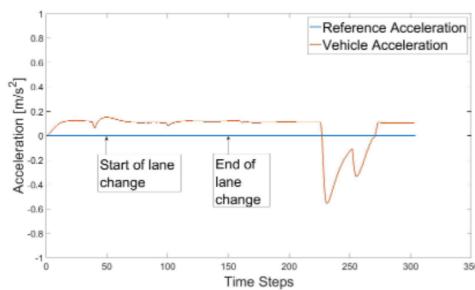
Figure 6.10: The scenario when the controlled vehicle approached a preceding vehicle with lower velocity



(a) The distance between the two vehicles



(b) The velocities of the two vehicles



(c) The output acceleration reference from the MPC

Figure 6.11: The scenario when the controlled vehicle made a lane change with a preceding vehicle in the other lane

big deviations from the references if a preceding vehicle is driving slower than the velocity reference for our vehicle. To incorporate surrounding vehicles in the reference generation and to generate the references on-line would eliminate this problem but the on-line generation of references would also increase the computational expense.

In the test scenario above the vehicles were driving on a straight road and the preceding vehicle had a constant speed. In order to demonstrate a more complicated scenario we tested a scenario where the controlled vehicle did a lane change to the left lane while the other vehicle was driving in the left lane. Furthermore, the other vehicle started at the same velocity as the controlled vehicle, 10 m/s, but decreased its velocity to 9 m/s. In Figure 6.11 the distance between the vehicles, the velocity of the vehicles and the output acceleration reference from the MPC is presented. As can be seen the controller was able to slow down the vehicle and made it keep the distance to the preceding vehicle.

6.4 Follow a Real Road

As a final test we used a part of a real road for the simulations. For this road we were given references for the position, X, Y , the velocity, v_x , and the heading, ψ . However, we needed to space the position coordinates according to the distances the vehicle travelled for one sample time. This was done according to Chapter 4 but the given position references were already spaced with a longer distance than we wanted. Hence, the references had to be interpolated before the approach in Chapter 4 could be used.

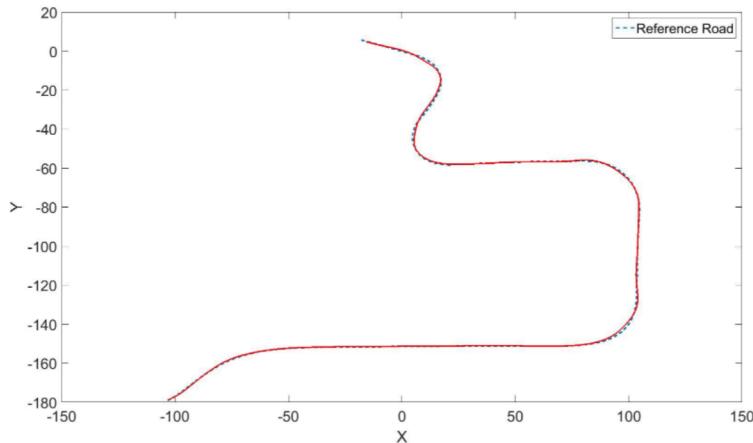


Figure 6.12: The position references and the vehicle position for the real road

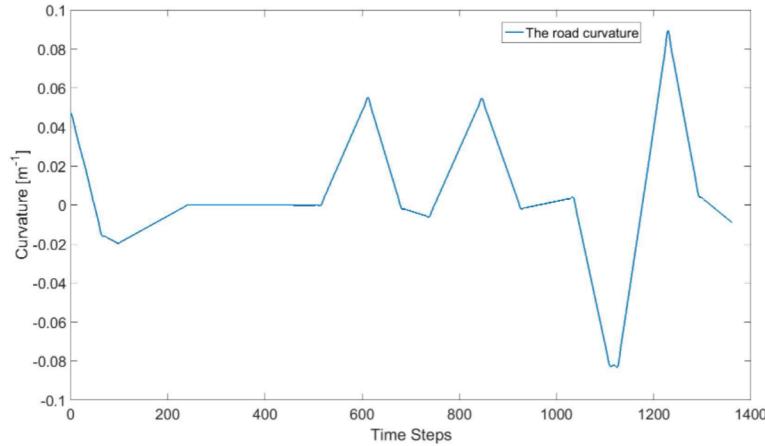
Since the real road is long and the Four Wheel Simulink model of the vehicle is rather slow we decided to test the controller with the Bicycle model. In order to make the vehicle follow the road we had to tune the weight matrices. The matrices we used are presented in table 6.4.

| | | |
|--|--|--------------------------|
| $Q = \text{diag}[40, 40, 35, 10, 1, 15]$ | $Q_f = \text{diag}[80, 80, 70, 20, 2, 30]$ | $R = \text{diag}[1, 16]$ |
|--|--|--------------------------|

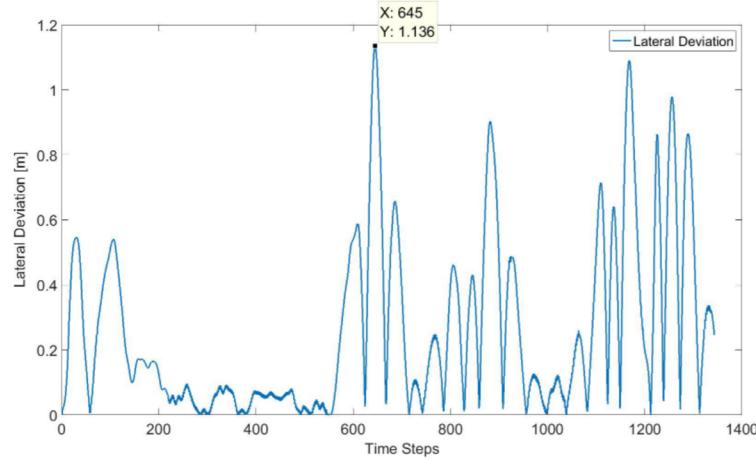
Table 6.4: The weight matrices used in the simulation with a real road

In Figure 6.12 the reference road and the position of the vehicle is demonstrated. As can be seen the vehicle was able to follow the real road. However, the deviation in the lateral direction was big, as presented in Figure 6.13.

The average deviation from the trajectory in the lateral direction, Δ_y , was 0.047 meters and in the longitudinal direction, Δ_x , it was 2.7 meters. The maximum deviation in the longitudinal direction was 3.7 meters and the maximum deviation in the lateral direction was 1.1 meters as can be seen in Figure 6.13.



(a) Curvature



(b) Lateral deviation

Figure 6.13: The curvature of the real road and the lateral deviation of the vehicle

For the real road the deviation from the references are bigger than for the generated roads used for the other scenarios. There are several reasons for this. One is that the real road has sharper turns than the generated trajectories. Another reason is the tuning of the weight matrices. We noticed an improved behaviour when the matrices were tuned but since the tuning was done mainly by trial and error it was time consuming. By using a systematic method to tune the matrices the behaviour could have been enhanced. Finally, since the real road scenario was not the main task for this thesis less time was spent on generating good references and to tune the matrices which contributed to the bigger deviation.

Chapter 7

Conclusion

The main objective of this thesis was to design an MPC for a truck participating in the GCDC. In order to participate in GCDC the vehicle should be able to autonomously perform three tasks: follow a straight road, change lane and make a turn in an intersection. It was specified that the lateral deviation should be less than 0.2 meters. The constructed MPC was able to perform all the mentioned tasks and fulfil the specifications. The maximum deviation in the lateral direction was 0.096 m and occurred during the turn scenario when the Four Wheel model was used to model the vehicle.

In order for a controller to be used in a real time scenario, as in a vehicle, the computational expenses need to be kept low. The time it takes to perform one iteration of the control loop should be less than the sampling time used for the vehicle. In this thesis we managed to construct a constrained LTV-MPC where one iteration of the control loop could be done in 0.020 s when a horizon of 18 was used. However, for the controller to be useful for implementation in real vehicles it not only has to be fast it also has to well predict the behaviour of the vehicle. To see how well our prediction model worked we used it in simulations where a more complicated vehicle model was used. The controller was able to control the more complicated vehicle system.

In addition to the above mentioned specifications the controller has to be able to deal with constraints. To control a vehicle constraints, both due to the mechanical system of the vehicle and the environment, need to be addressed. The implemented controller had constraints on the states, control signals and the distance to preceding vehicles. It was showed through simulations that the vehicle stayed in the constrained area for all the scenarios.

If the controller should be implemented to a real vehicle it needs to be able to deal with both measurement noise and disturbances. The latter could for example be a deviation from the trajectory. This could occur by a strong side-wind or if the driver manually steer the vehicle away from the trajectory. To see how the controller dealt with this type of disturbances we studied an error in the lateral direction for the start position. This scenario is comparable to the two situations mentioned above. The maximum error in the start position the vehicle could handle was 1.3 m. To see how the controller could handle noise white noise with different standard deviations was added to the position updates. The controller was able to handle noise with a maximum standard deviation of 0.3 m.

Finally, the controller was tested in a simulation with a real road. The vehicle could follow the road but the deviation both in the lateral and longitudinal direction was big. Since this was not the main task of the thesis less time was spent on decreasing the deviations. To generate the references differently or to tune the weight matrices more might improve the behaviour.

Over all, the MPC could perform the tasks stated in the objective while fulfilling the specifications on both constraints and performance. The next step would be to see how it behaves when implemented to a real vehicle.

Chapter 8

Future Work

One extension of this thesis would be to implement the controller to a real vehicle. However, testing on real vehicles is expensive and often time consuming why it would be of interest to develop the Four Wheel Simulink model and do further tests with it before implementing to a real vehicle. One aspect to improve with the Four Wheel model is the gear changing. Now the gear change happens instantly and no clutch is modelled. This could be further developed. Another improvement for the Four Wheel model would be to find a way of implementing it without initializing it for each iteration. One way of doing this would be to do the whole control loop in Simulink instead of MATLAB.

For this thesis we have chosen to use a simpler model to predict the behaviour of a more advanced system. This was done to keep the complexity of the prediction model low and hence avoid too high computational expenses. However, as seen in the results the prediction model could not predict the behaviour of the cruise controller, as expected, which made us forced to manipulate the input to the cruise controller. By adding an extra state to the prediction model the behaviour of the cruise controller could be modelled. It would be interesting to do so and compare the result regarding the computational burden and the accuracy with the prediction model used in this thesis.

As mentioned in the objective there was another project ongoing at the same time focusing on the supervisory layer of the control. To incorporate that supervisory layer with the controller derived in this thesis would be appealing. That would also include the advantages of cooperative driving to the controller, which is a focus area of the GCDC 2016.

The cooperative driving could also be addressed when generating the references. In this thesis the trajectories were generated without considering other vehicles, by considering other vehicles when the trajectories are generated the cooperation between vehicles could be improved. One way of doing this would be to generate the references inside the control loop. However, this would increase the computational expenses for the control algorithm.

Finally, the tuning of the weight matrices was done mostly by trial and error. To find a systematic way to generate good weight matrices would improve the behaviour of the controller.

Bibliography

- Assad Alam. *Fuel-Efficient Heavy Duty Vehicle Platooning*. PhD thesis, KTH School of Electrical Engineering, 2014.
- Rachid Attia, Rodolfo Orjuela, and Michel Basset. Combined longitudinal and lateral control for automated vehicle guidance. In *Vehicle System Dynamics*, volume 52, pages 261–279, 2014.
- Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control*. Springer, second edition edition, 2007.
- Jiajia Chen, Pan Zhao, Tao Mei, and Huawei Liang. Lane change path planning based on piecewise bezier curve for autonomous vehicle. In *Vehicular Electronics and Safety (ICVES), 2013 IEEE International Conference on*, pages 17–22, July 2013. doi: 10.1109/ICVES.2013.6619595.
- DARPA. Urban challenge, February 2016. URL <http://archive.darpa.mil/grandchallenge/index.html>.
- Jonas Didoff. Specification of scenarios. <http://gcdc.net/i-game>, March 2014.
- T. Keviczky J. Asgari D. Hrovat F. Borrelli, P. Falcone. Mpc-based approach to active steering for autonomous vehicle systems. In *INt. J. Vehicle Autonomous Systems*, pages 265–291, 2005.
- P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *Control Systems Technology, IEEE Transactions on*, 15(3):566–580, May 2007a. ISSN 1063-6536. doi: 10.1109/TCST.2007.894653.
- P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H.E. Tsengz. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2980–2985, Dec 2007b. doi: 10.1109/CDC.2007.4434137.
- P. Falcone, F. Borrelli, H.E. Tsengz, J. Asgari, and D. Hrovat. A hierarchical model predictive control framework for autonomous ground vehicles. In *American Control Conference, 2008*, pages 3719–3724, June 2008. doi: 10.1109/ACC.2008.4587072.
- H. Fritz. Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: experimental results from the chauffeur project. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, volume 2, pages 1348–1352, 1999. doi: 10.1109/CCA.1999.801168.
- Torkel Glad and Lennart Ljung. *Reglerteori, Flervariabla och olinjära metoder*. Studentlitteratur AB.
- i GAME GCDC. The project, September 2015. URL <http://gcdc.net/i-game>.
- Institute for Automotive Engineering IKA. Research projects from a to z, konvoi, September 2015. URL <http://www.ika.rwth-aachen.de/en/research/projects/driver-assistance-vehicle-guidance/1636-konvoi.html>.
- Anders Johansson. Iqfleet - intelligent styrning av fordon och flottor. Fordons Strategisk Forskning och Innovation, February 2014.

- T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat. Predictive control approach to autonomous vehicle steering. In *American Control Conference, 2006*, pages 6 pp.–, June 2006. doi: 10.1109/ACC.2006.1657458.
- R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R.S. Tabar, N.V. Irukulapati, C. Englund, P. Falcone, S. Papanastasiou, L. Svensson, and H. Wymeersch. Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):994–1007, Sept 2012. ISSN 1524-9050. doi: 10.1109/TITS.2012.2186513.
- R. Kianfar, M. Ali, P. Falcone, and J. Fredriksson. Combined longitudinal and lateral control design for string stable vehicle platooning within a designated lane. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1003–1008, Oct 2014. doi: 10.1109/ITSC.2014.6957819.
- Uwe Kiencke and Lars Nielsen. *Automotive Control Systems, For Engine, Driveline, and Vehicle*. Springer, second edition edition, 2005.
- J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 1094–1099, June 2015. doi: 10.1109/IVS.2015.7225830.
- Felipe Kuhne, Walter Fetter Lages, and João Manoel Gomes sa Silva Jr. Mobile robot trajectory tracking using model predictive control. In *II IEEE latin-american robotics symposium*, September 2005.
- Pamela I. Labuhn and William J. Chundrlik Jr. Adaptive cruise control, October 3 1995. URL <https://www.google.se/patents/US5454442>. US Patent 08/143,961.
- MathWorks. quadprog, January 2016. URL <http://se.mathworks.com/help/optim/ug/quadprog.html>.
- D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000. ISSN 0005-1098. doi: [http://dx.doi.org/10.1016/S0005-1098\(99\)00214-9](http://dx.doi.org/10.1016/S0005-1098(99)00214-9). URL <http://www.sciencedirect.com/science/article/pii/S0005109899002149>.
- J. Mårtensson, A. Alam, S. Behere, M.A.A. Khan, J. Kjellberg, Kuo-Yun Liang, H. Pettersson, and D. Sundman. The development of a cooperative heavy-duty vehicle for the gcdc 2011: Team scoop. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1033–1049, Sept 2012. ISSN 1524-9050. doi: 10.1109/TITS.2012.2204876.
- Per-Erik Nordström. Andra generationens euro 6-motorer med lägre bränsleförbrukning, Mars 2013.
- PATH, September 2015. URL <http://www.path.berkeley.edu/about>.
- Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, second edition edition, 2012.
- SARTE. Project final report, December 2012.
- R.R. Teetor. Speed control device for resisting operation of the accelerator, August 22 1950. URL <http://www.google.com/patents/US2519859>. US Patent 2,519,859.
- Tesla Motors Team. Your autopilot has arrived, 14 2014. URL <https://www.teslamotors.com/blog/your-autopilot-has-arrived>.
- V. Turri, A. Carvalho, H.E. Tseng, K.H. Johansson, and F. Borrelli. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pages 378–383, Oct 2013. doi: 10.1109/ITSC.2013.6728261.
- Volvo Car Corporation. Driver support, vårt bästa förarstöd, 2015. URL <http://www.volvcars.com/se/om-volvo/innovationer/intellisafe/driver-support>.

Peidong Wang, Zhenping Sun, Jun Tan, Zhenhua Huang, Qi Zhu, and Wei Zhao. Development and evaluation of cooperative adaptive cruise controllers. In *Mechatronics and Automation (ICMA), 2015 IEEE International Conference on*, pages 1607–1612, Aug 2015. doi: 10.1109/ICMA.2015.7237725.

M. Williams. Prometheus-the european research programme for optimising the road transport system in europe. In *Driver Information, IEE Colloquium on*, pages 1/1–1/9, Dec 1988.

Vincent Winstead. Autonomous lane changing using model predictive control. In *Intelligent Vehicle Initiative (IVI) Technology 2005– Advanced Controls and Navigation Systems*, April 2005.

Yunlong Ying, Tao Mei, Yan Song, and Yue Liu. A sliding mode control approach to longitudinal control of vehicles in a platoon. In *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, pages 1509–1514, Aug 2014. doi: 10.1109/ICMA.2014.6885923.

TRITA EE 2016:017

ISSN 1653-5146