EXAMENSARBETE INOM ELEKTROTEKNIK,
AVANCERAD NIVÅ, 30 HP
STOCKHOLM, SVERIGE 2016

Trajectory Planning for Autonomous Vehicles and Cooperative Driving

BENJAMIN NORDELL

KTH
SKOLAN FÖR ELEKTRO- OCH SYSTEMTEKNIK

Abstract

Autonomous vehicles have been the subject of intense research, resulting in many of the latest cars being at least partly self driving. Cooperative driving extends this to a group of vehicles called a platoon, relying on communication between the vehicles in order to increase safety and improve the flow of traffic. This thesis is partly done in context of Grand Cooperative Driving Challenge (GCDC) 2016 where KTH has participated with a Scania truck and the Research Concept Vehicle (RCV), an electric prototype car.

Trajectory planning is investigated for the longitudinal control of both the truck and the RCV. This planner is to ensure that the vehicles reached a position in a given time and a desired velocity. This is done using Pontryagin's minimum principle and interpolation.

A more advanced planner based on Model Predictive Control (MPC) is used to avoid collisions in two different scenarios. One considers obstacle avoidance in the form of an overtake and the other a lane change scenario were the vehicle needs to decide how to position itself relative to the other vehicles.

Simulations of the longitudinal control and planning of the truck did show that it could time the position and speed with a position error of less than 2m and speed error less than 0.2 m/s, assuming a distance of 120-200 m, a time interval of 40s and goal speed of 7m/s. The same simulation for the RCV had a distance error of less than 0.3m and a speed error below 0.2m.

Simulations of the RCV using MPC planners showed that overtaking and lane changes could be performed. When performing the lane change the RCV managed to maintain a longitudinal distance of at least 1m, even if the other vehicles are slowing down or increasing their speed. The overtaking could also be successfully performed although with small margins, having a lateral distance of 0.5 m to the vehicle being overtaken.

Sammanfattning

Autonoma fordon har länge varit ett intensivt forskningsområde vilket resulterat i att många av de senaste bilana är åtminstone delvis självkörande. Cooperative driving utvidgar detta till en grupp fordon som kommunicerar med varandra för att öka säkerheten och få trafiken att flyta bättre. Den här uppsatsen baseras på Grand Cooperative Divers Challange (GCDC) 2016 där KTH deltog med en Scania lastbil och en elbil kallad Research Concept Vehicle (RCV).

Rörelseplannering har undersökts för longitudinell kontrol av lastbilen och RCV. Den här planeraren ska se till att fordonet når en given position inom en viss tid och med en önskad hastighet. För detta ändamål används vad som kallas "Pontryagin's minimum principle" och interpolation.

En mer avancerad planerare baserad på Model Predictive Control (MPC) används för att undvika kollisioner i två olika situationer. Den ena simulerar en omkörning och den andra ett filbyte med flera andra fordon i den intilliggande filen.

Simuleringar av den longitudinella kontrollen av lastbilen visade att den kunde nå en position och hastighet med ett fel mindre än 2m respektive 0,2m/s då sträckor mellan 120-200m, ett tidsintervall på 40s och önskad hastighet på 7m/s används. Samma simuleringar med RCV hade ett positionsfel mindre än 0,3m och hastighetsfel under 0,2m/s.

Simuleringar med RCV då MPC används visade att omkörningar och filbyten kunde genomföras. Filbyten kunde genomföras med ett longitudinellt avstånd på minst 1m, även då övriga fordon saktar ner eller ökar farten. Omkörningar kunde också genomföras om än med små marginaler. Det laterala avståndet var 0,5m till det omkörda fordonet.

Acknowledgments

The author would like to thank examiner Jonas Mårtensson, supervisors Rui Oliveira, Lars Svensson, Stefanos Kokogias, Pedro Lima and fellow master students Jonathan Fagerström and Gonçalo Collares Pereira.

Contents

Chapter 1

Introduction

Significant progress has been made regarding the safety of vehicles, both to protect the occupants in the event of a crash and to reduce the risk of an accident occurring. This includes passive safety features such as airbags and crumble zones intended to prevent injuries caused by a collision. The increased capacity of embedded systems has made it possible to automatize an increasing number of functions, making active safety features such as lane keeping assist and stability systems more prevalent. Instead of simply protecting the occupants in the event of a crash the vehicle may prevent it from happening in the first place by assisting or warning the driver.

One approach towards increased safety could be the use of cooperative driving relying on extensive communication between vehicles. This also has the potential to improve the flow of traffic as the vehicle can drive in a well coordinated manner, having the same speed and maintaining proper distance to each other. A strictly autonomous vehicle on the other would be less dependent on the communication with other vehicles but instead require more advanced algorithms for estimation and decision making.

In the GCDC the cars drive together in three different scenarios, which are further explained below. These different scenarios serve as basis for the simulations done in this thesis. A longitudinal planner is used for testing the timing of position and speed while a more advanced planner allow the vehicle to perform more complex maneuvers without colliding with other vehicles .

1.1 Background

In the Grand Cooperative Driver Challenge (GCDC) a group of partly autonomous vehicles are suppose to drive together and handle different scenarios that requires cooperation. KTH participated in the GCDC contest with a modified Scania truck and the Research Concept Vehicle (RCV). The Scania R450 tractor unit was used to perform longitudinal control while steering was still done by a human driver. The RCV was meant to perform both longitudinal and lateral control but battery problems prevented it from participating in the contest. Instead simulations of the RCV and the truck were done for this thesis using Matlab and Simulink. A supervisory layer had been made by a group of students in a previous project which performs the tracking of vehicles and the overall decision making needed to participate in the contest.

In the GCDC the vehicles use cooperative driving, meaning that the vehicles communicate with each other to drive in different formations depending the required maneuver. The information including GPS position and speed is transmitted between the vehicles using a wireless Vehicle-to-Vehicle (V2V) communication system. The supervisory layer is used to make the basic decisions, for example when a lane changing maneuver is to be performed or when the vehicle is allowed to cross an intersection.

In the intersection scenario the vehicles are required to reach the Competition Zone (CZ) at a given time with a certain velocity. The CZ is simply the area at the intersection where there is a risk of collision as the vehicles perform their crossing maneuvers. This synchronization ensures that all vehicles start at the same position and velocity before the cooperative maneuvers begin. The longitudinal trajectory planing is used in this thesis to time the position of a vehicle, simulated using the RCV and a Scania truck.

When performing a lane change, such as the merging of two lanes performed in GCDC, a lateral trajectory is also needed. In this case the timing requirement is less strict but the trajectory planner must generate proper references values for the lateral control without causing too sharp turns. The generated trajectory for the lane change should make the maneuver as fast as possible while staying within parameters such as maximum lateral acceleration and keeping the distance to the other vehicles. A lane change is simulated for the RCV in this thesis but without the cooperative approach.

When using cooperative driving it is common to use platooning. Here a group of vehicles are driving together while maintaining the same speed and distance to each other. By driving really close to each other this could reduce fuel consumption due to the reduced air drag, in particular when using heavy duty commercial vehicles. The preferred inter vehicle distance is calculated as

$$d(t) = r(t) + h_{safe}v(t) \tag{1.1}$$

where $r(t)$ is the preferred stand still distance and h_{safe} is the headway proportional to the vehicles speed $v(t)$. This is a form of Adaptive Cruise Control (ACC) while the front vehicle maintains a reference speed with a simple Cruise Controller (CC). Although platooning is not the focus of this thesis, a proper distance keeping relative other vehicles is one of the main problems that have been investigated.

The general architecture used in GCDC is seen in Figure 1.1. The supervisory layer manages the communication with the other vehicles and co-

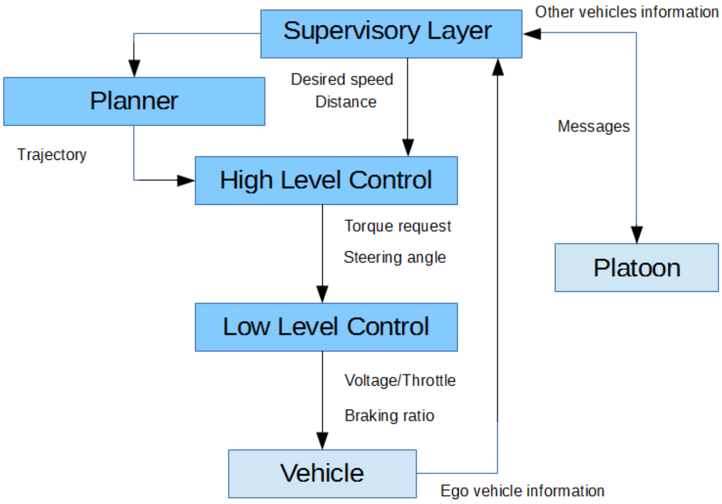

Figure 1.1 The overall structure of the control system used in GCDC and this thesis.

ordinates the different control agents. The high level controllers calculate the desired torque and steering angle, while the low level controller then applies the corresponding input to the motor and actuators. The planner is activated in the intersection and merging scenario, otherwise the reference speed and distance to the other vehicles is given by the supervisory layer. This is essentially the same architecture used for this thesis except the planner is active all the time. The low level controllers are not simulated, instead assuming that the desired torque and steering angle are always applied.

1.2 Problem Definition

In order for a self driving vehicle to safely navigate through traffic it is necessary to not only perform the basic controls but also to make proper decisions based on other vehicles positions and various traffic rules. While a manually driven vehicle has its own speed readings and more recently, the GPS position, an autonomous vehicle is very dependent on several different sensor data. As a bare minimum, it is necessary to know the ego vehicles position relative the road centerline and the distance to the other vehicles. This would be sufficient when driving on the highway when the ego vehicle can simply adjust the steering to stay in the middle of the lane and control the speed to maintain the distance to the vehicle in front.

However, many situations in traffic requires much more than simply maintaining speed and heading. The ego vehicles must be able to deter-

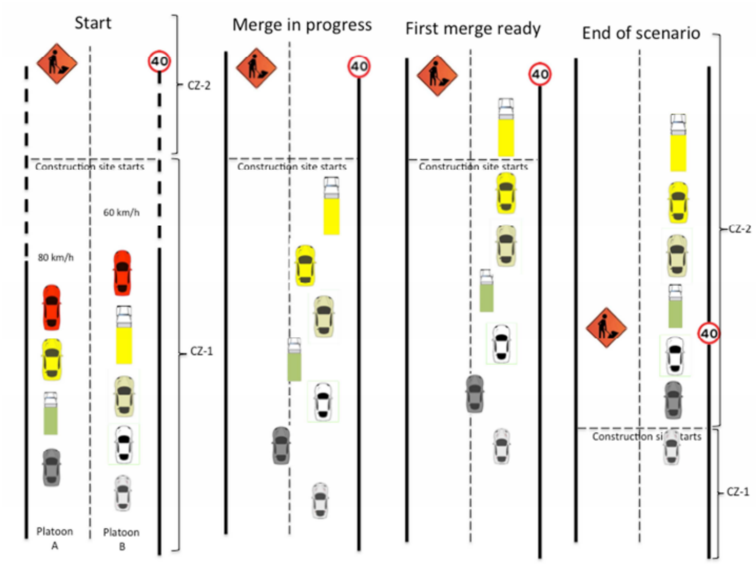

Figure 1.2 The highway scenario where the merging of two lanes is required. The vehicles in the right lane can be seen opening slots between for the other platoon to occupy. The merging is then done before the start of the construction work. [Semsar-Kazerooni et al., 2015]

mine when it is safe to change lane or drive through an intersection using information about other vehicles and the traffic infrastructure. Using cooperative driving, the vehicles solve this problem by exchanging information between each other. These messages contain not only speed and position but also intended actions of the vehicles, such as an request to change lane or the intended direction in an intersection. The vehicles may also tell each other when it is safe to perform a given maneuver.

The main drawback of cooperative driving is that a standardized protocol for communication and decision making has to be shared between the vehicle. If a vehicle is to drive in todays traffic with mostly human drivers, such a protocol would not exist. This thesis will therefore focus on a purely autonomous approach to investigate how trajectory planning can be used to navigate a vehicle through todays traffic. In this case the vehicle would need to make its own decisions using available information from its sensors and GPS data. It is assumed that the position and speed of the other vehicles can be estimated with sufficient accuracy using radar or computer vision.

1.2.1 GCDC Highway Scenario

When one lane is blocked, for example due to construction work, the cars in the left lane need to switch to the right. This requires the vehicles driving on the right lane to open the gap in order to safely perform the maneuver.

The scenario begins with two platoons of vehicles driving side by side in each lane as seen in Figure 1.2. At some point the vehicles will receive a message warning of a construction work ahead that blocks the left lane. This requires the vehicles in the left lane to change lane before reaching an imaginary line called the competition zone (CZ). In order to safely perform this maneuver the vehicle in the right lane must adapt the longitudinal distance to each other.

A vehicle that intends to merge will first adapt its distance to the vehicle in front and in the adjacent lane. Once this has been done a request to merge is sent to the rest of the platoon. The vehicle behind and in the adjacent lane then adapt its longitudinal distance to the vehicle intending to merge. When the distance is large enough a Safe to Merge (STOM) message is sent and the maneuver is performed.

1.2.2 GCDC Emergency Vehicle Scenario

In the case an emergency vehicle needs to get through, all the other cars keep to the edge of the road so that the emergency vehicle may drive along the centerline, see Figure 1.4. This scenario is not judged at the competition but was implemented as a demonstration of the effectiveness of cooperative driving.

1.2.3 GCDC Intersection Scenario

In a T-shaped intersection three different vehicles will come from their own direction and should get through without colliding, see Figure 1.3. First each vehicle needs to reach the competition zone (CZ) at a given time with a certain speed. At this point the vehicles will proceed with the crossing by keeping a virtual distance to each other, calculated by the difference in distance to a midpoint in the CZ.

1.2.4 Autonomous Lane Change

In GCDC, a merging of two lanes is done using V2V communication between the vehicles to make decisions and adapt the distance to each other. This is particularly effective in dense traffic where a good coordination between several vehicle is required. The main problem with cooperative driving is that it requires a standard communication protocol and supervisory layer for all vehicles.

However, one could also consider a scenario were this cooperation between the vehicles does not exist to the same degree as outlined above. This is a more autonomous form of driving where each decision is done independently. This scenario is shown in Figure 1.5. Rather than two lanes being reduced to one, this has a single vehicle intending to change to the adjacent lane occupied by a number of other vehicles which could be autonomous or

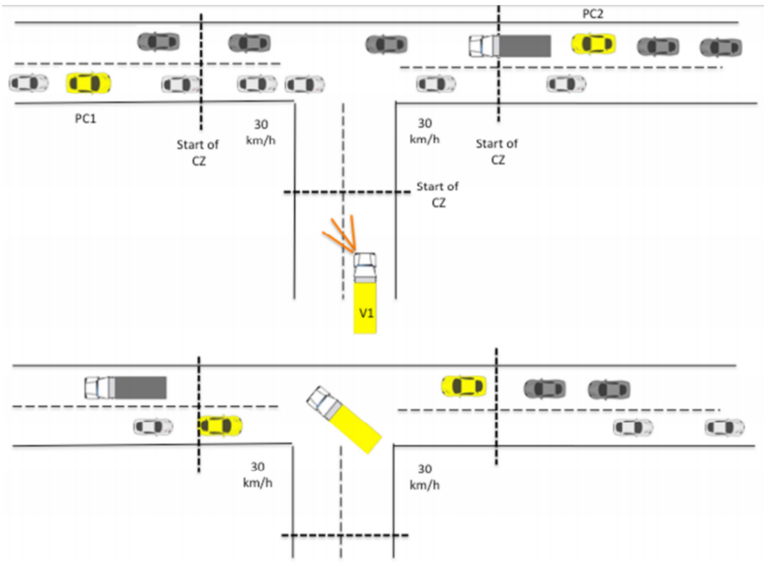

Figure 1.3 Illustration of the intersection scenario. The yellow vehicles arrive at the CZ simultaneously with the same speed. The crossing is then done by maintaining the virtual distance between the vehicles. [Semsar-Kazerooni et al., 2015]

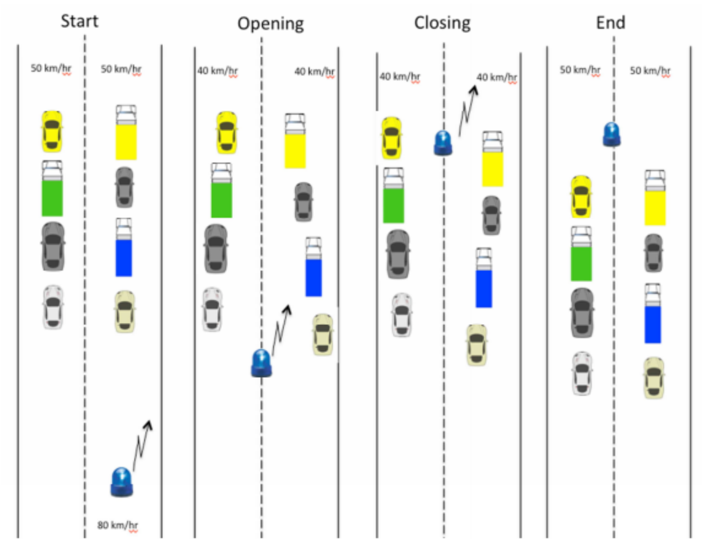

Figure 1.4 The emergency vehicle driving between the lanes. This is made possible by having the other vehicles driver closer to the road delimiter. [Semsar-Kazerooni et al., 2015]

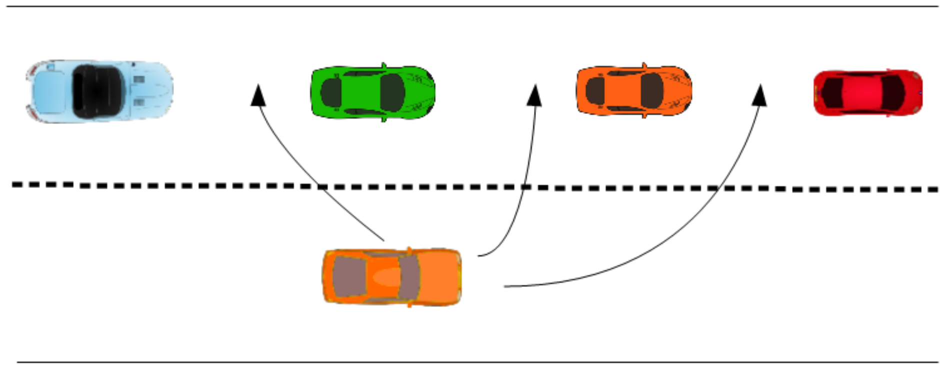

Figure 1.5 Lane change scenario where a vehicle is to find a slot between the vehicles in the adjacent lane.

human driven. Such a scenario requires more decision making by individual vehicles and less assumptions can be made by the other vehicles position and intended actions.

In this scenario the vehicle that is to perform the lane change must first decide which slot between the cars that is the most feasible. Once this has been done the car should adapt its longitudinal position so that it is adjacent to the preferred slot, then perform the lane change.

1.2.5 Autonomous Overtaking

Aside from the scenarios described for the GCDC there are many other situations that an autonomous vehicle must be able to handle. The state machine used in the supervisory layer for the GCDC cannot make decisions for an overtaking scenario, thus a decision rule must be made for that. This planner needs to take into account the speed and position of the vehicle ahead as well as those in the adjacent lane, which could have approaching traffic or not. If a collision with the opposing car is deemed too likely the car should wait behind the obstacle before going around it. This scenario would be particularly relevant in the case where one lane is being blocked by an obstacle, such as a stationary vehicle, or a very slow moving one e.g. a tractor.

1.3 Objective

The main objective in this thesis is to investigate how trajectory planning can be used in autonomous vehicles. Two kinds of planners are developed, one for longitudinal control and another one which also does lateral planning. These planners are tested using simulation of the RCV and a Scania truck. In

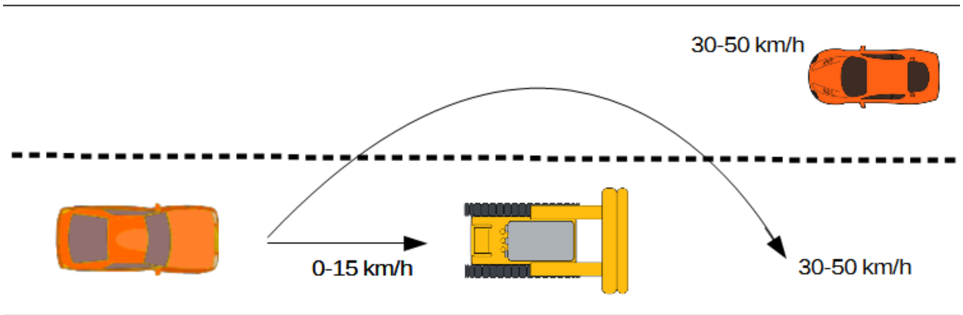

Figure 1.6 The overtaking scenario simulated in addition to the GCDC scenarios. The vehicle in the lower lane encounter a stationary or slower vehicle while another one is driving in the opposite direction.

all cases it is assumed that the speed and position are known for all vehicles.

The planner for longitudinal control is used to ensure that the vehicle can reach a given position and speed at a specific time instant. This planner should give a speed reference without excess acceleration and jerk in order to make it easy to follow.

In order to safely drive in traffic with other vehicles a more advanced planner is needed to control both the lateral and longitudinal position. This planner do not consider timing requirements but instead ensure that the ego vehicles can avoid collisions with other vehicles. The planner must be able to generate a trajectory without collision in a lane change and a overtaking scenario. Aside from the planner this also require a decision rule that can determine when the given maneuver is safe.

1.4 Related Work

A trajectory may be generated by using interpolation between different states at the given time instants. This has been used by [You et al., 2015] to generate a trajectory for a lane changing maneuver. Similarly to GCDC, this was done in a cooperative manner where the vehicles communicate with each other and share a supervisory layer. The final time instant t_f should be selected in a manner that makes the lane change quick yet ensure the comfort and safety for the occupants of the car. This final time could be calculated as the expected time interval using maximum acceleration and jerk in order to make the maneuver as fast as possible.

This would then be an approximation of a time optimal planner while allowing for recalculations in order to be more flexible for the current state of the vehicle. The motivation for using a time optimal planner in the lane changing scenario is to make the maneuver as quick as possible while still being comfortable and safe. The limits on the lateral acceleration and jerk of the vehicle can not be guaranteed with this method, however. By scaling the interpolation coefficients with maximum acceleration or jerk as done in [Erkorkmaz et al., 2011] the trajectory will stay within these constraints at the cost of increased time for the maneuver. As interpolation has been proven to work by [You et al., 2015] it could be useful for the highway scenario in GCDC.

An energy optimal planner is used to generate a trajectory from an initial state $x(t_0) = x_0$ to the final state $x(t_f) = x_f$ while minimizing the required energy. Such problems are commonly solved using Pontryagin's minimum principle as done in [Tokekar et al., 2014] to find an optimal trajectory for a car like robot, having a given starting and goal position as well as orientation. For the mobile robot the energy is calculated using the voltage and current of the DC-motors. By assuming a predefined path that the robot is to move on the problem is reduced to a one dimension for the trajectory with $x(t)$ being the distance traveled on the path and $v(t)$ the speed. Aside from the energy minimization this approach is also useful when a given state is to be achieved at a specific time instant. This is the case in the intersection scenario in GCDC which requires the vehicles to have a certain speed and position at some time instant. The simplification to one dimension can also be made here since the vehicle will follow the centerline of the lane so the trajectory only needs to give a proper velocity profile. While this could also be done using interpolation, the minimal squared acceleration ensure that the trajectory is easy to follow. The energy optimal planner could also be used for the lane changing maneuver but this does not necessarily give an acceleration equal to zero at the end point unless an additional constraint is put on the final states which increase the complexity on the calculations.

In [Falcone et al., 2008] Model Predictive Control (MPC) has been used both for trajectory planning and control of autonomous ground vehicles. A trajectory is first calculated off-line for the vehicle to follow. During the drive this trajectory can be recalculated when triggered by certain events, such as a large tracking error or pop-up obstacles. The trajectory is calculated by minimizing a cost function of deviation from the desired path and velocity as well as a penalty on the acceleration. An MPC was also used as an Active Front Steering to select the steering angle that minimizes the heading error. As noted by the authors the MPC is useful for handling the constraints on inputs and states but the computational complexity limits the use to lower speed.

A similar problem has been studied in [Farrokhsiar et al., 2013] where MPC is used for both motion planning and high level control of an autonomous robot. The trajectory here is calculated in two steps, first the nominal trajectory which the robot would ideally follow, then an auxiliary controller which adjusts the trajectory with regard to the current state of the robot. As MPC is already used for the control of the RCV in GCDC a more simple trajectory generator is preferable to limit the computational cost. It is however simulated for the autonomous lane change scenario.

As shown by [Grancharova et al., 2014] it is also possible to plan the motion of several vehicles simultaneous, in this case rotary wing UAVs. Aside from constraints on acceleration, velocity and position the trajectory planning also considered path losses for the radio communication. The latter is relevant in cases where there are significant distances between vehicles and possibly obstacles in the communication path. This should not be the case with cooperative driving as other vehicles are assumed to be close to each other, instead assuming that the communication works without any problem.

The trajectory planning of autonomous vehicles along a predefined path has also been investigated by [Li et al., 2014]. This was done using a sampled based approach to generate feasible trajectories in a dynamic environment. From the predefined path a number of final states including position (x, y), heading θ and curvature κ are sampled. A path is then generated for each final state using a numerical optimization based on the input constraints. The optimal path is then determined by a weighting function of deviation from the path, smoothness, obstacle proximity and length. The trajectory is then generated by adding a velocity profile to this path. The motivation for this approach is that a navigation system using a GPS could generate a path to a final destination but not account for vehicles and other dynamical obstacles. Thus a local trajectory planner is needed which could adjust for such problems and also generate proper velocity profiles.

1.5 Report Outline

In Chapter 2 the overall theory of the methods used for trajectory planning is explained, including MPC, PMP and interpolation. The implementation details are outlined in Chapter 3, including description of the hardware, system architecture, vehicle model and simulation setup. This is followed by the results which are discussed in Chapter 4 and the thesis ends with a conclusion and comments on future works.

Chapter 2

Trajectory Planning

Trajectory planning, also called motion planning, generates a path as a function of time. Unlike a path which is simply a sequence of coordinates (x_k, y_k) that can be followed with an arbitrary speed, the trajectory can be expressed as

$$(x_k, y_k, t_k), \quad k = 0, 1, 2...N \tag{2.1}$$

In many applications the trajectory functions as a time varying reference value for position and/or velocity being fed to a controller. Trajectory planning occur mainly in two different areas, which is industrial robots and autonomous vehicles. There are several reasons for having a trajectory rather than a constant speed reference. In many cases a strong acceleration and jerk is undesired, which may be prevented by having the trajectory within the given limits of acceleration and jerk. In addition, the trajectory planner may also function as a path finder in order to avoid obstacles in the environment. Finally, a movement may have timing requirements for the position and speed which make trajectory planning necessary.

In this thesis, a Model Predictive Control (MPC) was used for the simulation of overtaking and lane change scenario. For the longitudinal control more simple planners were selected as it was only required to control speed and position in one dimension and other vehicles were not considered.

2.1 Model Predictive Control

Model Predictive Control (MPC) is a form of optimal control, solving an optimization problem with constraints. This approach is particularly useful when the states and inputs are saturated. In the case of the autonomous vehicles this could include a maximum allowed acceleration or a minimum distance to other vehicles. The optimization problem may be formulated as

$$\begin{array}{rl} \text{Minimize} & \sum_{k=1}^{N} J(x_{t+k}, u_{t+k}) \\ \text{subj.} & x_{t+k+1} = f(x_{t+k}, u_{t+k}) \\ & g(x_{t+k}, u_{t+k}) \leq b \\ & l_x \leq x_{t+k} \leq u_x \\ & l_u \leq u_{t+k} \leq u_u \end{array} \tag{2.2}$$

Here $f(x_{t+k}, u_{t+k})$ is a discrete time state space model of the system defined as

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + v_1(t) \\ y(t) &= Cx(t) + v_2(t) \end{aligned} \tag{2.3}$$

with states $x(t)$, output $y(t)$ and input $u(t)$ while A, B, and C are matrices. The cost function $J(x_{t+k}, u_{t+k})$ that is to be minimized is

$$J(x_{t+k}, u_{t+k}) = Q_1(x_{t+k} - r_{t+k})^2 + Q_2 u_{t+k}^2 \tag{2.4}$$

where r_{t+k} is the reference value for the states while Q_1 and Q_2 are the weight matrices for the states and inputs, respectively. The process and measurement noise, v_1 and v_2, is modeled as zero mean Gaussian noise. The maximum like-hood prediction for the future states and output are then

$$\begin{aligned} \hat{x}(t+k+1|t) &= Ax(t+k|t) + Bu(t+k) \\ \hat{y}(t+k|t) &= C\hat{x}(t+k|t) \end{aligned} \tag{2.5}$$

Assuming that $C = I$ These predictions may be formulated as [Glad and Ljung, 2000]

$$\begin{aligned} \begin{bmatrix} \hat{x}(t+N|t) \\ \vdots \\ \vdots \\ \hat{x}(t+1|t) \end{bmatrix} &= \begin{bmatrix} A^N \\ \vdots \\ \vdots \\ A \end{bmatrix} x(t|t) + \begin{bmatrix} B & \dots & A^{N-1}B \\ 0 & \ddots & A^{N-2}B \\ \vdots & & \vdots \\ 0 & \dots & B \end{bmatrix} u(t) \\ &= Sx(t) + Gu(t) \end{aligned} \tag{2.6}$$

The planner assumes a simple model on the form

$$\begin{aligned} \dot{X} &= v\cos(\theta) \\ \dot{Y} &= v\sin(\theta) \\ \dot{v} &= a \\ \dot{\theta} &= \omega \\ \dot{\omega} &= u \end{aligned} \tag{2.7}$$

Here X and Y are the road coordinates while v is the speed, θ the heading and ω the yaw rate. The linear and angular acceleration a and u is treated as input. Since the input are integrated twice they may be zero for most of the prediction horizon while still causing a change in the system states. The amount of prediction steps M for the input can then be much fewer than for the output, thus reducing the computational cost. In order to use MPC the state space model needs to be linearized and transformed to discrete time using Euler forward and Taylor approximation. The state space model used is then

$$\begin{aligned} X(k+1) &= X(k) + T_s[v(k)\cos(\theta_0) - \Delta\theta(k)v_0\sin(\theta_0)] \\ Y(k+1) &= Y(k) + T_s[v(k)\sin(\theta_0) + \Delta\theta(k)v_0\cos(\theta_0)] \\ v(k+1) &= v(k) + T_s a \\ \Delta\theta(k+1) &= \Delta\theta(k) + T_s\omega \\ \omega(k+1) &= \omega(k) + T_s u \end{aligned} \tag{2.8}$$

where v_0 and θ_0 are the point of linearizion and $\Delta\theta(k) = \theta(k) - \theta_0$.

MPC As a QP Problem

By having all predicted states and inputs in a vector z this can be formulated as the Quadratic Problem (QP)

$$\begin{aligned} \text{Minimize} \quad & zHz^T \\ \text{subj.} \quad & A_{eq}z = b_{eq} \\ & Az \leq b \\ & z_{min} \leq z \leq z_{max} \end{aligned} \tag{2.9}$$

which may be solved with numerical algorithms such as the simplex method. Having the vector z on the form

$$z = [x_{k+1} - r_{k+1} \dots x_{k+N} - r_{k+N} \; u_k \dots u_{k+M}]^T \tag{2.10}$$

the equality constraint yields

$$\begin{aligned} b_{eq} &= -Sx(t) \\ A_{eq} &= [-I \; G] \end{aligned} \tag{2.11}$$

The Hessian matrix H is selected to give the cost function $\sum_{k=1}^{N} J(x_{t+k}, u_{t+k})$. The saturation on z limits the speed, acceleration and ensure that vehicle maintains a proper distance and stays on the road. The inequality constraint is used in the overtaking scenario to force the vehicle to the other lane and go around the obstacle. This can be done by having the inequality constraint

$$X_{k+i} - \frac{d_{safe}}{d_{lane}} Y_{k+i} \leq X_{obstacle} - d_{safe}, \; i = 1, 2 \dots, N \tag{2.12}$$

where $X_{obstacle}$ is the position of the obstacle, d_{safe} the distance behind the obstacle where it should start performing the overtaking and d_{lane} is the width of the lane. This constraint can be visualized as a straight line going diagonally past the obstacle into the adjacent lane, see figure 2.1.

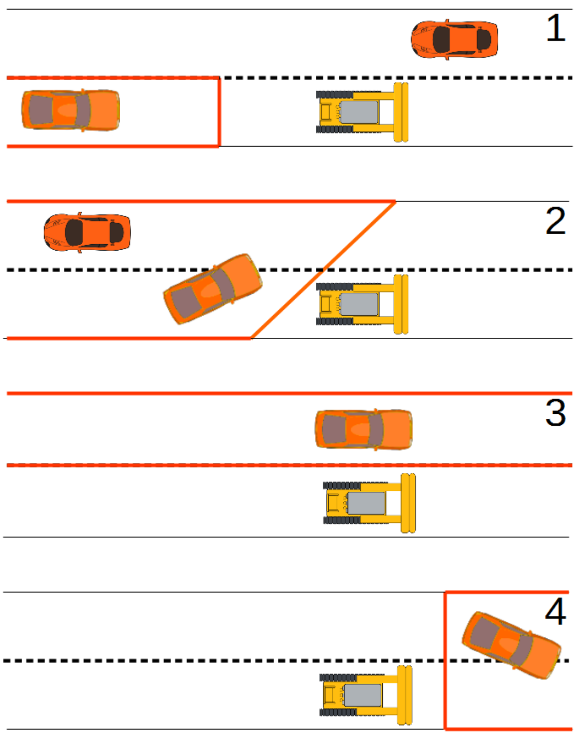

Figure 2.1 Illustration of the constraints on position, shown in red, when an overtaking is performed. 1: While waiting for the opposing vehicle to pass, the ego vehicle is contained in its own lane behind the obstacle. 2: When overtaking the obstacle these constraints are extended in order to guide the ego vehicle to the opposing lane. 3: While adjacent to the obstacle the ego vehicle is confined to the opposing lane. 4: Having gained enough longitudinal distance the ego vehicle may return to its own lane.

2.1.1 Generating the Constraints

The constraints on the ego vehicle's position are necessary in order to avoid collision and to keep the vehicle on the road. As seen in figure 2.1 these constraints are adapted to four different situations when an overtaking is performed:

1. If it is determined that an overtaking should not be done, the limits in the lateral and longitudinal position takes the form of a rectangle. The ego vehicle will then be confined in its own lane behind the obstacle.

2. When an overtaking is deemed to be safe, the constraint on the lateral position is expanded to include the entire road. In addition, the inequality constraints on the position forms a diagonal line behind the obstacle. Since the ego vehicle tries to maintain the reference speed it will drive to the opposing lane which allow it to drive further.

3. As son as the ego vehicle is in the opposing lane the constraints on the longitudinal position are removed to allow the ego vehicle to drive past the obstacle. Only the lateral constraints are used in order to avoid collision.

4. Having gained enough longitudinal distance to the obstacle the lateral constraints are expanded to its own lane. Since the reference set point for the lateral position is in its own lane, the ego vehicle will complete the overtaking by steering back to said lane.

As for the lane change scenario, the position constraints takes two different forms as seen in figure 2.2. The inequality constraint is not used here, only box constraints on the position.

1. While the longitudinal distance to the other vehicles are too small the ego vehicle will be confined to one lane. In this case the controller will adapt the longitudinal position to get the ego vehicle properly positioned between the vehicles since the set point for the position is in the middle of the available slot.

2. Having adapted the longitudinal position, the constraints on the lateral position is expanded to the whole road. Combined with the longitudinal constraints this forms a box around the ego vehicle to make sure that no collision occurs.

2.2 Longitudinal Planner

The longitudinal planner is used to time speed and position of the vehicle. It is simulated separately from the scenarios using MPC and does not take

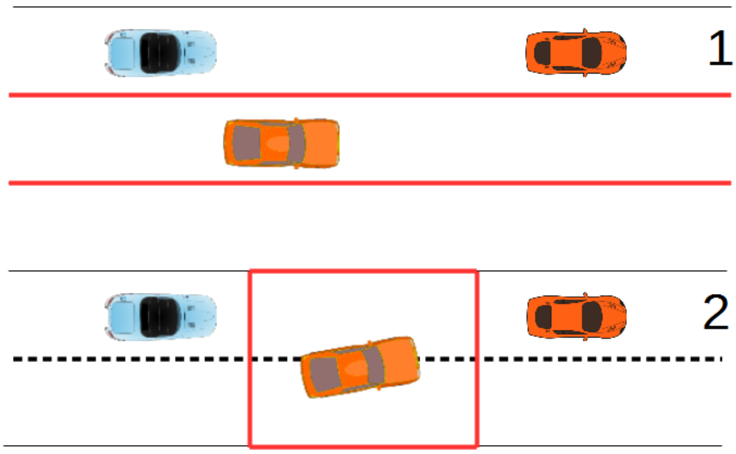

Figure 2.2 The constraints shown in red when performing a lane change. 1. The ego vehicle is confined to its own lane when adjacent to another vehicle. 2. Having adapted the longitudinal position th ego vehicle may perform the lane change.

the lateral position into account. Two different planers are considered in this thesis for longitudinal planning. One is based on Pontryagin's Minimum Principle (PMP) in order to minimize the required acceleration. A more straightforward method using interpolation is also used as a comparison. The interpolation method was initially intended for the lateral planning in the highway scenario used in GCDC but since the RCV could not participate in the contest it is instead compared with PMP.

2.2.1 Pontryagin's Minimum Principle

An energy optimal planner is used to generate a trajectory from an initial state $x(0) = x_0$ to the final state $x(t_f) = x_f$ while minimizing the required energy. Aside from the energy minimization this approach is also useful when a given state is to be achieved at a specific time instant. This is the case in the intersection scenario in GCDC which requires the vehicles to have a certain speed and position at some time instant. While this could also be done using interpolation, the minimal squared acceleration ensures that the trajectory is easy to follow.

The optimization problem is formulated as [Tokekar et al., 2014]

$$\begin{array}{rl} \min & \int_0^{t_f} f_0(x,u)dt \\ \text{subj.} & \dot{x} = f(x,u) \\ & x(0) = x_0 \\ & x(t_f) = x_f \end{array} \tag{2.13}$$

where the input u is the acceleration a and states $x = [p \quad v]^T$ are the position p and velocity v. By choosing $f_0(x,u) = u^2$ the square acceleration may be minimized. In the one dimensional case the vehicle dynamics $f(x,u)$ is modeled with a simple continuous time state space model as

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2.14}$$

To find the optimal input the Hamiltonian function is defined as

$$H(\lambda, x, u) = f_0(x,u) + \lambda^T f(x,u) = u^2 + \lambda_1 v + \lambda_2 u \tag{2.15}$$

where $\lambda = [\lambda_1 \quad \lambda_2]$ are the Lagrangian multipliers. These are defined by the differential equations

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = \begin{bmatrix} 0 \\ -\lambda_1 \end{bmatrix} \tag{2.16}$$

Since $\dot{\lambda}_1 = 0$ and therefore $\lambda_1 = c_1$ the solution for λ_2 is

$$\lambda_2 = -\lambda_1 t + c_2 = -c_1 t + c_2 \tag{2.17}$$

where c_1 and c_2 are unknown constants. The optimal control signal u^* is then found as the solution to

$$u^* = \underset{u}{\operatorname{argmin}} H(\lambda, x, u) = \underset{u}{\operatorname{argmin}} [u^2 + \lambda_1 v + \lambda_2 u] \tag{2.18}$$

which is given by

$$u^* = -\frac{\lambda_2}{2} = \frac{c_1 t - c_2}{2} \tag{2.19}$$

Inserting the optimal input into the state space model yields

$$\begin{aligned} \dot{p} &= v \\ \dot{v} &= \tfrac{c_1 t - c_2}{2} \end{aligned} \tag{2.20}$$

By integrating the acceleration to get the speed difference

$$v_f - v_0 = \int_0^{t_f} \frac{c_1 t - c_2}{2} = \frac{c_1 t^2}{4} - \frac{c_1 t}{2} \tag{2.21}$$

the parameter c_2 is obtained as

$$c_2 = \frac{2}{t_f} \left(\frac{c_1 t_f^2}{4} - v_f + v_0 \right) \tag{2.22}$$

In order to find c_1, the acceleration is integrated again to get the position as

$$p_f = v_0 t_f + \frac{c_1 t_f^3}{12} - \frac{c_2 t_f^2}{4} \tag{2.23}$$

Substituting 2.22 into 2.23 then yields the solution

$$c_1 = -12\frac{-(v_f - v_0)t_f - 2t_f v_0 + 2p_f}{t_f^3} \tag{2.24}$$

2.2.2 Interpolation Methods

A trajectory can be generated by using interpolation between different points at the selected time instant. This has been used by [You et al., 2015] to generate a trajectory for a lane changing maneuver. This interpolation could be done between the initial states

$$S_0 = (x_0, \dot{x}_0, \ddot{x}_0, y_0, \dot{y}_0, \ddot{y}_0) \tag{2.25}$$

and the final states

$$S_f = (x_f, \dot{x}_f, \ddot{x}_f, y_f, \dot{y}_f, \ddot{y}_f) \tag{2.26}$$

where x and its time derivatives is the longitudinal position, velocity and acceleration while y is the lateral equivalents. The longitudinal and lateral trajectories can be calculated separately assuming two different polynomials for the positions on the form

$$\begin{cases} x(t) = \sum_0^N a_i t^i \\ \\ y(t) = \sum_0^M b_i t^i \end{cases} \tag{2.27}$$

where N and M are the order of the polynomials while a_i and b_i are the coefficients. The time derivatives of these functions then give the velocities as

$$\begin{cases} f(\dot{x}, t) = \frac{d\sum_0^N a_i t^i}{dt} \\ \\ f(\dot{y}, t) = \frac{d\sum_0^M b_i t^i}{dt} \end{cases} \tag{2.28}$$

Similarly, the acceleration is given by a second derivation. By, for example, using $N = M = 6$ the longitudinal trajectory may be calculated as the least square solution to the equation

$$\begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix} = \begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0 & 2 & 0 & 0 \\ t_f^5 & t_f^4 & t_f^3 & t_f^2 & t_f & 1 \\ 5t_f^4 & 4t_f^3 & 3t_f^2 & 2t_f & 1 & 0 \\ 20t_f^3 & 12t_f^2 & 6t_f & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} \tag{2.29}$$

and similarly for the lateral trajectory. If used to perform a lane change maneuver the final states for the lateral trajectory would be equal to the centerline of the adjacent lane while the lateral speed and acceleration should be zero. In this case final time instant t_f should be selected in a manner that makes the lane quick, yet ensures the comfort and safety for the occupants of the car. This final time could be calculated as the expected time interval for the maximum acceleration, maximum jerk planner. However, in this thesis the interpolation is used for the same purpose as the PMP.

2.2.3 Constant Acceleration planner

Since the planner described above was not ready for implementation at the time the competition was held, a more simple planner was used instead. This planner calculates the constant acceleration required to reach the final velocity and then keep the v_f for the time required t_f travel the distance p_f. This acceleration $a(t)$ may be calculated as

$$\begin{aligned} p_f = & \frac{v_f T_1}{2} + v_f T_2 = \frac{v_f^2}{2a(t)} + v_f(t_f - \frac{v_f}{a(t)}) \\ \Rightarrow a(t) & = \frac{v_f^2}{2(v_f t_f - p_f)} \end{aligned} \tag{2.30}$$

Chapter 3

Implementation and Simulation

,

This chapter describes how the simulations were done in Matlab and Simulink. First, the vehicle models used in the simulations are described followed by the implementation of the planners.

3.1 Modeling the RCV

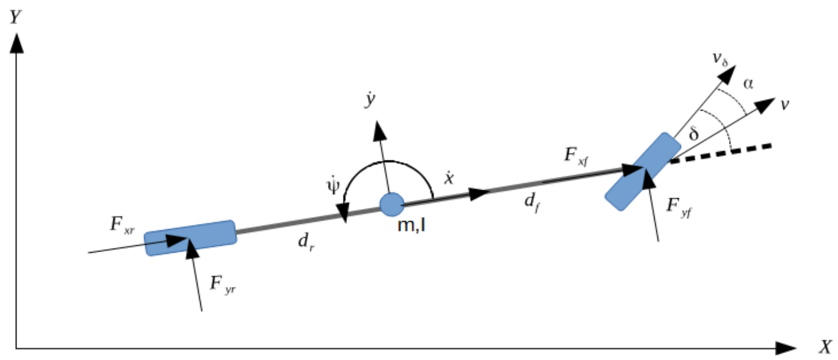

Figure 3.1 The bicycle model used to simulate the RCV.

The vehicle is simulated using the bicycle model as seen in Figure 3.1. This dynamical model assumes equal forces on the left and right wheels but different dynamics on the rear and front wheel pair [Turri, 2015]. Typically this is modeled as

$$\begin{aligned} m\ddot{x} &= F_{xf}\cos(\delta) + F_{xr} - F_{yf}\sin(\delta) - F_m - F_I\dot{x}^2 + m\dot{\Psi}\dot{y} \\ m\ddot{y} &= F_{xf}\sin(\delta) + F_{yf}\cos(\delta) + F_{yr} - m\dot{\Psi}\dot{x} \\ I\ddot{\Psi} &= d_f[F_{yf}\cos(\delta) + F_{xf}\sin(\delta)] - d_rF_{yr} \end{aligned} \tag{3.1}$$

where m is the vehicle mass, δ the steering angle, Ψ the heading while d_f and d_r is the distance to the center of gravity from the front and real axis, respectively. The x and y coordinates are the longitudinal and lateral position relative the vehicle. These can be transformed to the roads coordinate system (X, Y) as

$$\begin{aligned} \dot{X} &= \dot{x}\text{cos}(\Psi) - \dot{y}\text{sin}(\Psi) \\ \dot{Y} &= \dot{x}\text{cos}(\Psi) + \dot{y}\text{sin}(\Psi) \end{aligned} \tag{3.2}$$

The driving forces F_{xf} and F_{xr} on the front and rear wheel par is calculated as

$$\begin{aligned} F_{xf} &= k_{split}\frac{T}{r} \\ F_{xr} &= (1 - k_{split})\frac{T}{r} \end{aligned} \tag{3.3}$$

where T is the torque, k_{split} the percentage of torque to the front and r the wheel radius. Given a vehicle with the mass m, air resistance c and frontal area A, the resistive longitudinal forces F_m and F_i are calculated as

$$\begin{aligned} F_m &= mgf_r \\ F_I &= \tfrac{1}{2}\rho cA \end{aligned} \tag{3.4}$$

Here F_m is the motion resistance due to friction between the tires and the road while F_I is the proportional constant of the squared longitudinal speed assuming the gravitational constant g, air density ρ and friction coefficient f_r. The lateral forces on the front and rear axis, F_{yf} and F_{yr} respectively, are calculated as

$$\begin{aligned} F_{yf} &= -C_f\alpha_f \\ F_{yr} &= -C_r\alpha_r \end{aligned} \tag{3.5}$$

where C_f and C_r are the cornering stiffness for the front and rear axis while their slip angles are calculated as

$$\begin{aligned} \alpha_f &= \tan^{-1}\left(\tfrac{\dot{y}+\dot{\Psi}d_f}{\dot{x}}\right) - \delta \\ \alpha_r &= \tan^{-1}\left(\tfrac{\dot{y}-\dot{\Psi}d_r}{\dot{x}}\right) \end{aligned} \tag{3.6}$$

The discrete time state space model is then

$$\begin{aligned} X(k+1) &= X(k) + T_s[v_x(k)\text{cos}\Psi(k) - v_y(k)\text{sin}\Psi(k)] \\ Y(k+1) &= Y(k) + T_s[v_x(k)\text{cos}\Psi(k) + v_y(k)\text{sin}\Psi(k)] \\ \Psi(k+1) &= \Psi(k) + T_s\omega(k) \\ v_x(k+1) &= v_x(k) + \tfrac{T_s}{m}[F_{xf}\text{cos}(\delta) + F_{xr} - F_{yf}\text{sin}(\delta) - F_m - F_I v_x(k)^2 + m\omega(k)v_y(k)] \\ v_y(k+1) &= v_y(k) + \tfrac{T_s}{m}[F_{xf}\text{sin}(\delta) + F_{yf}\text{cos}(\delta) + F_{yr} - m\omega(k)v_x(k)] \\ \omega(k+1) &= \omega(k) + \tfrac{T_s}{I}[d_f[F_{yf}\text{cos}(\delta) + F_{xf}\text{sin}(\delta)] - d_rF_{yr}] \end{aligned} \tag{3.7}$$

where v_x and v_y is the longitudinal and lateral speed, respectively, while T_s is the sampling time and ω the angular velocity.

3.1.1 Parameter Values

The overtaking and file change scenarios are simulated in Matlab using the discrete time bicycle model of the RCV with a sample time of $Ts = 0.01s$. The parameters used in the simulation of the RCV are the following:

Parameter	Abbreviation	Value
Vehicle mass	m	608 kg
Moment of Inertia	I	1000 $\mathrm{kgm^2}$
Equivalent mass of wheel inertia	$\mathrm{m_j}$	40 kg
Gravity constant	g	9.81 $\mathrm{m/s^2}$
Distance from front axle to CoG	$\mathrm{l_f}$	1.0921 m
Distance from rear axle to CoG	$\mathrm{l_r}$	0.9079 m
Wheel radius	r	0.3 m
Air density	ρ	1.23
Projected frontal area	A	2.25 $\mathrm{m^2}$
Friction coefficient	$\mathrm{f_r}$	0.0927
Air drag coefficient	c	1.0834
Cornering Stiffness front wheel	$\mathrm{C_f}$	2.5669 $\cdot 10^{-4}$ N/rad
Cornering Stiffness rear wheel	$\mathrm{C_r}$	2.5669 $\cdot 10^{-4}$ N/rad

The other vehicles are assumed to have an approximately constant speed and stay in their lane. It is also assumed that their speed and position can be estimated with sufficient accuracy.

Recalling that $x = [X\ Y\ v\ \Psi\ \omega]^T$ and $u = [a\ u_\Psi]^T$ the weight matrices used for the lane change scenario are

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.8}$$

And for the overtaking

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.9}$$

More weight were selected for the speed in the lane change scenario since the RCV needs to keep up with the speed of the vehicles in the adjacent lane. The weight for the lateral position is to ensure that the RCV stays in the middle of the lane as much as possible. The prediction horizon used had $N = 20$ prediction steps for the states and $M = 1$ for the inputs, while the prediction step time T_{pre} where 0.2 seconds, giving a prediction horizon of 2 seconds.

3.1.2 Controllers

Simple lateral and longitudinal controllers were made to test the planners. The longitudinal controller output torque T proportional to the speed error

$$T = P_T(v_r - v) \tag{3.10}$$

where v_r is the speed reference given by the MPC and v the current speed. The lateral controller changes the steering angle δ proportionally to the error in yaw rate:

$$\delta = P(\omega_r - \omega) \tag{3.11}$$

where ω_r and ω is the preferred and current yaw rate, respectively. For the longitudinal controller P_t was set to 2000 while the lateral controller had $P = 10$ for the obstacle avoidance and $P = 5$ for the lane change.

Since the MPC planner use linear and angular acceleration as inputs while the control instead use speed and yaw rate, the MPC outputs the predicted vehicles state to the controllers rather than its input. Simple giving the vehicle the speed $v(k+1|k)$ and yaw rate $\omega(k+1|k)$ would give the preferred speed and yaw rate in 0.2s rather than how the vehicle should behave right now. By instead using the simulation sample time $T_s = 0.01$ the references for the controller is generated as

$$\begin{aligned} v_r &= v + T_s a \\ \omega_r &= \omega + T_s u_\Psi \end{aligned} \tag{3.12}$$

3.1.3 Decision Making and Constraints.

In the overtaking scenario the vehicle first decides if it should wait behind the obstacle or perform the overtaking. This is done by calculating the expected time it takes for the opposing vehicle to be adjacent to the obstacle. If this time is less than 10 seconds the MPC gets a constraint on the maximum longitudinal position equal to the obstacle position minus a safe distance of 8m. Otherwise the overtaking is performed using the inequality constraint to force the vehicle into the other lane. Once the vehicle is approximately adjacent to the obstacle this inequality is replaced by a lower limit on the lateral position.

In the lane changing scenario the vehicle is first confined to its own lane by the upper and lower limits on the lateral position. The slot closest to the vehicle is chosen as a reference position. When its longitudinal position is close enough this constraint is extended to the adjacent lane and the vehicle starts performing the lane change.

3.1.4 Modeling the Other Vehicles

The other vehicles simulated for the lane change and overtaking did not have a detailed model. The position $p_i(k)$ for vehicle i is calculated as

$$p_i(k+1) = p_i(k) + T_s v_i(k) \tag{3.13}$$

using the current speed $v_i(k)$ and sample time T_s.

3.2 Truck Model

In order to simulate the truck a Simulink function block written in C is used. This discrete time model provided by Scania takes the current and initial states of the truck as well as the speed request, using the trucks built in CC. This function block also takes a curvature request which was set to zero in these simulations since only longitudinal control was investigated. This model outputs the next states of the vehicle, in this case the longitudinal speed and position was used.

3.3 Implementing the Longitudinal Planner

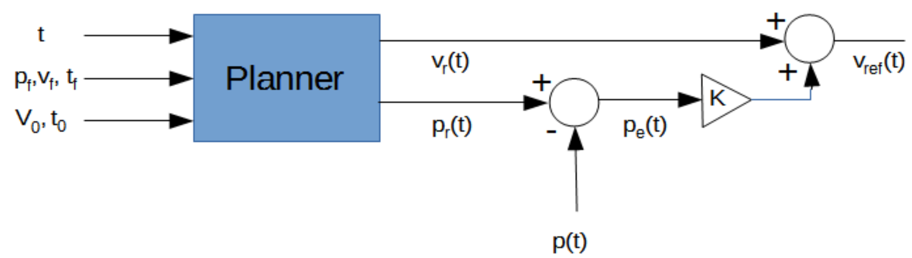

Figure 3.2 Block diagram illustrating how the longitudinal planner is implemented.

In order to calculate the longitudinal trajectory needed to reach the CZ with a given velocity in a certain time a Matlab function calculates the reference acceleration $a_r(t)$. This is done using PMP or interpolation as was shown in section 2.2, using a final position p_f, speed v_f and time t_f as well as initial speed v_0. In the case of PMP the function parameters c_1 and c_2 are calculated once then outputting the reference acceleration at each time instant as

$$a_r(t) = \frac{c_1 t - c_2}{2} \tag{3.14}$$

For some combinations of the final states t_f, p_f and v_f the reference acceleration may be initially negative, thus giving a negative speed reference which is generally not desirable. This typically happens when the average speed required by the vehicle is much lower than the final speed v_f. To prevent this the planner will wait until the the average speed required to reach the the final position p_f is at least a third of the final speed, that is

$$\frac{p_f}{t_f} \geq \frac{v_f}{3} \tag{3.15}$$

Since the vehicle uses a Cruise Controller, the reference acceleration is integrated to give the reference speed $v_r(t)$ as seen in figure 3.2. If the vehicle followed this speed all the time it would also arrive at the CZ within the preferred time instant. However, since the vehicle's speed is likely to differ from this reference, an error in position will accumulate and thus cause it to arrive at the CZ at the wrong time. To adjust for this error in position a simple feedback loop is added in order to adjust for the error in position. This adjustment is simply a P-controller, where the constant K determines how this error affects the adjusted speed reference $v_{ref}(t)$.

The reference position $p_r(t)$ is calculated simply by integrating $v_r(t)$ while the vehicles position $p(t)$ is calculated as

$$p(t) = d_0 - d(t) \tag{3.16}$$

where d_0 is the initial distance to CZ, used as p_f, and $d(t)$ is the current distance. The distance is calculated by a separate Matlab function using the GPS position of the vehicle and the CZ.

Chapter 4

Results

This chapter shows the main results that were obtained from simulations. First the Lane change scenario with RCV is showed, followed by the overtaking scenario. The lane change scenario is simulated in three different scenarios with constant, increasing and decreasing speed, respectively. The overtaking is performed both with a stationary target and a slow moving vehicle. Finally, longitudinal planning and control are shown for both the RCV and the truck, first with PMP and then interpolation.

4.1 Lane Change

4.1.1 Case 1: Constant Speed

In Figure 4.1 a simulation of the lane change using a MPC planner is shown. All vehicles have an initial speed of 10 m/s and those in the adjacent lane are spaced 10 m from each others point position. The controlled vehicle started at 40 m and selected the slot behind the lead vehicle in the adjacent lane. Before crossing the roads centerline, indicated by its position being plotted in blue, the ego vehicle is approximately adjacent to the lead vehicle. It then slows down to fit between the lead vehicle and the one right behind.

In figure 4.2 the lateral position of the ego vehicle is shown as it moves between the centerlines of each lane, a distance of 3m. The maneuver is done in approximately 10 seconds and the ego vehicle is 5m behind the lead vehicle when crossing the road centerline. Since this is the distance between the vehicles' point positions, the actual distance would be approximately 1m assuming a vehicle length of 4m.

4.1.2 Case 2: Slow Down

The same scenario is shown in 4.3, except here the speed of the other vehicles is not constant. After 10 second the speed slows down from the initial 10 m/s with an acceleration of -2m/s^2. This slow down starts approximately

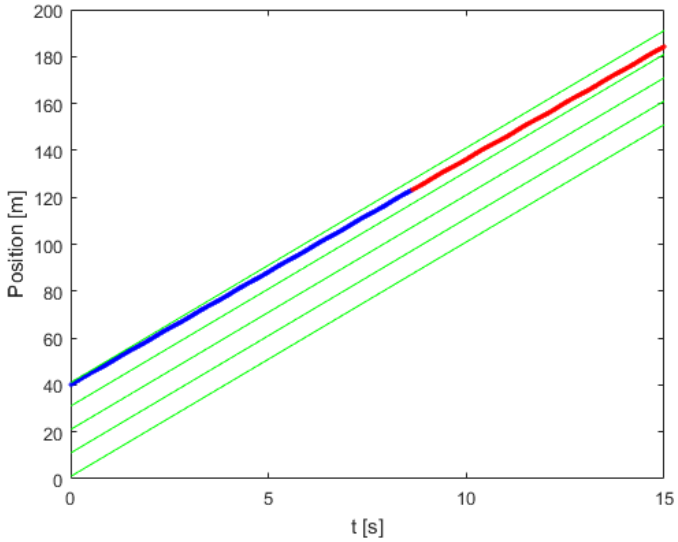

Figure 4.1 Simulation of a vehicle finding a slot between the other vehicles and changing lane, showing the longitudinal position as a function of time. The ego vehicle is shown in blue while in the its initial lane, then in red when crossing the road centerline. The other vehicles position are shown in green.

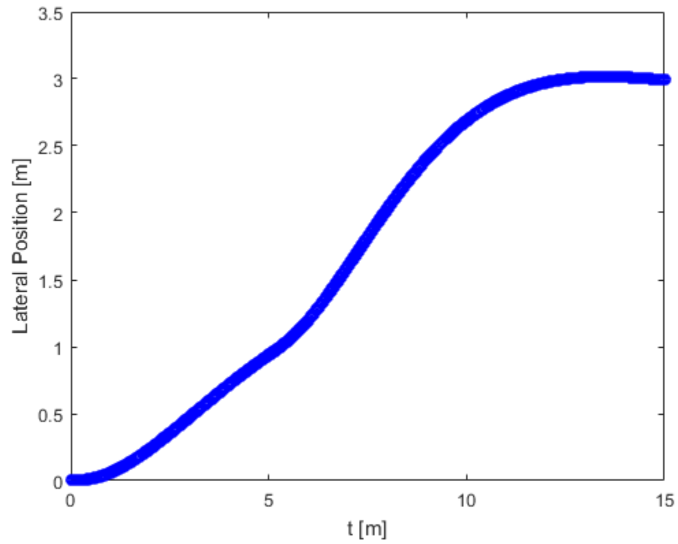

Figure 4.2 The lateral position of the ego vehicle during the lane change maneuver.

at the same time as the ego vehicle crosses the roads centerline. This slow down can also be seen at the lateral position shown in figure 4.4. Compared with case 1, the ego vehicle has greater difficulty in performing the lane change as it must also adapt the speed to the vehicles slowing down.

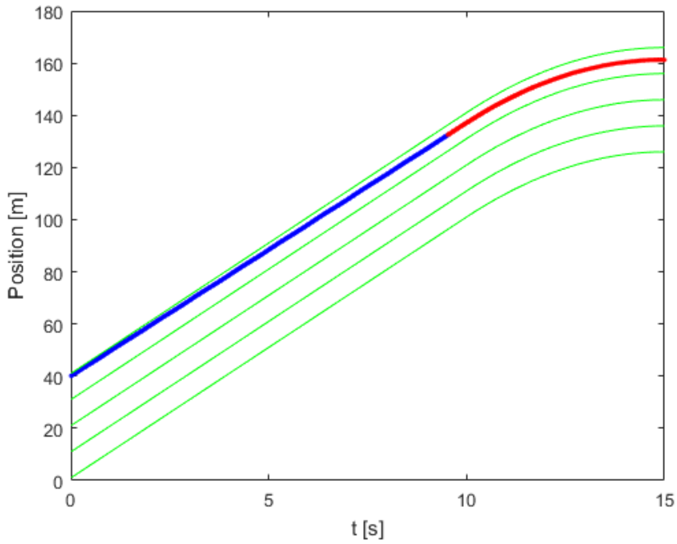

Figure 4.3 The lane change performed as in case 1, except for a slow down in speed after 10 seconds.

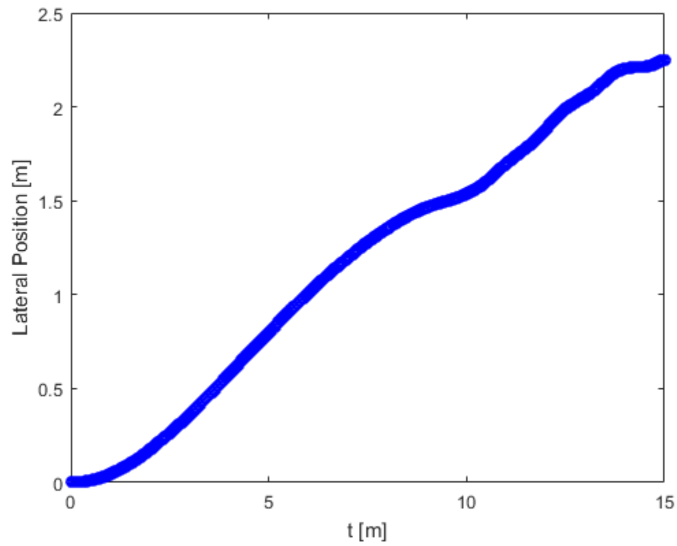

Figure 4.4 The lateral position of the ego vehicle when changing lane while slowing down.

4.1.3 Case 3: Increased Speed

The simulation was also done with an increase in speed after 10s, using an acceleration of 2 m/s^2. As seen in figure 4.5 the ego vehicle start cross the road centerline just before the other vehicles increases their speed. The ego vehicle completes the lane change as seen in 4.6 but does not reduce the lateral speed which might cause the vehicle to get too close to the road delimiter.

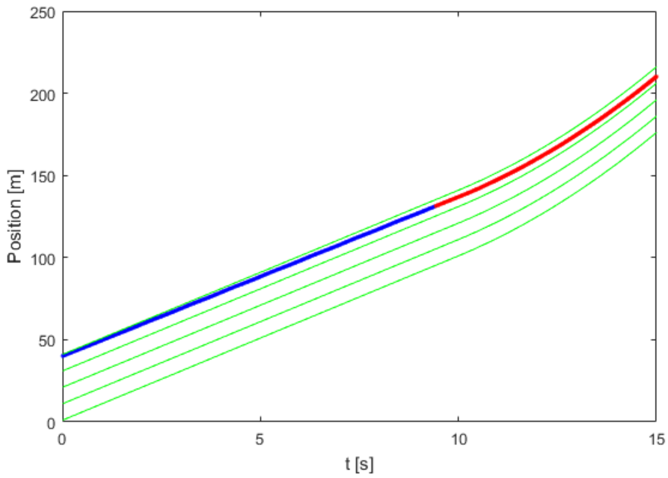

Figure 4.5 The lane change performed as in case 1, except for an increase in speed after 10 seconds.

4.2 Overtaking

A simulation of an overtaking scenario is shown in figure 4.7. The obstacle is stationary and initially 50 m ahead of the vehicle while an opposing car with a speed of 10 m/s (36 km/h) is driving in the opposite direction, starting 100 m in front of the car. The vehicle being guided by the MPC has the same initial speed as the opposing vehicle and drives past the obstacle at approximately 2.5 m distance from the cars central axis. Assuming that the cars are approximately 2m wide this should be enough although barely. As seen in the speed profile in Figure 4.9 the vehicle had to stop for a brief moment to let the opposing vehicle pass.

In Figure 4.8 a similar situation is shown except the obstacle is moving with a speed of 2 m/s. The RCV then takes longer time to pass the obstacle but maintain a lateral distance of approximately 2.5m. The ego vehicle does not completely stop as seen in figure 4.11, which is expected as the vehicle in front is not stationary. The turning is also less sharp compared to the stationary case as seen in figure 4.12.

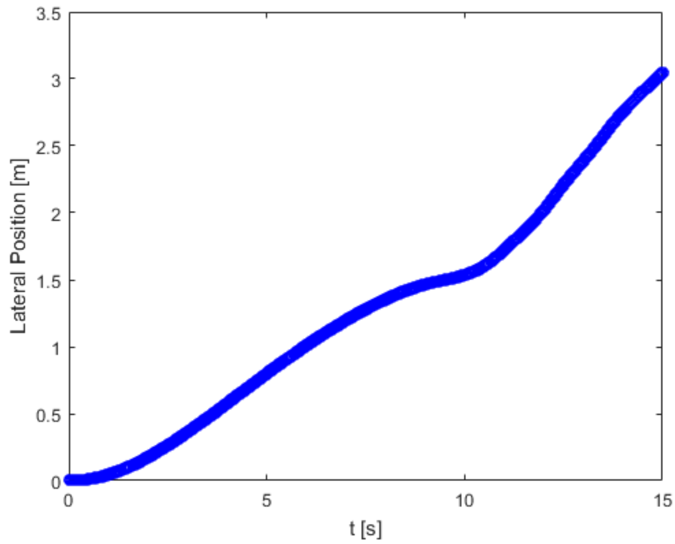

Figure 4.6 The lateral position of the ego vehicle when changing lane increasing the speed.

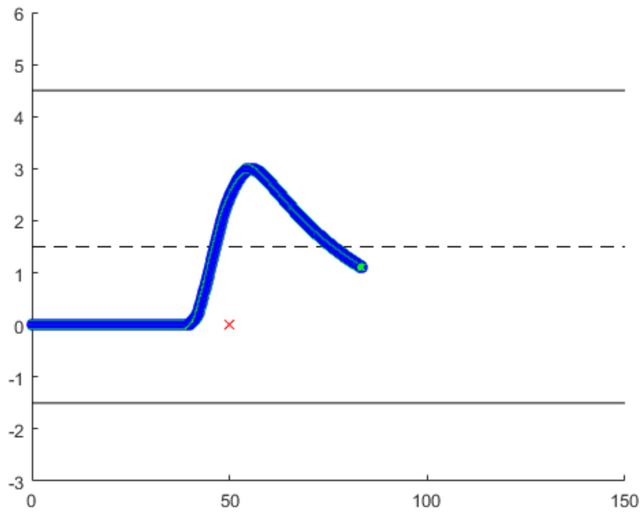

Figure 4.7 The vehicles trajectory seen in blue overtaking a stationary object marked in red.

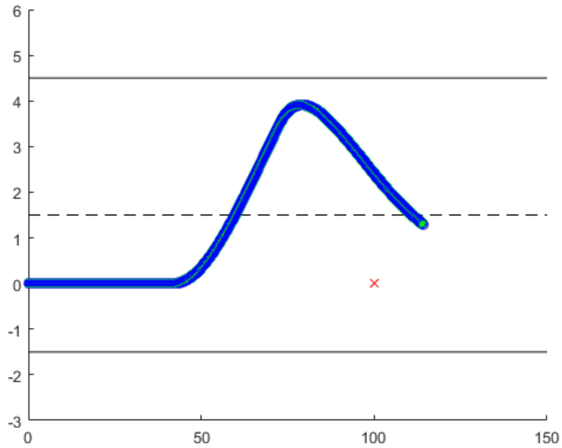

Figure 4.8 The overtaking scenario with a slow vehicle driving at 2 m/s.

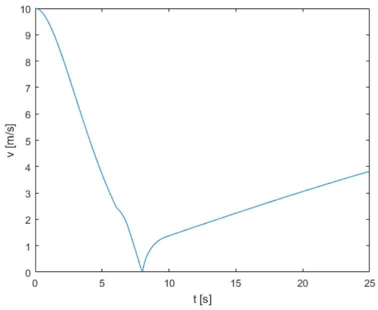

Figure 4.9 Speed for the vehicle when overtaking an stationary obstacle.

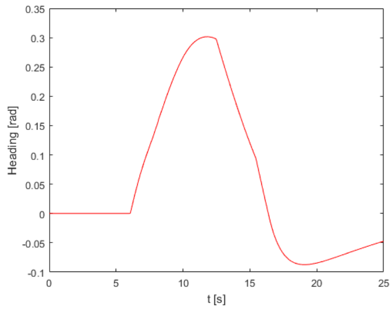

Figure 4.10 The heading of the vehicle relative the road in the same scenario.

4.3 Longitudinal Planning

4.3.1 Truck

Longitudinal control of the truck was simulated with the PMP to see how well it can achieve the desired time, speed and position for the intersection scenario. The truck is generally more difficult to control than the RCV, mainly due to the slow dynamics of its gearbox. The delay caused by the gear change can be seen in the speed profile in Figure 4.13 showing small intervals where the acceleration is low. The reference speed is approximatively 2 m/s for the first 17-18 seconds of the simulations as the planner waits for the time left to be smaller and thus give a better speed profile. The truck managed to reach the speed of 7 m/s after 40s and also reach the longitudinal position

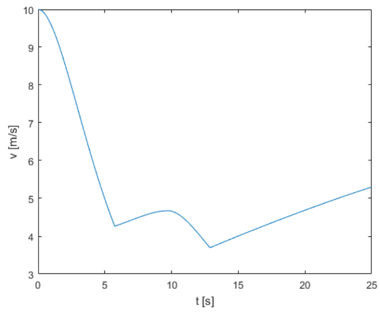

Figure 4.11 Speed for the vehicle when overtaking an moving vehicle.

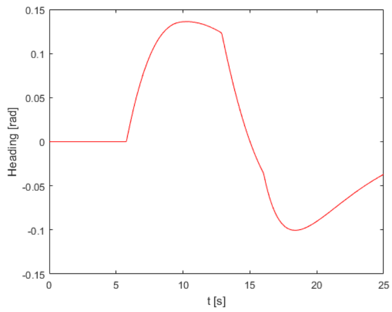

Figure 4.12 The heading of the vehicle relative the road in the same scenario.

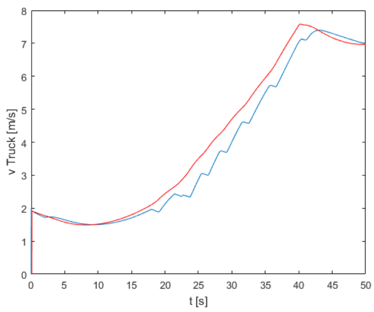

Figure 4.13 The speed of the truck in blue and the reference speed in red.

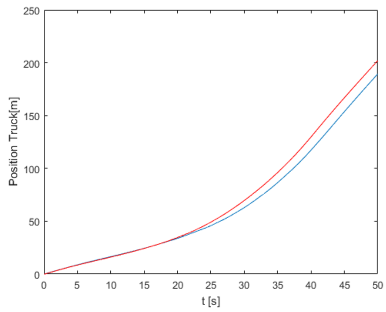

Figure 4.14 The position of the truck in blue and reference in red.

of 100m as seen in Figure 4.14.

In all simulations the final speed was 7m/s, goal time 40s and initial speed 2m/s, while the final position was varied. The simulations were then repeated using interpolation instead of PMP. As seen in figure 4.15 the position error is approximately the same for PMP and interpolation for distances up to 200m. At 240 and 280m the interpolation method does however give a position error that is at least 2m greater than for PMP. The exact errors for both speed and position are seen in the table below. While the error in speed is insignificant in both cases the planning using PMP performed better regarding the position error when a greater distance was covered. The improved performance could be a result of PMP minimizing the square acceleration.

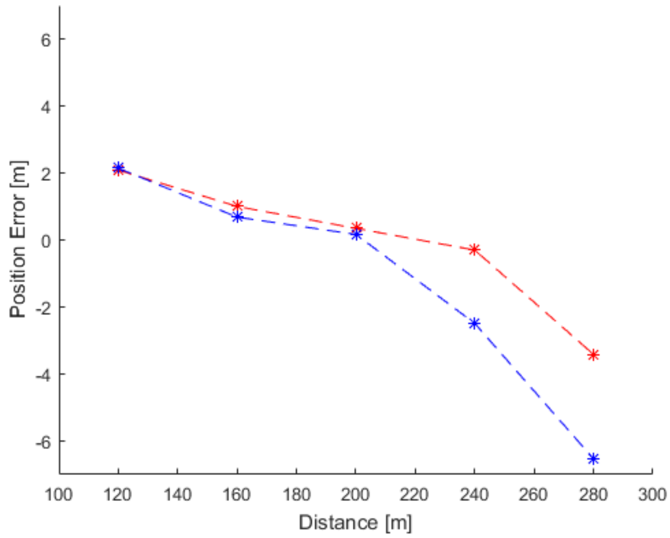

Figure 4.15 The distance error for the truck simulated with different goal positions. The errors using PMP are shown in red and the errors with interpolation are shown in blue.

Distance	Distance error, PMP	Speed error PMP
120m	2.0822m	-0.0633/s
160m	1.0015	-0.0229/s
200m	0.3474m	-0.0198m/s
240m	-0.3020	-0.1158m/s
280m	-3.4158m	-0.7890m/s
Distance	Distance error, interpolation	Speed error, interpolation
120m	2.14352m	-0.5999/s
160m	0.6818m	-0.0592/s
200m	0.1663m	-0.0198m/s
240m	-2.4886m	-0.2951m/s
280m	-6.5315m	0.4617m/s

4.3.2 RCV

Similar simulations were done with the RCV which however had an initial speed of 0m/s. The controller used was based on state feedback and set to perform CC. Using $t_f = 40s$, $p_f = 120m$ and $v_f = 7m/s$ the position error was 0.13m and the speed error 0.006 m/s. The speed and position are seen in Figure 4.16 and 4.17, respectively.

As seen in the table below the position errors are much smaller for the RCV compared to the truck's, confirming that its speed is easier to control than the truck. Also remarkable is how little the speed error varies as the

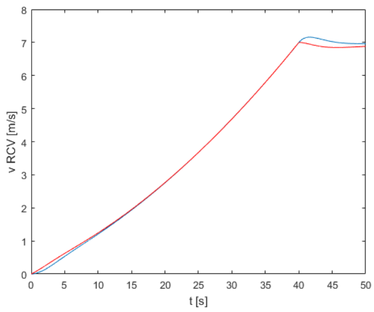

Figure 4.16 The speed of the RCV in blue and the reference speed in red.

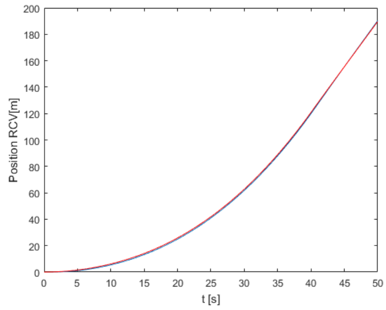

Figure 4.17 The position of the RCV in blue and reference in red.

distance is increased compared to the truck. As seen in figure 4.18 the position errors differ no more than 0.5m between PMP and interpolation.

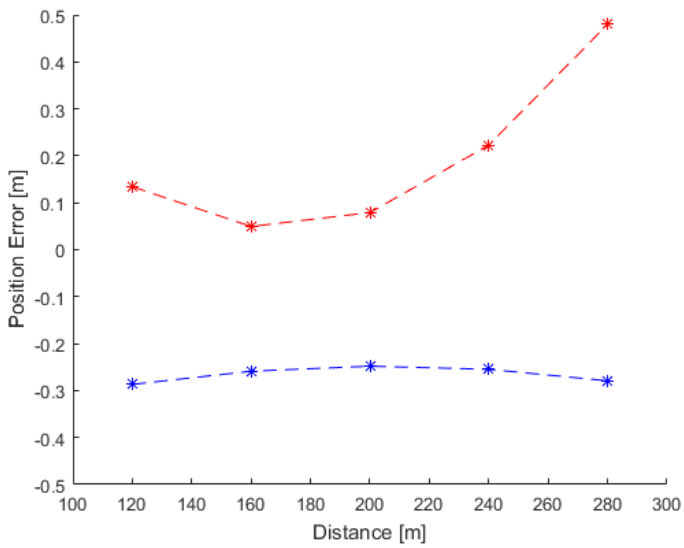

Figure 4.18 Distance error for the RCV simulated with different goal positions. The errors using PMP are shown in red while the errors for the interpolation methods are shown in blue.

Distance	Distance error PMP	Speed error PMP
120m	0.1342m	0.006m/s
160m	0.0495m	0.0048m/s
200m	0.0791m	0.0068m/s
240m	0.2227m	0.0118m/s
280m	0.4798m	0.0199m/s
Distance	Distance error Interpolation	Speed Interpolation
120m	-0.2868m	-0.0649m/s
160m	-0.2588m	0.0029m/s
200m	-0.2480m	0.0718m/s
240m	-0.2548m	0.1418m/s
280m	-0.2792m	0.2131m/s

Chapter 5

Conclusion

The purpose of this thesis was to investigate the use of trajectory planning for autonomous and cooperative driving. Since the RCV could not participate in GCDC no planner were implemented in a real vehicle. Simulations using the RCV with a MPC planner did however show that it was possible to perform lane change maneuvers, although the safety margins might not be good enough to be used in real traffic.

One of the must challenging aspects of developing the MPC was to set proper constraints on the ego vehicles position in order to avoid collision. These constraints had to be adapted depending on the intended maneuver and the other vehicles' positions. The main weakness of the planner is that it assumes constant speed of the other vehicles. If for example a vehicle in the front slows down, the constraints on the position over the prediction horizon might cause the ego vehicle to not reduce its speed enough.

A problem with the position constraints used by the MPC planner is that it often forbids more positions than what is actually necessary to avoid a collision. When the ego vehicle is adjacent to another vehicle the lateral constraints limits its position to the current lane even when a lane change is to be performed. The diagonal position constraint used for overtaking could also be a problem since it limits the longitudinal position even if the adjacent lane is free from other vehicles. This might cause the vehicles to slow down too much during the overtake.

As for longitudinal planning, both PMP and interpolation methods were proven to be useful for timing the position and speed of a vehicle. Simulations showed that the RCV was easier to control than the Scania Truck, which was expected as the truck has much slower acceleration and is affected by the gear changes. This could however also be a result of the RCV being simulated with a more simple model. Comparing PMP with interpolation showed that PMP was more effective in timing the position of the vehicles, in particular for the truck.

In GCDC it was shown that cooperative driving can be used to safely perform maneuvers such as merging of lanes and intersection crossings. It did however require that all vehicles share communication protocols and have a similar decision making process. While it is possible that cooperative driving is implemented in the future, it would require extensive standardization in order to be used on a larger scale.

5.1 Future Work

Further work on this project should focus on making a more flexible and universal planner. In GCDC the planners had a more limited role as they were only activated in specific situations. While the MPC planner that was simulated for this thesis was used all the time it had to be tunned for the different scenarios and also only considered the other vehicles as point particles. In a real traffic situation the planner would need to consider the actual euclidean distance between the vehicles bodies and use that as a constraint. This would give position constraints in the form of an oval around the nearest vehicle. The prediction horizon would then properly consider the different actions that the ego vehicle can take.

In order to make the simulations more realistic the other vehicles would need more advanced models and preferably some form of artificial intelligence in order to react properly to the ego vehicles actions. This planner should then be implemented on a actual vehicle, for example the RCV. In addition to the planner a more unified decision making is needed since different algorithms were used for each scenario. More advanced controllers should also be implemented since the real vehicle would be more difficult to control than the simplified bicycle model used in the simulations.

Bibliography

Erkorkmaz, K., Alzaydi, A., Elfizy, A., and Engin, S. (2011). Time-optimized hole sequence planning for 5-axis on-the-fly laser drilling. *CIRP Annals - Manufacturing Technology.*

Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., and Hrovat, D. (2008). A hierarchical model predictive control framework for autonomous ground vehicles. American Control Conference.

Farrokhsiar, M., Pavlik, G., and Najjaran, H. (2013). An integrated robust probing motion planning and control scheme: A tube-based mpc approach. *Robotics and Autonomous Systems.*

Glad, T. and Ljung, L. (2000). *Control Theory, Multivariable and Nonlinear Methods.* CRC Press.

Grancharova, A., Grøtli, E. I., Ho, D.-T., and Johansen, T. A. (2014). Uavs trajectory planning by distributed mpc under radio communication path loss constraints. *J Intell Robot Syst.*

Li, X., Sun, Z., Zhu, Q., and Liu, D. (2014). A unified approach to local trajectory planning and control for autonomous driving along a reference path.

Semsar-Kazerooni, E., Bengtsson, H. H., and Medina, A. M. (2015). Interactive protocol. http://gcdc.net/images/doc/D2.1.Interaction.Protocol.pdf.

Tokekar, P., Karnad, N., and Isler, V. (2014). Energy-optimal trajectory planning for car-like robots. *Auton Robot.*

Turri, V. (2015). *Fuel-efficient and safe heavy-duty vehicle platooning through look-ahead control.* PhD thesis, KTH Royal Instiute of Technology.

You, F., Zhang, R., Lie, G., Wang, H., Wen, H., and Xu, J. (2015). Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system. *Expert Systems with Applications.*

TRITA EE 2016:119
ISSN 1653-5146