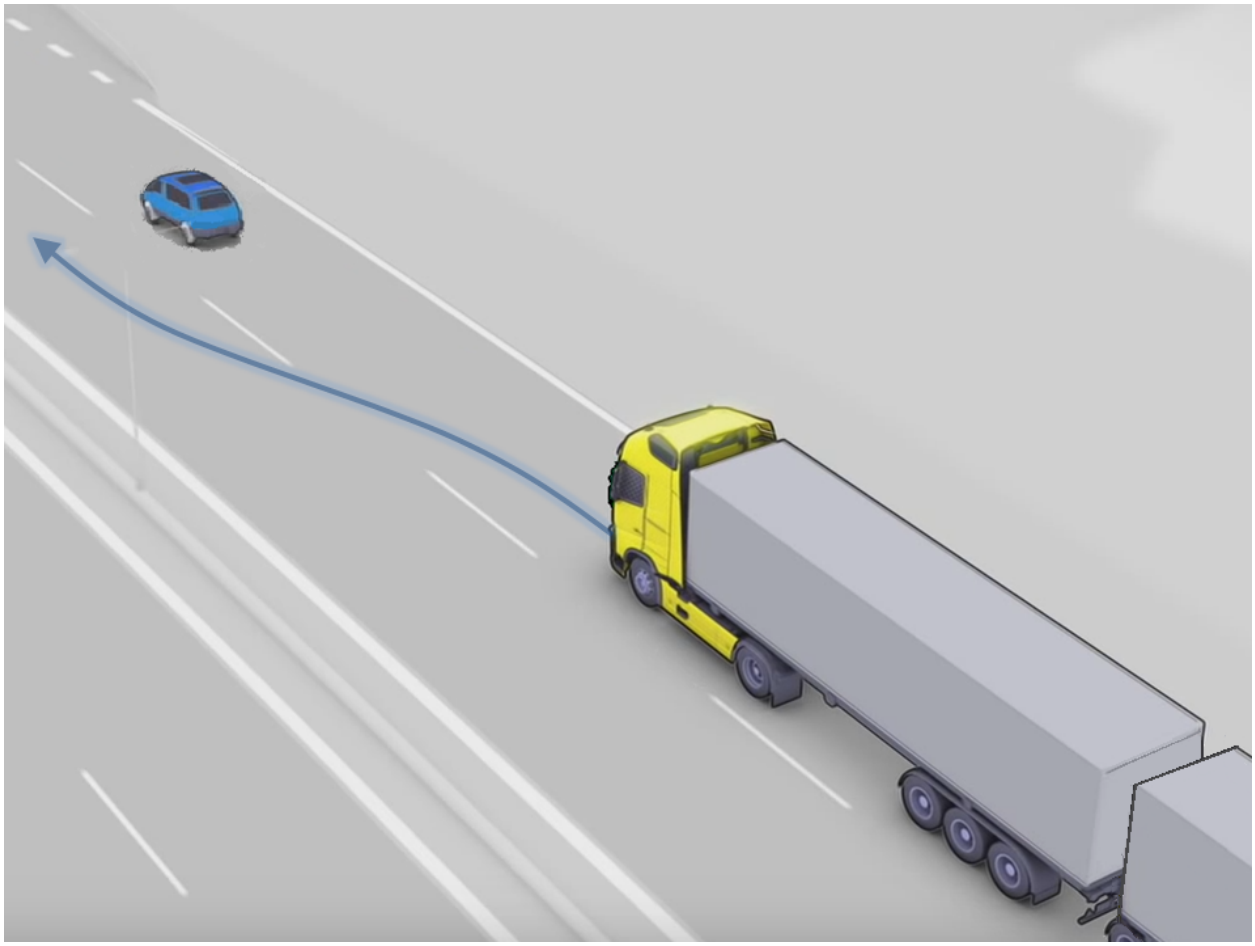
CHALMERS
UNIVERSITY OF TECHNOLOGY



Trajectory planning for automated highway driving of articulated heavy vehicles

Convex optimization using the model predictive control framework

Master's thesis in Systems, Control and Mechatronics

PATRIK WALLIN

PATRIK NILSSON

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018:EX025

Trajectory planning for automated highway driving of articulated heavy vehicles

Convex optimization using the model predictive control framework

PATRIK WALLIN PATRIK NILSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2018

Trajectory planning for automated highway driving of articulated heavy vehicles
Convex optimization using the model predictive control framework
PATRIK WALLIN PATRIK NILSSON

Supervisor: Peter Nilsson, Volvo AB/Chalmers
Examiner: Jonas Fredriksson, Chalmers, Systems and Control

Master's Thesis 2018:EX025
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Trajectory planned by algorithm to safely change lane, modified from [1].

Gothenburg, Sweden 2018

Trajectory planning for automated highway driving of articulated heavy vehicles
Convex optimization using the model predictive control framework
PATRIK WALLIN PATRIK NILSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

As the global level of CO_2 in the atmosphere continues to rise, ways to decrease and stop this rising trend are needed. One potential measure to combat rising green house gases could be to use long combination vehicles, LCVs, for heavy duty freight transports. However, because of the length, weight and motion characteristics of LCVs, they become more difficult to maneuver, especially in dense traffic. The development of advanced driver assist- or autonomous functions for controlling LCVs, could help ease LCVs inherent problem of maneuvering in dense traffic. In this project, the LCV considered is the A-double combination. An A-double combination consists of a tractor connected to a semi-trailer. Attached to this semi-trailer is a converter dolly onto which a second semi-trailer is attached.

The objective for this project is to develop a trajectory planner for an A-double combination to allow for automated driving in a highway environment. Trajectory planning for automated vehicles consist of generating a trajectory that spans both the longitudinal and the lateral dimension. This trajectory is calculated based on the vehicle state and control evolution in time, given some set of constraints. There are two core maneuvers that needs to be handled by the trajectory planner, lane keeping and lane changing.

The method used for developing the trajectory planner is based on numerical optimization due to its innate capability to handle constraints and its real-time compatibility. This way, safe and smooth trajectories can be generated. The control framework used is called model predictive control, MPC, and the trajectory planning problem is formulated as a quadratic problem with linear constraints. The trajectory planner was developed with the ACADO toolbox and then combined with the solver qpOASES.

The generation of results was done in Matlab/Simulink by creating s-functions for the trajectory planner, and then simulating it against a high-fidelity plant model of an A-double combination. Two trajectory planners, one with a prediction horizon of 2 seconds and one with a prediction horizon of 5 seconds, were simulated and discussed. Both the planners successfully completed a pre-defined highway test scenario regarding safety, smoothness, actuator limitations and computational time. However, smoother longitudinal characteristics could be concluded for the planner with a prediction horizon of 5 seconds compared to the trajectory planner with a prediction horizon of 2 seconds.

Overall, the proposed trajectory planner is able to handle the two main maneuvers of

highway driving, lane keeping and lane changing, while ensuring safety, smoothness and that the actuator limitations are not being violated. This while ensuring solution times suitable for a real time implementation when running on a notebook PC.

Keywords: long combination vehicles, A-double, trajectory planning, highway, quadratic programming, convex, model predictive control, Matlab/Simulink, ACADO, QPOASES.

Acknowledgements

The authors would like to thank our Volvo supervisor, Peter Nilsson, for guidance throughout the project, our examiner from Chalmers, Jonas Fredriksson, for his assistance with support and feedback and the former Volvo GTT master thesis student, Niels V. Duijkeren, for providing us with inspiration and a framework to build the trajectory planner upon. Furthermore, the authors would like to thank our fellow student colleagues, Johan Larsson and Hampus Berg, for a valuable peer-review. Finally, we would like to thank Volvo Group Truck Technology for providing an inspiring environment and the required tools to successfully realize this master thesis project.

Gothenburg, May 29 2018,
Chalmers University of Technology

Patrik Nilsson
Patrik Wallin

Contents

List of Figures

1
Introduction

As the global level of CO_2 in the atmosphere continues to rise, ways to decrease and stop this rising trend are needed [6]. The surface freight transportation industry in the OECD (Organization for economic cooperation and development) countries is responsible for about 35% of its total CO_2 emissions, and, is expected to rise to about 50% in the coming 35 years [7]. With climate goals such as the Kyoto protocol and the Paris agreement and the connection between global warming and CO_2 emissions, ways to decrease this is of high interest. One potential measure to combat rising green house gases could be to use long combination vehicles, LCVs. LCVs are vehicle combinations that typically are longer and heavier than current vehicle combinations allowed under European legislation. As described by the author in [4], LCVs often include two articulated joints or more and is most commonly between 27-32 meters in length. Furthermore, given their increased length and higher weight, compared to regular tractor semi-trailer combinations, LCVs can transport up to 2 times as much cargo in terms of weight and/or volume. As a consequence of this, the author claims that their energy consumption is generally 15-20% lower. In addition to contributing to lower energy consumption within the freight industry, LCVs could also help decrease traffic congestion as fewer vehicles would be needed to transport the same amount of goods and also decrease road transportation costs. However, because of the increased length, weight and additional articulation joints, LCVs become more difficult to maneuver, especially in dense traffic. The development of advanced driver assist- or autonomous functions for controlling LCVs, could help ease LCVs inherent problem of maneuvering in dense traffic.

In this project, the LCV considered is the A-double combination. An A-double combination consists of a tractor connected to a semi-trailer. Attached to this semi-trailer is a dolly onto which a second semi-trailer is attached. In fig. 1.1 an A-double combination and its articulation points is shown.

Figure 1.1: Setup for an A-double combination. Picture from [2].

Given the challenging environment many traffic situations can pose, highway driving is considered to be the first target for automated driving on public roads. This as it presents a rather controlled setting with one-way traffic and road users driving at similar speeds. Somewhat simplified, highway driving can be reduced to two main maneuvers, lane keeping and lane changing. With these two core maneuvers, further actions such as overtaking, entering and exiting the highway can also be produced, thus enabling complete highway control. However, major challenges exist when developing algorithms for dealing with these two maneuvers. This as they require extensive knowledge about the vehicle's surrounding environment and the current road profile, but also because systems for tactical decision making and trajectory planning are necessary.

In this project, the objective is to develop a trajectory planner for the A-double combination to allow for automated highway driving. Other key functionality for automated driving will not be investigated to any greater extent, this will be furthered discussed in section 1.3.

1.1 Background

Trajectory planning for automated vehicles consist of generating a trajectory that spans both the longitudinal and lateral dimension. This trajectory is calculated based on the vehicle state and control evolution in time, given some set of constraints.

As previously mentioned, there are two core maneuvers that needs to be handled by the trajectory planner, lane keeping and lane changing. When generating a trajectory for these maneuvers, there are some sets of constraints that needs to be taken into consideration. First, due to the heavy weight of an A-double combination, the longitudinal dynamics of the truck will be affected. Most notably will be the limited acceleration capabilities, even on flat roads. The additional length and articulation points of an A-double combination will also affect the lateral dynamics of the vehicle combination. For example, phenomena such as rearward amplification and lateral off-tracking are introduced. These will be further discussed in chapter 3.

Properties of the road also creates a set of important constraints for the trajectory planner as the road profile will have an impact on the vehicle dynamics. For example, the curvature of the road will add to the lateral acceleration of the vehicle combination and the road topography will affect the longitudinal acceleration and deceleration. Another important aspect that needs to be handled when planning different maneuvers is the available road area for the truck to drive on. Fellow traffic participants needs to be addressed to ensure that the truck will not come to close to surrounding vehicles and for the truck to be able to change lane. Finally there are limitations regarding the actuators of the truck, for example acceleration and steering limits.

All these limitations are typically connected to safety and ride comfort in terms of

keeping the vehicle motion within specified limits and keeping the actuation within capacity limits. This needs to be addressed by the trajectory planner to ensure that a safe and smooth trajectory is planned.

As described in [4], there are different methods for solving the trajectory planning problem and they differ in how the available search space is represented. One way to model it could be as a graph, where a transition between two nodes could represent a change of states. This way, graph search algorithms could be used to plan trajectories for different maneuvers. Another way of solving the problem could be by using numerical optimization. If the search space is represented with models subject to a set of constraints, the problem is modeled as an optimal control problem, OCP. By simulating these models forward in time, subject to the set of constraints, trajectories could be planned for a given time interval. In [8], numerical optimization techniques are used in order to successfully perform lane change maneuvers for a passenger car and in [2], non-linear numerical optimization have successfully been used in order to state a trajectory planner for an A-double vehicle combination.

1.2 Objective

The objective of this project is, as briefly mentioned before, to study, develop, implement and evaluate algorithms which will be used in a trajectory planner for an A-double combination. The trajectory planner should be able to maneuver an A-double combination driving in high speed in a highway application. Another part of the objective is that the trajectory planner should be implemented with real-time performance in mind.

The method used for developing the trajectory planner is based on numerical optimization due to its innate capability to handle constraints and its real-time compatibility. This way, trajectories that ensure both ride comfort and safety of the truck can be generated.

1.3 Prerequisites

Trajectory planning is a vital part in a system for automated driving functionality. In fig. 1.2, one can see a system overview which places the trajectory planning in a system context in a system for automated driving functionality. In this project, only the concept of trajectory planning will be addressed. Surrounding key functionality for automated driving, such as environment perception and decision making, is assumed to be given or simplified in order to obtain a complete system for the trajectory planner.

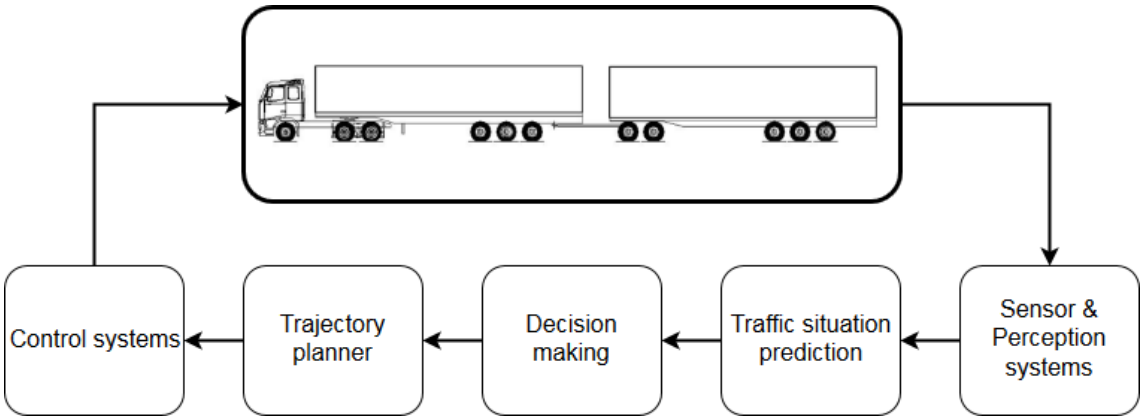


Figure 1.2: Envisioned system for automated driving functionality.

The algorithms in this project are developed under the following prerequisites:

- P1 Traffic situation predictions will be generated by using single integrator models of the surrounding vehicles. Furthermore, it is also assumed that the surrounding traffic travels at a constant velocity without any lane changes. In a more advanced system, this could be furthered developed to ensure that more accurate information is provided to the trajectory planner.
- P2 Traffic perception is assumed to be perfect in this project and the knowledge about the speed and position of fellow road participants is assumed to be known.
- P3 Road perception is assumed to be perfect in this project, i.e., knowledge about the curvature and topography is assumed to be known.
- P4 Decision making functionality is assumed to be given. This will feed the trajectory planner with information such as when to initiate a lane change and a longitudinal velocity reference.
- P5 Actuation control is assumed to be given. The trajectory planner will generate references that will be fed forward to an actuation control layer that will create control signals for actuation of the truck.

1.4 Limitations

In this thesis, the work has been carried out under the following limitations:

- L1 Only an A-double LCV has been considered in the thesis.
- L2 The developed algorithm is only intended for one way, multiple lane highway use in a velocity range of 30-90 $[km/h]$.
- L3 The models used for describing longitudinal and lateral dynamics of the LCV have been simplified to ensure linear dynamics.
- L4 The developed algorithms will not be verified using physical testing. Instead simulations against a high-fidelity model will be used to verify the algorithms.
- L5 Only convex numerical optimization will be evaluated.
- L6 The problem will be separated into two control problems, one for longitudinal control and one for lateral control.

L7 Only the ACADO toolbox will be evaluated when generating code for the optimal control problem formulation.
L8 Only qpOASES will be evaluated as the solver for the generated quadratic problem.

1.5 Outline of the thesis

Chapter 2 will briefly introduce the basics of numerical optimization and also introduce a control framework called model predictive control. This by introducing the concepts of convexity and receding horizon control.

Models will be used in both the trajectory planner and the tuning and evaluation of the trajectory planner. Chapter 3 will present the lateral model used for representing the lateral dynamics of the A-double combination. The model used for the longitudinal dynamics will also be presented along with the road representation and the modelling of the surrounding traffic.

In chapter 4, the trajectory planning problem is stated as an OCP. Physical limitations of the vehicle combination and an attempt to state the properties of an desirable trajectory are explained. Then, two optimal control problems are stated to describe the trajectory planning in both the longitudinal and lateral dimension. At the end of the chapter, the software ACADO, which is used for stating and generating C-code for the optimal control problem, is presented together with other implementation details.

Simulation results in a pre-defined highway scenario will be presented in chapter 5. In addition, a discussion regarding the performance of the trajectory generator with respect to lane changing capability, traffic handling and real-time performance is presented.

Lastly, chapter 6 presents the main conclusions as well as recommendations for future work.

2
Numerical optimization

The field of numerical optimization is broad and is used in many different areas, from airline companies deciding how to route their planes and utilize their crew in the most efficient manner [9], to how to control the temperature in the process of freezing food [10]. In this chapter, a brief introduction to optimization is given in section 2.1 and the property of convexity is explained in section 2.2. Furthermore, the basic concept behind model predictive control is presented in section 2.3.

2.1 Introduction to numerical optimization

The basis for numerical optimization is to find a minimum of a function f, called the *objective function*, with respect to the *optimization variable(s)*. The objective function could be a composition of different variables that should be minimized in a system. The objective function can also be subject to a set of constraints that often represents the dynamics and limits of the system. These constraints can be divided into two sets, equality constraints, often describing the dynamics of the system, and inequality constraints, often describing limitations in the system. This can more formally be described as

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0, \\ & g(x) \leq 0 \end{aligned} \tag{2.1}$$

where *f(x)* is the aforementioned objective function, x is the optimization variable, *h(x)* is the set of equality constraint and *g(x)* is the set of inequality constraints. The set of solutions satisfying the constraints is called the *feasible set*, $\mathcal{F}$, and the set not satisfying the constraints is called the *infeasible set*, $\mathcal{I}$. In cases where $\mathcal{F} = \emptyset$ the problem is said to be infeasible, which means no solution satisfying the constraints exists.

Depending on the formulation of the objective function and constraints, different types of optimization problems can be stated. One example is a *linear program*, where the objective function and the constraint functions are linear. This type of optimization problem can efficiently be solved with for example the Simplex method [11]. Optimization problems that are non-linear in the objective function and/or the constraints are called *non-linear programs* and one example of a non-linear program is the *quadratic program*, or *QP* for short. In a QP formulation, the objective function is quadratic and the constraints are linear. This is described as:

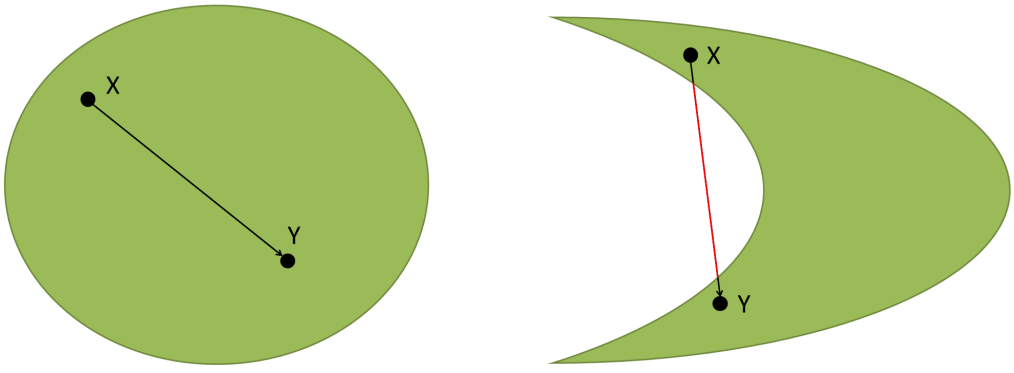
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}x^T Q x + c^T x \\ \text{subject to} \quad & Hx = k_{eq}, \\ & Gx \leq k_{in} \end{aligned} \tag{2.2}$$

In this formulation x is a vector containing the optimization variables, Q is a weight matrix and c is a vector for handling linear contributions to the problem. H and G are matrices used to state the equality and inequality constraints for the problem and vectors k_{eq} and k_{in} are used to define the equalities and inequalities. Finally, this formulation is said to be *convex* if Q is positive definite. To solve a QP, a variety of methods exists such as the *interior point method* or the *active set method* [12].

2.2 Convexity

To understand which types of optimization problems that can be efficiently solved, the notion of *convexity* needs to be introduced. Convexity is a mathematical property that applies to both functions and sets and can be similarly explained for both functions and sets.

A *convex set* can be seen as an area where a straight line can be drawn between any two points in the area, without leaving the area. Conversely, a *non-convex set* can be seen as an area where a straight line can be drawn from two points in the area with the line exiting the area at some instance. A graphical representation of a convex and a non-convex set can be seen in fig. 2.1 where a line, XY, has been drawn between two points, X and Y, in each set. The same concept applies to convex functions, where a straight line between any points on the curve of the function never should cross the curve.



(a) Convex set. **(b)** Non-convex set.

Figure 2.1: Difference between a convex and a non-convex set.

When optimizing over a convex set of solutions, if a minimum is found, it is the global minimum. When using non-convex sets, all the minimums need to be evaluated against each other, determining which of the minimums are only local minimums

and which one is the global minimum. The use of convex sets hence reduces the complexity and the algorithm will be more efficient. In this thesis, only convex sets and functions will be considered. This is because of the real-time applicability of convexity. For further reading on convexity, see [11].

2.3 Model predictive control

Model predictive control, or *MPC* for short, is a way of utilizing optimization in the area of control theory by stating the control problem as an optimization problem. This formulation is often referred to as an OCP. With this method, optimal control signals can be generated based on forward simulation of the system. More specifically, MPC utilizes the models and constraints of the system it is applied to, in order to optimize the control signal for the next time-step. By simulating the evolution of the system over a fixed interval, called the *prediction horizon*, an optimal control sequence is generated for that horizon. Also, the prediction horizon changes based on the *receding horizon idea*. The author of [13] explains the idea as follows: At the current time step, information about the current state of the system and the information about the future states from the prediction, are used to generate an optimal sequence of control signals. The first of these control signals, corresponding to the next time step, is then applied to the real system. At the next time instance, the prediction horizon has moved one time instance further and the procedure is repeated.

For a graphical interpretation of MPC, see fig. 2.2, where an MPC tries to minimize an offset to a reference trajectory. When the optimization at time step k is done, the control input calculated for time step k is applied to the system and the current time step is shifted to k+1 and the prediction horizon to k+p+1.

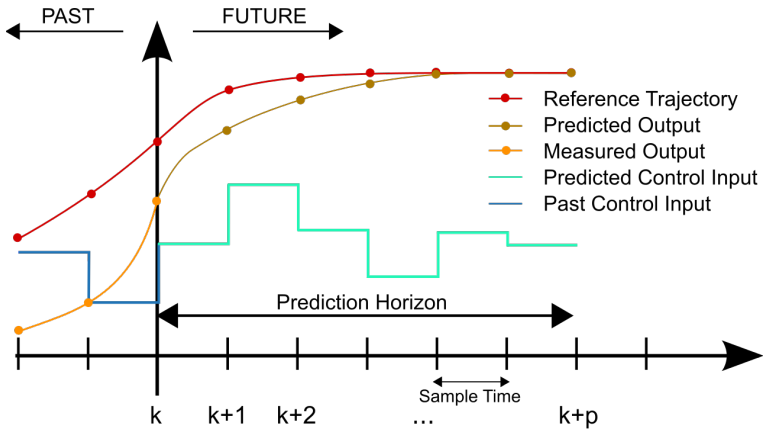


Figure 2.2: The principle of MPC for the current time step k, from [3].

In order to be able to steer the states of a system, x, to desired values, r, the optimization problem in the MPC is formulated as a QP, giving the following objective function

$$\underset{u}{\text{minimize}} \quad (x-r)^T Q(x-r) \tag{2.3}$$

This way, the lowest possible value of the objective function is obtained when the states are equal to the desired values. In practice this would mean zero offset tracking. An added benefit for this choice of objective function is that if the set of values satisfying the constraints is convex, the entire problem is convex and can thus be efficiently solved. This results in the following OCP formulation:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \frac{1}{2}\mathbf{z}^T Q\mathbf{z} + c^T\mathbf{z} \\ \text{subject to} \quad & A_{eq}\mathbf{z} = \mathbf{b_{eq}}, \\ & A_{in}\mathbf{z} \leq \mathbf{b_{in}} \end{aligned} \tag{2.4}$$

where $\mathbf{z} = \begin{bmatrix} x_{k+1}, x_{k+2}, \ldots, x_{k+p}, u_k, u_{k+1}, \ldots, u_{k+p-1} \end{bmatrix}^T$ is a vector with the states and control signals for the prediction horizon, c is a real valued vector, Q is a real symmetric matrix and c is a real valued vector. The matrix A_{eq} and the vector b_{eq} are used to impose equality constraints on the controller. Among these constraints, the actual differential equations used to describe the system are included along with other possible equality constraints. Finally, the matrix A_{in} and the vector vector b_{in} are used to impose in-equality constraints on the controller.

Depending on the linearity of the model dynamics and constraints used, MPC is usually divided into two groups, linear and non-linear MPC. If only linear dynamics and constraints are present, this is simply referred to as MPC whereas if non-linearities are present, it is usually referred to as *non-linear MPC*, or *NMPC* for short. The method for solving NMPC differs a bit from solving regular MPC and the most successful technique is the *sequential quadratic programming method*, or *SQP* for short [14]. For more information on how SQP works, the reader is referred to the work conducted in [14]. Finally, for the sake of clarity, when discussing MPC in this master thesis report, only linear MPC is considered.

2.4 Summary

In this chapter, the basics behind numerical optimization is explained and the concept of convexity was presented. These theories were then used in the model predictive control framework which is used for creating algorithms for the trajectory planner. In this way, a trajectory planner that will ensure that the constraints from the environment and the vehicle dynamics are kept, can be constructed.

In the next chapter, models for the lateral and longitudinal dynamics of the truck, the road and the surrounding traffic will be introduced. These models will be used in the algorithms for the trajectory planner, but also when the algorithms will be tuned and evaluated.

3
Modelling

As discussed in the previous chapter, a system model is a vital part in MPC. Furthermore, in order to test and verify the working capabilities of the final trajectory planner by simulation, a model description of the complete system is needed. However, since the necessary vehicle models used in this project already have been developed within Volvo GTT, this has not been the main focus in this project and the derivations will not be covered extensively. The vehicle modelling has been divided into two models, with one model describing the lateral dynamics and one model describing the longitudinal dynamics for the A-double combination.

In the first section of this chapter, the longitudinal and lateral models of the A-double combination, together with some important lateral phenomena, will be presented. After that the modelling approach for the road will be discussed in section 3.2 followed by the modelling of the vehicle positioning with respect to the road in section 3.3. In section 3.4, the modelling of fellow traffic participants is presented. The high-fidelity plant of the A-double combination that is used for simulations and tuning of the trajectory planner is presented in section 3.5. Finally, section 3.6 will provide a short summary of the chapter.

3.1 Vehicle modelling

The requirements for the model-fidelity of the lateral and longitudinal dynamics differ quite much. Highway maneuvers are more sensitive to lateral movements than longitudinal since the lateral safety margins are generally smaller than the longitudinal safety margins. For example, the lateral distance to road limits or other traffic participants is smaller than the longitudinal distance to vehicles ahead. Thus a more accurate lateral model is needed. When modelling the longitudinal motion, all forces, except the gravitational force, have been disregarded and the dynamics are therefore rather simple. Because of this, a double integrator has been chosen to model the longitudinal velocity and acceleration. The lateral motions and rotations, on the other hand, are much more complex to model as the model has to include several interconnected units with one degree of rotational freedom. As a consequence of this, a single track model [15] has been used in this project.

However, the single track model is highly non-linear as it for example uses the longitudinal velocity when calculating many of the vehicle states. Though, under a set of simplifications that can be assumed for the highway environment, the single track

model can be stated linearly. This makes it possible to use the MPC framework and state the trajectory planning problem as a convex QP.

Furthermore, given the length and articulation points of the A-double combination, some non-trivial lateral phenomena are introduced which needs to be handled. Before presenting a model for the lateral dynamics, these phenomena will first be discussed.

3.1.1 Lateral phenomena

The lateral phenomena that are introduced due to the length and articulation points of the truck are rearward amplification, RA, and lateral off-tracking. RA can generally be described as the lateral motion of the last unit divided by the lateral motion of the first unit [16]. In fig. 3.1, RA is demonstrated and it can be seen that the lateral acceleration for the fourth unit is amplified compared to the lateral acceleration of the first unit. This occurs when the truck executes maneuvers like lane changing or evasive lateral movements. In high speeds the RA can become quite high, especially when executing evasive maneuvers.

High RA increases the risk that any of the units in the vehicle combination will roll over due to the lateral acceleration, but, can also be dangerous as the swept path by the units will become larger and risk sliding into the wrong lanes or off road. Most notably, the last trailer faces this risk the most as the RA becomes greater further back in the vehicle combination, and, in [17], it was found that the RA for an A-double combination could be up to 1.74.

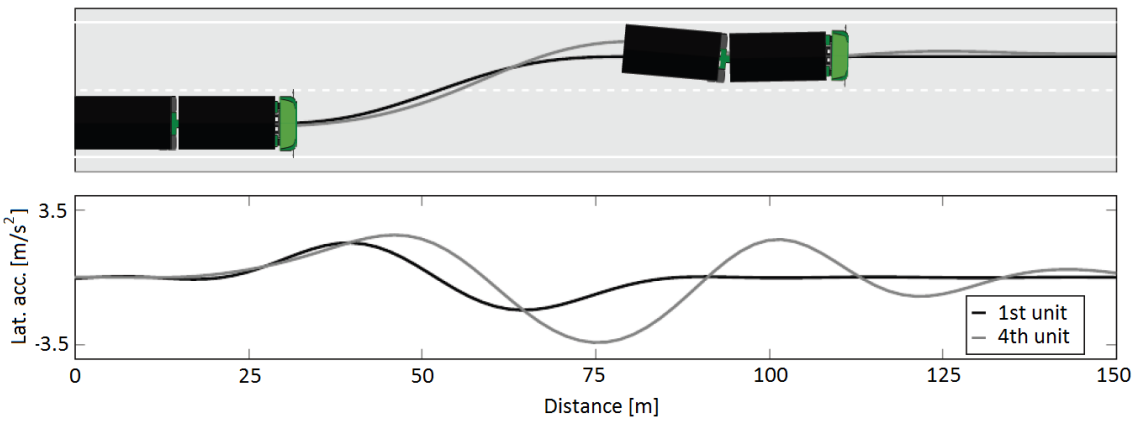


Figure 3.1: Demonstration of rearward amplification. Cropped from [4].

The second phenomenon introduced, lateral off-tracking, can be defined as the distance between the traveled path of the center point of the innermost axle and the traveled path of the center point of the outermost axle [15] when driving in a curve. In practice, this will most of the times be the distance between the front axle of the truck and the rear axle of the second semi-trailer and one example is seen in fig. 3.2. It shows how the front and rearmost axles of the truck travels different paths when

the truck is going through a curve at high speed.

One thing that is worth pointing out is that the direction of the off-tracking between the tractor and second semi-trailer will depend on the velocity that the truck is travelling in. In high speeds, off-tracking as seen in fig. 3.2 will occur, that is that the trailers will sweep outwards from the curve. However, when travelling at low speeds, the trailers will instead sweep inwards in the curve that the truck is travelling in. This inwards off-tracking typically starts to occur at speeds below 40 $[km/h]$ [2].

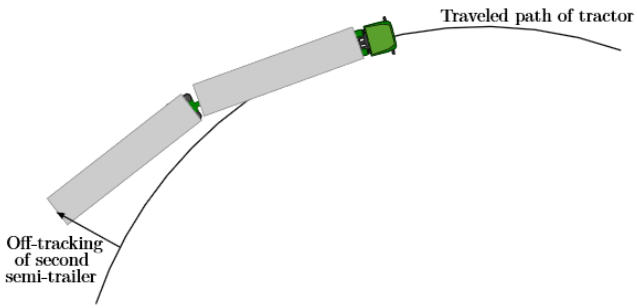


Figure 3.2: Different trajectories for the first axle and the last axle of the truck. From [2].

3.1.2 Lateral vehicle modelling

For the trajectory planner to be able to plan trajectories that are safe and smooth, it needs a vehicle model that captures the motion characteristics of the A-double combination in a highway environment. These motions could be the phenomena described earlier or other movements that occur when executing a lane following or a lane change maneuver. Also, for the trajectory planner to be able to run in real-time, the complexity of the model must be kept on a level so that it is able to be efficiently computed.

In this project, the lateral dynamics of the truck is described by a single track model including a linear tire model. The single track model is a common choice when modelling a vehicles basic cornering response [5] and is expected to provide a good compromise in complexity and performance for highway driving [2]. The single track model combines the tires of each axle into a virtual tire on one virtual axle, hence the name single track, see fig. 3.3.

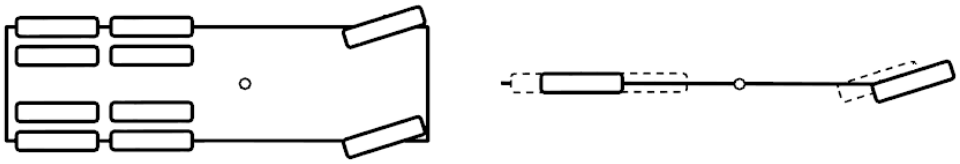


Figure 3.3: Simplification from a real truck, to the left, to a single track model, to the right. Modified from [4].

In this project, the single track model used for modelling the A-double combination is based on the model derived in [5]. The derived model is validated for high speed vehicle cornering, typically in the speed range of 30-80 $[km/h]$ with lateral and longitudinal acceleration levels below 1.5 $[m/s^2]$ and 3 $[m/s^2]$ respectively. The A-double single track model can be seen in fig. 3.4. Additionally, the parameter values used in this project for the single track model can be found in appendix A.

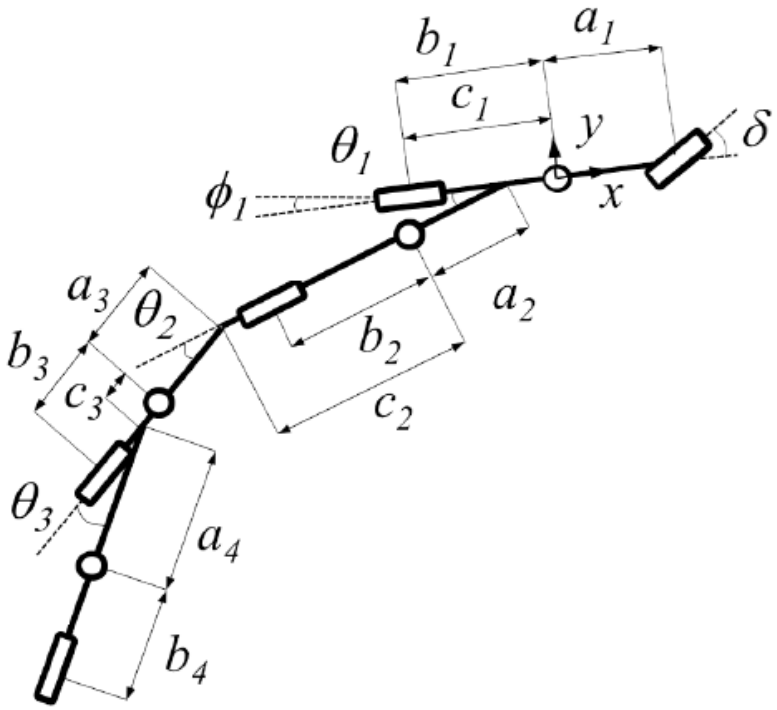


Figure 3.4: A-double single track model for an A-double combination with parameters. From [5].

In order to, if ever so slightly, familiarize the reader with the derivation of this model, a brief summary from [5] will now be presented. The model is derived by using the Euler-Lagrange approach to generate the equations of motion for the A-double combination. This results in a non-linear representation that can be linearized under a set of simplifications:

- As the states in the single track model depends on the longitudinal velocity of the truck, it will not be treated as a model state in the lateral model, but instead as a time varying constant.
- Articulation angles between the units and steering angles are assumed to be small in the highway environment. This allows for using the small angle approximation and also to set the products of steering and articulation angles and their time derivatives to zero.
- The tires are used far from their performance limits in the highway environment, thus linear tire forces can be assumed.

For a fully detailed derivation of the model and the simplifications, the reader is

referred to [5].

In this model, the following states are defined to describe the lateral dynamics of the truck:

- The lateral velocity of the tractor: $\mathbf{v_{y,1}}$, $[m/s]$.
- Yaw angle of the tractor: $\boldsymbol{\phi_0}$, $[rad]$.
- Yaw angle rate of the tractor: $\dot{\boldsymbol{\phi}}_\mathbf{0}$, $[rad/s]$.
- Articulation angle between tractor and the first trailer: $\boldsymbol{\theta_1}$, $[rad]$.
- Articulation angle rate between tractor and the first trailer: $\dot{\boldsymbol{\theta}}_\mathbf{1}$, $[rad/s]$.
- Articulation angle between the first trailer and the dolly: $\boldsymbol{\theta_2}$, $[rad]$.
- Articulation angle rate between the first trailer and the dolly: $\dot{\boldsymbol{\theta}}_\mathbf{2}$, $[rad/s]$.
- Articulation angle between the dolly and the second trailer: $\boldsymbol{\theta_3}$, $[rad]$.
- Articulation angle rate between the dolly and the second trailer: $\dot{\boldsymbol{\theta}}_\mathbf{3}$, $[rad/s]$.
- Steering angle of the truck: $\boldsymbol{\delta}$, $[rad]$.

These states are controlled by the steering angle rate $\dot{\boldsymbol{\delta}}$, $[rad/s]$, and the following state and control vectors can be stated:

$$\zeta_{\mathbf{lat}} = \left[v_{y,1}, \phi_0, \dot{\phi}_0, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \theta_3, \dot{\theta}_3, \delta\right]^T, \quad \mathbf{u_{lat}} = \left[\dot{\delta}\right]^T$$

The time derivatives of these states are based on the derived equations in [5] and its structure looks like:

$$\dot{\zeta_{\mathbf{lat}}} = \begin{bmatrix} a_{11} & 0 & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} & a_{110} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} & a_{310} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} & a_{510} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_{71} & 0 & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} & a_{710} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ a_{91} & 0 & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} & a_{910} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \zeta_{\mathbf{lat}} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{u_{lat}} \quad (3.1)$$

For a more detailed formulation of the states see appendix B.

3.1.3 Longitudinal vehicle modelling

As previously discussed, the modelling of the longitudinal dynamics of the A-double combination have been chosen to disregard all resistive forces but the gravitational force, and thus rather simple dynamics are obtained. This means that this model does not include any truck specific characteristics, such as gearbox dynamics, mechanical losses or engine efficiency. Moreover, it does not include any models of aerodynamic drag or rolling resistance. The reason for this is to try to keep the overall model complexity in the OCP formulation down, as a real-time implementation of the controller is in mind. The gravitational force is kept in the model in order to take the effects of road topography into consideration. Hence, the A-double combination has been modeled as a double integrator to describe the longitudinal velocity and acceleration. The reasoning behind choosing a double integrator is that

it provides a nice trade-off in model complexity and accuracy for highway driving.

The A-double combination is modelled to have three states:

- The longitudinal velocity of the tractor: $\boldsymbol{v_{x,1}}$, $[m/s]$.
- The longitudinal acceleration of the tractor: $\boldsymbol{a_{x,1}}$, $[m/s^2]$.
- The desired acceleration of the tractor: $\boldsymbol{a_{x,1des}}$, $[m/s^2]$.

The desired longitudinal jerk, $j_{x,1des}$, $[m/s^3]$, has been chosen as the control signal, which results in the following state and control vectors for the longitudinal model:

$$\gamma_{\mathbf{long}} = \Big[v_{x,1}, a_{x,1}, a_{x,1des}\Big]^T, \quad \mathbf{u_{long}} = \Big[j_{x,1des}\Big]^T$$

Furthermore, the time derivatives of each state can be stated as:

$$\frac{dv_{x,1}}{dt} = a_{x,1} - g\sin(\alpha_{road}) \tag{3.2}$$

where g is gravitational acceleration and α_{road} is the slope of the road measured in $[rad]$. In order to linearize the state equation for the time derivative, small angle approximation is used. Furthermore, a motivation on why the small angle approximation is possible is found in section 3.2.2. The change in acceleration of the truck is modelled with an inertia. This to simulate the time delay from request to actuation:

$$\frac{da_{x,1}}{dt} = \frac{a_{x,1} - a_{x,1des}}{\tau} \tag{3.3}$$

where τ is the response time for cruise controller of the truck. Finally, the time derivative of the desired acceleration is set to be the input of the system.

$$\frac{da_{x,1des}}{dt} = j_{x,1des} \tag{3.4}$$

3.2 Road modelling

In this project, highway driving in Sweden is considered as the operational domain for the controller. Given this, the road characteristics under which the controller will operate can be stated. This section is mainly based on [4] and [18].

3.2.1 Horizontal road geometry

When building highways in Sweden, [18] state that three different geometric shapes are used for horizontal design, namely straight lines, circular arcs and clothoids. Straight lines are defined as horizontal curves with infinite radii. Circular arcs are used for horizontal curves and have a constant curvature with a minimum radii of 800 [m]. When transitioning between straight line segments and horizontal curves, clothoids are used to create smooth transitions. A clothoid is a line segment in which the curvature of the line increases along the length of the line, which is how a smooth transition can be created. This concept is visualized in fig. 3.5 where line segment 1 is a straight line, segment 2 is a clothoid and segment 3 is a circular arc

with radii 30 $[m]$.

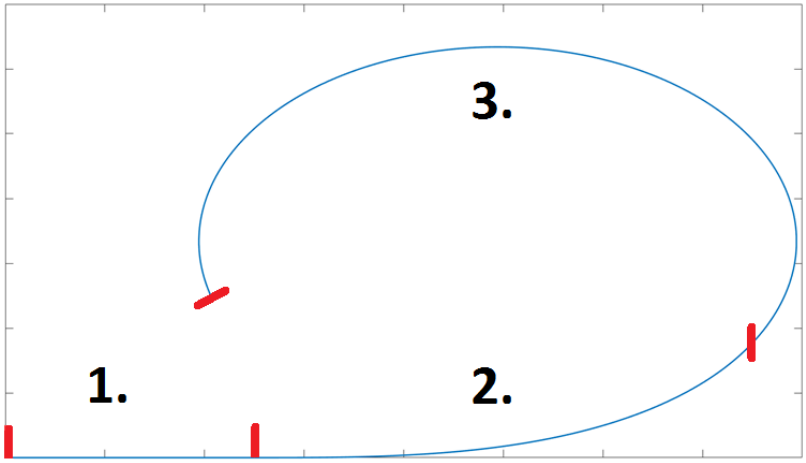


Figure 3.5: Design concept for horizontal curves on Swedish highways, where segment 1 is a straight line, segment 2 is a clothoid and segment 3 is a circular arc.

The geometric shape design principles are applied in this project when modelling the horizontal road profile. The road model used defines the curvature of the road at certain distances along the road geometry from a given origin point. With this description, a transformation from curvature to road heading in each point is then possible. This will later prove to be important when the model of the vehicle combination and the road are going to be connected, more on this in section 3.3.

3.2.2 Vertical road geometry

The requirements for vertical design are also stated in [18]. The allowed geometric shapes for vertical design are straight lines, circular arcs and parabolas, where only the latter two are allowed when designing vertical curves. The minimum concave radii allowed for vertical curves ranges from 2000-6500 $[m]$ depending on the length of the curve and whether or not road lighting exists. The vertical slope of the road, $slope_{road}$, is defined in [%] and is calculated as the relationship between the vertical distance traveled divided by the horizontal distance traveled. Furthermore, the maximum slope allowed for Swedish roads is 8%, and to obtain the angle which the road slopes, α_{road}, the following equation can be used:

$$\alpha_{road} = \tan^{-1}(slope_{road}) \tag{3.5}$$

This means that the maximum slope allowed for Swedish roads is 4.5739 degrees, which is why the small angle approximation has been used. The vertical road modelling in this project follows the same structure as the horizontal road modelling, i.e., that sloping angles are defined at certain distances from a given origin point.

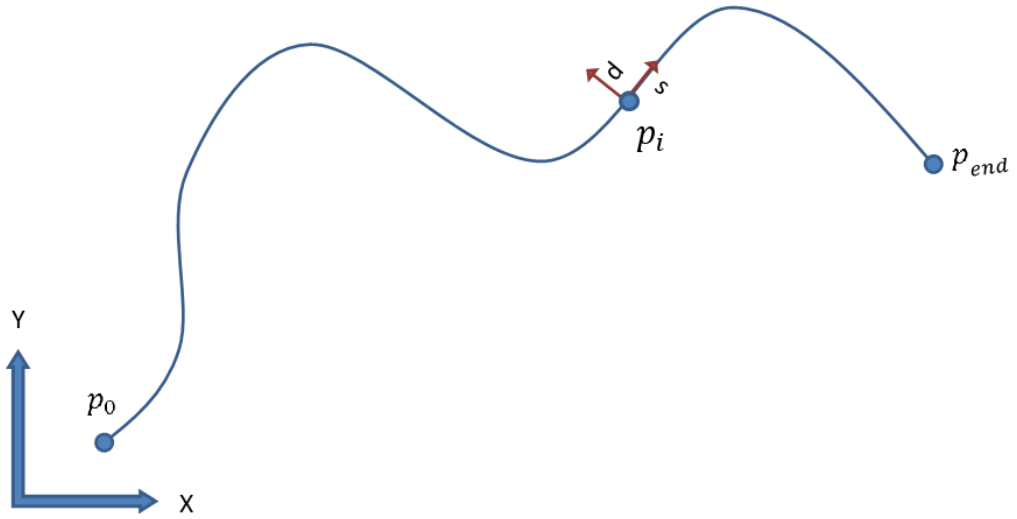


Figure 3.6: Visual representation of a road description in the global coordinate system from position point p_0 to p_{end}. Along the description an arbitrary position point p_i is visualized, demonstrating the local coordinate frame present in each point along the road description.

3.3 Vehicle positioning with respect to the road

Now that models for the vehicle and road description have been presented, they need to be connected. First, a measure to relate the position of the vehicle combination compared to the road geometry is needed. This is done by introducing the auxiliary states d_1, d_4, s_1 and s_4, all defined in meters.

The states s_1 and s_4 each describe the length along the description of the road from a given position p_0. If slightly simplified, it can be said that s_1 and s_4 are the distances travelled along the clothoid description of the road, from a given starting point p_0, for the center of mass of the tractor and the rearmost axle of the second semi-trailer respectively. The states d_1 and d_4 describe the perpendicular offset from the road geometry to the center of mass for the truck and to the center of the rearmost axle of the second semi-trailer respectively. In fig. 3.6, a visual representation of a road description together with an arbitrary point i can be seen. In the position point p_i, the local coordinate system describing the directions of s and d is visualized.

As mentioned in section 3.2.1, the road model is defined as curvatures at certain points along the road. By interpolating between these points, it is possible to obtain the curvature along the entire road. Then, by transforming the curvature into the angle the road has compared to the global X axis, the heading can thus also be obtained for the entire road. To describe the heading angle of the road in an arbitrary point p_i, this is done as:

$$\phi_{road}(p_i) = \sum_{k=1}^{i} \phi_{road,k-1} + (p_k - p_{k-1})c_k \tag{3.6}$$

where p_k is the total distance travelled along the road geometry from a starting point p_0 and c_k is the curvature at that point. With this formulation, the heading of the entire road can be placed in a look-up table that is mapped to the distance travelled along the road description.

The information regarding road heading is important as the states d_1 and d_4 depend on this information at the positions s_1 and s_4. However, before stating the expressions for d_1 and d_4, expressions for the states s_1 and s_4 evolution over time are required.

$$\frac{ds_1}{dt} = v_{x,1} \tag{3.7a}$$

$$\frac{ds_4}{dt} = v_{x,1} \tag{3.7b}$$

Their first order differential equations are chosen to simply be the longitudinal velocity. This however assumes that the truck follows the road description without any major deviations so that the longitudinal velocity of the truck also is the velocity in which it travels along the road geometry. With this information, the expressions for d_1 and d_4 can be formulated as following:

$$\frac{dd_1}{dt} = v_{x,1}\sin(\phi_0 - \phi_{road,1}) + v_{y,1}\cos(\phi_0 - \phi_{road,1}) \tag{3.8a}$$

$$\frac{dd_4}{dt} = v_{x,4}\sin(\phi_4 - \phi_{road,4}) + v_{y,4}\cos(\phi_4 - \phi_{road,4}) \tag{3.8b}$$

where $\phi_{road,1}$ and $\phi_{road,4}$ describes the heading angle of the road in the points defined by the states s_1 and s_4 and ϕ_4 defines the heading angle for the second semi-trailer.

$$\phi_{road,1} = \phi_{road}(s_1) \tag{3.9a}$$

$$\phi_{road,4} = \phi_{road}(s_4) \tag{3.9b}$$

Important to note, is that this formulation of the lateral offset is not linear as both the states include sinusoidal components. Furthermore, the heading angle and the longitudinal and lateral velocities for the second semi-trailer are not available in our state vector.

The sinusoidal components in the lateral offset states are removed by, again, using the assumption that the truck follows the road. Then the difference between ϕ_0 and $\phi_{road,1}$ and between ϕ_4 and $\phi_{road,4}$ become small and the small angle approximation is again applicable. The heading angle for the second semi-trailer can also be obtained rather easily, this by adding the heading angle for the truck with all the articulation angles between the tractor, trailers and the dolly.

$$\phi_4 = \theta_0 + \theta_1 + \theta_2 + \theta_3 \tag{3.10}$$

Since the truck is assumed to follow the road, $v_{x,4}$ is assumed to be equal to $v_{x,1}$ since all connection points are rigid. The only unknown variable left is thus the lateral velocity of the second semi-trailer. This can be obtained by propagating the

lateral velocity in the truck, through the vehicle combination, back to the second semi-trailer. This has been done in the same manner as in [2] and for a more in depth explanation the reader is advised to study the referenced material.

With these simplifications and re-writings, the resulting equations for dd_1/dt and dd_4/dt can be written linearly as:

$$\frac{dd_1}{dt} = v_{x,1}(\phi_0 - \phi_{road,1}) + v_{y,1} \tag{3.11a}$$

$$\begin{aligned}\frac{dd_4}{dt} = v_{x,1}(\phi_4 - \phi_{road,4}) + \Big(&- (a_2 + a_3 + a_4 + b_4 + c_1 + c_2 + c_3)\dot{\theta}_0 \\ &- (a_2 + a_3 + a_4 + b_4 + c_2 + c_3)\dot{\theta}_1 - (a_3 + a_4 + b_4 + c_3)\dot{\theta}_2 \\ &- (a_4 + b_4)\dot{\theta}_3 + v_{y,1} - \theta_1 v_{x,1} - \theta_2 v_{x,1} - \theta_3 v_{x,1}\Big)\end{aligned} \tag{3.11b}$$

Now, the linear auxiliary states can be added to the lateral model so that it is possible to relate the vehicle combination to the road. This results in the following, final, state and control vector for the lateral model.

$$\zeta_{\mathbf{lat}} = \Big[v_{y,1}, \phi_0, \dot{\phi}_0, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \theta_3, \dot{\theta}_3, \delta, s_1, d_1, s_4, d_4\Big]^T, \quad \mathbf{u_{lat}} = \Big[\dot{\delta}\Big]^T \tag{3.12}$$

3.4 Traffic modelling

Another aspect of highway driving that needs to be handled is modelling of other traffic participants, this as the highway very seldom is free from other vehicles. Only one major distinction has been made between different types of traffic participants, that is whether it is classified as a truck or a car. This difference in object classification will affect the objects retardation capabilities. In this project the values are set to be representative, where cars are modelled to be able to decelerate with 7 $[m/s^2]$ whereas trucks are modelled to only be able to decelerate with 5.9 $[m/s^2]$.

In this project, up to three objects will be taken into consideration for the trajectory planner. All objects are modelled with a single integrator to describe their longitudinal dynamics, to drive with constant velocity and are assumed to stay in the same lane as they start. This means that no lateral model is required and the following longitudinal model can be stated for an arbitrary object i:

$$\frac{d\Delta p_{x,i}}{dt} = \Delta v_{x,i} \tag{3.13a}$$

$$\Delta v_{x,i} = v_{x,ego} - v_i \tag{3.13b}$$

where $\Delta p_{x,i}$ is the distance from object i rearmost point to the front of the ego vehicle, measured in $[m]$, if the object is in front of the ego vehicle. If the object is behind the ego vehicle, $\Delta p_{x,i}$ is then the distance from the rearmost point of the second semi-trailer to the front of object i, measured in $[m]$. Note that in this case eq. (3.13b) changes sign. $\Delta v_{x,i}$ is the difference in velocity of object i compared to the velocity of the ego vehicle measured in $[m/s]$. These states are used as auxiliary

states in the longitudinal model. With this addition, the final state and control vector for the longitudinal model can be stated as:

$$\gamma_{\mathbf{long}} = \left[v_{x,1}, a_{x,1}, a_{x,1des}, \Delta p_{x,1}, \Delta p_{x,2}, \Delta p_{x,3}\right]^T, \quad \mathbf{u_{long}} = \left[j_{x,1des}\right]^T \tag{3.14}$$

3.5 High-fidelity plant

To evaluate and tune the trajectory planner, a high-fidelity plant model of an A-double has been used. This model is implemented in Matlab/Simulink and is developed and validated by Volvo Group Trucks Technology. It models all of the axles and its tires independently and provides a detailed description of the vehicle chassis, suspension, steering system and powertrain. Furthermore, the high-fidelity plant provides more realistic results compared to the single track plant and will thus give an indication on how the trajectory planner would perform in a real truck.

3.6 Summary

In this chapter, an introduction to the models that are used in this project were given. It describes the difference between the longitudinal and lateral modelling and the phenomena of lateral off-tracking and rearward amplification. Furthermore, how the horizontal and vertical geometries of the road has been modelled are discussed followed by a description on how the relationship between the road and the truck is modelled. In addition, the modelling of fellow traffic participants are presented. Finally, a brief introduction to the high-fidelity plant developed by Volvo Group Trucks Technology, which is used to evaluate and tune the trajectory planner, is presented.

In the next chapter, the theory in chapter 2, will be concatenated with the models described in this chapter to state the trajectory planning problem as an optimal control problem.

4

Trajectory planning as an optimal control problem

The size and weight of the A-double combination presents some challenges for the trajectory planner. These properties, along with the lateral phenomena discussed in section 3.1.1, the road description and traffic environment needs to be handled by the trajectory planner in order for it to plan safe and smooth trajectories. In this chapter, the theory and models in chapter 2 and 3 respectively, will be concatenated in order to formulate the trajectory planning problem as two MPCs.

In this project, real-time performance is a key property. Therefore, the longitudinal and lateral planning problem has been separated into two loosely coupled MPC formulations. The separation makes it possible to linearize the lateral vehicle model under the assumption of constant longitudinal velocity in each time step, while the longitudinal velocity can be modeled as a state in the longitudinal planner. Both the OCP in the longitudinal and in the lateral MPC will be stated as a convex QP problem with a quadratic objective functions and linear constraints.

This chapter will start with a description on how and why the trajectory planner has been separated into one longitudinal and lateral trajectory planner in section 4.1. After that, a discussion about the definitions of safe and smooth trajectories will be presented in section 4.2. Section 4.3 and 4.4 will present the formulation of the OCP for the longitudinal and lateral planner respectively. Thereafter, the complete trajectory planner will be presented in section 4.5. Lastly, a short summary of this chapter will be presented in section 4.8.

4.1 Separation of lateral and longitudinal trajectory planning problem

The fundamental problem the trajectory planner is designed to solve, is to plan trajectories in the lateral and the longitudinal dimension without violating any constraints. This is done by calculating the desired jerk of the truck in order to obtain some predefined, desired, longitudinal velocity. Similarly, the desired steering angle rate is calculated so that a, predefined, desired lateral position can be achieved. As presented in section 2.3, given a model and an objective function, a control signal can be calculated by numerical optimization and simulation of the model by using the MPC framework. The same concept also applies if two models are used and

two control signals were to be calculated by a single MPC. By inserting the lateral and longitudinal models, combined with auxiliary states, and a suitable objective function into a single MPC, the jerk and the steering angle rate of the truck could be calculated simultaneously.

However, as discussed in section 3.1.2, the lateral model needs a constant longitudinal velocity in order for the dynamics to be linear. If both the longitudinal and lateral model would be present in the same MPC, this would mean that the longitudinal velocity would be a vehicle state and linear lateral dynamics would thus not be possible, as discussed in section 3.1.2. To solve this issue, the trajectory planner has been separated into two, one planning the longitudinal trajectory and one planning the lateral trajectory. Recall from section 2.3 that the MPC calculates a control signal for each time step over the entire prediction horizon. This means, that by first calculating the desired jerk of the truck, the resulting longitudinal velocity is also available in each time step over the prediction horizon. These longitudinal velocities can then be passed to the lateral trajectory planner and be inserted into the problem at the correct time instant. In this manner, the longitudinal velocity can be used as a time varying constant in the lateral trajectory planning problem and linear dynamics can be ensured. In fig. 4.1, a schematic overview of the separation can be seen.

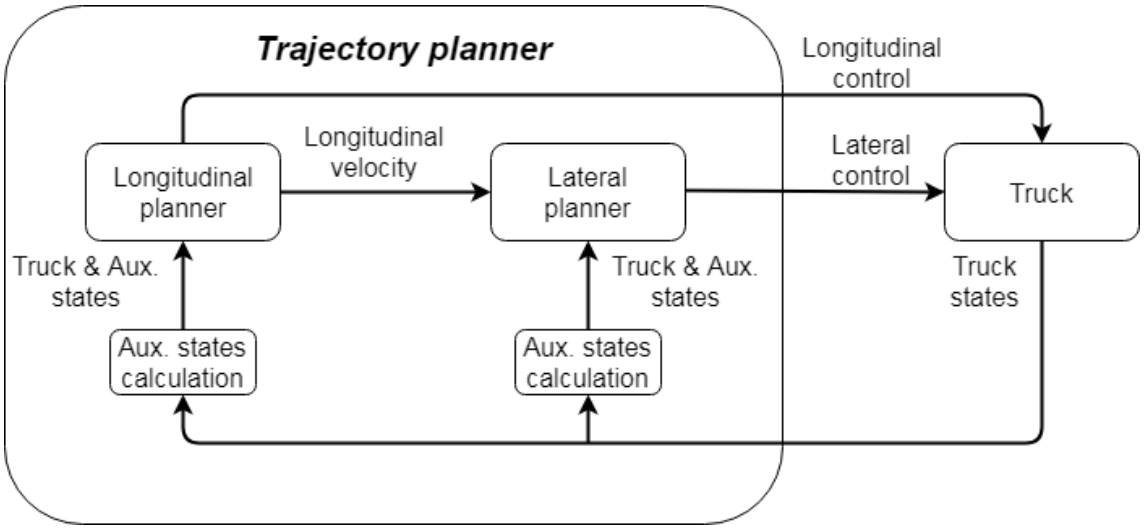


Figure 4.1: Separation of the trajectory planner into a lateral and longitudinal trajectory planning problem.

Important to note is that the auxiliary states calculated for the lateral planner are the auxiliary states added to the lateral model presented in section 3.3 and equivalently, the auxiliary states calculated for the longitudinal planner are the auxiliary states added to the longitudinal model in section 3.4. Furthermore, if not otherwise stated, the term *trajectory planner* will henceforth refer to the combination of both the lateral and the longitudinal trajectory planner.

4.2 Desirable trajectories

Now that the structure of the trajectory planner has been presented, definitions regarding what a desirable trajectory is needs to be presented for both the lateral and the longitudinal planning. These definitions have been divided into two categories, safe and smooth trajectories, and will now be presented. Furthermore, as the actuators have limited capabilities, a set of actuator constraints will also be presented.

4.2.1 Safe trajectories

Safety is a very important aspect which the trajectory planner must handle. To generate safe trajectories, there are some sets of constraints, both laterally and longitudinally, that needs to be taken into account. These constraints will now be presented together. Note however that they will only affect their corresponding trajectory planner.

4.2.1.1 Lateral acceleration

To limit the additional swept path of the trailers and to make sure that none of the trailers will roll over, the lateral acceleration will be constrained. The static roll over limit for passenger cars is typically around 1.1 g, however, due to the relatively higher center of gravity, the static roll over limit for heavy trucks can be as low as 0.35 g [4]. This is equal to an acceleration of roughly 3.44 $[m/s^2]$. The absolute value of the lateral acceleration, in both the truck and the second semi-trailer, is therefore conservatively limited to be below 2.5 $[m/s^2]$.

The reason for why a limit on both the truck and second semi-trailer is imposed is because of RA. If a lateral acceleration of only 2 $[m/s^2]$ would be applied to the truck, the RA could amplify this acceleration so that an acceleration of 3.48 $[m/s^2]$ would affect the second semi-trailer. This acceleration would thus risk the trailer to roll over and is why the additional acceleration bound is present. Note that no bound is placed on any of the units between the truck and the semi-trailer. This because the RA is highest at the rearmost unit and thus provides a robust bound for all units.

It should be noted that the lateral acceleration for the tractor, $a_{y,1}$, is present in the lateral state vector as the state derivative for $v_{y,1}$. Then, in the same manner as discussed in section 3.3, the lateral acceleration for the second semi-trailer can be obtained by propagating $a_{y,1}$ through the truck. This results in the following expression for $a_{y,4}$:

$$\begin{aligned} a_{y,4} = &-(a_2 + a_3 + a_4 + b_4 + c_1 + c_2 + c_3)\ddot{\theta}_0 \\ &- (a_2 + a_3 + a_4 + b_4 + c_2 + c_3)\ddot{\theta}_1 - (a_3 + a_4 + b_4 + c_3)\ddot{\theta}_2 \\ &- (a_4 + b_4)\ddot{\theta}_3 + \dot{v}_{y,1} - \dot{\theta}_0 v_{x,1} \end{aligned} \tag{4.1}$$

4.2.1.2 Distance constraints

When the truck moves in a lane, with no intentions to leave, there are some limits to where the truck can move. These limits stem from the lane markings for the road and safety distances to fellow traffic participants that drives in the same lane as the truck. Intuitively, the lane markings on the road provides a natural set of lateral constraints for the auxiliary states d_1 and d_4. However, as other vehicles may be present in neighboring lanes, these constraints have also been set conservatively so that the truck and second semi-trailer stay within the current lane markings plus an additional safety margin of 0.2 meters, which is added to each side. The width of the tractor is approximately 2.5 $[m]$ and the lane width on Swedish highways are between 3.25-3.5 $[m]$ [18]. This means that the states d_1 and d_4 are allowed to deviate 0.175-0.3 $[m]$ in both directions from the lane center.

The longitudinal safety distance that should be kept to an obstacle i ahead is not fixed, but is instead velocity dependent. This as a constant safety distance becomes very conservative in low speeds if tuned for high velocities and dangerously low in high speeds if tuned for low speeds. This safety distance is instead designed to allow the truck to stop 5 $[m]$ behind the obstacle ahead, even if that obstacle brakes as hard as possible, and is calculated as:

$$d_{obs,i}^{safe} = v_{x,1}(t_{brake} + t_{actuate}) \tag{4.2}$$

where $t_{actuate}$ is a time constant for the system to detect that the obstacle ahead has started to brake and for the ego vehicle to start braking. This is dependent on the system setup on the truck and has therefore arbitrarily been chosen to be 0.1 $[s]$. The constant t_{brake} can be seen as the time headway the truck needs to have to the preceding obstacle, minus the actuation time, in order to be able to stop 5 $[m]$ behind the obstacle ahead if it needs to fully brake. Note that this constant will be different depending on the obstacle type that is in front of the truck as passenger cars are defined to have a higher deceleration capabilities than trucks.

4.2.1.3 Steering constraints

As the intended working domain of the controller are Swedish highways, this means that the curves have a large radius and it can be assumed that the general longitudinal velocity will be quite high. This means that the use of the full steering range of the truck will not only be unnecessary, but could also be dangerous. Large steering angles in high velocities could cause the vehicle combination to roll over or cause understeer if the road surface has low friction. The absolute value of the steering angle, δ, has therefore been constrained to be below 0.1 $[rad]$ and the absolute value of the steering angle rate, $\dot{\delta}$, has been constrained to be below 0.05 $[rad/s]$.

4.2.2 Smooth trajectories

Another important property is the smoothness of the generated trajectories from the trajectory planner. Furthermore, the truck needs to drive in a smooth, human like behaviour that ensures ride comfort for potential passengers and so it does

not cause any discomfort for other traffic participants. In this project, smoothness refers to keeping the lateral acceleration and the longitudinal jerk of the truck within moderate levels.

4.2.2.1 Tuning of lateral controller

When tuning the lateral controller, two different approaches can be used, either a tuning for precise reference tracking or a tuning for smooth lane changes. Precise reference tracking is desirable as that would imply that the vehicle combination stays within the lane boundaries and drives in the middle of the ego lane. However, when receiving a lane change request, the reference quickly changes from one lane to another. This would cause the lateral distance of the truck to be far from it's reference and would result in large control actions in order to bring the state closer to the reference quickly, which in turn would cause rapid, jerky, movements. A controller tuned for smooth lane change maneuvers would however not generate such large control signals when being far off from the reference, and a much smoother lane change can be made. This would however mean that the lane following performance of such a controller would be worse as it would not track the middle of the lane as well. Therefore, it is always a trade-off when tuning the controller for the different maneuvers. It should be noted that the parameters in the controller simply could be changed online to provide good performance for the intended maneuver. This has however not been done in this project. Instead, to be able to execute smooth lane changes while still using the tuning setup used for lane following, a pre-defined optimal trajectory is laid out and used as the lateral reference while executing a lane change. In this way, how smooth the lane change is will mostly depend on how smooth the reference trajectory is as the controller is tuned for reference tracking.

This reference trajectory is a fifth degree polynomial, also called quintic polynomial, that defines the optimal path between two points with regards to minimum lateral jerk. The polynomial can be derived by solving a minimization problem that minimizes the jerk required for a point mass, modelled with a triple integrator, to transition between a start point and an end point. The optimal jerk profile can then be used to calculate the lateral distance reference as a function of how far the truck has travelled along the path between the points. For a more detailed description on how this has been derived the reader is referred to [2] as the same implementation is used. The resulting reference polynomial for lane changing is then:

$$\begin{aligned} d_{1,ref}(s) = &-\frac{6(d_{1,0}-d_{1,1})(s-s_{lc})^5}{\Delta t_{lc}^5 v_{x,lc}^5} + \frac{5(d_{1,0}-d_{1,1})(s-s_{lc})^3}{\Delta t_{lc}^3 v_{x,lc}^3} \\ &- \frac{15(d_{1,0}-d_{1,1})(s-s_{lc})}{\Delta t_{lc} v_{x,lc}} + \frac{1}{2}(d_{1,0}+d_{1,1}) \end{aligned} \tag{4.3}$$

where $d_{1,0}$ and $d_{1,1}$ represent the lateral start and end points respectively, s_{lc} is a pre-defined longitudinal distance ahead for when the lane change should occur. Note that this distance will decrease as the truck approaches this point. s is a vector with longitudinal distances ahead, from the current position, that are mapped to a lateral distance reference, Δt_{lc} is a pre-determined time for a comfortable lane change and

$v_{x,lc}$ is the longitudinal velocity during the lane change. In this project, Δt_{lc} has been chosen to be 7 seconds.

4.2.2.2 Longitudinal jerk

The absolute value of the longitudinal jerk of the truck has been constrained to be below 2 $[m/s^3]$ so that the generated trajectories are smooth and don't cause any unnecessary discomfort for the surrounding drivers and/or any possible person(s) in the truck.

4.2.3 Actuator constraints

Apart from the set of constraints that appear from ensuring smooth and safe trajectories, an additional set needs to be added to ensure that the capabilities of the actuators are not violated. This results in two additional constraints, namely on the longitudinal velocity and on the longitudinal acceleration. These constraints depend solely on the truck the controller is implemented on and will differ from truck to truck. Therefore, the actuator constraints used in this project are not aimed at replicating any particular truck, but are instead estimations.

4.2.3.1 Longitudinal velocity

The upper boundary condition for the longitudinal velocity could be chosen to be the maximum speed limit for the truck, but has instead been set to represent the maximum capability of the truck. This enables the desired longitudinal velocity of the truck to be set freely within actuator boundaries. This means that the trajectory planner does not ensure that legal speed limits are followed. In addition, as the derived model of the A-double is validated in the speed range of 30-80 $[km/h]$, the lower bound has been set to 30 $[km/h]$ (8.33 $[m/s]$). However, the upper velocity limit has been set to 90 $[km/h]$ (25$[m/s]$) to better represent the capabilities of the A-double and as the model performance is found to be acceptable in that velocity region.

4.2.3.2 Longitudinal acceleration

The acceleration capability of a truck depend on a number of things, for example the current gear, the slope of the road and the power of the engine. Since the operational domain of the controller is on Swedish highways, the acceleration capabilities are assumed to be rather low as the velocity generally is high on highways. In addition, given the heavy weight of a fully loaded A-double combination, the acceleration capability is even further limited. Because of this, the acceleration is assumed to be no higher than 0.25 $[m/s^2]$ and the upper acceleration constraints is therefore also set to 0.25 $[m/s^2]$. This however means that the maximum slope the vehicle combination can climb, while still maintaining a constant velocity, is limited and can be calculated by using eq. (3.2). The resulting slope is then around 2.5 [%], or approximately 0.025 $[rad]$, which is significantly lower than the maximum allowed

slope on Swedish highways of 8 [%]. This is an issue that has not been dealt with in this project and a hill with greater slope than 2.5 [%] has not been used. Though, it should be noted that the current acceleration limit of 0.25 $[m/s^2]$ is based on that the truck drives in high speeds and therefore in a high gear. Higher acceleration capabilities are possible if lower gears are used, but it is not modeled in this project.

The deceleration capabilities however, are much greater and can be as high as 5.9 $[m/s^2]$. The lower constraint for the longitudinal acceleration has therefore been set to this limit.

4.3 Formulation of the longitudinal planner

In this section, the formulation of the longitudinal planner is presented. The optimal trajectory generated by the longitudinal planner follows a velocity request, whilst making sure that the longitudinal dynamics and limits of the truck are not violated. The velocity request is supplied by some external decision making functionality which also supply the longitudinal planner with lane change requests. In case of a lane change request, the longitudinal trajectory planner is responsible for making sure that the target lane for the lane change is free from other traffic participants. This information is then passed forward to the lateral planner, which will plan the lane change if the longitudinal planner allows it. For an overview of the inputs and outputs of the longitudinal planner, see fig. 4.2.

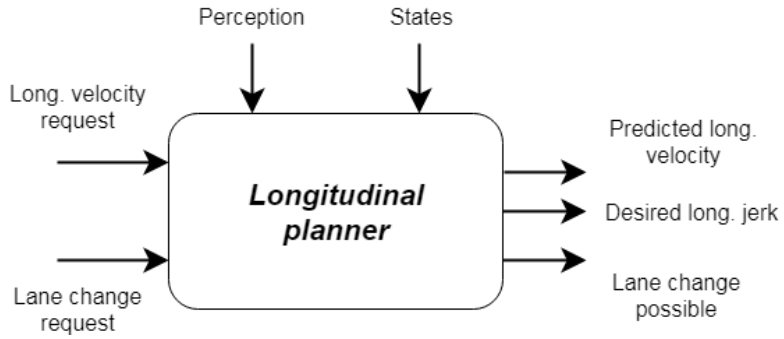


Figure 4.2: Longitudinal planner and its interface. The longitudinal velocity and lane change requests, current traffic situation, road and the state measurements are used as inputs. A predicted longitudinal velocity, information whether a lane change is possible or not and a desired longitudinal jerk are passed as outputs.

4.3.1 Objective function

The objective function is stated as quadratic function that should be minimized:

$$\begin{aligned} \underset{j_{x,1des}}{\text{minimize}} \quad & \int_{t=t_0}^{T} \frac{1}{2}\Big(K_1^{long}(v_{xreq}(t) - v_{x,1}(t))^2 + K_2^{long} a_{x,1des}(t)^2 \\ & + K_3^{long} j_{x,1des}(t)^2\Big) dt \end{aligned} \tag{4.4}$$

where T is the length of the prediction horizon in $[s]$, v_{xreq} is the requested velocity, $v_{x,1}$ is the velocity of the tractor, $a_{x,1des}$ is the desired acceleration of the tractor, $j_{x,1des}$ is the desired longitudinal jerk of the tractor and K_i, $i = 1, 2, 3$, are the tuning parameters for the objective function. The states and tuning parameters in the objective function have been chosen in order to make the trajectory planner able to plan safe and smooth trajectories. The first term, $(v_{xreq} - v_{x,1})$ arises to ensure that it is optimal to follow the velocity reference. Furthermore, the terms $a_{x,1des}$ and $j_{x,1des}$ are added to give an incentive for the trajectory planner to generate smooth trajectories.

4.3.2 Constraints

The constraints for the longitudinal trajectory planner are added to the OCP formulation to state the dynamics and limits of the system, as described in section 2.1. These constraints are separated into equality and inequality constraints. The equality constraints are used in the OCP to represent the dynamics of the system. This way, the OCP can predict the system state evolution in time. The inequality constraints are used in the OCP formulation to state the limits of the system.

Equality constraints

In the longitudinal planner, the system consist of the longitudinal vehicle model, presented in section 3.1.3 and the auxiliary states for relating the vehicle to the surrounding traffic, presented in section 3.4. Using these models, the current measurement of vehicle, the road states and the surrounding traffic, the longitudinal planner can simulate how the vehicle states will evolve in time, relative to the surrounding traffic. The equality constraints can be described as:

$$\frac{d\zeta_{\mathbf{long}}}{dt} = f(\zeta_{\mathbf{long}}, \mathbf{u_{long}}) \tag{4.5}$$

where $\zeta_{\mathbf{long}}$ is the state vector and $\mathbf{u_{long}}$ is the control vector:

$$\zeta_{\mathbf{long}} = \left[v_{x,1}, a_{x,1}, a_{x,1des}, \Delta p_{x,1}, \Delta p_{x,2}, \Delta p_{x,3}\right]^T, \quad \mathbf{u_{long}} = \left[j_{x,1des}\right] \tag{4.6}$$

Furthermore, to avoid discontinuities in the states, an initial value constraint is added to ensure that the initial values of the states in the OCP, for each new time instance, correspond to the current state values.

$$\zeta_{\mathbf{long}}(0) = \zeta_{\mathbf{long}}^{\mathbf{current}} \tag{4.7}$$

Inequality constraints

From section 4.2, the inequality constraints for the longitudinal planner are summarized as follows. For a more detailed explanation of the constraints, the reader is referred to section 4.2.

$$8.33 \leq v_{x,1} \leq 25 \tag{4.8}$$

where the velocity is stated in $[m/s]$.

$$-5.9 \leq a_{x,1des} \leq 0.25 \tag{4.9}$$

where the acceleration is stated in $[m/s^2]$.

$$-2 \leq j_{x,1des} \leq 2 \tag{4.10}$$

where the longitudinal jerk is stated in $[m/s^3]$.

$$\Delta p_{x,1} \geq d_{obs,1}^{safe} \tag{4.11}$$

where $\Delta p_{x,1}$ is defined as the relative distance to closest preceding vehicle in our lane.

$$\Delta p_{x,2} \geq d_{obs,2}^{safe} \tag{4.12}$$

where $\Delta p_{x,2}$ is defined as the relative distance to closest preceding vehicle in an adjacent lane.

$$\Delta p_{x,3} \leq -15 \tag{4.13}$$

where $\Delta p_{x,3}$ is defined as the relative distance to closest trailing vehicle in an adjacent lane and its constraint is defined in $[m]$. Furthermore, $d_{obs,i}^{safe}$ is also defined in meters.

The relative distance constraints, $d_{obs,i}^{safe}$ $i = 2, 3$, are only active when a lane change is in progress and only refer to the vehicles in the target lane. When no lane change is in progress, these two constraints are disregarded. The distance constraints are changed depending on which speed the ego vehicle has, and this functionality is described in 4.2.1.2.

4.3.3 Final OCP formulation

The final OCP formulation is a combination of the objective function and the equality and inequality constraints. This becomes an initial value problem where the initial value for the states is the measurement of the state in the current time step. The formulation thus becomes the following:

$$\begin{aligned}
\underset{j_{x,1des}}{\text{minimize}} \quad & \int_{t=t_0}^{T} \frac{1}{2}\Big(K_1^{long}(v_{xreq}(t) - v_{x,1}(t))^2 + K_2^{long} a_{x,1des}(t)^2 \\
& + K_3^{long} j_{x,1des}(t)^2\Big) dt \\
\text{subject to} \quad & \frac{d\zeta_{\mathbf{long}}}{dt} = f(\zeta_{\mathbf{long}}, \mathbf{u_{long}}), \\
& \zeta_{\mathbf{long}}(0) = \zeta_{\mathbf{long}}^{\mathbf{current}}, \\
& 8.33 \leq v_{x,1}(t) \leq 25, \\
& -5.9 \leq a_{x,1des}(t) \leq 0.25, \\
& -2 \leq j_{x,1des}(t) \leq 2, \\
& \Delta p_{x,1}(t) \geq d_{obs,1}^{safe}(t), \\
& \Delta p_{x,2}(t) \geq d_{obs,2}^{safe}(t), \\
& \Delta p_{x,3}(t) \leq -15
\end{aligned} \tag{4.14}$$

where $\zeta_{\mathbf{long}}$ is the state vector and $\mathbf{u}_{\mathbf{long}}$ is the control vector:

$$\zeta_{\mathbf{long}} = \left[v_{x,1}, a_{x,1}, a_{x,1des}, \Delta p_{x,1}, \Delta p_{x,2}, \Delta p_{x,3}\right]^T, \quad \mathbf{u}_{\mathbf{long}} = \left[j_{x,1des}\right] \tag{4.15}$$

4.4 Formulation of lateral planner

In this section, the formulation of the lateral planner is presented. The optimal trajectory follows a lateral reference, for example the center of the lane or a reference for changing lane. This, while ensuring that limits from the road, actuators and lateral dynamics of the truck are kept. As previously stated, when a lane change request is received, the longitudinal planner determines whether a lane change is possible or not. This is communicated to the lateral planner by the input *lane change possible*. For a overview of the inputs and outputs of the lateral planner, see fig. 4.3.

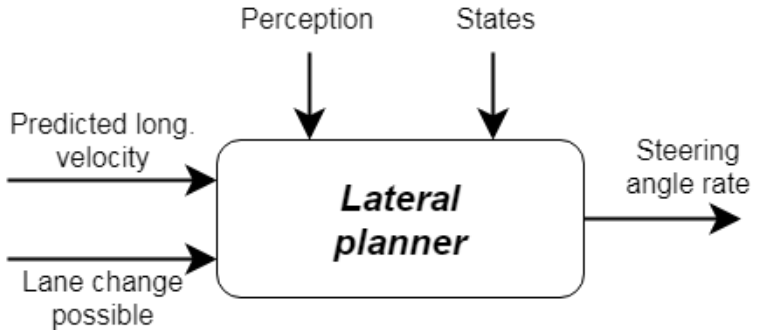


Figure 4.3: Lateral planner and its interface. The predicted longitudinal velocity, the signal lane change possible, the current traffic situation and the state measurements are used as inputs. The steering angle rate is passed as the output.

4.4.1 Objective function

Similarly as for the longitudinal planner, the objective function is stated as a quadratic function,

$$\begin{aligned} \underset{\dot{\delta}}{\text{minimize}} \quad & \int_{t=t_0}^{T} \frac{1}{2}\Big(K_1^{lat}(d_{1ref}(t) - d_1(t))^2 + K_2^{lat}(d_{4ref}(t) - d_4(t))^2 \\ & + K_3^{lat}\dot{\delta}(t)^2\Big)dt \end{aligned} \tag{4.16}$$

The term T is the length of the prediction horizon in $[s]$, d_{1ref} and d_{4ref} are the lateral distance references for unit 1 and 4 respectively, d_1 and d_4 are the lateral distance offsets from the road geometry of unit 1 and 4 respectively, $\dot{\delta}$ is the steering angle rate and K_i, $i = 1, 2, 3$, are the tuning parameters for the objective function. The states and tuning parameters has been chosen in order to make the trajectory planner able to plan safe and smooth trajectories. The first and second term, $d_{1ref} - d_1$ and $d_{4ref} - d_4$, are stated to make it optimal to follow the lateral references for the trajectory planner. Furthermore, the last term, $\dot{\delta}$ is stated to give an incentive to keep the steering angle rate as low as possible to allow for a smooth behaviour.

4.4.2 Constraints

As discussed in 2.1 and 4.3.2, constraints are used in the OCP to describe the system dynamics and limits and are separated into equality and inequality constraints. The equality constraints are used in the OCP to predict the system state evolution in time and the inequality constraints are used by the OCP to state the limits for the system models in the equality constraints.

Equality constraints

In the lateral planner, the system consist of the lateral vehicle model, presented in section 3.1.2, and auxiliary states for connecting the vehicle to the road, presented in section 3.3. Using these models and the current measurement of the states, the lateral planner can simulate how the vehicle states will evolve in time, relative the road. The equality constraints can be described as:

$$\frac{d\gamma_{\mathbf{lat}}}{dt} = f(\gamma_{\mathbf{lat}}, \mathbf{u_{lat}}) \tag{4.17}$$

where $\gamma_{\mathbf{lat}}$ is the state vector and $\mathbf{u_{lat}}$ is the control vector:

$$\gamma_{\mathbf{lat}} = \left[v_{y,1}, \phi_0, \dot{\phi_0}, \theta_1, \dot{\theta_1}, \theta_2, \dot{\theta_2}, \theta_3, \dot{\theta_3}, \delta, s_1, d_1, s_4, d_4\right]^T, \quad \mathbf{u_{lat}} = \left[\dot{\delta}\right] \tag{4.18}$$

Also, as described when stating the equality constraints for the longitudinal planner, to avoid discontinuities in the states, an initial value constraint is added.

$$\gamma_{\mathbf{lat}}(0) = \gamma_{\mathbf{lat}}^{\mathbf{current}} \tag{4.19}$$

Inequality constraints

For the lateral planner, the following inequality constraints are used in the OCP. For a more detailed description, one is refereed to 4.2.

$$\underline{d_1} \leq d_1 \leq \overline{d_1} \tag{4.20}$$

$$\underline{d_4} \leq d_4 \leq \overline{d_4} \tag{4.21}$$

where $\underline{d_1}$ is the lower limit and $\overline{d_1}$ is the upper limit of the distance from the tractor to the reference and $\underline{d_4}$ is the lower limit and $\overline{d_4}$ is the upper limit of the distance from the rearmost axle to the reference.

The values of the upper and lower limits for d_1 and d_4 depends on the current maneuver of the truck. If the truck is following the lane, the upper and lower limits of d_1 and d_4 will be the same as described in section 4.2.1.2. However, if the truck is executing a lane change, the limit in the direction of the lane change will be relaxed so that the trajectory planner is able to plan a trajectory into the target lane without breaking any constraints. For example, if the lane change is to the left, the limits in the left direction of d_1 and d_4 are extended so that they also include the target lane, whilst the limits in the right direction of d_1 and d_4 still are the limits in the departure lane. When the lane change is completed, the limits in the right

direction of d_1 and d_4 are moved to the target lane.

The constraints for the steering angle and the steering angle rate are defined in $[rad]$ and $[rad/s]$ correspondingly, and are presented below.

$$-0.1 \leq \delta \leq 0.1 \tag{4.22}$$

$$-0.05 \leq \dot{\delta} \leq 0.05 \tag{4.23}$$

The last set of constraints for the lateral planner is the lateral acceleration of the tractor and for the rearmost axle. These are defined in $[m/s^2]$ and are presented below:

$$-2.5 \leq a_{y,1} \leq 2.5 \tag{4.24}$$

$$-2.5 \leq a_{y,4} \leq 2.5 \tag{4.25}$$

4.4.3 Final OCP formulation

The final OCP formulation is a combination of the objective function and the equality and inequality constraints. Just as the longitudinal formulation, this becomes an initial value problem where the initial value for the states is the measurement of the state in the current time step. This results in the following formulation:

$$\begin{aligned}
\underset{\dot{\delta}}{\text{minimize}} \quad & \int_{t=t_0}^{T} \frac{1}{2}\Big(K_1^{lat}(d_{1ref} - d_1(t))^2 + K_2^{lat}(d_{4ref} - d_4(t))^2 \\
& + K_3^{lat}\dot{\delta}(t)^2\Big)dt \\
\text{subject to} \quad & \frac{d\gamma_{\mathbf{lat}}}{dt} = f(\gamma_{\mathbf{lat}}, \mathbf{u_{lat}}), \\
& \gamma_{\mathbf{lat}}(0) = \gamma_{\mathbf{lat}}^{\mathbf{current}}, \\
& \underline{d_1} \leq d_1(t) \leq \overline{d_1}, \\
& \underline{d_4} \leq d_4(t) \leq \overline{d_4}, \\
& -0.1 \leq \delta(t) \leq 0.1, \\
& -0.05 \leq \dot{\delta}(t) \leq 0.05, \\
& -2.5 \leq a_{y,1} \leq 2.5, \\
& -2.5 \leq a_{y,4} \leq 2.5
\end{aligned} \tag{4.26}$$

where $\gamma_{\mathbf{lat}}$ is the state vector and u_{lat} is the control vector:

$$\gamma_{\mathbf{lat}} = \Big[v_{y,1}, \phi_0, \dot{\phi}_0, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, \theta_3, \dot{\theta}_3, \delta, s_1, d_1, s_4, d_4\Big]^T, \quad \mathbf{u_{lat}} = \Big[\dot{\delta}\Big]^T \tag{4.27}$$

4.5 Complete trajectory planner

Now that the longitudinal and lateral trajectory planners have been presented separately, the combination of the two will be presented as the complete trajectory planner. In fig. 4.4, a schematic overview of the complete trajectory planner is presented. The longitudinal and the lateral planner are treated as two separate

problems. However, as the lateral planner needs the longitudinal velocity in its model of the lateral dynamics, the predicted longitudinal velocity is used as input to the lateral planner. Also, as fellow traffic participants are handled by the longitudinal planner, information describing if a lane change is possible to initiate needs to be given as an input to the lateral planner.

Furthermore, the complete trajectory planner needs auxiliary state calculations for both planners, as both the lateral and longitudinal planner include auxiliary information such as the relative distance to fellow traffic participants and the connection to the road. The planner also needs measurements of the states of the truck, the road and information about fellow traffic participants, and these can be seen as inputs in fig. 4.4. Lastly, requests of lane changes and longitudinal velocity are also passed as inputs and are generated by some external decision making functionality.

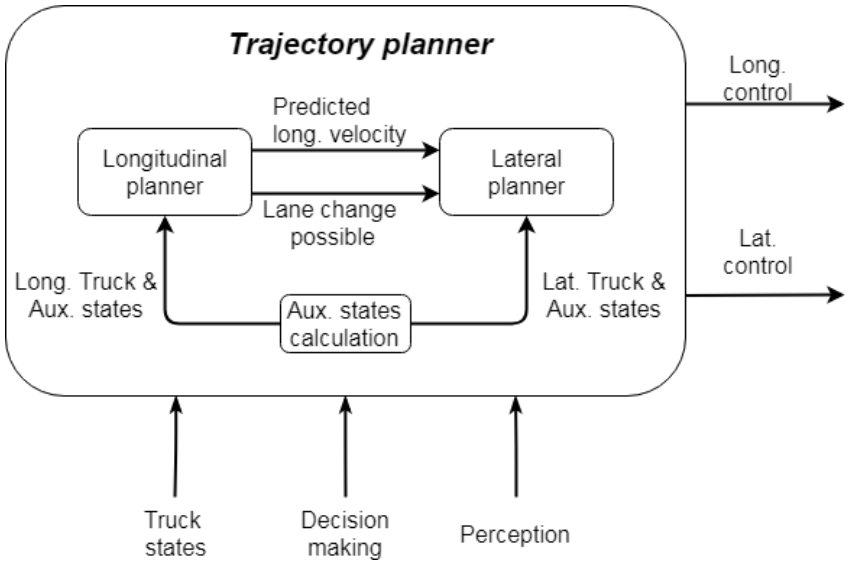


Figure 4.4: Complete trajectory planner including the longitudinal planner and longitudinal auxiliary state calculation and the lateral planner with corresponding calculation of auxiliary states. Measurements of the road and traffic are used as inputs to the trajectory planner through perception systems. Requests of lane change and longitudinal velocity are given from a decision making functionality and are also passed as inputs. Lastly, the measurements of the truck states are also passed as inputs to the trajectory planner.

The output from the trajectory planner is then the combination of the longitudinal control, $j_{x,1des}$, and the lateral control, $\dot{\delta}$.

4.6 Prediction horizon and sampling rate

As MPC is based on the receding horizon idea, a design choice of prediction horizon, T_{ph} defined in $[s]$, must be made. As described in section 2.3, the prediction horizon can be viewed as a fixed time-window of T_{ph} seconds that moves forward in time,

over which the models are simulated.

When choosing the value of T_{ph} there is a trade-off. Generally, longer prediction horizon includes more information into the OCP formulation, which in turn results in better control signals. However, this additional information also increases the computational time as more data needs to be calculated in each time step. Furthermore, the effect of model miss-match and integration errors increase with longer prediction horizon and the actions of other traffic participants become virtually impossible to predict over long time intervals.

In [19], it was found that the lane change intent of other traffic participants could be predicted 2 $[s]$ in advance. This indicates the complexity with using models for predicting how fellow traffic participants will move in time.

Because of these reasons, two different prediction horizons will be used in order to evaluate the performance of the trajectory planner. Furthermore, it should also be noted that the length of the prediction horizon in the longitudinal MPC does not necessarily need to be the same as the one in the lateral MPC. In this project however, the length of the prediction horizons will be the same in both MPCs.

Another design choice is the sampling time, T_{sr}, defined in $[s]$, which is the discretization step used when discretizing the OCP problem. This way, T_{sr} determines the resolution of the data in the OCP, i.e. with a large T_{sr} less data is included in the problem. A large T_{sr} makes it possible to choose a larger prediction horizon, while having the same computational time as when having a small T_{sr} and short prediction horizon. However, the sampling time can not be chosen completely freely, there is a rule of thumb when choosing the sampling time of a system. The sampling time should be chosen to be between a factor of 5 to 30 times faster than the fastest dynamics in the system [20]. By investigating the eigenvalues of the system, the characteristics of the system dynamics can be determined. This has been done in [2], where the fastest dynamics of the same linear A-double model used in this project was identified to be 2 $[Hz]$. Also, as will be discussed in chapter 5, the longest solution time for the trajectory planner is around 0.012 $[s]$. The sampling time must be set to a value that is higher than the solution time, but also short enough to give a good resolution. The sampling time has therefore been set to 0.05 $[s]$.

4.7 Implementation tools

Since the OCP uses quadratic programming, the size of the OCP grows polynomically with the states and control signal and quickly becomes hard to handle manually. Tools such as *FORCES* [21], *YALMIP* [22] and *ACADO* [23] have therefore been developed to generate the necessary structure required by QP solvers. This by simply formulating the mathematical description of the OCP in the tool. In this project, ACADO, or *Automatic Control and Dynamic Optimization*, has been chosen because it is an open-source software available for free online, the simple syntax

used when stating the OCP and the possibility to generate C code for the stated OCP.

ACADO uses a simple mathematical description of the OCP and then generates C code which then can be efficiently solved by a QP solver. The QP solver used in this project is *qpOASES*, which uses an active set method to solve the QP [24]. It should be noted that the OCPs for the lateral and longitudinal trajectory planning problems are implemented separately in ACADO. However, the implementation of each trajectory generator follows these same steps:

- The states, control signal, objective function and constraints for the OCP are stated in ACADO-syntax together with some code generation and solver settings. The QP solvers supported for code generation by ACADO are qpOASES and FORCES [25].
- ACADO then generates C code for the OCP and also interfaces the generated code with the chosen solver, in this case qpOASES.
- Support functions, to handle for example traffic and road related updates, and the communication between the longitudinal and lateral trajectory planner have then been written in C++ code. This to interface the inputs to the trajectory planner with the generated C code from ACADO.

The code generation settings in ACADO enables the user to choose which discretization and integration types that are to be used in the OCP and also how many integration steps that are to be used. The discretization strategy used for the system in this project is a *multiple shooting node* strategy which partitions the state and control trajectories over the prediction horizon. Between each discretization node, the system is then integrated with a Runge-Kutta 4 method with 5*N integration points, where N is the number of points in the prediction horizon.

4.8 Summary

This chapter has introduced the formulation of two OPC:s, one for the lateral planner and one for the longitudinal planner. The theory from chapter 2 and the models derived in chapter 3, are combined with constraints to state the OCP:s which will generate safe and smooth trajectories for the truck. Why the trajectory planner has been divided into two separate MPCs was discussed followed by definitions of what a desirable trajectory is for both planners.

The resulting trajectory planner consists of one longitudinal and one lateral planner, which were presented separately. The lateral planner uses information about the longitudinal velocity and if a requested lane change is possible to execute. This information is supplied by the longitudinal planner, otherwise the planners are completely separated.

The choice of prediction horizon and the sampling interval was presented and the chapter ended with a description of the tools used in the implementation of the trajectory planner.

In the next chapter, the trajectory planner will be tested for some scenarios and the results will be presented.

5

Results and discussion

Now that the modeling and the complete trajectory planner has been presented, the performance of the trajectory planner needs to be evaluated. In order to do so, simulations are performed in Matlab/Simulink on a notebook PC and the states and computational times are measured.

In the first section of this chapter, the simulation environment where the results are generated from is presented, thereafter the plant model used when testing the trajectory planner will be stated. This will be followed by section 5.3 where the pre-defined highway scenario used to test most of the functionality of the trajectory planner will be explained. In section 5.4, the results and analyses for the simulations testing the general performance of the trajectory planner will be presented when a prediction horizon of 2 seconds is used. Section 5.5 will present the corresponding results and analyses when a trajectory planner with a prediction horizon of 5 seconds is used instead. This is followed by a simulation that shows the trajectory planners ability to cope with a sloped road in section 5.6. Lastly, section 5.7 will present a summary of the chapter.

5.1 Simulation environment

The simulation software used to generate the results was Matlab/Simulink 2015b and was executed on a notebook PC. The specifications for the PC can be found in appendix C. Both the trajectory planner application and the simulation software were running on the same notebook PC during the simulations. Note that other applications were running in the background during the simulations when the results were generated which could influence the solution time for the trajectory planner.

The code for the trajectory planner has been written in C/C++ and has been interfaced with Matlab/Simulink by wrapping the C/C++ code into an S-function. S-functions are used to create custom blocks in Simulink which execute the C/C++ code. In Simulink, the trajectory planner is combined with a plant model, which represents the truck and its control layer, a road model, which feeds the trajectory planner with information about the curvature and slope, and a traffic model which feeds the trajectory planner with information about the relative positions of the surrounding traffic. An overview of the simulation environment can be seen in fig. 5.1.

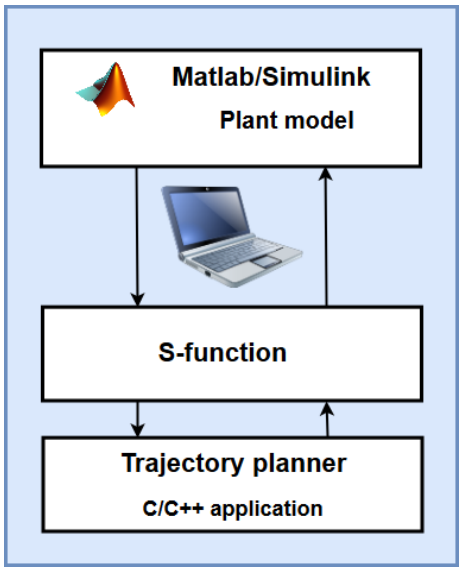


Figure 5.1: Graphical overview of the simulation environment.

5.2 Plant model

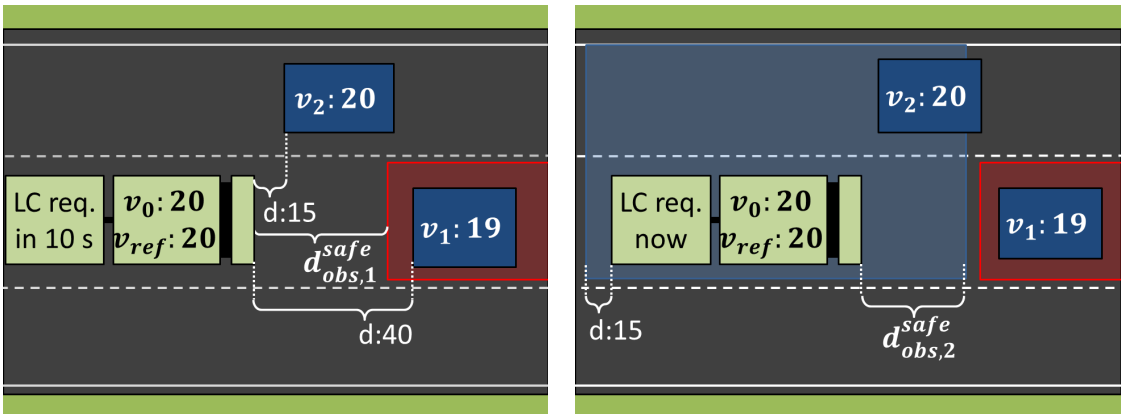
When generating the results for the trajectory planner, the high-fidelity plant described in section 3.5 is used as a plant model. This model is a more detailed representation of the real truck compared to the prediction model and gives an indication of how the performance of the trajectory planner would be if it would be implemented on the real truck. Note that the prediction model in the trajectory planner and the plant model are not the same. This introduces a model miss-match which in turn could mean that the trajectories generated by the trajectory planner are sub-optimal for the real plant. If the miss-match is large, the trajectory planner could generate trajectories which can not be tracked when actuating the real plant.

Furthermore, the slope of the road is not implemented in the high-fidelity model, but only in the prediction model. Therefore, when testing the trajectory planners ability to cope with sloped roads, the trajectory planner is evaluated using the prediction model as the plant model. This test procedure and the test results will be presented separately in section 5.6.

5.3 Scenario

The scenario that has been simulated aims to test all the challenges the trajectory planner needs to handle when driving on a highway, except sloped roads. The traffic scenario that has been used in the simulations is shown in fig. 5.2. Note that these figures are not proportionally correct, but only aim to give the reader a better graphical understanding of the traffic situation.

The truck has an initial velocity of 20 $[m/s]$ and a reference velocity of 20 $[m/s]$. The vehicle ahead of the truck, called obstacle 1, starts 40 $[m]$ in front, is modelled



(a) Before lane change is requested, only the car in front of the truck is of interest. The vehicle ahead starts 40 meters in front of the truck and the distance constraint to it is velocity dependant.

(b) When a lane change is requested, the target lane also becomes of interest. A lane change will not be initiated until no vehicles are inside the safety box represented in blue.

Figure 5.2: Lane change scenario which has been simulated when testing the trajectory planner. The truck will have an initial velocity of 20 m/s and a velocity reference of 21 m/s. The other vehicles are modelled as cars with a constant velocity.

as a car and drives at a constant velocity of 19 $[m/s]$. This will cause the truck to catch up with obstacle 1 and must therefore also adjust its own velocity so that it does not come to close to obstacle 1. The distance constraint to obstacle 1, $d^{safe}_{obs,1}$, is, as described in section 4.2.1.2, dependent on the velocity of the truck. To the left of the truck, another vehicle is present, called obstacle 2. Obstacle 2 is initiated to start 15 meters in front of the truck and is also modelled as a car. Obstacle 2 has a constant velocity of 20 $[m/s]$ and will thus drive past obstacle 1 after 25 seconds.

After 10 seconds into the simulation, the truck will receive a request to change lane to the left. For the truck to be able to change lane, no obstacle can be inside a safety box encompassing the truck and the target lane, see fig. 5.2b. The safety box is defined from 15 meters behind the rear of the truck and a distance $d^{safe}_{obs,i}$ ahead of the truck, where $d^{safe}_{obs,i}$ is calculated as in eq. 4.2. When driving at 19 $[m/s]$, $d^{safe}_{obs,i}$ will be equal to approximately 30 meters. The reason 15 meters was chosen as the safety margin backwards was to increase the comfort of surrounding drivers. This is a design parameter and can be changed if the choice of 15 meters is too conservative. The reason that $d^{safe}_{obs,i}$ was chosen as the safety margin forward was to simply allow the truck to be able to brake and avoid a crash if the vehicle ahead, in the target lane, would fully decelerate.

At the time of the lane change request, the truck will not be able to change lane as obstacle 2 will be inside its safety box and must therefore wait until obstacle 2 has exited the safety box, see fig. 5.2b. Note that this means that the truck would never be able to change lane if an obstacle inside the safety box would be driving at the

same speed as the truck. However, since obstacle 2 is driving faster than obstacle 1, the safety box for the truck will eventually become free of obstacle 2 and the truck can then execute the lane change.

During the lane change, two distance constraints are active. The safety distance to obstacle 1 must still be kept during the lane change and the safety box will add an additional distance constraint. This means that the distance $d_{obs,2}^{safe}$ to obstacle 2 must also be kept. After the lane change, the truck will accelerate until the reference velocity of 20 m/s has been reached.

The road used for the simulation is an S-curved road with three lanes, with the truck starting in the center lane. The radii for both curves are 800 meters and the road geometry can be seen in fig. 5.3. The reason why this road geometry has been chosen is to show the trajectory planners ability to both follow and execute a lane change on a curved road. Note that the truck will not necessarily use the entire length of the road during the simulation.

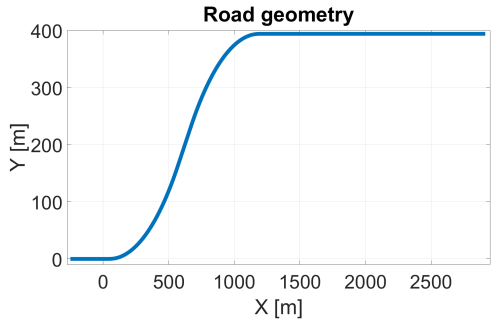


Figure 5.3: Geometry of the road used in the simulations expressed in the global coordinate system.

5.4 Simulation results, prediction horizon of 2 seconds

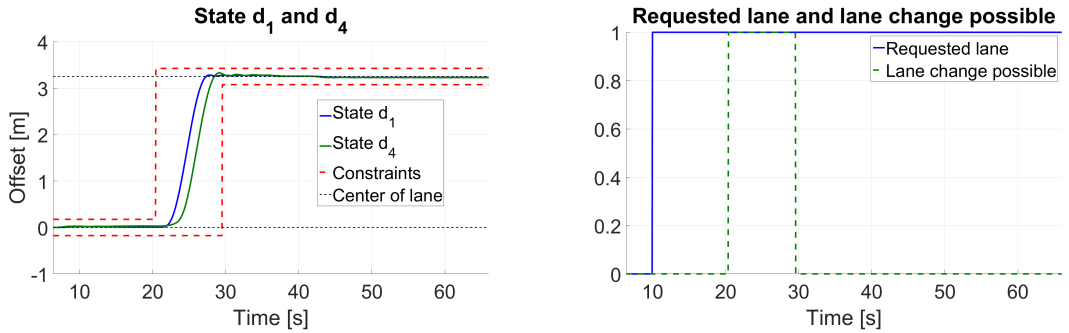
One of the parameters in the MPC formulation that has the largest impact on the computational load is the length of the prediction horizon. The length of the prediction horizon also affects the control signal as the MPC can predict the state behavior further into the future, resulting in better control signals. The tuning parameters used in the controller can be found in appendix D. Note that the trajectory planner with a prediction horizon of 2 seconds have been tuned differently compared to the one with a prediction horizon of 5 seconds. This has been done to ensure good overall performance.

As discussed in section 4.6, lane change predictions for fellow traffic participants can be made 2 seconds in advance. In this section, a prediction horizon of 2 seconds has therefore been used and the results from simulations of the scenario described in section 5.3 will be presented.

Please note that due to initial transient behaviour present in the high-fidelity plant model, the trajectory planner is started after 6 seconds of simulation. This to avoid that inaccurate control signals are generated by the trajectory planner. This has also been implemented when simulating the trajectory planner with a 5 second prediction horizon.

5.4.1 Lateral offset and lane change request

The results presented in fig. 5.4 display the evolution of the lateral offset states, d_1 and d_4, over the simulation and also the signals *requested lane* and *lane change possible*.



(a) Auxiliary states d_1 and d_4 during a lane change maneuver.

(b) Requested lane by decision maker and whether the lane change is possible to execute or not.

Figure 5.4: The state evolution for the lateral offset and the signals for *requested lane* and *lane change possible*, generated with a prediction horizon of 2 seconds.

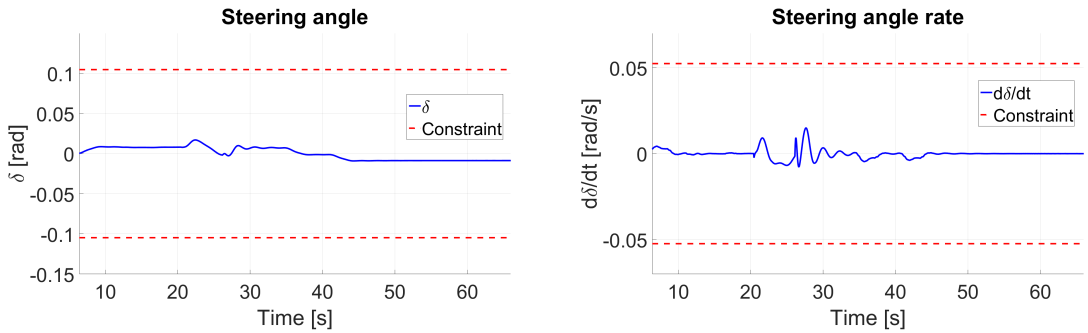
In fig. 5.4b one can see that the requested lane changes from current lane, with id 0, to the adjacent left lane, with id 1, 10 seconds into the simulation. However, one can also see that the lane change is not possible until approximately 20 seconds into the simulation. This because obstacle 2 is present in the safety box, described in section 5.3, up until that time. In fig. 5.4a, the lateral offset from the initial lateral position can be seen. At second 20, when the lane change is possible, the lateral offset constraints are relaxed and the reference for the lateral offset is changed to allow for a smooth transfer to the center of lane 1.

When studying fig. 5.4a, it can be seen that the lane change maneuver starts roughly 3 seconds after the lane change is possible. This is so that the trajectory planner can make a smooth transition between lane following and starting a lane change. When the actual lane change maneuver starts, it can be seen that it takes approximately 7 seconds to move the states d_1 and d_4 to the target lane. As Δt_{lc} was defined to be 7 seconds, this serves as an indication that the pre-optimized reference trajectory for lane changing is working properly.

It can also be observed in fig. 5.4a that both of the states follow the reference, which is the middle of the lane, really well. The offset between the states during the lane change is due to that the vehicle combination has multiple connection points and a delay on the second semi-trailer is thus introduced. The trajectory planner can with ease stay within the distance constraints during, and after the lane change and no significant overshoot for any of the states d_1 or d_4 after the lane change is visible.

5.4.2 Steering and lateral acceleration

In fig. 5.5, one can see the steering angle and the steering angle rate. As one can see, both signals are well within their actuation constraints and that the change in steering angle is smooth. At approximately 23 seconds into the simulation, the steering angle rate changes because of the requested lane change. This results in a change of steering angle during that same time period.



(a) Steering angle during the stimulation. **(b)** Steering angle rate during the stimulation.

Figure 5.5: The state evolution for the steering angle, steering angle rate and generated with a prediction horizon of 2 seconds.

The lane change is also visible in fig. 5.6 where the lateral acceleration for the tractor and second semi-trailer is plotted.

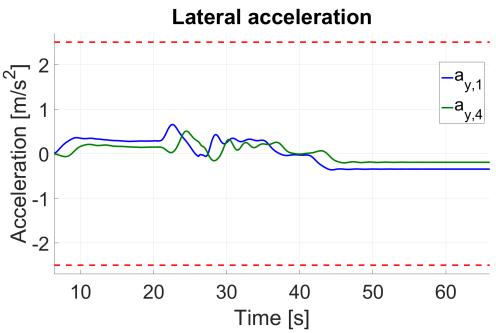
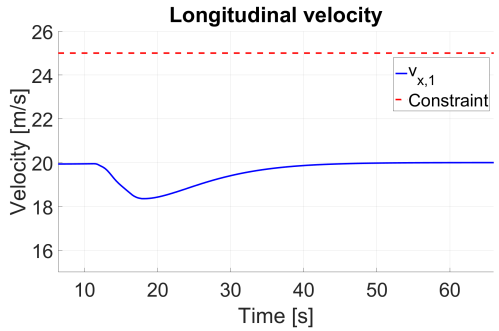


Figure 5.6: Lateral acceleration for tractor and second semi-trailer during the stimulation.

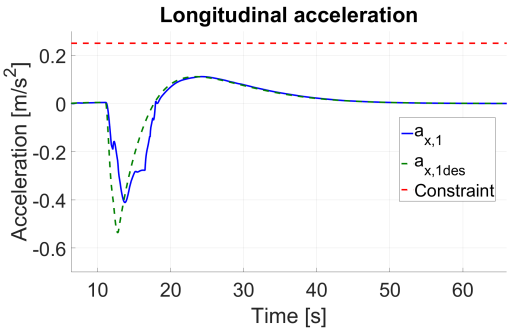
It can be seen that during the initial part of the lane change, the lateral accelerations for both units increases as the truck starts to turn. Note that the effect of RA is not visible in this plot. This is because RA is more prominent when executing rapid maneuvers. Before and after the lane change maneuver, the truck is following lane. Because of the s-shaped geometry of the road, an offset for the steering angle and for both the lateral accelerations can be observed. These appear because the truck needs to turn in order to follow the road geometry.

5.4.3 Longitudinal movement

In fig. 5.7, the evolution of the longitudinal vehicle states over the simulation are displayed. These are highly correlated with the plots in fig. 5.8 which displays the relative distance to the two surrounding obstacles from the front of the truck.



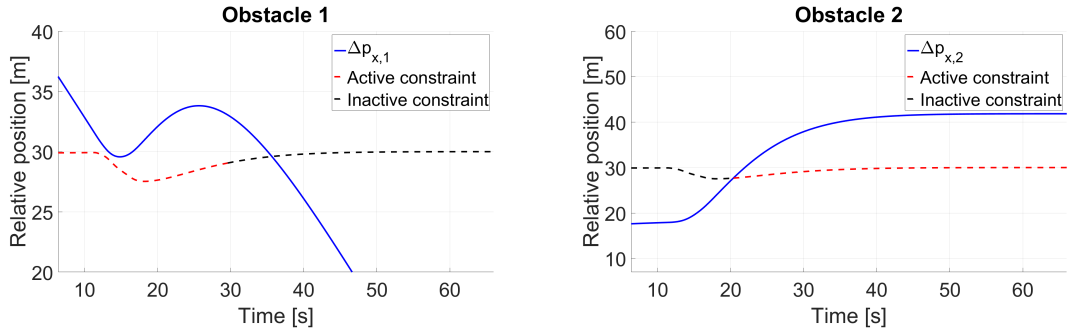
(a) Longitudinal velocity.



(b) Longitudinal acceleration and desired acceleration.

Figure 5.7: The state evolution for the longitudinal velocity, desired acceleration and the actual acceleration with a prediction horizon of 2 seconds.

As described in section 5.3, the initial velocity and the velocity reference for the truck is 20 m/s. However, as obstacle 1 has a velocity of 19 m/s, the truck needs to slow down. This is clearly visible in fig. 5.7a where the truck decelerates from 20 m/s to just over 18 m/s in roughly 8 seconds. This can also be seen in fig. 5.8a, where the relative distance to obstacle 1 is moving close to the constraint. As a result of this, a negative desired jerk is requested, see fig. 5.9, which results in a rapid deceleration visible in fig. 5.7b. This deceleration is then immediately followed by a positive requested jerk as the current velocity is far from the reference velocity. This creates a jerky movement of the truck and a more desirable trajectory would be if a smooth adaption to the velocity of obstacle 1 could be made. The reason for this jerky behaviour is the short prediction horizon. The trajectory planner simply cannot predict far enough into the future to generate a velocity profile which gently adjust the ego velocity to the velocity of obstacle 1.



(a) Relative position to obstacle 1. **(b)** Relative position to obstacle 2.

Figure 5.8: The figures show the state evolution for obstacle 1 and obstacle 2 and the constraints. Observe that the constraints can be active or inactive. This has been generated with a prediction horizon of 2 seconds.

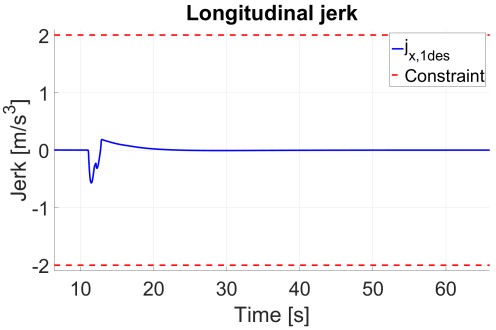


Figure 5.9: Longitudinal jerk generated with a prediction horizon of 2 $[s]$.

As can be seen in fig. 5.8a, the distance constraint to obstacle 1 is active until approximately 30 seconds. At that time, the lane change is completed and the truck has moved into the new lane. This is also depicted in fig. 5.4b when the lane change possible signal becomes inactive again. Figure 5.8b shows how the constraint for obstacle 2 becomes active at approximately 23 seconds, when the lane change becomes possible to execute and the safety box adds a distance constraint to obstacle 2. The additional constraint imposed by the safety box is visible as both the constraint for obstacle 1 and 2 are active during the lane change maneuver. After the lane change has been completed, only the obstacle in the ego lane is of interest and thus the constraint for obstacle 2 is the only active constraint.

When the lane change is complete, obstacle 1 will no longer limit the truck to drive at 19 $[m/s]$ and at the end of the simulation, one can observe that the velocity reference is successfully tracked, see fig. 5.7a. This can also be observed in fig. 5.8b, as the relative distance to obstacle 2 levels out as they have the same velocity.

5.4.4 Solution time

The solution time for the trajectory planner is critical to keep low as real-time performance is of interest. In fig. 5.10 the solution time for the complete trajectory

planner is shown.

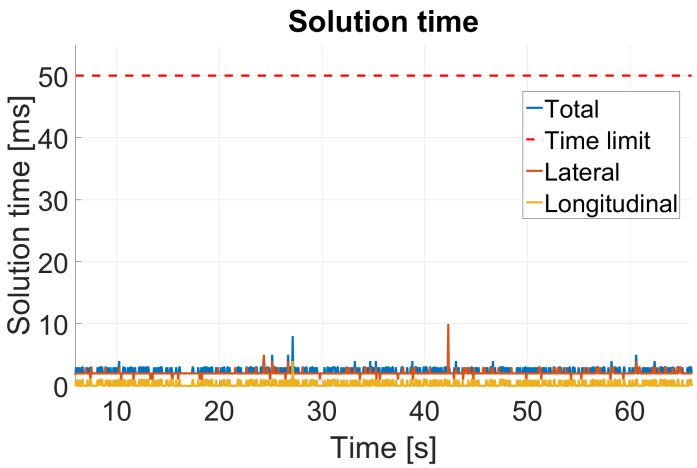


Figure 5.10: The solution time for the lateral and longitudinal planner in every time step. It also shows the summation of these times and the limit for the solution time. This has been generated with a prediction horizon of 2 seconds.

As discussed in section 4.6, the sampling rate of the trajectory planner was chosen to be 50 $[ms]$. To be able to calculate a new set of control signals between each sample, the solution time for the complete trajectory planner must therefore be below 50 $[ms]$.

As can be seen, the solution time for complete trajectory planner is significantly lower than 50 $[ms]$ and is approximately 3 $[ms]$ in mean. The deviations at approximately second 27 and second 44 are most likely caused by background processes. By looking closely, it can be noted that the solution time for the lateral trajectory planner contributes the most to the total time. This is unsurprising as the models used in the lateral trajectory planner are more complex than the models used in the longitudinal trajectory planner.

5.5 Simulation results, prediction horizon of 5 seconds

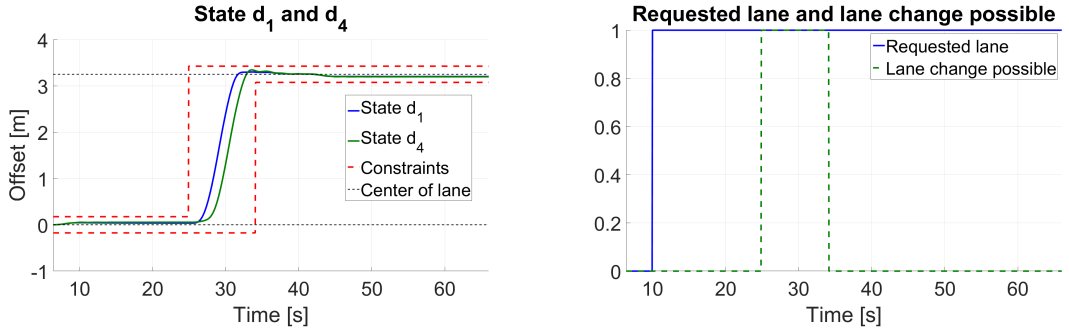
Now that the results for the trajectory planner with a 2 second prediction horizon have been presented, the prediction horizon will be increased to 5 seconds. This to provide the trajectory planner with more information so that smoother trajectories can be generated. The tuning parameters used in the controller can be found in appendix E. Once again, please note that the trajectory planners have been tuned differently to ensure good overall performance.

Please note once more, that the initial 6 seconds of the simulation has been discarded due to the initial transient behaviour in the high-fidelity model.

5.5.1 Lateral offset and lane change request

In fig. 5.11 the states d_1 and d_4 are once again presented together with the signals *requested lane* and *lane change possible*.

It can be observed that no major differences in the behaviour for the states d_1 and d_4 can be seen compared to in fig. 5.4 and that the states are well within the constraints during the entire simulation. Once again, no significant overshoot in either of the lateral offset states can be observed and the references are tracked nicely. The only major difference between fig. 5.11 and fig. 5.4 is that the lane change becomes possible to execute later when using a prediction horizon of 5 seconds. The reason for this will be further discussed in section 5.5.3.



(a) Auxiliary states d_1 and d_4 during a lane change maneuver.

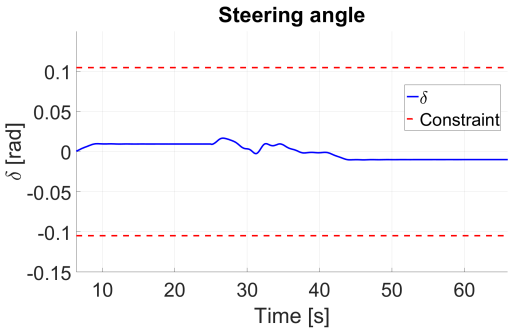
(b) Requested lane by decision maker and whether the lane change is possible to execute or not.

Figure 5.11: The state evolution for the lateral offset and the signals for *requested lane* and *lane change possible*, generated with a prediction horizon of 5 seconds.

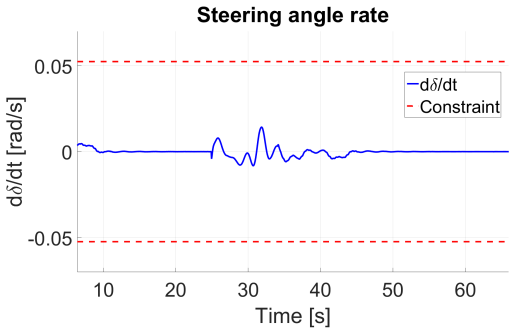
5.5.2 Steering and lateral acceleration

In fig. 5.12, the steering angle and the steering angle rate can be seen. As in the case of using a prediction horizon of 2 seconds, both signals are well within their actuation constraints and the character of the steering angle is smooth. Furthermore, the characteristics for both signals are strikingly similar as in fig. 5.5 and no major differences can be observed.

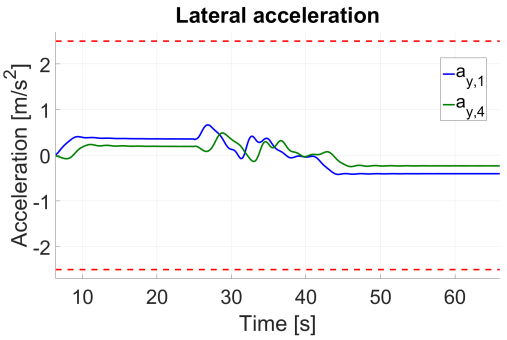
This is also true for the lateral acceleration of the tractor and the second semi-trailer, visible in fig. 5.12. All in all, it can be concluded that the lateral dynamics of the truck were not greatly affected by the increase of prediction horizon.



(a) Steering angle during the simulation.



(b) Steering angle rate during the simulation.



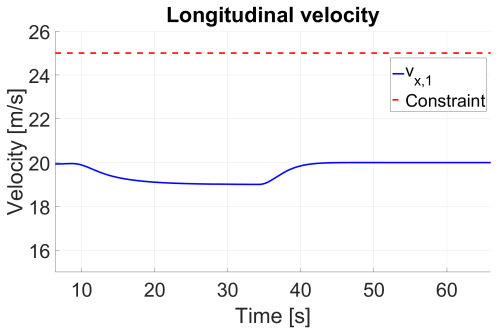
(c) Lateral acceleration for tractor and second semi-trailer during the simulation.

Figure 5.12: The state evolution for the steering angle, steering angle rate and lateral acceleration of the tractor and rearmost axle of the second semi trailer generated with a prediction horizon of 5 seconds.

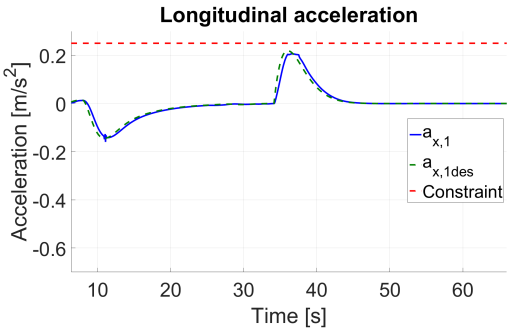
5.5.3 Longitudinal movement

In fig. 5.13, the evolution of the longitudinal vehicle states over the simulation are displayed. Once again, these are highly correlated with the results displayed in fig. 5.15.

When comparing these results to the results generated with a prediction horizon of 2 seconds, the same general conclusions can be drawn. The longitudinal velocity of the truck is adapted depending on the velocity of obstacle 1, the truck can not execute the lane change until obstacle 2 has moved out of the safety box and the distance constraints to obstacle 1 and obstacle 2 become active and inactive in the same manner as described in section 5.4.3.



(a) Longitudinal velocity.



(b) Longitudinal acceleration and desired acceleration.

Figure 5.13: The state evolution for the longitudinal velocity, desired acceleration and actual acceleration generated with a prediction horizon of 5 seconds.

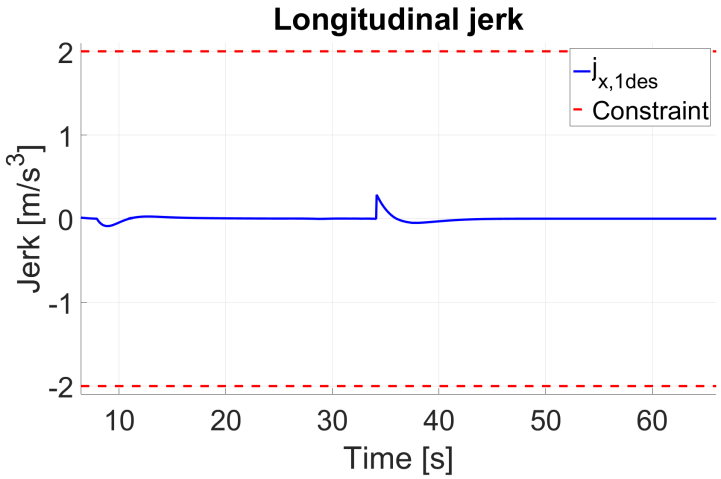
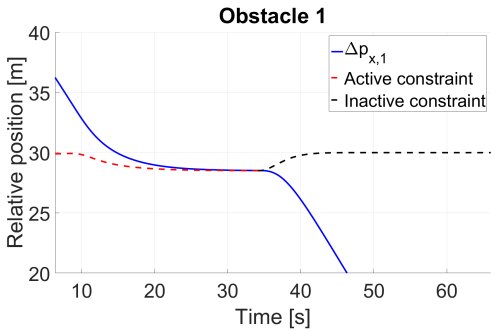
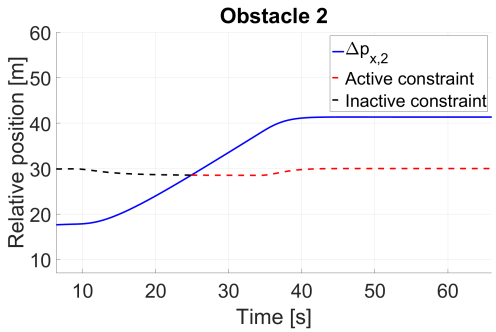


Figure 5.14: Longitudinal jerk generated with a prediction horizon of 5 $[s]$.



(a) Relative position to obstacle 1.



(b) Relative position to obstacle 2.

Figure 5.15: The state evolution for obstacle 1 and obstacle 2 and their constraints. Observe that the constraints can be active or inactive. This has been generated with a prediction horizon of 5 seconds.

However, some major differences in how the longitudinal trajectory is planned, mostly in terms of smoothness, can be seen. One can observe that the longitudinal velocity is adapted more gently to the velocity of obstacle 1, see fig. 5.13a, and that the trajectory planner does not cause the truck to brake as much as in fig. 5.7. This causes the relative distance to obstacle 1 to decrease more smoothly and the truck does not "bounce" of the constraint as in fig 5.8, but instead drives as close to obstacle 1 as the distance constraint allows, see fig. 5.15a. Because of this, the overall longitudinal velocity is higher during the following of obstacle 1. This causes obstacle 2 to exit the safety box later, compared to the simulation with the trajectory planner based on a prediction horizon of 2 seconds, which is why the lane change is initiated later. Also note that, as the desired acceleration changes more slowly, the actual acceleration of the truck follows the desired acceleration much better. This performance is much more desirable than the jerky behaviour displayed when using a prediction horizon of 2 seconds.

The reason the trajectory planner now can adjust the longitudinal velocity much better, is because of the longer prediction horizon. This makes it possible for the trajectory planner to plan its longitudinal trajectory based on more information, compared to the case with a prediction horizon of 2 seconds, and thus smoother trajectories can be generated. Furthermore, the additional information has a greater impact on the longitudinal planner as the response time for the longitudinal dynamics is larger than for the lateral dynamics. This because of the delay present in the state derivative for the longitudinal acceleration. From these results, it can be established that increase in prediction horizon positively affected the performance of the longitudinal planner.

5.5.4 Solution time

In fig. 5.16 the solution time for the complete trajectory planner based on a prediction horizon of 5 seconds is shown.

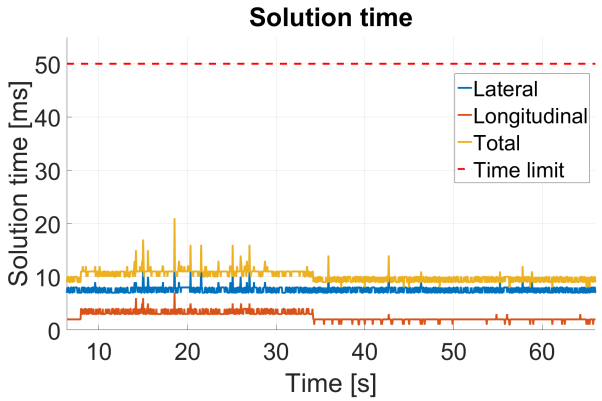


Figure 5.16: The solution time for the lateral and longitudinal planner in every time step. It also shows the summation of these times and the limit for the solution time. This has been generated with a prediction horizon of 5 seconds.

As can be seen, the total solution time for this trajectory planner is almost a factor

3 larger than for the trajectory planner based on a prediction horizon of 2 seconds. Once more, it is the solution time for the lateral planner that contributes with the largest part. However, the total solution time is still far from the time limit of 50 $[ms]$. The explanation for the increased solution time is that the trajectory planner now must process the additional information resulting from the larger prediction horizon.

Furthermore, one thing to note is that the solution time for the longitudinal planner is slightly higher in the region of second 8 to second 34. The longer computational time can be attributed to the fact that the trajectory planner needs to adjust the longitudinal velocity, and deviate from its reference, in order to stay behind obstacle 1. It can be seen that when the constraint to obstacle 1 becomes inactive, the solution time for the longitudinal planner decreases. This as the planner does not need to adjust its velocity from the reference in order to stay behind obstacle 2.

Lastly, the spikes in the solution time for both of the planners are dismissed as disturbances from background processes.

5.6 Simulation with sloped road

The trajectory planner need to model how much acceleration capability it has in every time instance in order to be able to plan safe and smooth trajectories. As explained in section 5.2, the slope of the road was not modeled in the high-fidelity plant. Thus, to demonstrate the trajectory planners ability to cope with a sloped road, simulations when the prediction model was used as the plant model were conducted. Furthermore, this test has also been simulated with two different prediction horizons, namely 2 and 5 seconds.

5.6.1 Scenario

To demonstrate the trajectory planners ability to use topographic information of the road, the following scenario is simulated:
The initial velocity for the truck is set to be 20 $[m/s]$ with the velocity reference set to 22 $[m/s]$. The truck starts on a level road and after 275 meters the slope of the road starts to increase and after an additional 50 meters the slope is the maximum slope that the truck can manage while driving at a constant longitudinal velocity. This maximum slope is based on the acceleration capability of the truck and the maximum slope is discussed in section 4.2.3.2. As the maximum acceleration capability of the truck is 0.25 $[m/s^2]$, the maximum slope that the truck can keep a constant velocity in is roughly 2.5%. This slope will then be kept constant during the entire simulation.

5.6.2 Result

As can be seen in fig. 5.17, as the slope of the road increases, the velocity levels out at the current speed for the trajectory planner with a prediction horizon of 5

seconds. However, the velocity starts to decrease and eventually levels out for the trajectory planner with a prediction horizon of 2 seconds. This is partly because of the shorter prediction horizon, which does not catch the increase in slope as fast as the trajectory planner with a prediction horizon of 5 seconds, but also because the trajectory planners are tuned differently. The difference in tuning can be seen in fig. 5.18, where the planner with a prediction horizon of 5 seconds is tuned to increase the acceleration demand quicker than the planner with a prediction horizon of 2 seconds. Moreover, the tracking of the velocity reference can not be achieved since all the acceleration capability of the truck is required to maintain its current speed when moving uphill, as can be seen in fig. 5.17.

All in all, this result shows that the trajectory planner can adapt to the topography of the road and use this information when generating trajectories.

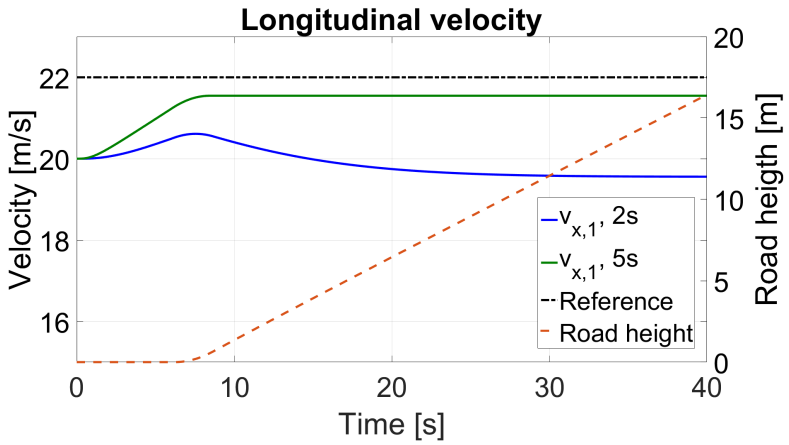


Figure 5.17: The longitudinal velocity of the truck, the reference velocity and the height of the road in a global coordinate frame.

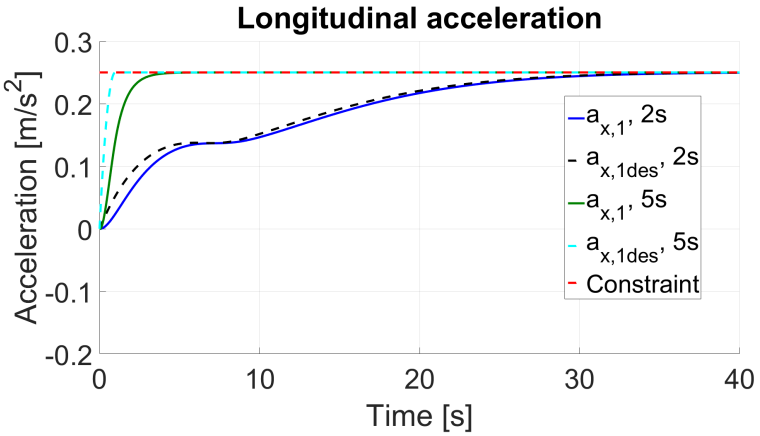


Figure 5.18: The longitudinal acceleration of the truck and the desired acceleration for a trajectory planner with a prediction horizon of 2 and 5 seconds.

5.7 Summary

This chapter started with a presentation of the simulation environment used for generating the results, the plant model used in the simulation and a detailed description of the scenario that is used to demonstrate the capabilities of the trajectory planner. Thereafter, results from the simulations of the scenario with a trajectory planner based on a prediction horizon of 2 seconds and one trajectory planner with a prediction horizon of 5 seconds was presented and discussed. At the end of the chapter, a simulation which demonstrates the ability of the trajectory planner to cope with a sloped road was presented.

It was concluded that an obvious trade-off between the solution time and smoothness of the generated longitudinal trajectory existed. This was visible when comparing the results from the trajectory planner with a prediction horizon of 2 seconds with the trajectory planner with a prediction horizon of 5 seconds. In the next chapter the conclusions of the results will be presented followed by our recommendations for future work.

6
Conclusions and future work

This chapter will start with the conclusions that can be drawn from the results and discussion generated by simulations with the high-fidelity plant model. This will be followed by a recommendation for future work within this project.

6.1 Conclusions

The main conclusions from the simulations of the trajectory planner, tested with both a prediction horizon of 2 seconds and 5 seconds, are the following:

- The trajectory planner can execute the two main maneuvers for an automated truck in a highway environment, lane keeping and lane changing. This by using the method of dividing the trajectory planning problem into two separate problems.

- All the constraints regarding, safety, smoothness and actuators are kept for both lane keeping and lane changing maneuvers on curved roads. It is thus also concluded that the trajectory planner works on straight roads as curved roads present a greater challenge when generating the trajectory.

- The simulation results with the high-fidelity plant model were successful, both with a prediction horizon of 2 and 5 seconds. This indicates that there is small model miss-match between the prediction models used in the trajectory planner and the high-fidelity plant model. This suggests that the trajectory planner could work well if implemented in a real truck.

- The proposed objective functions and tunings used in the lateral and longitudinal planners, together with the suggested pre-optimized trajectory for the lane change maneuver, allows for a smooth behaviour when executing the lane keeping and lane change maneuvers.

- The solution times for the trajectory planner, with different prediction horizons are both below the time limit of 50 $[ms]$. However, as these results were generated on a notebook PC with background processes running, the calculation times when running on embedded hardware in the truck are hard to estimate. Nonetheless, as the calculations times are well below the time limit, the proposed trajectory planner shows good potential to work in a real-time

system.

- The main trade-off when the designing the trajectory planner has been identified to be between smoothness and computational time. This mainly depends on the length of the prediction horizon as the generated longitudinal trajectories becomes smoother to a cost of increased computational time when using a longer prediction horizon.

- It was shown that the trajectory planner was able to adjust the trajectory depending on the slope of the road, regardless of the proposed prediction horizons. This further indicates that the trajectory planner could be successfully implemented in a real truck.

From this, it can be concluded that the main trade-off when designing the trajectory planner is between the smoothness of the generated trajectories and the solution time. It was also concluded that the modelling of the lateral dynamics required a more complex model compared to the model used for the longitudinal dynamics of the truck. As a result of this, the computational time for the lateral trajectory planner was shown to be higher than for the longitudinal trajectory planner.

The length of the prediction horizon was concluded to be of greater importance for the longitudinal planner than for the lateral planner. The reason for this was that the delay present in the longitudinal dynamics is handled better with longer prediction horizon. From this conclusion, a prediction horizon of 2 $[s]$ for the lateral planner and 5 $[s]$ for the longitudinal planner could give a good trade-off between smoothness and total solution time. This is however a solution that has not been evaluated in this project.

Nonetheless, in this project it has been shown that the trajectory planner developed, with a prediction horizon of 5 seconds, both allows for planning of smooth trajectories, while still keeping the computational time for the trajectory planner far below the time limit. The resulting trajectory planner is able to handle the two main maneuvers of highway driving, lane keeping and lane changing, while ensuring safety, smoothness and that the actuator limitations are not being violated.

6.2 Future work

Since the chosen method for developing the trajectory planner shows good potential, both in the performance of the generated trajectories but also in the solution time, recommendations for future work will now be presented.

- Evaluate a trajectory planner with a different prediction horizons in the lateral and longitudinal planners. This could further decrease the computational time while still ensuring good performance.

- Evaluate the solution time with embedded hardware to further examine the real-time performance.

- Extend the functionality of the trajectory planner. This could for example be to implement functional safety such as a robust abort maneuver when changing lane and also implement more robust signal management, for example how to handle lost sensor signals.

- Implement a less conservative safety box. This could further increase the use ability of the trajectory planner as the current implementation could be somewhat conservative. This could be done by for example using a convex parallelogram as the shape of the safety box instead.

- Implement a more advanced prediction model of the surrounding traffic. This would mean that the trajectory planner could become less conservative and that better trajectories could be generated. This in turn could also improve the performance and the use-ability of the trajectory planner.

- A more advanced prediction model for the longitudinal dynamics of the truck could be implemented. This to better model the resistive longitudinal forces and to get a better representation of the capabilities of the truck. This could for example be to include a model of the gearbox in the truck to better represent the acceleration capability of the truck.

- Evaluate other implementation tools and solvers than ACADO and qpOASES in order to further improve the real-time performance of the trajectory planner.

Bibliography

[1] Volvo Truck Corporation, Filename: F2012_0002_se.mp4, accessed on 28-05-18.
URL http://images.volvotrucks.com/

[2] N. J. van Duijkeren, Real-time receding horizon trajectory generation for long heavy vehicle combinations on highways, Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering (3mE), Delft University of Technology, Netherlands (2014).

[3] M. Behrendt, A basic working principle of model predictive control (2009).
URL https://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg

[4] P. Nilsson, Traffic situation management for driving automation of articulated heavy road transports - from driver behaviour towards highway autopilot, Ph.D. thesis, Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Sweden (2017).

[5] P. Nilsson, K. Tagesson, Single-track models of an a-double heavy vehicle combination, Tech. rep., Department of Applied Mechanics, Chalmers University of Technology, Sweden (2013).

[6] NASA, Graphic: The relentless rise of carbon dioxide (Accessed on 18-05-04).
URL https://climate.nasa.gov/climate_resources/24/graphic-the-relentless-rise-of-carbon-dioxide/

[7] N. Murgovski, B. Egardt, M. Nilsson, Cooperative energy management of automated vehicles, Control Engineering Practice 57 (2016) 84 – 98. doi:https://doi.org/10.1016/j.conengprac.2016.08.018.

[8] J. Nilsson, M. Brännström, J. Fredriksson, E. Coelingh, Lane change maneuvers for automated vehicles, IEEE Transactions on Intelligent Transportation Systems 18 (5).

[9] C. Barnhart, H. Jiang, L. Marla, Optimization approaches to airline industry challenges: Airline schedule planning and recovery.

[10] H. Karloff, Linnear Programming, 1st Edition, Birkhauser Boston, 1991.

[11] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge University Press, 2015.

[12] J. Nocedal, S. J. Wright, Numerical optimization, 2nd Edition, Springer, 2006.

[13] W. H. Kwon, S. Han, Receding Horizon Control, Springer, 2005.

[14] P. T. Boggs, J. W. Tolle, Sequential quadratic programming, Acta Numerica 4 (1995) 1–51. doi:10.1017/S0962492900002518.

[15] B. Jacobsson, Vehicle dynamics compendium for course mmf062 (2015).

[16] Q. Wang, Y. He, A study on single lane-change manoeuvres for determining rearward amplification of multi-trailer articulated heavy vehicles with active trailer steering systems, Vehicle System Dynamics 54 (1) (2016) 102–123. arXiv:https://doi.org/10.1080/00423114.2015.1123280, doi:10.1080/00423114.2015.1123280.
URL https://doi.org/10.1080/00423114.2015.1123280

[17] J. Aurell, T. Wadman, Vehicle combinations based on the modular concept, 2007.
URL http://www.nvfnorden.org/lisalib/getfile.aspx?itemid=1589

[18] Trafikverket, Krav för vägars och gators utformning (2015).

[19] M. G. Ortiz, F. Kummert, J. Schmüdderich, Prediction of driver behavior on a limited sensory setting, IEEE Intelligent Transportation Systems Magazine (2012) pp. 638–643.

[20] W. S. Levine, The control handbook, CRC Press, 1996.

[21] FORCESpro (Accessed on 18-05-04). [link].
URL https://www.embotech.com/FORCES-Pro

[22] YALMIP (Accessed on 18-05-04). [link].
URL https://yalmip.github.io/

[23] B. Houska, H. Ferreau, M. Diehl, ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization, Optimal Control Applications and Methods 32 (3) (2011) 298–312.

[24] qpOASES (Accessed on 18-05-04). [link].
URL https://projects.coin-or.org/qpOASES

[25] B. Houska, H. Ferreau, M. Vukov, R. Quirynen, ACADO Toolkit User's Manual, http://www.acadotoolkit.org (2009–2013).

A

A-double parameters

Vehicle parameters for the A-double combination, from [5].

Parameter	**Symbol**	**Value**	**Unit**
Mass, unit 1	m_1	9841	kg
Mass, unit 2	m_2	33601	kg
Mass, unit 3	m_3	2700	kg
Mass, unit 4	m_4	33801	kg
Yaw movement of inertia, unit 1	J_1	$20 \cdot e3$	kgm^2/rad
Yaw movement of inertia, unit 2	J_2	$543 \cdot e3$	kgm^2/rad
Yaw movement of inertia, unit 3	J_3	$2 \cdot e3$	kgm^2/rad
Yaw movement of inertia, unit 4	J_4	$546 \cdot e3$	kgm^2/rad
Distance from COM to front axle, unit 1	a_1	1.45	m
Distance from front connection point to COM, unit 2	a_2	4.43	m
Distance from front connection point to COM, unit 3	a_3	4.55	m
Distance from front connection point to COM, unit 4	a_4	4.65	m
Distance from COM to rear axle, unit 1	b_1	2.23	m
Distance from COM to rear axle, unit 2	b_2	3.27	m
Distance from COM to rear axle, unit 3	b_3	0.65	m
Distance from COM to rear axle, unit 4	b_4	3.05	m
Distance from COM to rear connection point, unit 1	c_1	1.95	m
Distance from COM to rear connection point, unit 2	c_2	5.97	m
Distance from COM to rear connection point, unit 3	c_3	0.00	m
Front axle cornering stiffness, unit 1	C_{1f}	$4.07 \cdot e5$	N/rad
Rear axle cornering stiffness, unit 1	C_{1r}	$2.07 \cdot e6$	N/rad
Rear axle cornering stiffness, unit 2	C_{2r}	$1.24 \cdot e6$	N/rad
Rear axle cornering stiffness, unit 3	C_{3r}	$1.17 \cdot e6$	N/rad
Rear axle cornering stiffness, unit 4	C_{4r}	$1.42 \cdot e6$	N/rad

B

Differential equations for the lateral dynamics

The time derivatives of the states presented in section 3.1.2 are based on the derived equations in [5] and by inserting the parameter values of the truck, see appendix A. This results in the following equations:

$$\begin{aligned}\frac{dv_{y,1}}{dt} =& -\frac{70.6191}{v_{x,1}}v_{y,1} - \frac{v_{x,1}^2 - 9.7314}{v_{x,1}}\dot{\phi}_0 + \frac{21.9217}{v_{x,1}}\dot{\theta}_1 + 1.9775\theta_1 \\ &+ \frac{4.4014}{v_{x,1}}\dot{\theta}_2 + 0.8494\theta_2 - \frac{0.017}{v_{x,1}}\dot{\theta}_3 - 0.0022\theta_3 + 45.9558\delta\end{aligned} \tag{B.1a}$$

$$\begin{aligned}\frac{d\dot{\phi}_0}{dt} =& \frac{27.5489}{v_{x,1}}v_{y,1} - \frac{174.2882}{v_{x,1}}\dot{\phi}_0 - \frac{21.0338}{v_{x,1}}\dot{\theta}_1 - 1.8974\theta_1 + \frac{4.2231}{v_{x,1}}\dot{\theta}_2 \\ &- 0.815\theta_2 + \frac{0.0164}{v_{x,1}}\dot{\theta}_3 + 0.0021\theta_3 + 25.0956\delta\end{aligned} \tag{B.1b}$$

$$\frac{d\phi_0}{dt} = \dot{\phi}_0 \tag{B.1c}$$

$$\begin{aligned}\frac{d\dot{\theta}_1}{dt} =& -\frac{36.4048}{v_{x,1}}v_{y,1} + \frac{165.4516}{v_{x,1}}\dot{\phi}_0 - \frac{10.5324}{v_{x,1}}\dot{\theta}_1 - 3.9082\theta_1 + \frac{12.8600}{v_{x,1}}\dot{\theta}_2 \\ &+ 2.4818\theta_2 - \frac{0.0498}{v_{x,1}}\dot{\theta}_3 - 0.0065\theta_3 - 25.4638\delta\end{aligned} \tag{B.1d}$$

$$\begin{aligned}\frac{d\dot{\theta}_2}{dt} =& \frac{19.7904}{v_{x,1}}v_{y,1} - \frac{216.8786}{v_{x,1}}\dot{\phi}_0 - \frac{170.0741}{v_{x,1}}\dot{\theta}_1 + 2.2622\theta_1 - \frac{125.6565}{v_{x,1}}\dot{\theta}_2 \\ &- 22.9024\theta_2 - \frac{7.1692}{v_{x,1}}\dot{\theta}_3 - 0.9311\theta_3 + 0.5539\delta\end{aligned} \tag{B.1e}$$

$$\begin{aligned}\frac{d\dot{\theta}_3}{dt} =& -\frac{12.4638}{v_{x,1}}v_{y,1} + \frac{195.8250}{v_{x,1}}\dot{\phi}_0 + \frac{168.7766}{v_{x,1}}\dot{\theta}_1 + 5.0960\theta_1 + \frac{68.1597}{v_{x,1}}\dot{\theta}_2 \\ &+ 22.7324\theta_2 - \frac{54.6629}{v_{x,1}}\dot{\theta}_3 - 7.0991\theta_3 - 0.1851\delta\end{aligned} \tag{B.1f}$$

$$\frac{d\theta_i}{dt} = \dot{\theta}_i, \quad i = 1, 2, 3 \tag{B.1g}$$

$$\frac{d\delta}{dt} = \dot{\delta} \tag{B.1h}$$

C

Specifications for notebook PC

Specifications for PC used for desktop simulations:

Model	Dell Precision M6700
CPU	Intel i7-3520M @ 2.9 GHz
Memory	16 GB 1600 MHz DDR3
OS	Windows 7 64 bit

D

Tuning parameters for trajectory planner with a prediction horizon of 2 seconds

Parameter	**Symbol**	**Value**
Longitudinal vehicle speed	K_1^{long}	5/2
Longitudinal desired acceleration	K_2^{long}	13/2
Longitudinal desired jerk	K_3^{long}	50/0.75
Steering angle rate	K_1^{lat}	0.5
d_1	K_2^{lat}	0.5
d_4	K_3^{lat}	0.5

E

Tuning parameters for trajectory planner with a prediction horizon of 5 seconds

Parameter	**Symbol**	**Value**
Longitudinal vehicle speed	K_1^{long}	5/2
Longitudinal desired acceleration	K_2^{long}	13/2
Longitudinal desired jerk	K_3^{long}	25
Steering angle rate	K_1^{lat}	0.5
d_1	K_2^{lat}	0.5
d_4	K_3^{lat}	0.5