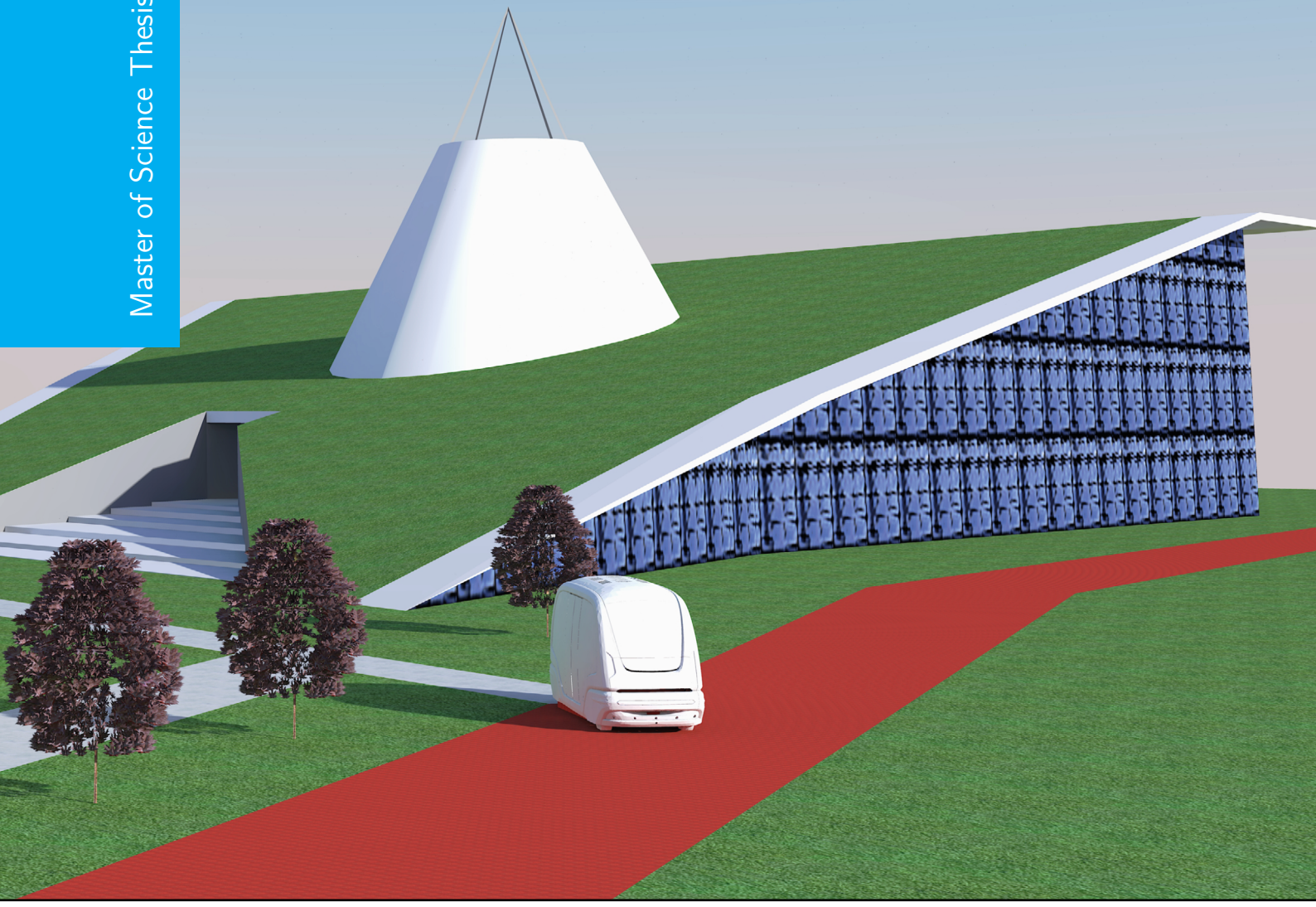
Collision-free predictive trajectory generation for automated four-wheel-steered mini-buses

N.J. Brouwer

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Collision-free predictive trajectory generation for automated four-wheel-steered mini-buses

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

N.J. Brouwer

May 1, 2018

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

The work in this thesis was supported by 2getthere Automated Transit Systems. Their cooperation is hereby gratefully acknowledged.

Abstract

Now that the use of private vehicles is near its limits to efficiently meet the desire for affordable freedom in personal mobility, mobility-on-demand systems distinguish themselves as a sustainable and economically more attractive alternative. 2getthere markets and develops these kind of sustainable Automated Transit Networks for personal and group transportation. To accommodate the requirements of future projects, their current Group Rapid Transport (GRT) system needs to be evolved to support dynamic motion planning.

In this thesis a Model Predictive Control-based, human-like planning approach is proposed, that is capable of dynamic re-routing the GRT using both safe and comfortable trajectories. Moreover, the proposed solution is capable of planning these trajectories at update rates in the millisecond range.

A front and rear wheel steered linear single-track model is introduced to formulate an Optimal Control Problem (OCP) defining the open-loop lateral planner. Resulting is an optimal steering policy as input to the GRT. Additionally the closed-loop interconnection of the open-loop planner and a low-level controller has been formulated in an OCP of which the resulting solution is a re-routed optimal trajectory. Furthermore, a longitudinal OCP is introduced to re-plan the GRT's velocity along the planned lateral trajectory.

Hard constraints are formulated so that planned trajectories do not exceed road boundaries, lateral acceleration and lateral jerk limits or violate actuator limitations. An extended notion of a standard ellipse is introduced as collision avoidance constraint to achieve human-like behaviour in the trajectory generator. Surrounding traffic is modeled using constant-velocity physics-based prediction models.

The proposed planner is implemented in the `ACADO toolkit` modeling environment exploiting the Real-Time Iteration scheme to solve the Nonlinear Program (NLP) using a Sequential Quadratic Programming (SQP) approach. Intermediate SQP solutions are computed using the `qpOASES` Quadratic Program (QP) solver providing new references to the GRT.

The planner is wrapped into a C MEX s-function and implemented in the PreScan driven, Simulink test environment. Additionally, a Simulink state-flow state-machine is implemented to simulate a simple rule-base used to allow or abort an overtake manoeuvre and enable the corresponding collision avoidance constraints.

The dynamic re-routing algorithm is tested in closed-loop simulations with a PreScan high fidelity model. Four different scenarios were defined to test the vehicle in vehicle-following, overtaking and overtake-abort situations. Both trajectory generators control the GRT as intended providing fully collision-free and followable trajectories. Furthermore, no violation of comfort constraints is observed. Execution times for the closed-loop planner need some future work, whereas the open-loop planner has been proven to be real-time feasible therefore providing a satisfying first step to dynamic re-routing on the GRT.

Table of Contents

List of Figures

List of Tables

Acknowledgements

It is with mixed feelings that I am writing this final section of my thesis. On the one hand I am relieved and filled with curiosity to what the future will hold. On the other hand, it means my time in Delft is coming to an end. A time in which I have learned so much, both on a professional and personal level. A time in which many pleasant experiences have passed. A time which I have fond memories of. I met amazing people and made new friends. For that reason, I would like to thank the people that have contributed to this work and experience.

First and foremost, I would like to thank my supervisor Tamás Keviczky for his assistance while working on this thesis. His valuable advise and constructive criticism has helped me to deliver the targets set. His willingness to give his time, advise and feedback has been very much appreciated.

Additionally, I would like to thank Dimitrios Kotiadis from 2getthere Automated Transit Systems for the fruitful discussions we had. He always made time to discuss various interesting aspects of automated driving. His expertise in system engineering has provided me with valuable input in- and feedback on this thesis.

Advise given by Athanasios Tasoglou has been a great help in setting up the planner for 2gettheres purposes. Discussions with him gave me new insights in various interesting topics concerning vehicle automation. Also, the feedback from Jeroen Ploeg in the late stage of writing this thesis has helped me to improve my academic writing and helped me to deliver a thesis I am proud of.

In particular, I would like to thank 2getthere's system engineering team for their support to help me understand the overall system architecture as present in their current and future vehicles. They made me feel right at home in their group and provided the pleasant distractions during my days in Utrecht.

I would like to thank my friends and family whom I could count on for support. I would not have had the opportunity of writing this work if it wasn't for my parents. They therefore deserve my gratitude for backing my choices unconditionally. Last, but certainly not least, I would like to thank Esmée for her endless love and for being my tower of strength in the past year.

Delft, University of Technology — N.J. Brouwer
May 1, 2018

"We crave for new sensations but soon become indifferent to them. The wonders of yesterday are today common occurrences."

— *Nikola Tesla*

Chapter 1

Introduction

The times that owning a private vehicle offered affordable freedom have passed. Especially during rush hours, traffic jams cause big delays on the Dutch road-network [3]. Meanwhile, vehicles are burning non-renewable energy sources and produce more greenhouse-gasses than the Earth's natural systems can absorb. Moreover, it is known that owning a private vehicle is poor value for money as these are typically used for less than 10% of the time [4]. Yet, public transport is usually still an unfavourable alternative as the last-mile from the transport hub to destination is more often than not poorly connected.

Mobility on demand systems have emerged, possibly filling the gap of driving the last-mile, or as a flexible alternative to scheduled public transport [5]. These kinds of systems enable having public transport at 24/7 operation while keeping energy consumption and operation cost at a minimum. Furthermore, applications are often driven by electric motors, reducing air pollution. 2getthere markets and develops these kind of sustainable automated transit networks for personal and group transportation [6]. To accommodate the requirements of future projects that take place in a less controlled environment, their current Group Rapid Transport (GRT) system needs to be evolved to support dynamic motion planning.

A human-like planning approach is needed, capable of dynamic re-routing the GRT in a safe way. This thesis proposes a Model Predictive Control (MPC) based algorithm that achieves both safe and comfortable trajectories and is capable of planning these trajectories in the millisecond range. The next section will explain what the notion of 'Motion Planning' relates to in this thesis. Furthermore, the system for which motion planning is desired is introduced in more detail, as well as its limitations. Thereafter, a short introduction is given to the typical driving scenario concerned for this thesis. For this particular driving scenario, the thesis goal is then defined, followed by the thesis main contributions. The chapter is concluded by a description of the thesis outline.

Table 1-1: Main specifications of a Group Rapid Transport vehicle.

Characteristic		Value
Length	L_{ego}	6000 [mm]
Width	W_{ego}	2100 [mm]
Height	H_{ego}	2800 [mm]
Weight	m	5000 [kg]

1-1 Motion Planning

Motion planning is the problem of planning a trajectory from A to B. Planning for autonomous or intelligent driving is often divided into four hierarchical classes [7] being:

1. Route planning
2. Path planning
3. Manoeuvre planning
4. Trajectory generation or control planning

The first of these is a high level planning approach deciding which road network nodes to connect to get from origin to destination. As this type of motion planning concerns the planning of long-distance paths only, it is also referred to as global motion planning.

The other three classes fall under the notion of local motion planning, which is concerned with planning short-distance paths. The second class is concerned with finding a collision-free driving corridor. The third class concerns the decision making of what manoeuvre to execute within this driving corridor. Examples are the choice whether or not to overtake, follow a target vehicle or make a left turn. The final class concerns the time-parametrized planning of the manoeuvre execution. These last three classes can be combined to one big problem taking into account vehicle dynamics, target vehicles, road geometry and traffic interaction, but can also be split into individual sub-problems.

This thesis work focusses on a combination of problem classes 2 and 4. Therefore the planner is both concerned with finding a collision-free path as well as defining the ego vehicles' time-parametrized states along this path in order to obtain, or maintain, a desired velocity. Prediction horizons are typically in the order of several (dozens of) meters. Since this thesis focusses solely on local motion planning, from hereon the notion of motion planning will cover local planning only.

1-2 Group Rapid Transport

The Group Rapid Transport (GRT) vehicles as produced by 2getthere are four wheel steered mini-buses. This type of Automated People Mover (APM) can carry up to 24 passengers. Due to the four-wheel steering gear, the vehicle is capable of driving small radius corners at lower speeds, while driving with minimum yaw velocity at higher speeds, also known as crabbing. The GRT's main specifications are summarized in Table 1-1.

Table 1-2: Most important design requirements when creating trajectories for a GRT.

Constraint variable		Maximum absolute value
Longitudinal Velocity	v_x	40/3.6 [m/s]
Longitudinal Acceleration	a_x	0.16 [g]
Longitudinal Jerk	j_x	0.10 [g/s]
Lateral Acceleration	a_y	0.10 [g]
Lateral Jerk	j_y	0.06 [g/s]
Maximum steering angle	δ	0.09 [rad]
Maximum steering rate	$\dot{\delta}$	0.26 [rad/s]

The buses are deployed on projects all over the world often driving in low-density open spaces, or on a segregated two-way track. By driving in such a controlled environment, the trajectories for the GRT can be predefined. Therefore step 1 till 4 of Section 1-1 have to be computed only once. Moreover, this can be done off-line. Trips are saved and directly passed onto the GRT by a monitoring system.

In order to provide comfortable trips, the American Society of Civil Engineers (ASCE) sets requirements on the vehicles' driving smoothness during these trips. In their Automated People Mover standards [8], the ASCE requires that during a normal trip:

- The maximum longitudinal acceleration of the vehicle should not exceed 0.16 [g]
- The maximum lateral acceleration of the vehicle should not exceed 0.10 [g]
- The maximum longitudinal jerk should not exceed 0.1 [g/s]
- The maximum lateral jerk should not exceed 0.06 [g/s]

Since fixed trajectories are used, the trips should only be evaluated once off-line. After a trip has been dispatched to a GRT, the low level controller follows the predefined trips as close as possible, taking into account the ASCE requirements. The full list of comfort requirements and mechanical limitations is listed in Table 1-2.

The Society of Automotive Engineers (SAE) classifies APMs like the GRT as level 4 type Automated Driving System (ADS). Therefore the GRT is classified as an high driving automation ADS-Dedicated Vehicle (ADS-DV) [1]. This level of automated systems perform the entire Dynamic Driving Task (DDT) and DDT fallback autonomously. Moreover, such a system is capable of achieving a minimal risk condition autonomously in case of an ADS failure. In other words:

> *"... the user of an engaged level 4 ADS feature is a passenger who need not respond to requests to intervene ..."* [9]

Although the current GRT successfully performs this DDT, it is too conservative in the DDT fallback and reaching a minimal risk condition. Furthermore, the DDT does only include following the predefined trajectory while monitoring its environment. No options for deviation from this trajectory are considered in the case the nominal trajectory is blocked. If an obstacle is encountered, the GRT comes to a full stop and waits until track has been cleared. This causes unwanted and unnecessary delays in the system that could be avoided with dynamic re-routing incorporated into the DDT.

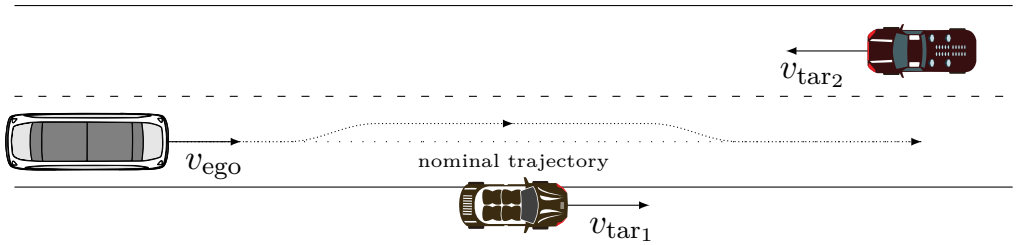


Figure 1-1: Schematic representation of the concerned driving scenario.

1-3 Driving Scenario

Throughout this thesis, the scenarios used are similar to the setting depicted in Figure 1-1. For this setting the following conditions are true:

- The GRT deviates with respect to the nominal trajectory
- The GRT does not exit its own lane and does therefore not encounter oncoming traffic
- The GRT shall always return to the nominal trajectory

Furthermore it is assumed that passing a target vehicle is allowed by a higher level decision making algorithm. Although generating such an algorithm is not within scope of this thesis, a simple rule-base will be implemented for numerical simulations.

1-4 Thesis Goal

From the need of incorporating a re-routing module in the GRT's DDT the main goal of this work can be formulated:

Design and implement a real-time trajectory generation approach that, in the standard driving scenario, is guaranteed to find a collision-free trajectory that is feasible regarding the vehicle's dynamics and brings a GRT, without exceeding comfort and mechanical constraints, from an initial to a desired final state. [10]

From this main goal the main requirements for the trajectory generator can be derived:

- The trajectory generator shall run in real-time
- The generated trajectory shall be feasible regarding the vehicle's dynamics
- The generated trajectory shall give passengers a comfortable ride

Throughout the thesis, the trajectory generator will be designed for, and tested against, those requirements.

1-5 Thesis Contributions

This thesis work introduces the first dynamic trajectory generator for 2getthere's GRT. The algorithm is developed within an existing state-of-the-art MPC framework. The author's contribution in this work is twofold.

First, and foremost, the notion of an ellipse shaped collision avoidance constraint has been extended by introducing a flexible shell around a standard ellipse shaped collision avoidance constraint. To the author's knowledge, the ellipse shaped collision avoidance constraint with flexible shell has not been used in literature before [10]. Due to the introduced flexible shell, the extended constraint is applicable to Sequential Quadratic Programming based solving techniques. Additionally, the flexible shell formulation has several advantages over using the standard ellipse shaped collision avoidance constraint.

Secondly, all work done throughout this thesis is scoped on implementing the trajectory generation algorithm on 2getthere's GRT. The solution strategy proposed and models used are tailored to the GRT's specifications and system architecture. Moreover, the closed-loop planner is designed, based on control laws similar to the control approach used in the real GRT, increasing the compatibility of generated trajectories over the standard open-loop trajectory generation.

1-6 Thesis Outline

The remainder of this thesis has the following structure; in Chapter 2 several motion planning frameworks are introduced. Furthermore it is argued why the MPC motion planning framework is appropriate for motion planning for the GRT. The prediction models needed to predict the motion of the ego vehicle and its surroundings are then introduced in Chapter 3. Additionally, the constraints that are set to ensure safe motion planning within this framework are handled in the same chapter as well. The chapter concludes by defining a two step trajectory generation approach. Chapter 4 introduces the tools used to solve the resulting OCPs and presents the simulation setup that is used to test the planner. The results are presented and analysed in Chapter 5. Chapter 6 concludes this work with an overall summary, conclusions of the results and an outlook for future work.

Chapter 2

Trajectory generation framework

The main purpose of trajectory generation is to provide the vehicle with a safe and collision-free time-parametrized path that brings the vehicle from the current initial state as close as possible to a target state. While planning this time-parametrized path, the GRT's dynamics and manoeuvring capabilities as well as traffic rules and road boundaries have to be taken into account. In order to have a collision-free path, the presence and manoeuvring of obstacles have to be taken into account additionally.

Generating a safe and collision-free trajectory consists of three consecutive steps subsequently consisting of several sub-tasks:

1. Environment Sensing - Obtain data from the environment and reference systems
    - Obtain sensor information
    - Obtain nominal trajectory
2. Planning and Control - Generate safe, comfortable and feasible trajectories
    - Predict surrounding traffic
    - Re-evaluate nominal trajectory, re-route if necessary
    - Plan inputs to follow this trajectory
3. Low Level Actuation - Operate the vehicle such that it follows the planned trajectory
    - Control steering angles via steering interface
    - Control vehicles acceleration via acceleration interface

in which sub-task 2 typically includes the path planning (class 2) until trajectory planning (class 4) classes as introduced in Section 1-1. After the low level controls have been applied, the system receives new environmental data and the cycle starts over again. This typical loop for an Automated Driving System (ADS) is depicted in Figure 2-1.

According to literature, algorithms running the planning and control task can be classified in algorithms that scope on feasible path planning, optimal path planning, or a combination of

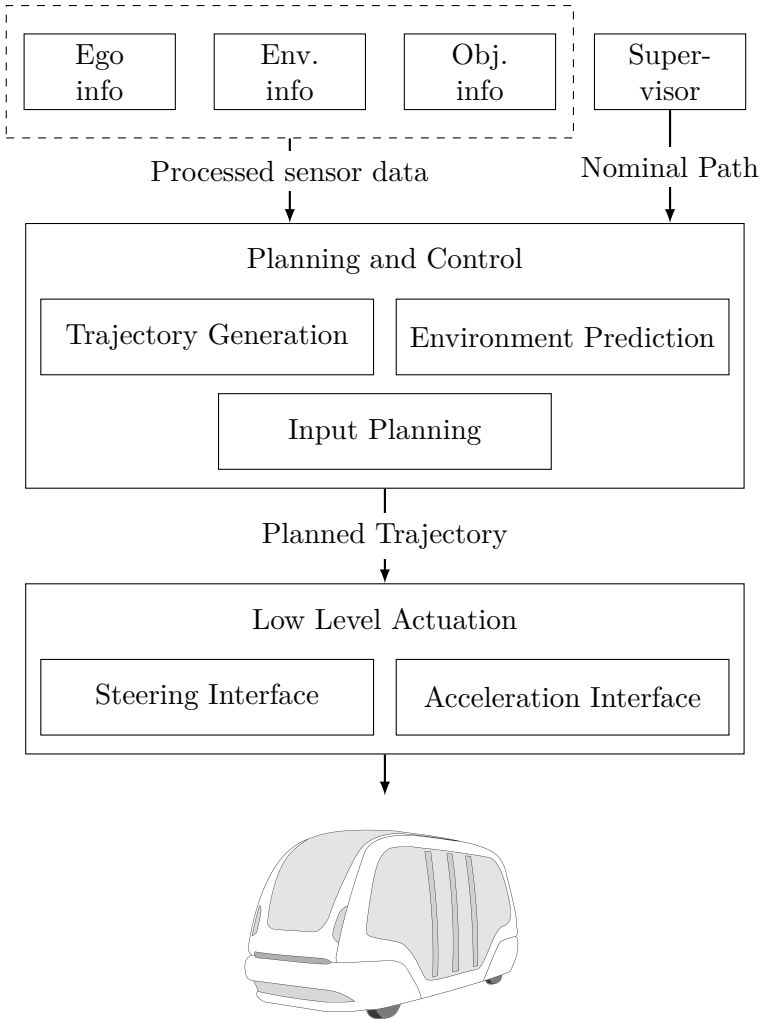


Figure 2-1: Schematic representation of the system architecture of an Automated Driving System.

both [11]. When concerning feasible path planning, one refers to finding a suitable path that satisfies the stated problem constraints. This does not necessarily take into account whether a solution is optimal or not, in the sense that some performance criterion is minimized. Optimal path planning on the other hand refers to the problem of finding a path that optimizes some performance criterion subject to constraints. Most classical approaches from literature are scoped on feasible path planning whereas extensions to original algorithms should be made to achieve optimal paths [12].

For the remainder of this chapter, some background on path planning and state-of-the-art approaches used for trajectory generation will be discussed in Section 2-1. In Section 2-2 a detailed explanation is given of working principle of the Model Predictive Control (MPC) based trajectory generation framework. The Optimal Control Problem (OCP) formulation solved every MPC iteration is introduced as well as the Karush-Kuhn-Tucker (KKT) optimality conditions for a general Nonlinear Program (NLP). The chapter concludes with a short summary in Section 2-3.

2-1 State-of-the-art trajectory generation

From the very beginning of the era of robotics, the problem of finding a feasible, collision-free path has been the interest of many research groups. Path planning has undergone a lot of development since the invention of Dijkstra's Algorithm [13]. However, most path planning algorithms such as the Rapidly-Expanding Random Tree (RRT) or potential field algorithm only scope on finding a feasible path. Along this path then another planner is needed in order to convert the static points in space to time-parametrized paths.

With the distinction between path planning and trajectory generation made in Chapter 1, not so many options for a trajectory planner remain. In their review paper, Katrakazas et al. distinguish two main categories for trajectory generation [7] being either geometric curve based planning or Model Predictive Control (MPC)-based planning.

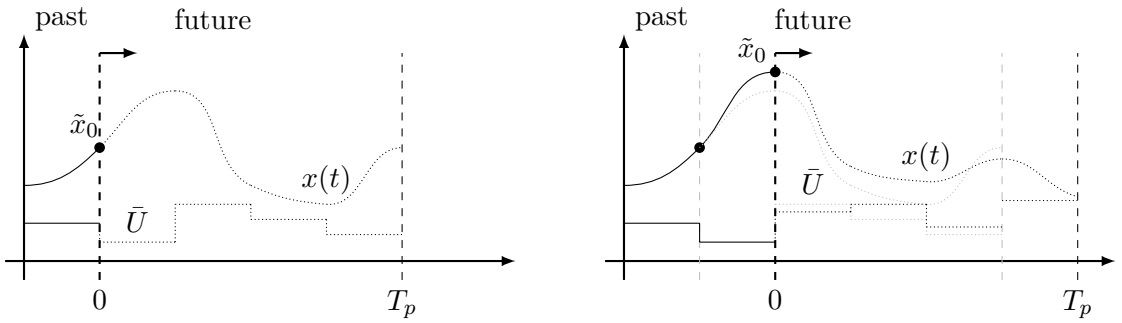
Geometric based curve planners optimize a certain geometric curve between two or more control points. This type of planners is low in computational cost and can provide continuous concatenations of curves. Typically, higher order polynomials [14], Bézier- [15] or spline curves [16] are used to represent the trajectory generated by the planner. The main downside is that control points used by the planner need to be placed in advance by another algorithm such as for RRT, thus the optimality of the generated trajectory is heavily depending on these underlying routines.

In Model Predictive Control (MPC) based planning, a dynamic model of the system is used to predict its future evolution of states [17]. Two typical consecutive steps of solved MPC instances are shown in Figure 2-2. Based on a prediction model, at each time step t, an OCP is solved over the prediction horizon T_p subject to a set of constraints and the initial state $\tilde{x}_0$. The optimization routine returns a policy future inputs ($\bar{U}$) minimizing a certain cost function (J). When a solution is found, the first step of the solution is applied, the problem is then solved again at a new sampling instance to find and apply the optimal input. This way the planned trajectory can easily avoid collisions or stay close to the center-lane. Since the prediction horizon is moving with the sampling instance, MPC is also often referred to as Receding Horizon Control (RHC).

2-2 Model Predictive Control

In this thesis a Model Predictive Control (MPC) based framework is build to generate both feasible and optimal trajectories. The major benefits of using MPC lie in the ease to extend the problem. Constraints can be added so that the vehicle will keep safe minimum distances to objects while not exceeding jerk or acceleration constraints [18]. For geometric based planners, these type of constraints have to be set based on the curvature and slope of the trajectory without knowing whether the vehicle will be able to follow this trajectory. The use of MPC is beneficial since hard constraints on the collision avoidance can be set and thus collision avoidance is ensured as long as the OCP remains feasible.

The MPC cost function can be tuned based on the vehicle state, leading to state dependent optimal trajectories. Furthermore due to the updating and replanning in a receding manner, MPC is already a feedback controller on its own and thus robust to disturbances and



(a) Initial MPC iteration. The optimal piecewise input policy is computed and first input is applied

(b) Next MPC iteration. New initial state is obtained and used to plan new piecewise input policy

Figure 2-2: Two consecutive MPC steps with receding horizon

measurement inaccuracies. Finally, due to the predictive nature of planning and constraint handling, the algorithm corresponds to a human driver planning a driveable trajectory.

2-2-1 Optimal Control Problem formulation

Optimization-based trajectory generation results from iteratively solving a constrained Optimal Control Problem. The general OCP problem formulation that is solved in MPC, has the following form:

$$\underset{y(\cdot),u(\cdot)}{\text{minimize}} \quad \underbrace{\int_{t_0}^{t_0+T_p} \left(\|y(t) - y_{\text{ref}}(t)\|_Q^2 + \|u(t) - u_{\text{ref}}(t)\|_R^2 \right) dt + \|y(t_0+T_p) - y_{\text{ref}}(t_0+T_p)\|_P^2}_{J} \tag{2-1a}$$

subject to

$$x(t_0) = \tilde{x}_0 \qquad \text{initial condition} \tag{2-1b}$$
$$\dot{x}(t) = f(x(t), u(t)) \qquad \text{vehicle dynamics} \tag{2-1c}$$
$$y(t) = h(x(t), u(t)) \qquad \text{output mapping} \tag{2-1d}$$
$$\underline{x}(t) \leq x(t) \leq \overline{x}(t) \qquad \text{state constraints} \tag{2-1e}$$
$$\underline{u}(t) \leq u(t) \leq \overline{u}(t) \qquad \text{actuator constraints} \tag{2-1f}$$
$$g(x(t), u(t)) \geq 0 \qquad \text{collision avoidance constraints} \tag{2-1g}$$

$$Q \in \mathbb{R}^{n_y}_{\succeq 0}, \quad R \in \mathbb{R}^{n_u}_{\succ 0}, \quad P \in \mathbb{R}^{n_y}_{\succeq 0}. \tag{2-1h}$$

The solution of the OCP is the result of minimizing the cost function J as in (2-1a) in which Q, R and P are non-negative definite weighting matrices penalizing the output (y) and input (u) deviations from their reference values (y_{ref}, u_{ref}). These weighting matrices can therefore be used as tuning parameters of the solution of the OCP. The OCP of (2-1) consists of three main parts:

- Choosing and setting up prediction models as in (2-1c)
- Designing constraints as in (2-1e) till (2-1g)
- Tuning MPC weights from (2-1h)

These three separate parts of designing the MPC-Framework will be handled in the coming Chapters of this thesis. Chapter 3 handles the modeling required for the equality constraints of (2-1c). Additionally, this chapter lists the constraints used in this work. Chapter 4 explains why the choice is made to solve the designed framework within the `ACADO Toolkit`. Furthermore it describes how the original MPC formulation of (2-1) is approximated by a discretized variant and how the discrete step time (T_s) and prediction horizon (N_p) are chosen. Finally Chapter 5 lists the costs matrices as used in this work and shows the results of numerical simulations.

2-2-2 Karush-Kuhn-Tucker optimality conditions for a NLP

The first-order necessary conditions for a solution of a NLP to be optimal are given by the Karush-Kuhn-Tucker (KKT) conditions. These are listed below:

$$\nabla_{y,u} J\left(y^{\text{opt}}, u^{\text{opt}}\right) + \sum_{i=1}^{m} \mu_i^{\text{opt}} \nabla_{y,u} g_i\left(y^{\text{opt}}, u^{\text{opt}}\right) + \sum_{j=1}^{p} \lambda_j^{\text{opt}} \nabla_{y,u} h_j\left(y^{\text{opt}}, u^{\text{opt}}\right) = 0 \tag{2-2a}$$

$$\mu_i^{\text{opt}} g_i\left(y^{\text{opt}}, u^{\text{opt}}\right) = 0, \qquad i = 1, \ldots, m \tag{2-2b}$$

$$\mu_i^{\text{opt}} \geq 0, \qquad i = 1, \ldots, m \tag{2-2c}$$

$$g_i\left(y^{\text{opt}}, u^{\text{opt}}\right) \leq 0, \qquad i = 1, \ldots, m \tag{2-2d}$$

$$h_j\left(y^{\text{opt}}, u^{\text{opt}}\right) = 0, \qquad i = 1, \ldots, p \tag{2-2e}$$

in which g_i are the inequality constraints of (2-1g) augmented with the system bounds in (2-1e) and (2-1f) and h_j are the equality constraints describing the vehicle dynamics in (2-1c). Furthermore $(y^{\text{opt}},\ u^{\text{opt}})$ and $(\mu^{\text{opt}},\ \lambda^{\text{opt}})$ are the primal and dual optimal points respectively. Any solution to (2-1) for which there exists a μ and λ that satisfy these conditions, is an optimal solution.

In case of a convex primal problem, the KKT conditions are also sufficient conditions for optimality. In some special cases, the KKT conditions can be solved analytically, however, this is generally not the case [19]. In most algorithms, the KKT conditions are used either for a measure of optimality, and thus quality, of the solution, or as a termination tolerance.

2-3 Summary

This chapter introduced the planning loop of an intelligent vehicle as an iterative process of sensing, planning and low-level actuation. It is explained that this thesis focusses specifically on this second step.

Several state-of-the-art trajectory planning approaches have been introduced, highlighting MPC for this thesis for its advantages in extending the trajectory generation problem and the possibility of fine-tuning resulting trajectories on comfort and safety.

The continuous time OCP solved every MPC iteration is introduced as well as the first-order necessary conditions for solving this NLP. The next chapter will provide further ingredients for modeling both the equality and inequality constraints for this OCP.

Chapter 3

Modeling and Problem setup

Model Predictive Control (MPC) requires a high-fidelity vehicle model in order to make accurate predictions. In order to obtain safe and comfortable re-routing as well, additional constraints have to be set up. In this chapter, three different models will be handled and constraints used in the problem setup will be introduced. Additionally three Optimal Control Problems will be formulated.

First an introduction is given to the coordinate frames used in Section 3-1, next a slightly more detailed section on the used curvilinear framework is presented in Section 3-2. Section 3-3 then describes how other vehicles are represented in this work. This is followed by two sections on the modeling of the GRT's dynamics. Section 3-4 concerns the vehicle's longitudinal dynamics and states a low order model describing these dynamics. Section 3-5 presents a detailed model for the lateral vehicle dynamics that is used for the open-loop lateral planner and proposes two control laws for the low-level lateral controller such that a closed-loop lateral planner can be formulated. Next, Section 3-6 lists the modeling of the constraints used throughout this thesis and states how they differ from the often used approach in literature. Then Section 3-7 binds it all together formulating three Optimal Control Problems for the two-step trajectory generation approach. This chapter concludes with a short summary in Section 3-8.

Note that all coordinate reference frames for the vehicle modeling will adhere to the Z-UP coordinate frame according to the Society of Automotive Engineers (SAE) standards presented in [1]. The definition of this coordinate frame is depicted in Figure 3-1 in which ψ, θ and ϕ are yaw-, pitch- and roll rotational motions respectively.

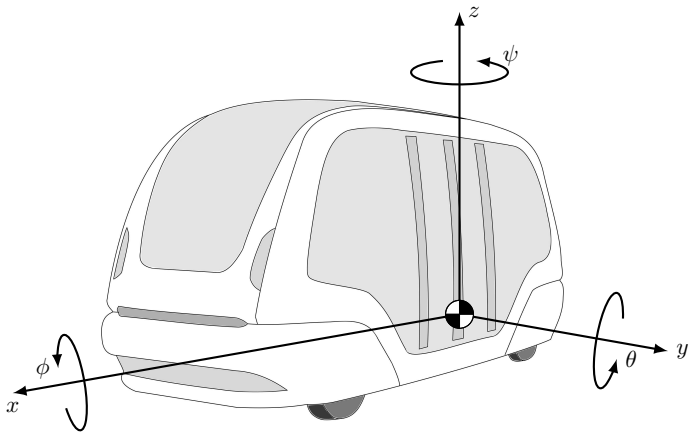


Figure 3-1: Definition of the Z-UP coordinate frame as in [1]

3-1 Coordinate Frames

In order to effectively re-plan the GRT's route four different Frenet coordinate frames are used:

- Global coordinate frame $(\vec{e}_x^G, \vec{e}_y^G)$
- Road coordinate frame $(\vec{e}_x^s, \vec{e}_y^s)$
- Vehicle coordinate frame $(\vec{e}_x^v, \vec{e}_y^v)$
- Movement coordinate frame $(\vec{e}_x^m, \vec{e}_y^m)$

The global coordinate frame is earth-fixed, the vehicle frame is vehicle-fixed, and road frame is a frame moving along the road, whose x-axis is directed according to the road. These three Frenet frames are depicted in Figure 3-2. The relation between all three coordinate systems can be described by a combined operation of translation and rotation. The relation between the vehicle coordinate frame and movement coordinate frame consists only of a rotation over angle β and is depicted in Figure 3-3.

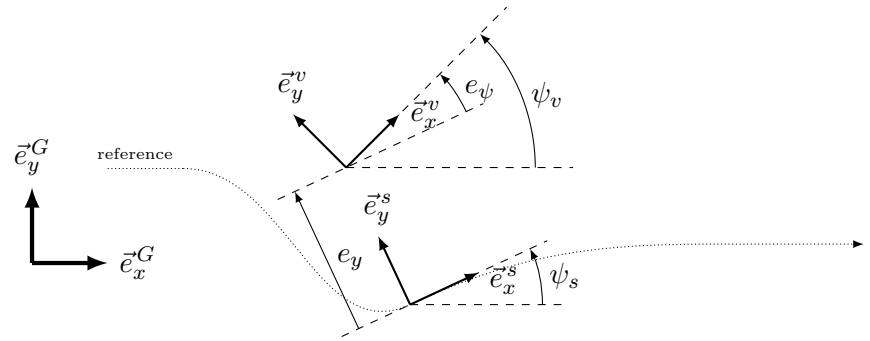


Figure 3-2: Schematic representation of the relation between the global $(\vec{e}^G)$, road $(\vec{e}^s)$ and vehicle $(\vec{e}^v)$ Frenet frame used in this work

3-2 Curvilinear Framework

It is convenient to place obstacles and compute the vehicle's position relative to the road coordinate system instead of the global coordinate system. The movement of the vehicle's center of gravity with respect to the road coordinate system can be described by the equations of a curvilinear framework [20, 21]:

$$e_\psi = \psi_v - \psi_s \tag{3-1a}$$
$$\dot{e}_y = v_x \sin(e_\psi) + v_y \cos(e_\psi) \tag{3-1b}$$
$$\dot{s} = \frac{v_x \cos(e_\psi) - v_y \sin(e_\psi)}{1 - e_y \kappa} \tag{3-1c}$$

in which κ is the road curvature as obtained by the vehicle's sensors. s is the coordinate along $\vec{e}_x^s$ and $\dot{s}$ is its derivative. Quantities e_y and e_ψ are the respective lateral and rotational offset with respect to the road coordinate frame. Both are depicted in Figure 3-2 in general as well as Figure 3-3 for the single-track model. In order to simplify the formulations of the MPC, considered scenarios will include straight roads only, therefore $\kappa = 0$, $\psi_s = 0$ is substituted in (3-1) from which it follows that:

$$e_\psi = \psi_v \tag{3-2a}$$
$$\dot{e}_y = v_x \sin(e_\psi) + v_y \cos(e_\psi) \tag{3-2b}$$
$$\dot{s} = v_x \cos(e_\psi) - v_y \sin(e_\psi) \tag{3-2c}$$

Still, scenarios can be easily extended to include curved roads as well by substituting the notions of (3-1).

3-3 Surrounding traffic prediction

In order to provide safe trajectory generation, the movement of surrounding vehicles needs to be predicted. For this generally three different classes of models are distinguished. Models are classified as being either physics-based, manoeuvre-based or interaction-aware [22]. For the purpose of re-routing, physics-based class provides enough insight in the movement of surrounding vehicles.

As the obstacles information will be updated at the same update rate as the replanning will be executed, a constant heading, constant velocity model is sufficient to predict the obstacle's state evolution. Therefore, the surrounding traffic prediction can be formulated as:

$$s_{\text{obj}}(t) = s_{\text{obj}_0} + \int v_{x_{\text{obj}}} dt \tag{3-3a}$$
$$e_{y_{\text{obj}}}(t) = y_{\text{obj}_0} + \int v_{y_{\text{obj}}} dt \tag{3-3b}$$

Note that this approach implicitly also assumes that the heading of the object is aligned with the road coordinate frame and that target lateral motion is thus always perpendicular to the road center lane.

3-4 Longitudinal dynamics

In order to create a trajectory generator, one should not only simulate at constant velocity. Therefore, a simple, but efficient, model is used to describe the longitudinal dynamics of the vehicle. The model used in this thesis is formulated as follows:

$$\dot{v}_x = \frac{F_{\text{drive}} - F_{\text{roll}} - F_{\text{air}}}{m} \tag{3-4a}$$

$$\dot{s} = v_x \cos(e_\psi(t)) - v_y(t) \sin(e_\psi(t)) \tag{3-4b}$$

in which F_{roll} and F_{drag} are the roll and drag resistance respectively:

$$F_{\text{roll}} = C_{rr}\ m\ g \tag{3-5a}$$

$$F_{\text{drag}} = \frac{\rho_{\text{air}}\ C_d\ A_f}{2} \cdot v_x^2 \tag{3-5b}$$

with C_{rr}, C_d, ρ_{air}, A_f, m, and g being the roll and drag resistance coefficient, air density, vehicle frontal surface, vehicle mass and gravitational constant respectively. Note that while planning longitudinal motion, the evolution of $v_y(t)$ and $e_\psi(t)$ is received from the lateral planner. The system of (3-4) is used for the longitudinal planning of the GRT.

3-5 Lateral dynamics

For the dynamic re-routing, the lateral dynamics are the fastest dynamics present in the dynamical system. Therefore it is of high importance to describe these dynamics accurately. This section exists of three subsections. First it is explained how the nominal system is modelled, secondly a feedforward and feedback law for controlling the vehicle along the rerouted trajectory is introduced. Finally the nominal system and the two control laws are combined into a single closed-loop system which is used to plan the trajectory while taking into account the low-level controller dynamics.

3-5-1 Open-loop representation

The GRT's lateral dynamics are described by the linearized single-track model. The single-track or 'Bicycle' model is originally invented by Dr. Riekert and Dr. Schunck [23]. It includes several important omissions and simplifications of the original vehicle. These simplifications serve to greatly reduce the model's degree of freedom, but do not significantly affect the vehicle dynamic behaviour results in the linear range. It can be used to quickly and easily analyse vehicle behaviour. The single-track model can also be easily converted to a computer simulation model. The model includes the following two important simplifications [24]:

- It is assumed that the height of the vehicle's center of gravity is at the level of the road surface, as a result, vertical tire forces at the inner and outer wheels remain the same during cornering and vertical forces at the front and rear wheels remain the same during longitudinal acceleration. Consequently, the rolling motion during cornering is not modelled.
- The equations of motion of the single-track model are linearized. This linearity, which is valid for angles smaller than 4 [°], applies not only to steering angle functions ($\sin(\alpha) \approx \alpha$ and $\cos(\alpha) \approx 1$ for small angles), but also to the tire side slip angles. As a consequence tire characteristics are assumed to operate in the linear regime.

According to [25] this single-track model neatly represents the vehicles dynamics for lateral accelerations below 0.4 [g]. Since the GRT's requirements state that the vehicle is not allowed to exceed even a fourth of this magnitude, it is safe to use the vehicle model for our planning purposes.

Classical single-track model with front and rear wheel steering

Figure 3-3 depicts the single-track representation of the GRT including both front and rear wheel steering. Typically, the system states consist of the lateral velocity (v_y) and rotational velocity ($\dot{\psi}$) as such:

$$\underbrace{\begin{bmatrix} \dot{v}_y \\ \ddot{\psi} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -\frac{C_r+C_f}{mv_x} & \frac{C_r l_r - C_f l_f}{mv_x} - v_x \\ \frac{C_r l_r - C_f l_f}{I_z v_x} & -\frac{C_r l_r^2 + C_f l_f^2}{I_z v_x} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{C_f}{m} & \frac{C_r}{m} \\ \frac{C_f l_f}{I_z} & -\frac{C_r l_r}{I_z} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \delta_f \\ \delta_r \end{bmatrix}}_{u} \tag{3-6}$$

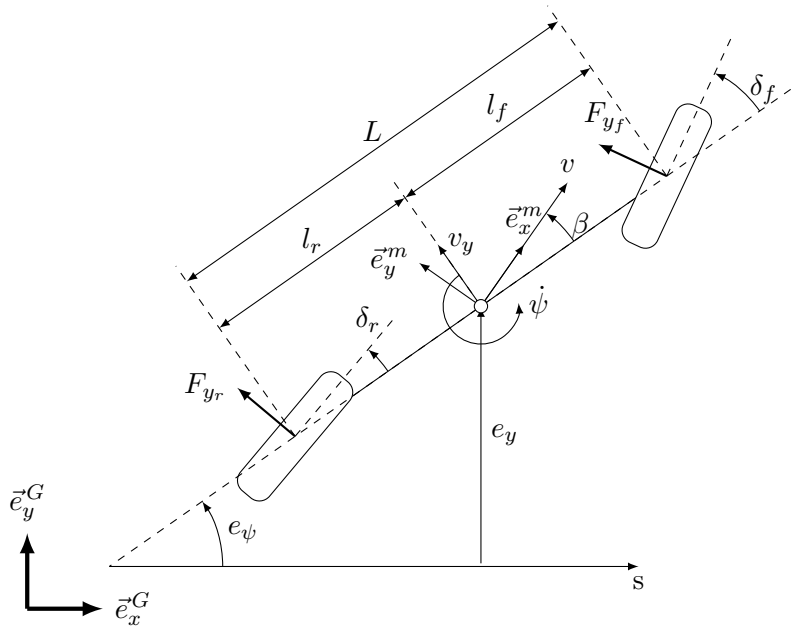


Figure 3-3: Schematic representation of a front and rear wheel steered single-track model

in which m denotes the vehicle's mass and I_z denotes the rotational inertia around the z axis. l_f and l_r denote the distances from the vehicle's center of gravity to the front and rear wheel respectively. C_f and C_r denote the tire stiffness of the front and rear wheel and δ_f, δ_r their respective steering angles.

For planning purposes, the curvilinear framework of (3-2) can be included in this system as well resulting in:

$$\begin{pmatrix} \dot{v}_y \\ \ddot{\psi} \\ \dot{e}_\psi \\ \dot{e}_y \\ \dot{s} \\ \dot{v}_x \end{pmatrix} = \begin{pmatrix} \left(-\frac{C_r+C_f}{mv_x}\right) v_y + \left(\frac{C_r l_r - C_f l_f}{mv_x} - v_x\right) \dot{\psi} + \frac{C_f}{m}\delta_f + \frac{C_r}{m}\delta_r \\ \left(\frac{C_r l_r - C_f l_f}{I_z v_x}\right) v_y - \left(\frac{C_r l_r^2 + C_f l_f^2}{I_z v_x}\right) \dot{\psi} + \frac{C_f l_f}{I_z}\delta_f - \frac{C_r l_r}{I_z}\delta_r \\ \dot{\psi} \\ v_x \sin(e_\psi) + v_y \cos(e_\psi) \\ v_x \cos(e_\psi) + v_y \sin(e_\psi) \\ 0 \end{pmatrix} \tag{3-7}$$

The linearized system of (3-7) has two poles due to the vehicle dynamics, whose position change based on the velocity of the vehicle. Furthermore it has three poles at the origin due to the development of the error within the model. This makes the open-loop system marginally stable [26]. The linearized system (3-7) is fully controllable making it usable for trajectory generation. Note that during the lateral planning, the longitudinal velocity (v_x) is updated every iteration, but kept constant while planning. The following two subsections will elaborate on how the system (3-7) can be used to create a closed-loop planner, but first the system will be augmented such that in a open-loop planning case, bounds can be set on the steering rate additionally.

Steering rate input

The GRT requirements also limit the steering rate as input to the system. In order to take this into account in the MPC based trajectory generator, the system state can be augmented with the steering angle [δ_f, δ_r] so that the steering rate can be used as input:

$$\underbrace{\begin{pmatrix} \dot{v}_y \\ \ddot{\psi} \\ \dot{e}_\psi \\ \dot{e}_y \\ \dot{s} \\ \dot{v}_x \\ \dot{\delta}_f \\ \dot{\delta}_r \end{pmatrix}}_{\dot{x}} = \underbrace{\begin{pmatrix} \left(-\frac{C_r+C_f}{mv_x}\right) v_y + \left(\frac{C_r l_r - C_f l_f}{mv_x} - v_x\right) \dot{\psi} + \frac{C_f}{m}\delta_f + \frac{C_r}{m}\delta_r \\ \left(\frac{C_r l_r - C_f l_f}{I_z v_x}\right) v_y - \left(\frac{C_r l_r^2 + C_f l_f^2}{I_z v_x}\right) \dot{\psi} + \frac{C_f l_f}{I_z}\delta_f - \frac{C_r l_r}{I_z}\delta_r \\ \dot{\psi} \\ v_x \sin(e_\psi) + v_y \cos(e_\psi) \\ v_x \cos(e_\psi) + v_y \sin(e_\psi) \\ 0 \\ \dot{\delta}_f \\ \dot{\delta}_r \end{pmatrix}}_{f(x,u)} \tag{3-8}$$

Note that augmenting the system places two additional open-loop poles at zero for the linearized system, but does not affect the controllability of the system. The system (3-8) can now be used to plan trajectories that suffice the GRT steering rate constraints. The system of (3-8) is therefore used for the open-loop lateral planning of the GRT.

3-5-2 Low-level control of the open-loop plant

A low-level controller has been designed to predict the steering angles based on the reference trajectory and to regulate a possibly developed rotational or translational error back to zero. Variables describing the reference trajectory are denoted with the subscript ($\star_{\text{ref}}$) whereas internal system states have no particular subscript. The low-level controller that is implemented in this thesis consists of two control laws:

- A feedforward rule describing how the reference trajectory can be converted to an estimated front and rear steering angle
- A feedback rule steering the remaining error to zero over time

The proposed interconnection of the feedforward and feedback gain with the vehicle system is depicted in Figure 3-4. In this thesis the interconnection is used for two purposes:

1. As a validation to test if the computed reference trajectory can be followed by the GRT, taking into account the low-level controller
2. As a combination of which the closed-loop dynamics can be used to predict a new trajectory, taking into account the low-level dynamics already in the planning phase

Results for point 1 are shown in Chapter 5. A detailed explanation of the closed-loop interconnection of the two control laws and the vehicle dynamics is handled in Section 3-5-3.

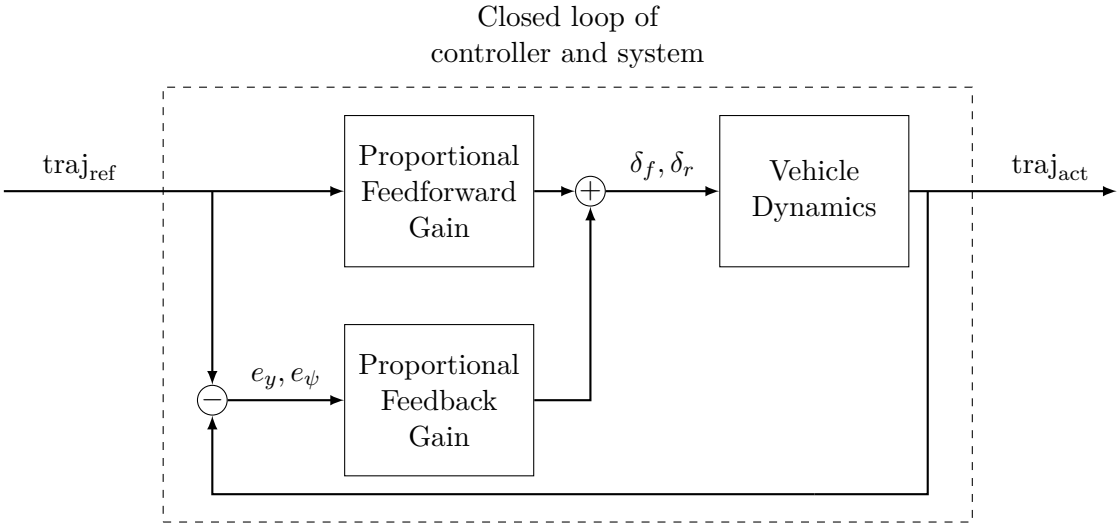


Figure 3-4: Schematic setup of the proposed controller used for feedforward and feedback control.

Feedforward control law

The feedforward control law is used to estimate a steering angle based on the shape of the reference trajectory. A typical feedforward term is based on steering angle at steady-state cornering. For front wheel steering this can be formulated as follows [27]:

$$\delta_{\text{ffw}} = \underbrace{L\kappa}_{\delta_A} + \underbrace{\frac{m}{L}\left(\frac{l_r}{C_f} - \frac{l_f}{C_r}\right)}_{U} \underbrace{v_x^2\kappa}_{a_y} = \delta_A + Ua_y \tag{3-9}$$

in which, L is the wheelbase, κ is the road curvature and U is the understeer/oversteer gradient. Equation (3-9) has to be adapted to be applicable for both front and rear wheel steering. This can be achieved by re-deriving the feedforward control term using the following relation on the sideslip angle:

$$\begin{aligned} \beta &= \tan^{-1}\left(\frac{l_f \tan\delta_r + l_r \tan\delta_f}{L}\right) \\ &\approx \frac{l_f}{L}\delta_r + \frac{l_r}{L}\delta_f \end{aligned} \tag{3-10}$$

It then follows that (3-9) for front and rear wheel steering can be written as:

$$\begin{aligned} \delta_{f_{\text{ffw}}} &= l_f\kappa_{\text{ref}} + \frac{l_f m}{L^2}\left(\frac{l_r}{C_f} - \frac{l_f}{C_r}\right) v_x^2\kappa_{\text{ref}} + \tan\beta_{\text{ref}} \\ &\approx \underbrace{\left(l_f + \frac{l_f m}{L^2}\left(\frac{l_r}{C_f} - \frac{l_f}{C_r}\right) v_x^2\right)}_{X_1} \kappa_{\text{ref}} + \beta_{\text{ref}} \end{aligned} \tag{3-11a}$$

$$\begin{aligned} \delta_{r_{\text{ffw}}} &= -l_r\kappa_{\text{ref}} - \frac{l_r m}{L^2}\left(\frac{l_r}{C_f} - \frac{l_f}{C_r}\right) v_x^2\kappa_{\text{ref}} + \tan\beta_{\text{ref}} \\ &\approx \underbrace{\left(-l_r - \frac{l_r m}{L^2}\left(\frac{l_r}{C_f} - \frac{l_f}{C_r}\right) v_x^2\right)}_{X_2} \kappa_{\text{ref}} + \beta_{\text{ref}} \end{aligned} \tag{3-11b}$$

in which β_{ref} and κ_{ref} are the slope and curvature of the computed reference trajectory respectively.

Feedback control law

The feedback rule introduced next, is used to drive the lateral error (e_{y_2}) and orientation error (e_{ψ_2}) with respect to the computed trajectory ($e_{y_{\text{ref}}}$,$e_{\psi_{\text{ref}}}$) back to zero. A schematic of the error handling is depicted in Figure 3-5 and Figure 3-6. The control laws steering those depicted errors back to zero are shown below:

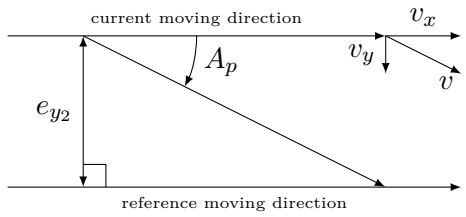


Figure 3-5: Steering lateral error to zero assuming no orientation error.

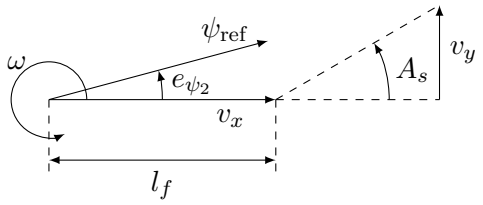


Figure 3-6: Steering orientation error to zero assuming no lateral error.

$$\delta_{f_{\text{fb}}} = \frac{l_r}{v_x \tau_1} e_{\psi_2} + \frac{1}{v_x \tau_2} e_{y_2} \tag{3-12a}$$

$$\delta_{r_{\text{fb}}} = \frac{-l_f}{v_x \tau_1} e_{\psi_2} + \frac{1}{v_x \tau_2} e_{y_2} \tag{3-12b}$$

These control laws are both velocity dependent as well as tunable by time constants τ_1 and τ_2. Furthermore they are dependent on the vehicle's kinematics. Using this feedback law, the orientation error is regulated to zero using counter steering (A_s) over τ_1 [s]. The lateral offset to the reference trajectory is regulated to zero using a crabbing manoeuvre over crab angle (A_p) over τ_2 [s].

Control laws combined

In order to combine both the feedforward and feedback control law, it is necessary to have all the inputs to the system consisting of a system state. Since β_{ref} and κ_{ref} are the first and second derivative of the reference trajectory with respect to the distance driven, both laws are not instantly comparable. In order to make both the feedforward and feedback rule comparable, β_{ref} and κ_{ref} are estimated as follows:

$$\beta_{\text{ref}} \approx \frac{lr}{L} \delta_{f_{\text{ref}}} + \frac{lf}{L} \delta_{r_{\text{ref}}} \tag{3-13a}$$

$$\begin{aligned} \kappa_{\text{ref}} &\approx \frac{a_n}{v_x^2} = \frac{v_x \left(\dot{\psi}_{\text{ref}} + \dot{\beta} \right)}{v_x^2} = \frac{\left(\dot{\psi}_{\text{ref}} + \dot{\beta} \right)}{v_x} \\ &\approx \underbrace{-\frac{C_r + C_f}{m v_x^3}}_{\frac{A_{11}}{v_x^2}} v_{y_{\text{ref}}} + \underbrace{\frac{C_r l_r - C_f l_f}{m v_x^3}}_{\frac{A_{12}+v_x}{v_x^2}} \dot{\psi}_{\text{ref}} + \underbrace{\frac{C_f}{m v_x^2}}_{\frac{B_{11}}{v_x^2}} \delta_{f_{\text{ref}}} + \underbrace{\frac{C_r}{m v_x^2}}_{\frac{B_{12}}{v_x^2}} \delta_{r_{\text{ref}}} \end{aligned} \tag{3-13b}$$

in which a_n is defined as the centripetal acceleration and the approximation of κ_{ref} is obtained by plugging in $\beta = \frac{v_y}{v_x}$ and v_y coming from the open-loop single-track model (3-6). Furthermore A_{11}, A_{12}, B_{11} and B_{12} are the corresponding entries from matrix A and B in (3-6). With (3-11) till (3-13) known, the total input $u = [\delta_f,\ \delta_r]^T$ to (3-7) is computed as:

$$\begin{bmatrix}\delta_{f_{\text{ffw}}}\\ \delta_{r_{\text{ffw}}}\end{bmatrix} = \underbrace{\begin{bmatrix} X_1\frac{A_{11}}{v_x^2} & X_1\frac{A_{12}+v_x}{v_x^2} & X_1\frac{B_{11}}{v_x^2}+\frac{lr}{L} & X_1\frac{B_{12}}{v_x^2}+\frac{lf}{L} & 0 & 0\\ X_2\frac{A_{11}}{v_x^2} & X_2\frac{A_{12}+v_x}{v_x^2} & X_2\frac{B_{11}}{v_x^2}+\frac{lr}{L} & X_2\frac{B_{12}}{v_x^2}+\frac{lf}{L} & 0 & 0\end{bmatrix}}_{\text{FFW}} \underbrace{\begin{bmatrix} v_{y_{\text{ref}}}\\ \dot{\psi}_{\text{ref}}\\ \delta_{f_{\text{ref}}}\\ \delta_{r_{\text{ref}}}\\ e_{\psi_{\text{ref}}}\\ e_{y_{\text{ref}}}\end{bmatrix}}_{x_{\text{ref}}} \tag{3-14a}$$

$$\begin{bmatrix}\delta_{f_{\text{fb}}}\\ \delta_{r_{\text{fb}}}\end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{lr}{v_x\tau_1} & \frac{1}{v_x\tau_2}\\ 0 & 0 & 0 & 0 & \frac{-lf}{v_x\tau_1} & \frac{1}{v_x\tau_2}\end{bmatrix}}_{\text{FB}} \underbrace{\begin{bmatrix} v_{y_{\text{ref}}}-v_y\\ \dot{\psi}_{\text{ref}}-\dot{\psi}\\ \delta_{f_{\text{ref}}}-\delta_f\\ \delta_{r_{\text{ref}}}-\delta_r\\ e_{\psi_{\text{ref}}}-e_\psi\\ e_{y_{\text{ref}}}-e_y\end{bmatrix}}_{x_{\text{ref}}-x} \tag{3-14b}$$

$$\begin{aligned}\begin{bmatrix}\delta_f\\ \delta_r\end{bmatrix} &= \begin{bmatrix}\delta_{f_{\text{ffw}}}\\ \delta_{r_{\text{ffw}}}\end{bmatrix} + \begin{bmatrix}\delta_{f_{\text{fb}}}\\ \delta_{r_{\text{fb}}}\end{bmatrix}\\ &= \begin{bmatrix} X_1\frac{A_{11}}{v_x^2} & X_1\frac{A_{12}+v_x}{v_x^2} & X_1\frac{B_{11}}{v_x^2}+\frac{lr}{L} & X_1\frac{B_{12}}{v_x^2}+\frac{lf}{L} & \frac{lr}{v_x\tau_1} & \frac{1}{v_x\tau_2}\\ X_2\frac{A_{11}}{v_x^2} & X_2\frac{A_{12}+v_x}{v_x^2} & X_2\frac{B_{11}}{v_x^2}+\frac{lr}{L} & X_2\frac{B_{12}}{v_x^2}+\frac{lf}{L} & \frac{-lf}{v_x\tau_1} & \frac{1}{v_x\tau_2}\end{bmatrix}\begin{bmatrix} v_{y_{\text{ref}}}\\ \dot{\psi}_{\text{ref}}\\ \delta_{f_{\text{ref}}}\\ \delta_{r_{\text{ref}}}\\ e_{\psi_{\text{ref}}}\\ e_{y_{\text{ref}}}\end{bmatrix}\\ &\quad - \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{lr}{v_x\tau_1} & \frac{1}{v_x\tau_2}\\ 0 & 0 & 0 & 0 & \frac{-lf}{v_x\tau_1} & \frac{1}{v_x\tau_2}\end{bmatrix}\begin{bmatrix} v_y\\ \dot{\psi}\\ \delta_f\\ \delta_r\\ e_\psi\\ e_y\end{bmatrix}\end{aligned} \tag{3-14c}$$

3-5-3 Closed-loop representation

With the nominal lateral dynamics introduced in Section 3-5-1 and the low level control laws introduced in Section 3-5-2 all necessities are available for computing the closed-loop system interconnection of the vehicle and controller. For this the result of (3-14c) is combined with the open-loop system (3-6) and augmented with the curvilinear framework of (3-2), which directly results in the new system $\hat{\dot{x}} = \hat{f}(x, x_{\text{ref}})$:

$$\begin{bmatrix} \dot{v}_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\left(B_{11}\frac{lr}{v_x\tau_1} + B_{12}\frac{-lf}{v_x\tau_1}\right) & A_{11} & A_{12} & -\left(B_{11}\frac{1}{v_x\tau_2} + B_{12}\frac{1}{v_x\tau_2}\right) \\ -\left(B_{21}\frac{lr}{v_x\tau_1} + B_{22}\frac{-lf}{v_x\tau_1}\right) & A_{21} & A_{22} & -\left(B_{21}\frac{1}{v_x\tau_2} + B_{22}\frac{1}{v_x\tau_2}\right) \end{bmatrix} \begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} + \tag{3-15a}$$

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} X_1\frac{A_{11}}{v_x^2} & X_1\frac{A_{12}+v_x}{v_x^2} & X_1\frac{B_{11}}{v_x^2} + \frac{lr}{L} & X_1\frac{B_{12}}{v_x^2} + \frac{lf}{L} & \frac{lr}{v_x\tau_1} & \frac{1}{v_x\tau_2} \\ X_2\frac{A_{11}}{v_x^2} & X_2\frac{A_{12}+v_x}{v_x^2} & X_2\frac{B_{11}}{v_x^2} + \frac{lr}{L} & X_2\frac{B_{12}}{v_x^2} + \frac{lf}{L} & \frac{-lf}{v_x\tau_1} & \frac{1}{v_x\tau_2} \end{bmatrix} \begin{bmatrix} v_{y_{\text{ref}}} \\ \dot{\psi}_{\text{ref}} \\ \delta_{f_{\text{ref}}} \\ \delta_{r_{\text{ref}}} \\ e_{\psi_{\text{ref}}} \\ e_{y_{\text{ref}}} \end{bmatrix}$$

$$\begin{pmatrix} \dot{e}_\psi \\ \dot{e}_y \\ \dot{s} \\ \dot{v}_x \end{pmatrix} = \begin{pmatrix} \dot{\psi} \\ v_x \sin(e_\psi) + v_y \cos(e_\psi) \\ v_x \cos(e_\psi) + v_y \sin(e_\psi) \\ 0 \end{pmatrix} \tag{3-15b}$$

Note that for typesetting reasons this system has been divided in a linear part (3-15a) and a partially non-linear part (3-15b). Similar to the open-loop planner (3-8), the longitudinal velocity v_x in this system is obtained and updated every iteration. The v_x terms in the denominator of this dynamical system causes the system to be singular for $v_x = 0$. Therefore, at velocities below $v_x = 1$, $v_x = 1$ will be used instead for lateral planning. The system of (3-15) is used for the closed-loop lateral planning of the GRT.

3-6 Inequality constraints

The latter sections handled the models that are used for accurate predictions in the MPC design. In order to meet the GRT requirements on comfort and safe re-routing, constraints have to be designed. The inequality constraints that are set in this work can be divided in bounds and general constraints. These two types are handled in the upcoming two subsections.

First upper and lower bounds on the system states and inputs will be handled. These may be varying over time but are mostly fixed values. Secondly the general constraints are handled. These path constraints are generally functions consisting of combinations of several system states or other variables. Path constraints are therefore highly dependent on the initial state of the optimization.

3-6-1 Bounds

The bounds on the Optimal Control Problem (OCP) are set on single states only. Most bounds follow directly from the requirements for the Group Rapid Transport (GRT). Some of these can be directly states as box constraints. Others have to be computed slightly different. Because of the two individual models dealing with longitudinal and lateral control, also the bounds have to be assigned to either of these models. And will be handled separately below.

Longitudinal planner

The longitudinal bounds are summarized below:

$$\underline{v}_x \leq v_x \leq \overline{v}_x \qquad \text{(speed limits)} \tag{3-16a}$$
$$\underline{a}_x \leq a_x \leq \overline{a}_x \qquad \text{(longitudinal acceleration limit)} \tag{3-16b}$$

Note that no longitudinal jerk limit is applied to the longitudinal controller since this controller is only used to keep the GRT as close as possible to the maximum speed instead of having a planner that is only operational at constant velocity. Furthermore including box constraints on the longitudinal jerk would demand a more complex model than the one introduced in Section 3-4.

Lateral planner

The lateral bounds are summarized as follows:

$$\underline{e}_y \leq e_y \leq \overline{e}_y \qquad \text{(lane boundaries)} \tag{3-17a}$$
$$\underline{\delta}_f \leq \delta_f \leq \overline{\delta}_f \qquad \text{(front steering limitation)} \tag{3-17b}$$
$$\underline{\delta}_r \leq \delta_r \leq \overline{\delta}_r \qquad \text{(rear steering limitation)} \tag{3-17c}$$
$$\underline{\dot{\delta}}_f \leq \dot{\delta}_r \leq \overline{\dot{\delta}}_f \qquad \text{(front steering rate limitation)} \tag{3-17d}$$
$$\underline{\dot{\delta}}_r \leq \dot{\delta}_r \leq \overline{\dot{\delta}}_r \qquad \text{(rear steering rate limitation)} \tag{3-17e}$$

These generally describe the bounds on the GRT's position on the road and the typical actuator limits. Note that the actuator steering rate bounds are only applicable to the open-loop model, since the steering rate ($\dot{\delta}$) is not available in the state or input of the closed-loop system. However, this steering rate will be limited in the simulation and testing environment.

Recall that in the introduction also lateral acceleration and lateral jerk were mentioned to remain bounded. Since these are not explicitly available in the state or input, they have to be modelled as path constraints as explained in the following subsection.

3-6-2 General constraints

The path constraints of the OCP are set mainly for collision avoidance. However, due to the model formulation of Section 3-5, also the lateral acceleration and lateral jerk are included in the planner as path constraints. For both the longitudinal and lateral planner an ellipse shaped collision avoidance constraint is utilized as listed below:

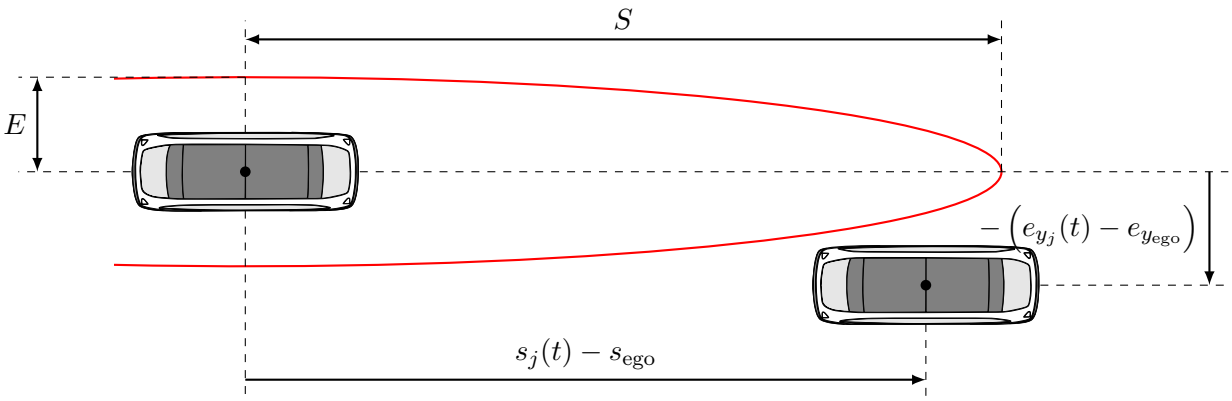


Figure 3-7: Visualisation of the standard ellipse shaped collision avoidance constraint of (3-18).

$$\left(\frac{s_j(t) - s_{\text{ego}}}{S}\right)^2 + \left(\frac{e_{y_j}(t) - e_{y_{\text{ego}}}}{E}\right)^2 \geq 1 \quad \forall j \in \{1, \ldots, n\} \tag{3-18}$$

$$S = \frac{L_{\text{ego}} + L_j}{2} + L_{\text{safety}}, \qquad L_{\text{safety}} = f(v_x)$$

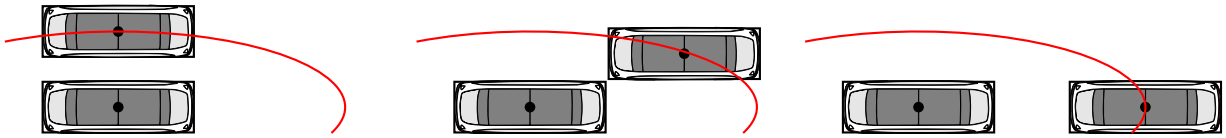
$$E = \frac{W_{\text{ego}} + W_j}{2} + W_{\text{safety}}, \qquad W_{\text{safety}} = f(v_x)$$

where the subscript j is used to denote the fact that the constraint is applied to n target vehicles. S and E are the size of the semi-major and semi-minor axis of the ellipse respectively. These are defined based on the GRT's and obstacles width (W) and length (L) plus a minimum safety distance. Note that this safety distance can be set as function of the longitudinal velocity. The time dependence is added since the target's position is predicted over the full prediction horizon using the physics based modeling approach from Section 3-3. A graphical representation of the collision avoidance constraint is depicted in Figure 3-7.

There are two main reasons for using an ellipse shaped collision avoidance constraint:

- The natural shape of the ellipse resembles the distance holding approach applied by a human driver; the longitudinal margin is larger than the lateral margin.
- Using an ellipse requires no switching of constraints when overtaking a vehicle whereas switching is required for e.g. affine constraints.

This constraint has been used in other literature as well [28, 29], but for this thesis an adapted representation of this collision avoidance constraint will be introduced since the standard constraint does not suit the solving framework that will be introduced in the following chapter. The collision avoidance constraint can be rotated to match the GRT's orientation if necessary. However, since the angles driven in overtake scenarios are small, no additional rotation is added. Although the longitudinal and lateral planner use the same constraint type, the final shape of both constraints differs slightly. The shaping of the constraints for both the longitudinal and lateral case is discussed below.



(a) E_{min} based on passing only is starting point for S, E.

(b) Critical collision avoidance condition.

(c) S_{min} follows from Figure 3-8a and Figure 3-8b.

Figure 3-8: Minimal sizes of a standard elliptical constraint depicted in three Figures.

Longitudinal planning

For the longitudinal planning (3-18) is slightly adjusted by substituting the GRT's lateral offset ($e_{y_{\text{ego}}}$) by the time-parametrized lateral GRT position ($e_{y_{\text{ego}}}(t)$) as planned by the lateral planner. The modified constraint is listed below:

$$\left(\frac{s_j(t) - s_{\text{ego}}}{S_{\text{long}}}\right)^2 + \left(\frac{e_{y_j}(t) - e_{y_{\text{ego}}}(t)}{E_{\text{long}}}\right)^2 \geq 1 \quad \forall j \in \{1, \ldots, n\} \tag{3-19}$$

$$S_{\text{long}} = \frac{L_{\text{ego}} + L_j}{2} + L_{\text{safety}}, \qquad L_{\text{safety}} = f(v_x)$$

$$E_{\text{long}} = \frac{W_{\text{ego}} + W_j}{2} + W_{\text{safety}}, \qquad W_{\text{safety}} = f(v_x)$$

The modification is performed such that the longitudinal planner re-plans a trajectory over the planned lateral offset provided by the lateral planner. This way less conservative longitudinal planning achieved over using the initial lateral offset only. The lateral safety distance is set as a function of the GRT's initial longitudinal velocity. The minimum lateral safety distance is depicted in Figure 3-8a. For this thesis it is chosen to keep this value fixed at a minimum passing distance of 0.5 [m]. Therefore E_{min} is set to be:

$$E_{\text{min}} = \frac{W_{\text{ego}} + W_j}{2} + 0.5 \tag{3-20}$$

The longitudinal safety distance consists of both a static and a dynamic part. The longitudinal static part is deduced from the case that is depicted in Figure 3-8b assuming the lateral passing distance is the exact minimum of 0.5 [m]. The situation depicted in this Figure shows the case where the bounding boxes around the GRT and target vehicle are touching corners. This determines the minimum of size S_{long} being:

$$S_{\text{min}} = \frac{L_{\text{ego}} + L_j}{2} + L_{\text{stat}} = \sqrt{\frac{\left(\frac{L_{\text{ego}}+L_j}{2}\right)^2}{1 - \frac{\left(\frac{W_{\text{ego}}+W_j}{2}\right)^2}{E_{\text{min}}^2}}} \tag{3-21}$$

Note that for a typical GRT-to-GRT overtake maneuver, this would result in $L_{\text{stat}} = 4$ [m]. The longitudinal dynamic part can be deduced from the stable ACC time gap which for this

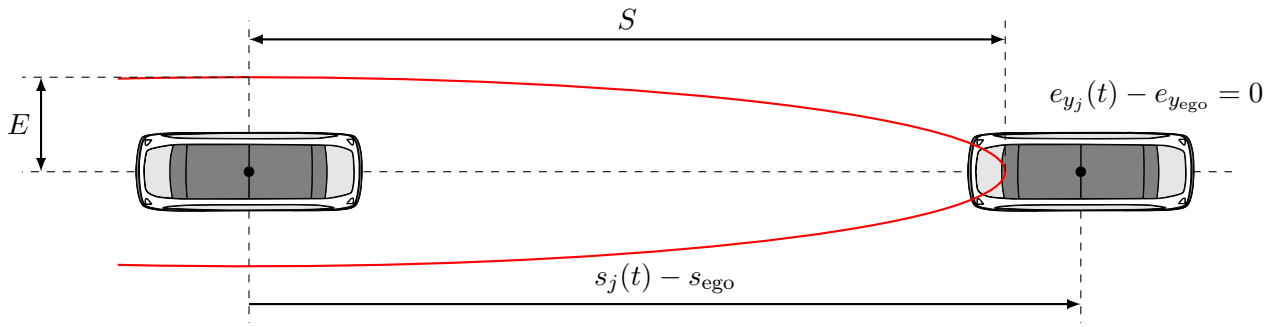


Figure 3-9: Situation sketch of an infeasible scenario where the Quadratic Program solver does not find a feasible solution for the standard ellipse shaped collision avoidance constraint of (3-18).

work is set at $\tau_{\text{ACC}} = 1.0$ [s], which is a typical value for string stable ACC [26]. Therefore the dynamic safety distance can be as:

$$L_{\text{dyn}} = v_x \cdot \tau_{\text{ACC}} \tag{3-22}$$

As long as $L_{\text{dyn}} > L_{\text{stat}}$, this value can be used as safety distance (L_{safety}). Otherwise the static constraint length is used such that pure collision avoidance is ensured.

Next to the constraint used for overtaking cases, an additional general constraint is designed for the longitudinal planner that can be enabled in case of an aborted overtake or a standard following manoeuvre. This constraint resembles the general collision avoidance constraint of (3-18), but does not include the notion of a lateral offset such that the GRT is forced to return to a following scenario. As this constraint might be enabled while an overtake manoeuvre has already been initiated, there must be an option to fully relax the constraint. The general constraint that is used for this purpose is:

$$\left(\frac{s_j(t) - s_{\text{ego}}}{S_{\text{long}} + 4}\right)^2 \geq (1 - \epsilon_2) \quad \forall j \in \{1, \ldots, n\}, \quad 0 \leq \epsilon_2 \leq 1 \tag{3-23}$$

in which the slack variable ϵ_2 is introduced to ensure the possibility of full relaxation of the constraint at high cost. Throughout the rest of this thesis, this constraint will be referred to as the distance holder. Note that the distance holder is only enabled additional to the general avoidance constraint of (3-19) and that the general constraint's semi-minor axis (E_{long}) has to adopt to the size of the lateral safety constraint's semi-minor axis in order to avoid planning an infeasible trajectory.

Lateral planning

For the lateral planning, the collision avoidance constraint has to be redesigned slightly for two main reasons:

- If the exact same collision avoidance constraint would be used as for longitudinal planning, both would become active at the exact same moment. Therefore the GRT may start braking before initiating an overtake manoeuvre, which is unwanted.

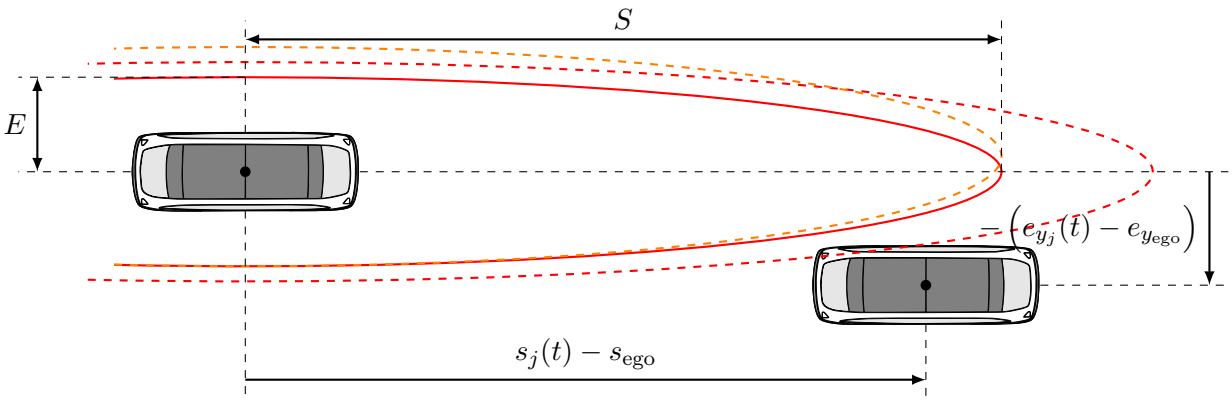


Figure 3-10: The ellipse shaped collision avoidance constraint with flexible shell as in (3-24). The thick red ellipse shows the original representation, the dashed red line shows the flexible zone created by adding $\epsilon_1 = 0$ to A. The dashed orange line shows the maximum e_y shift for $\epsilon_1 = 1$ of the flexible constraint to retain a minimum passing distance of 0.5 [m].

- Due to the quadratic term on the lateral offset in the collision avoidance constraint as depicted in (3-18), linearizing and condensing the problem in a situation where the center of the GRT and target are exactly aligned ($e_{y_j} - e_{y_{\text{ego}}} = 0$) results in zero rows in the condensed constraint matrix ($\tilde{G}$, see Appendix A-1). Such a situation is depicted in Figure 3-9. Zero rows from the constraint matrix cannot be added to the solver's active set of constraints since the solver used in this work uses a null space approach to solve the resulting condensed problem [30].

The slightly redesigned constraint is visualized in Figure 3-10 and listed below:

$$\left(\frac{s_j(t) - s_{\text{ego}}}{S_{\text{lat}}}\right)^2 + \left(\frac{\left(e_{y_j}(t) - 0.4\epsilon_1\right) - e_{y_{\text{ego}}}}{E_{\text{lat}}}\right)^2 \geq 1 \quad \forall j \in \{1, \ldots, n\} \tag{3-24}$$

$$S_{\text{lat}} = \frac{L_{\text{ego}} + L_j}{2} + L_{\text{safety}} + 4\left(1 - \epsilon_1\right), \qquad L_{\text{safety}} = f(v_x), \quad 0 \leq \epsilon_1 \leq 1$$

$$E_{\text{lat}} = \frac{W_{\text{ego}} + W_j}{2} + W_{\text{safety}}, \qquad W_{\text{safety}} = f(v_x)$$

The introduction of the new slack variable ϵ_1 has several advantages:

- The semi-major axis (S_{lat}) to both front and rear now has a flexible zone where an obstacle can be situated only at a very high cost.
- Using the flexible zone, there is first a need to re-plan laterally before longitudinal replanning is necessary.
- The flexible zone will only be used when there is no other option to move out of a collision state. Using this flexible zone also automatically moves the ellipse constraint slightly to the right which virtually moves the vehicles apart and creates a solvable Quadratic Program.

- Because of the coupling of the flexible zone and allowing $0 \leq \epsilon_1 \leq 1$ only, in a typical case where vehicles are center aligned, overtaking is only done on the left of the target vehicle.

Note that E_{lat} has to be increase by at least 0.4 [m] over $E_{\min}$ to guarantee the minimum passing distance as is used in the longitudinal planner for an overtake manoeuvre.

Additionally to the collision avoidance constraint, the lateral planner contains two additional general constraints coming from the GRT requirement. These two constraints are the upper and lower limit on the lateral acceleration (a_n) and jerk (j_n). Assuming $\dot{v}_x = \ddot{\beta} = 0$ it follows that:

$$\underline{a}_n \leq \underbrace{\dot{\psi} \cdot v_x + v_y}_{a_n} \leq \overline{a}_n \qquad \text{(lateral acceleration)} \tag{3-25a}$$

$$\underline{j}_n \leq \underbrace{\ddot{\psi} \cdot v_x}_{j_n} \leq \overline{j}_n \qquad \text{(lateral jerk)} \tag{3-25b}$$

in which v_y and $\dot{\psi}$ can be obtained directly from the system state. $\ddot{\psi}$ can be obtained from the system lateral dynamics model of (3-8) and (3-15) for the open-loop and closed-loop planner respectively.

3-7 Problem setup

Now that all models and constraints have been introduced, the three different OCPs can be formulated. Note that the following subsections will use a subscript $\star_{\text{nom}}$ denoting the reference values along the nominal trajectory, whereas the subscript $\star_{\text{ref}}$ will be used to denote the reference values along the newly computed trajectory. Internal system states are not denoted by a particular subscript.

The trajectory generating problem is solved using a two-step approach. First a lateral trajectory is planned based on the current initial velocity. Thereafter a longitudinal planner re-evaluates this trajectory in order to keep the GRT's velocity as close as possible to its reference velocity.

3-7-1 Step 1: Lateral planning

First a lateral trajectory is planned, based on the sensed environment, nominal trajectory and the current initial state. For this, an open-loop and closed-loop OCP will be introduced below.

Lateral open-loop planner

With the system state x governed by the dynamics of the open-loop lateral system (3-8), the OCP for open-loop lateral planning can be formulated as follows:

$$x = \left[v_y,\, \dot{\psi},\, e_\psi,\, e_y,\, s,\, v_x,\, \delta_f,\, \delta_r\right]^T \tag{3-26a}$$

$$y = \left[v_y,\, \dot{\psi},\, e_\psi,\, e_y,\, \delta_f,\, \delta_r\right]^T \tag{3-26b}$$

$$u = \left[\dot{\delta}_f,\, \dot{\delta}_r,\, \epsilon_1\right]^T \tag{3-26c}$$

$$\begin{aligned}\min_{y,u} \int_{t_0}^{t_0+T_p} \Big(& K_{v_y,1}\left(v_y(t) - v_{y_{\text{nom}}}(t)\right)^2 + K_{\dot{\psi},1}\left(\dot{\psi}(t) - \dot{\psi}_{\text{nom}}(t)\right)^2 + K_{e_\psi,1}\left(e_\psi(t) - e_{\psi_{\text{nom}}}(t)\right)^2 \\ & + K_{e_y,1}\left(e_y(t) - e_{y_{\text{nom}}}(t)\right)^2 + K_{\delta_f,1}\left(\delta_f(t) - \delta_{f_{\text{nom}}}(t)\right)^2 + K_{\delta_r,1}\left(\delta_r(t) - \delta_{r_{\text{nom}}}(t)\right)^2 \\ & + K_{\dot{\delta}_f}\dot{\delta}_f(t)^2 + K_{\dot{\delta}_r}\dot{\delta}_r(t)^2 + K_{\epsilon_1,1}\epsilon_1(t)^2 \Big)\, dt \\ & + K_{v_y,2}\left(v_y(T_p) - v_{y_{\text{nom}}}(T_p)\right)^2 + K_{\dot{\psi},2}\left(\dot{\psi}(T_p) - \dot{\psi}_{\text{nom}}(T_p)\right)^2 \\ & + K_{e_\psi,2}\left(e_\psi(T_p) - e_{\psi_{\text{nom}}}(T_p)\right)^2 + K_{e_y,2}\left(e_y(T_p) - e_{y_{\text{nom}}}(T_p)\right)^2 \\ & + K_{\delta_f,2}\left(\delta_f(T_p) - \delta_{f_{\text{nom}}}(T_p)\right)^2 + K_{\delta_r,2}\left(\delta_r(T_p) - \delta_{r_{\text{nom}}}(T_p)\right)^2\end{aligned} \tag{3-26d}$$

The OCP of (3-26) is always subject to the bounds for lateral planning (3-17) as introduced in Section 3-6-1 and the elliptical collision avoidance constraint with flexible shell (3-24) as introduced in Section 3-6-2. Additionally the OCP is subject to the path constraints limiting the lateral acceleration and lateral jerk (3-25). In (3-26) the weighting factors $K_\star$ scale the penalty on separately contributing terms. Penalty cross-terms have been disregarded as tuning based on states only enables more intuitive tuning compared to the additional tuning on cross-terms. An additional end-term is included in the longitudinal trajectory generation problem to encourage the planner to return to the center lane as soon as possible.

The solution of the OCP of (3-26) is an optimal policy for the steering rates $(\dot{\delta}_f,\, \dot{\delta}_r)$ steering the GRT away from possible hazard, while minimizing the actual offset to the nominal trajectory. The enabled collision avoidance constraint is possibly relaxed by the slack variable (ϵ_1). From this optimal policy, the first entry is applied to the dynamical system in a feedforward manner directly controlling the dynamical system's steering angles. Furthermore, the predicted planner states are fed to the longitudinal planner.

Lateral closed-loop planner

With the system state x governed by the dynamics of the closed-loop lateral system (3-15), the OCP for closed-loop lateral planning can be formulated as follows:

$$x = \left[v_y,\, \dot{\psi},\, e_\psi,\, e_y,\, s,\, v_x\right]^T \tag{3-27a}$$

$$y = \left[v_y,\, \dot{\psi},\, e_\psi,\, e_y\right]^T \tag{3-27b}$$

$$u = \left[v_{y_{\text{ref}}},\, \dot{\psi}_{\text{ref}},\, \delta_{f_{\text{ref}}},\, \delta_{r_{\text{ref}}},\, e_{\psi_{\text{ref}}},\, e_{y_{\text{ref}}},\, \epsilon_1\right]^T \tag{3-27c}$$

$$\begin{aligned}\min_{y,u} \int_{t_0}^{t_0+T_p} \Big(& K_{v_y,3}\left(v_y(t) - v_{y_{\text{nom}}}(t)\right)^2 + K_{\dot{\psi},3}\left(\dot{\psi}(t) - \dot{\psi}_{\text{nom}}(t)\right)^2 + K_{e_\psi,3}\left(e_\psi(t) - e_{\psi_{\text{nom}}}(t)\right)^2 \\ & + K_{e_y,3}\left(e_y(t) - e_{y_{\text{nom}}}(t)\right)^2 + K_{v_{y_{\text{ref}}}} v_{y_{\text{ref}}}(t)^2 + K_{\dot{\psi}_{\text{ref}}} \dot{\psi}_{\text{ref}}(t)^2 + K_{\delta_{f_{\text{ref}}}} \delta_{f_{\text{ref}}}(t)^2 \\ & + K_{\delta_{r_{\text{ref}}}} \delta_{r_{\text{ref}}}(t)^2 + K_{e_{\psi_{\text{ref}}}} e_{\psi_{\text{ref}}}(t)^2 + K_{e_{y_{\text{ref}}}} e_{y_{\text{ref}}}(t)^2 + K_{\epsilon_1,2}\epsilon_1(t)^2 \Big)\, dt \\ & + K_{v_y,4}\left(v_y(T_p) - v_{y_{\text{nom}}}(T_p)\right)^2 + K_{\dot{\psi},4}\left(\dot{\psi}(T_p) - \dot{\psi}_{\text{nom}}(T_p)\right)^2 \\ & + K_{e_\psi,4}\left(e_\psi(T_p) - e_{\psi_{\text{nom}}}(T_p)\right)^2 + K_{e_y,4}\left(e_y(T_p) - e_{y_{\text{nom}}}(T_p)\right)^2 \end{aligned} \tag{3-27d}$$

Like in the open-loop case, the OCP of (3-27) is always subject to the bounds for lateral planning (3-17) as introduced in Section 3-6-1 and the elliptical collision avoidance constraint with flexible shell (3-24) as introduced in Section 3-6-2. Additionally the OCP is subject to the path constraints limiting the lateral acceleration and lateral jerk (3-25). Again, the weighting factors $K_\star$ in (3-27) scale the penalty on separately contributing terms. Penalty cross-terms have been disregarded as tuning based on states only enables more intuitive tuning compared to the additional tuning on cross-terms. An additional end-term is included in the longitudinal trajectory generation problem to encourage the planner to return to the center lane as soon as possible.

The solution of the OCP of (3-27) is an optimal policy for a new reference trajectory ($v_{y_{\text{ref}}}$, $\dot{\psi}_{\text{ref}}$, $\delta_{f_{\text{ref}}}$ $\delta_{r_{\text{ref}}}$ $e_{\psi_{\text{ref}}}$ $e_{y_{\text{ref}}}$) steering the GRT away from possible hazard, while minimizing the actual offset to the nominal trajectory. The enabled collision avoidance constraint is possibly relaxed by the slack variable (ϵ_1). The new optimal trajectory is fully fed to the low-level controller that takes care of following the trajectory. Furthermore, the predicted planner states are fed to the longitudinal planner.

3-7-2 Step 2: Longitudinal planning

Now that a lateral trajectory is defined by either of the lateral planners, a longitudinal trajectory can be generated, taking into account the expected lateral states. With the system state x governed by the dynamics of the longitudinal system (3-4), the OCP for longitudinal planning can be formulated as follows:

$$x = \begin{bmatrix} v_x,\ s \end{bmatrix}^T \tag{3-28a}$$

$$y = \begin{bmatrix} v_x \end{bmatrix} \tag{3-28b}$$

$$u = \begin{bmatrix} F_x,\ \epsilon_2 \end{bmatrix}^T \tag{3-28c}$$

$$\begin{aligned}\min_{y,u} \int_{t_0}^{t_0+T_p} & \left(K_{v_x,1}\left(v_x(t) - v_{x_{\text{nom}}}(t)\right)^2 + K_{F_x} F_x(t)^2 + K_{\epsilon_2}\epsilon_2(t)^2 \right) dt \\ & + K_{v_x,2}\left(v_x(T_p) - v_{x_{\text{nom}}}(T_p)\right)^2 \end{aligned} \tag{3-28d}$$

The OCP of (3-28) is always subject to the bounds for longitudinal planning (3-16) as introduced in Section 3-6-1 and the elliptical collision avoidance constraint (3-19) as introduced

in Section 3-6-2. Recall that this collision avoidance constraint is based on the expected lateral offset ($e_{y_{\text{ego}}}(t)$) as planned by the lateral planner. Note that in a target following, or an overtake abort manoeuvre, the OCP is additionally subject to the distance holder constraint (3-23). In (3-28) the weighting factors $K_\star$ scale the penalty on separately contributing terms. Penalty cross-terms have been disregarded as tuning based on states only enables more intuitive tuning compared to the additional tuning on cross-terms. An additional end-term is included in the longitudinal trajectory generation problem to encourage the planner to reach the set reference velocity.

The solution of the OCP of (3-28) is an optimal policy for the longitudinal force (F_x) bringing the GRT to the desired velocity. The enabled distance holder constraint is possibly relaxed by the slack variable (ϵ_2). From this optimal policy, the first entry is applied to the dynamical system.

3-8 Summary

This chapter handled the modeling of the equality and inequality constraints for the OCP. Three different dynamical prediction models have been introduced. First the coordinate frames used in this work are introduced, as well as the curvilinear framework. It is proposed to model the surrounding traffic according to a constant velocity physics-based prediction model.

The longitudinal planner is proposed to predict its future state evolution based on a second-order model incorporating rolling resistance and drag. For the lateral planner two different dynamical systems have been proposed. The first system is based on the open-loop configuration of a GRT converting a steering rate input to lateral movement. The second proposed model is based on a closed-loop interconnection of the GRT dynamical system and a feedforward/feedback based proportional low-level controller.

Furthermore the longitudinal and lateral comfort and safety constraints have been introduced. A distinction is made between bounds and general constraints. In this work, most of the bounds can be set as box constraints. Two collision avoidance constraints have been introduced. The first constraint is an adapted variant of a standard ellipse shaped collision avoidance constraint. This general constraint is always active during operations for both the lateral and longitudinal planner with a slight distinction in constraint sizes. The second constraint is only enabled for the longitudinal planner in following or overtake-abort situations and does only include a notion on the longitudinal spacing between the GRT and the target vehicle. Additionally, the lateral comfort constraints have been introduced to ensure a comfortable ride.

The latter section formulated the three different Optimal Control Problems necessary for the trajectory generation of the GRT and introduced the two-step solving approach. With the modeling of the OCP presented in the latter two chapters, it is time to introduce the implementation of the OCP in a modeling environment. Additionally the solution strategy, as well as solver settings, used for solving this OCP have to be chosen. Furthermore the next chapter explains how the numerical simulations are set up and what tools are used to do so.

Chapter 4

Implementation and Simulation setup

The latter chapter introduced all models and constraints necessary to fully design the Model Predictive Control (MPC) trajectory generation Optimal Control Problem (OCP). This chapter elaborates on the implementation of the OCP and provides insight into the simulation environment.

Section 4-1 introduces the modeling environment in which the OCP is set up. This section is continued by background information on the Quadratic Program (QP) solver used in this work. Additionally this section explains the general solver setup and introduces a simple rule-base for driving mode switching. Section 4-2 describes the software packages used and their implemented interconnection for numerical simulations.

4-1 Implementation

This section handles the implementation of the defined OCPs for the longitudinal and the two lateral planners. First, the modeling environment for setting up the Nonlinear Program (NLP) introduced. The modeling environment used in this work is the `ACADO Toolkit`. This toolkit does not solve the problem itself, but prepares the NLP for a QP solver in a sequential manner. Next `qpOASES` is introduced, followed by brief explanation of the settings used within the modeling framework. After the general introduction of the solver and the settings used in this work, it is explained what measures are taken in case the QP solver does not find a feasible solution. Finally the last subsection scopes on how the switch between several modes of the NLP is implemented.

4-1-1 ACADO Toolkit

The OCP formulation is implemented in within the `ACADO Toolkit`. The toolkit is a software environment for automatic control and dynamic optimization, completely written in C++. `ACADO` is an effective, open-source and user friendly tool that supports auto generation of

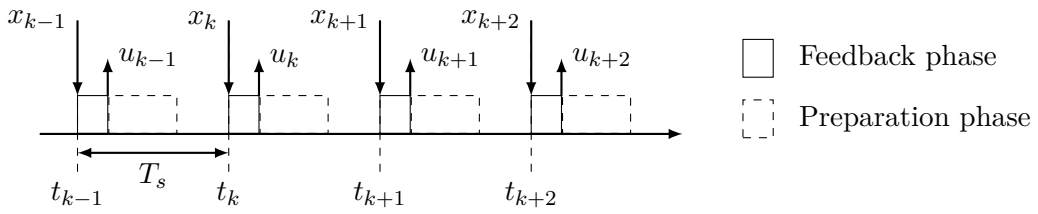


Figure 4-1: Task scheduling of the RTI-scheme. After the preparation phase the algorithm waits for the incoming new initial state x_k every T_s and executes feedback phase. The solution (u_k) found is directly applied to the system, inspired by [2].

highly efficient, self contained C-code [31]. In this work, the tool is used as a modeling environment, to state the OCP in a syntax that resembles the one in (2-1).

One of the ingenious novelties introduced in the `ACADO Toolkit` is the RTI-scheme, that splits one iteration in two parts:

- A preparation phase, where the NLP is linearized, discretized and condensed.
- A feedback phase, where the condensed QP is solved by a QP solver.

Figure 4-1 shows the typical task scheduling in the RTI-scheme. The solution found in the feedback phase is then applied to the system and the next iteration starts again with the linearization of the problem, this iterative approach is also known as Sequential Quadratic Programming (SQP).

Typically the preparation phase can be time consuming since it consists of lots of floating point operations, but since operations are independent from the initial state of the optimization $(\tilde{x}_0)$, preparations can be executed before this measurement arrives, leaving the framework to only solve the problem in the feedback phase after a new initial state is measured. This way real-time performance in the mili or microsecond range can be achieved [32].

When modeling the OCP with the `ACADO Toolkit`, one first has to set the prediction horizons and shooting interval. Furthermore one should choose for either a single shooting or multiple shooting approach [33]. Thereafter the non-linear dynamical systems, as introduced in Chapter 3, can be implemented in continuous time and bounds and constraints can be set. Finally the weights for the cost function can be set.

The `ACADO Toolkit` takes the prepared linearized continuous time OCP and discretizes the OCP over an equidistant grid from T_0 till T_p with N_p shooting intervals. Therefore, the OCP cost of (2-1) problem is then approximated by:

$$\begin{aligned}\min_{\substack{x_{k_0},\ldots,x_{k_0+N_p}\\ y_{k_0},\ldots,y_{k_0+N_p}\\ u_{k_0},\ldots,u_{k_0+N_p}}} \quad & \frac{1}{2}\sum_{k=k_0}^{k_0+N_p-1} (y_k - y_{\text{ref}})' Q (y_k - y_{\text{ref}}) + (u_k - u_{\text{ref}})' R (u_k - u_{\text{ref}}) \\ & + \frac{1}{2}\left(y_{k+N_p} - y_{\text{ref}}\right)' P \left(y_{k+N_p} - y_{\text{ref}}\right)\end{aligned} \tag{4-1}$$

in which $Q \in \mathbb{R}^{n_y}_{\succeq 0},\ R \in \mathbb{R}^{n_u}_{\succ 0},\ P \in \mathbb{R}^{n_y}_{\succeq 0}$. From this discretized form, one of two approaches can be taken:

- Either the resulting QP is solved in the current form using structure exploiting solvers such as FORCES [34]
- Or the resulting structured QP undergoes a procedure called condensing (see Appendix A-1) and the resulting condensed form is solved by a dense QP solver

In this work, the discretized QP is further condensed and the resulting QP problem is solved using `qpOASES`.

4-1-2 QP OASES

The `ACADO Toolkit` sends the linearized and discretized (and optionally condensed) OCP to a QP solver. Various interfaces are available. In this work the interface with `qpOASES3.0` is used. `qpOASES` is a dense, parametric active-set QP solver that supports warm starting of the optimization routine. The algorithm is capable of effectively solving the condensed QP of the form:

$$\min_{\tilde{u}} \quad \frac{1}{2}\tilde{u}'\tilde{H}\tilde{u} + \tilde{u}'\underbrace{\tilde{F}\tilde{x}_0}_{g} \tag{4-2a}$$

$$\text{subject to} \quad \tilde{G}\tilde{u} \geq \underbrace{\tilde{l} - \tilde{E}\tilde{x}_0}_{w} \tag{4-2b}$$

in which $\tilde{H}$ and g are the condensed Hessian and gradient of the QP respectively and $\tilde{G}$ is the condensed constraint matrix. For the condensed QP problem the KKT conditions as listed in (2-2) become:

$$\tilde{H}\tilde{u} + g - \tilde{G}'\mu = 0 \tag{4-3a}$$

$$\mu_i \left(G_i\tilde{u} - w_i\right) = 0, \qquad i = 1, \ldots, m \tag{4-3b}$$

$$\mu \geq 0 \tag{4-3c}$$

$$\tilde{G}\tilde{u} - w \geq 0 \tag{4-3d}$$

The active-set algorithm solves (4-3) by finding a solution for the active subset ($\tilde{G}_A$) of $\tilde{G}$. For this active subset, the solution to the dense inequality constrained QP of (4-2) is reduced to solving an equality constrained QP. The solution to the equality constrained QP can then be found by solving the following Linear Program (LP):

$$\begin{bmatrix} \tilde{H} & \tilde{G}'_{\mathrm{A}} \\ \tilde{G}_{\mathrm{A}} & 0 \end{bmatrix} \begin{bmatrix} \tilde{u}^{\mathrm{opt}} \\ -\mu_{\mathrm{A}}^{\mathrm{opt}} \end{bmatrix} = \begin{bmatrix} -g \\ w_{\mathrm{A}} \end{bmatrix} \tag{4-4}$$

whose matrix is invertible. Older versions of the QP solver could only deal with positive definite Hessian matrices. By applying a regularization procedure, `v3.0` can also solve (4-4) for positive semi-definite Hessian matrices [35].

In order to reach real-time requirements `qpOASES` offers two variables that can be used [36]:

- Limiting the number of Maximum Working Set Recalculations
- Early termination of the optimization procedure

For the open-loop planning model (3-8), none of the variables have to be used to reach real-time performance. The closed-loop planning model of (3-15) on the other hand produces a larger optimization problem. Therefore, using either early termination or limiting the number of working set calculations might be beneficial.

Note that for the MPC design with the open-loop model of (3-8) and prediction horizon N_p, `qpOASES` delivers feasible results at a decent speed. For longer prediction horizons or the larger systems such as the closed-loop model of (3-15), structure exploiting QP solvers e.g. `qpDUNES` might be beneficial [37]. In order to use a single approach for both systems, `qpOASES` is used for both the open-loop and closed-loop planner.

4-1-3 Solver settings

The two main important settings in the solver framework are the prediction horizon (N_p) and the sampling time (T_s). These two determine how often a new trajectory is generated. Furthermore, since the used framework is only applicable for equidistant grids, the combination of N_p and T_s also directly fixes the lookahead time.

The choice for an optimal T_s and N_p is not trivial. The optimization is required to run at real-time, and thus faster than T_s (see Figure 4-1). Increasing the prediction horizon has a positive correlation with the total computation time. However, the product of T_s and N_p determine the lookahead time. When driving at a certain v_x, this lookahead time has a minimal required value in order to brake the vehicle according to the requirements in Table 1-2. Therefore a trade-off between a short sampling time and prediction horizon has to be found.

Figure 4-2 and Table 4-1 show the braking distances for a GRT decelerating from a certain initial speed (v_{init}) to a final target speed (v_{tar}). From this Figure, it can be seen that at $v_{\text{init}} = 10$ [m/s] an upper-bound of the braking distance is 40 [m]. The actual braking distance is slightly lower than that. Taking into account the maximum longitudinal jerk and acceleration the vehicle comes to a full stop in 39.69 [m]. Since the low level controller of the GRT is running at 10 [Hz], it is logical not to have a sampling time exceeding this frequency.

Table 4-1: Braking distance (s_{brake}) [m] as a function of initial (v_{init}) and target velocity (v_{tar}) [m/s].

v_{init} \ v_{tar}	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
5	11.80	11.48	10.52	8.93	6.66	0.00	0.00	0.00	0.00	0.00	0.00
6	16.10	15.78	14.83	13.24	11.01	8.09	0.00	0.00	0.00	0.00	0.00
7	21.04	20.72	19.77	18.18	15.95	13.08	9.52	0.00	0.00	0.00	0.00
8	26.62	26.30	25.35	23.75	21.53	18.66	15.16	10.95	0.00	0.00	0.00
9	32.84	32.52	31.56	29.97	27.74	24.87	21.37	17.23	12.38	0.00	0.00
10	**39.69**	39.37	38.41	36.82	34.59	31.73	28.22	24.08	19.31	13.80	0.00

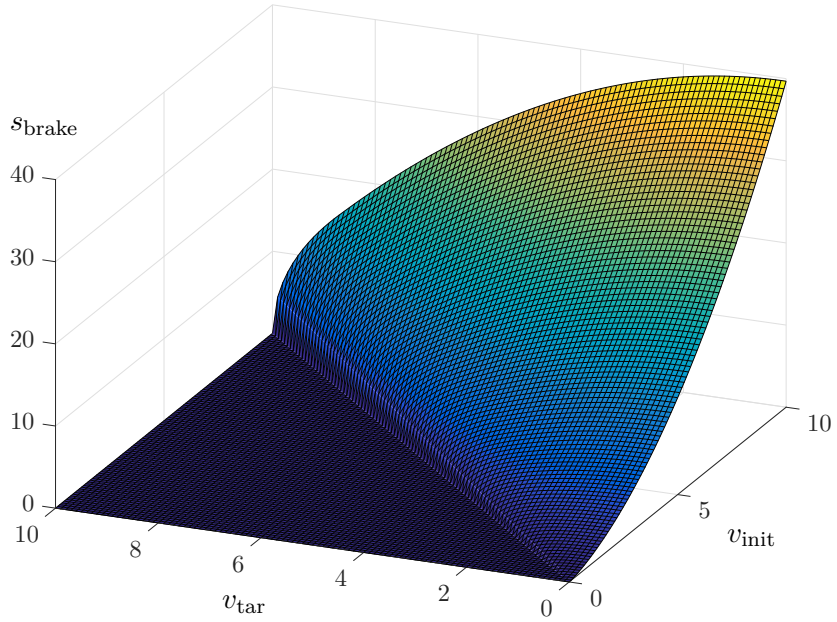


Figure 4-2: Braking distance (s_{brake}) [m] as a function of initial (v_{init}) and target velocity (v_{tar}) [m/s].

Therefore, it is chosen to set the sampling time to the same frequency as the low level control cycle. From this all N_p can then be computed as:

$$N_p = \frac{s_{\text{brake}}}{T_s \cdot v_{x_{\max}}} \tag{4-5}$$

Thus for $T_s = 0.1$ [s], $v_{x_{\max}} = 10$ [m/s] and $s_{\text{brake}} = 40$ [m], it follows that $N_p = 40$. This means that at the maximum GRT speed the OCP has an equidistant grid size of $T_s \cdot v_{x_{\max}} = 1$ [m] a prediction step.

Next to the prediction horizon and sampling time, more changes can be made to the SQP solver in order to speed up computation times. A summary of the settings used is listed in Table 4-2.

Table 4-2: `ACADO Toolkit` solver settings.

Variable	Value
Prediction Horizon (N_p)	40
Sampling Time (T_s)	0.1 [s]
Number of SQP iterations	1
Max. Number of Working Set Recalculations	90
Shooting Type	Multiple Shooting
Condensing Type	Full Condensing
Hessian Approximation	Gauss Newton
MPC integrator	Implicit Runge Kutta 2^{nd} order
Termination Tolerance	$2.221 \cdot 10^{-7}$

4-1-4 Handling of Infeasibility

Although the planner has been designed to return feasible solutions for every scenario, it can occur that the NLP or QP becomes infeasible. Generally there are two types of infeasibility:

- The non-linear on-line optimization itself can be infeasible for example due to new initial state that violates the constraints set in the OCP.
- The underlying QP problem becomes infeasible, for example due to linearization of constraints.

Currently the solution of the QP solver is only used when the returned success flag is either 0 or 1 meaning the solution was either successful or the maximum number of working set recalculations was reached. Also in this second case, using the solution of the QP solver is useful since the solution is both feasible and closer to the optimum than it was at the last iteration. The hotstart of the next iteration will then try to bring the solution to an optimum again.

In case any other return flag is given than 0 or 1, the solution of the QP is discarded and the solution of last successful iteration is used. This is not a pure guarantee that the system will recover from this state, but it is assumed that this approach is a better alternative compared to choosing an infeasible solution.

4-1-5 Decision making

Decision making is an essential part of a trajectory generator. The decision whether or not to overtake a vehicle is generally very complex. Moreover, lots of decision variables play a role in making this decision. Even while overtaking, a normal driver is continuously checking its surroundings and re-evaluating the decision made. This decision making is generally modeled by using multi layered models such as MDPs [7]. However, developing such an intelligent decision making process is beyond the scope of this thesis, therefore, a rule-base is introduced to switch between different modes. For both the longitudinal planner and lateral planner, three modes are introduced:

- *Standby* - Obstacle following (ACC)
- *Overtake* - Initiate and execute overtake
- *Abort* - Abort overtake and follow

The system is initialized in the *Standy* mode and can switch to the *Overtake* mode if:

- The target vehicle is not moving in lateral direction.
- There is enough spacing between target and road boundaries.
- Overtake can be executed over a distance shorter than a certain amount of meter.

All rules in this rule-base have to be true in order to enable an overtake manoeuvre. The last rule in this base is the most important one since it ensures that both the relative velocity

between target and GRT and the GRT's velocity are both large. If during an overtake one of the rules is not true any more, the system switches to the *Abort* state. The switch to this state is also executed if either the longitudinal or lateral planner returns an infeasibility flag for more than 5 consecutive times. From the *Abort* mode, there is no way to return to the *Overtake* mode again other than executing a reset.

The longitudinal and lateral collision avoidance constraint of (3-19) and (3-24) respectively are always enabled for all modes. The distance holder of (3-23) is only enabled in the *Standby* and *Abort* mode and thus disabled while overtaking.

4-2 Simulation Setup

This section discusses simulation setup of the trajectory generator in order to give the reader insight on how the trajectory generator is tested. The general testing layout is visualised in Figure 4-3. The entire scheme runs on one single notebook pc. The contents of all blocks, except the PreScan Core, PreScan Visualisation and Preprocessing, have been discussed in earlier chapters. This chapter introduces the interaction and coupling of separate blocks.

In Section 4-2-1 a short introduction is given to the PreScan software simulation package [38]. Section 4-2-2 explains how the trajectory generator is wrapped into a C MEX s-Function and is connected to the PreScan environment within SIMULINK.

4-2-1 PreScan

PreScan is a physics-based development and simulation platform developed for Co-Simulation of Advanced Driver Assistance Systems (ADAS). The package aims to aid the user in designing scenarios such that the user can focus on designing Automated Driving System (ADS) software. Using PreScan, one is enabled to easily set a priori known trajectories for target vehicles. PreScan supports standard and customizable sensor classes for environment sensing. Furthermore, scenarios can be made more realistic by including buildings and other obstacles in the test environment that can for example block a sensor's line of sight. Due to the main interface with MATLAB/SIMULINK rapid prototyping of ADAS supported.

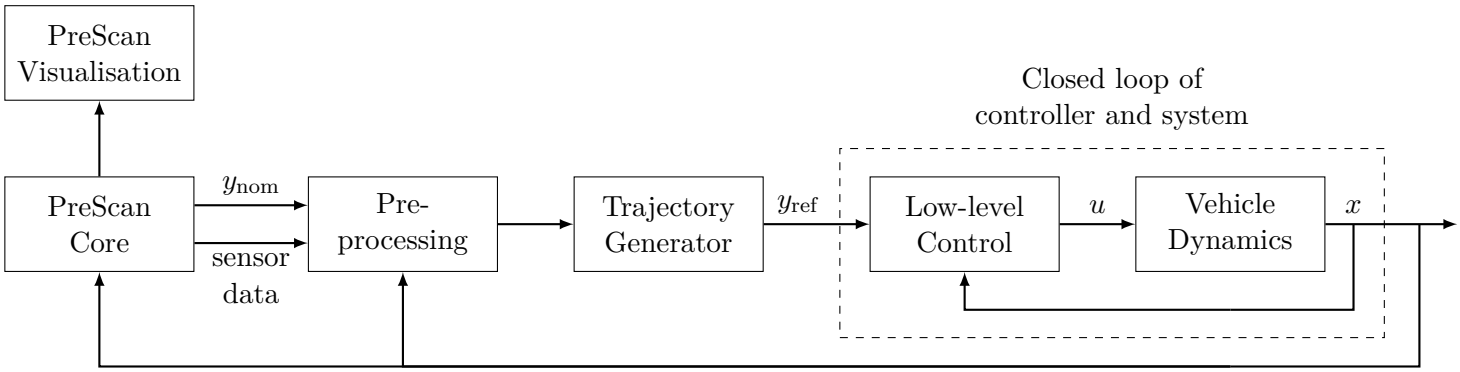


Figure 4-3: Simplified schematic setup of the SIMULINK system interaction.

Table 4-3: PreScan Idealized Actor Information Receiver (AIR) sensor settings.

Variable	Value
Detection Range	300 [m]
Detection Angle	360 [°]
Detection Type	Bounding Box Center

The typical phases carried out while setting up PreScan Software in the Loop (SIL) tests are:

1. Build a scenario within the PreScan GUI
2. Set up sensor models in the PreScan GUI
3. Include user specified control system in a SIMULINK Compilation Sheet
4. Run the experiment and process results

The scenarios set up and results from the tested trajectory generator will be discussed in Chapter 5. The sensor model will be discussed directly hereafter.

GRT sensor model

The GRT consists of several sensors to create a mapping of the environment. This way a 360° surround view is created. In this thesis work the sensor fusion algorithms to create this 360° view are not included in the SIL environment. In order to achieve the 360° surround view the Idealized Actor Information Receiver (AIR) sensor is placed in the GRT bounding box center and set up as in Table 4-3.

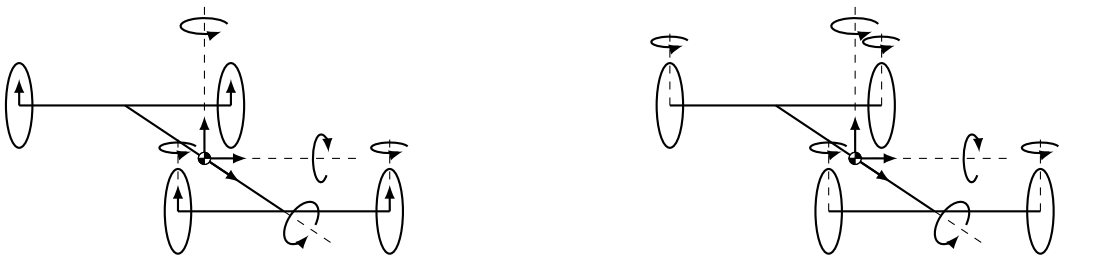
The Sensor then returns the following characteristics of the objects within range:

- Object Range [m]
- Object Azimuth (θ) [°]
- Object Elevation (ϕ) [°]
- Numerical ID [-]
- Velocity [m/s]
- Heading [°]

Sensor signals are then mapped in such a way that the obstacle detection is equivalent to the obstacle positioning as used in the collision avoidance constraints as introduced in Section 3-6.

PreScan high fidelity model

PreScan supplies the user with the high fidelity vehicle model as depicted in Figure 4-4a. For this work the standard PreScan vehicle model has been slightly adjusted in order to also take into account the rear wheel steering capabilities of the GRT. The degrees of freedom of the adapted vehicle model is depicted in Figure 4-4b. Note that vertical tire movement has been removed from the vehicle model. The adapted vehicle model includes of:



(a) PreScan Standard Model.

(b) Adapted PreScan model with linear tire dynamics.

Figure 4-4: Standard and adapted PreScan high-fidelity model.

- Longitudinal load transfer
- Slip angle and camber angle tire relaxation
- Velocity dependent rolling resistance
- Aerodynamic drag

Note that the PreScan high fidelity model thus only simulates the GRT's vehicle dynamics, actuator dynamics are therefore not taken into account in the simulation environment. Nevertheless, the vehicle model describes the GRT's dynamics more accurately than the model used in the trajectory generator. Therefore a model mismatch between the trajectory generator and testing vehicle is present.

4-2-2 Simulink

Simulations are run in the Simulink run mode of PreScan. This way Simulink runs as a slave of the PreScan simulation Core. PreScan dictates the execution times of Simulink and feeds the obstacle and environment information to the underlying blocks. Furthermore, PreScan feeds the offset to the road center lane combined with the predefined preferred velocity profile as the nominal trajectory (y_{nom}) to the trajectory generator. Based on the sensor inputs and the predefined nominal trajectory, the trajectory generator computes the new optimal trajectory which is passed on to the closed loop of the low-level controller and vehicle dynamics system. The measured state is fed back into the PreScan core for visualisation. All necessary mappings between the blocks shown in Figure 4-3 are executed within the Simulink environment as well.

Note that the PreScan Core in Simulink needs to run at a higher sampling time than the trajectory planner update rate in order to effectively simulate the PreScan high-fidelity model. Different sampling times in simulation are listed in Table 4-4.

s-Function compilation of all planner instances

In order to speed up simulation time and keeping the simulation environment tidy, the exported C-code by the `ACADO Toolkit` for the trajectory generator is compiled in a C MEX s-Function using the Microsoft Windows SDK 7.1 (MSSDK71) mex compiler. As long as no

Table 4-4: SIMULINK simulation environment sample times.

Environment	Sample time [Hz]
SIMULINK core	200
Trajectory Generator	10
Visual World	20

explicit delay is implemented, the relation between the input and output of the s-Function is instantaneous. This way the trajectory generator block can easily be tested both in open-loop and in closed-loop situations. References, Tuning Matrices and Variable algorithm settings can be send in row-major format to the trajectory generator. Constant algorithm settings such as the initial state and sampling times can be set as block parameters.

Note that the s-Function is assumed to send new trajectories every T_s, but even if the solver needs more time to compute a new trajectory, this is no problem for the SIL testing. The SIMULINK instance will just wait until the new trajectory is provided and execute other processes after that, even if computation time exceeds T_s.

Setting up the simulation environment like this results in a so-called 'black-box' trajectory generator. However, executions and reasoning of the 'black-box' can still be checked by breaking the execution of the s-Function in VisualStudio, given it is compiled in debug modus.

Simulink state flow for state machine

The simple decision making rule-base as introduced in Section 4-1-5 is implemented in SIMULINK stateflow. Stateflow is especially developed for modeling combination and sequences of decision logic. This preprocessing block is implemented as preprocessor between the PreScan core and the trajectory generator (see Figure 4-3). Stateflow visualizes the current mode. If-else checks are continuously executed to evaluate the necessity of mode switching. The use of stateflow simplifies the debugging and tracking of mode switching of the trajectory generator.

4-3 Summary

This chapter introduced the modeling environment and QP solver to solve the OCP defined in the latter chapters. Additionally the solver setup has been handled and measures taken when no feasible solution is found have been introduced. The implementation section is concluded by a simple decision making rule-base to enable the earlier introduced collision avoidance constraints for the appropriate situations.

PreScan and SIMULINK and their testing interconnection have been described. A description of the modeling of the GRT surround view is given. Furthermore a description of the standard and adapted PreScan high fidelity vehicle models on which the planner is tested is given as well. It is mentioned how the longitudinal and lateral planners are wrapped into a C MEX s-Function in order to test their planning capabilities in numerical simulations. Finally it is mentioned how the rule-base for mode switching is implemented in SIMULINK state-flow.

With the OCP for both the longitudinal and lateral planner wrapped in C MEX s-Functions and implemented in Simulink, numerical simulations can be executed. The next chapter shows the results and discussion of simulating the designed trajectory planner in several scenarios.

Chapter 5

Results and Discussion

The previous chapters described the necessities to the algorithm to provide a solution of the Model Predictive Control (MPC) trajectory planning problem. Furthermore the testing environment has been introduced. This chapter presents the results from simulations executed using this closed-loop interconnection of MPC based planner and the high fidelity model. Figures showing the results for every simulated scenario are shown in the corresponding sections. These figures do not only show the GRT's state evolution but also performance measures of the planning algorithm stated in terms of computation time. Every scenario contains both the results obtained when planning using the open-loop extended model of (3-8) and the closed-loop model of (3-15). The parameters used in these models are the nominal parameters of the GRT. These parameters are summarized in Table 5-1.

The weights of the Optimal Control Problem (OCP) are kept constant for all the results of the numerical simulations presented in this chapter. In Table 5-2 these weights are depicted for both the open-loop and closed-loop lateral and longitudinal planner.

Table 5-1: Parameters used for the GRT dynamical system and safety margin size.

Parameter		**Maximum absolute value**
Vehicle mass	m	5000 [kg]
Moment of Inertia	I_z	16500 [kg$\cdot$ m^2]
GRT and target Length	L_{ego},L_{obj}	6 [m]
GRT and target Width	W_{ego},W_{obj}	2 [m]
Minimum lateral constraint size	$B_{\min}$	2.5 [m]
Minimum longitudinal constraint size	$A_{\min}$	10 [m]
CoG to front axle	l_f	1.85 [m]
CoG to rear axle	l_r	1.85 [m]
Front cornering stiffness	C_f	$2 \cdot 160000$ [N/rad]
Rear cornering stiffness	C_r	$2 \cdot 160000$ [N/rad]
Rotational error time constant	τ_1	12 [s]
Lateral error time constant	τ_2	15 [s]

Table 5-2: Standard tuning configuration for the planner for both the open-loop and closed-loop system.

Parameter	Symbol	Open-loop Value		Closed-loop Value	
		Initial	Terminal	Initial	Terminal
Lateral planner weights					
Steering angle front	K_{δ_f}	0.0001	0.1	n/a	n/a
Steering angle rear	K_{δ_r}	0.001	0.1	n/a	n/a
Heading angle	K_{e_ψ}	100	100	0.05	10
Lateral offset from trajectory	K_{e_y}	0.1	10	0.001	0.05
Lateral velocity	K_{v_y}	10	10	0.01	0.01
Yaw rate	$K_{\dot{\psi}}$	1	1	1	100
Steering rate front	$K_{\dot{\delta}_f}$	5	n/a	n/a	n/a
Steering rate rear	$K_{\dot{\delta}_r}$	1	n/a	n/a	n/a
Relaxation variable 1	K_{ϵ_1}	10	n/a	0.1	n/a
Reference Steering angle front	$K_{\delta_{f_{\text{ref}}}}$	n/a	n/a	$5 \cdot 10^{-20}$	n/a
Reference Steering angle rear	$K_{\delta_{r_{\text{ref}}}}$	n/a	n/a	$5 \cdot 10^{-20}$	n/a
Reference Heading angle	$K_{e_{\psi_{\text{ref}}}}$	n/a	n/a	$1 \cdot 10^{-10}$	n/a
Reference Lateral offset from trajectory	$K_{e_{y_{\text{ref}}}}$	n/a	n/a	$1 \cdot 10^{-20}$	n/a
Reference Lateral velocity	$K_{v_{y_{\text{ref}}}}$	n/a	n/a	$1 \cdot 10^{-15}$	n/a
Reference Yaw rate	$K_{\dot{\psi}_{\text{ref}}}$	n/a	n/a	$1 \cdot 10^{-11}$	n/a
Longitudinal planner weights					
Longitudinal velocity	K_{v_x}	50	1000	50	1000
Applied force	K_{F_x}	$1 \cdot 10^{-5}$	n/a	$1 \cdot 10^{-5}$	n/a
Relaxation variable 2	K_{ϵ_2}	$1 \cdot 10^{6}$	n/a	$1 \cdot 10^{6}$	n/a

Recall the most important design parameters for the GRT's trajectories as introduced in Chapter 1. The box constraints are deduced from this original table. The constraints active for all these scenarios are depicted in Table 5-3. Note that the scenario lane width is 3.10 [m] but since the planning is executed based on the center of gravity of the GRT, half of W_{ego} has to be subtracted before entering the constraint value. For every scenario it is verified whether the scenario meets these constraints listed in Table 5-3. In addition, it is checked whether the results of the numerical simulations also meet the trajectory generation requirements as formulated in Section 1-4. Therefore it is checked whether:

- The trajectory generator can be run in real-time
- The generated trajectory is both feasible regarding the vehicle's dynamics
- The generated trajectory gives passengers a comfortable ride

In order to test these constraints and other requirements, four different scenarios have been designed. These scenarios represent typical situations a GRT can encounter when driving on access roads in inner-city traffic. The access road is depicted in Figure 5-1. It contains a single driving lane with a width of 3.7 [m]. On both sides advisory bicycle lanes of 1.25 [m] are added, which mark the actual lane boundaries as depicted in Table 5-3. Note that no oncoming traffic is modeled in these scenarios.

Table 5-3: Box constraints set on the OCP.

Constraint variable		Maximum absolute value
Lane boundary	e_y	2.10 [m]
Longitudinal Velocity	v_x	10 [m/s]
Longitudinal Acceleration	a_x	0.16 [g]
Longitudinal Jerk	j_x	0.10 [g/s]
Lateral Acceleration	a_y	0.10 [g]
Lateral Jerk	j_y	0.06 [g/s]
Maximum steering angle	δ	0.09 [rad]
Maximum steering rate	$\dot{\delta}$	0.26 [rad/s]

First of all Section 5-1 shows the working principle for both the open-loop and closed-loop longitudinal planner, which is implemented in a simple obstacle following scenario. Section 5-2 shows an overtake manoeuvre of an obstacle moving at an arbitrary velocity. In this scenario both the lateral and longitudinal planner have to update trajectories simultaneously. Section 5-3 shows the results a complex re-routing with minimal velocity reduction. Deviations from the nominal trajectory reach approximately up to 3 [m]. For both the open-loop and the closed-loop planner a safe and comfortable trajectory is generated. Finally Section 5-4 shows the scenario where the overtake manoeuvre is aborted due to a sensed lateral velocity of the target vehicle. A safe return to an obstacle following scenario is obtained.

Aside from the time-parametrized state and input results, also snapshots of several important moments during the execution of the scenario are included in the results. This way the reader obtains a visual insight in the progress of the scenario over time. In this graphical visualisation the elliptical constraint is visualized in orange. The GRT is depicted in blue for the open-loop planner and in dark orange for the closed-loop planner. The target vehicle is depicted in red. For both the GRT and target vehicle, the planned trajectories are shown in blue and red for the open-loop and closed-loop planning case respectively. The trajectory driven is depicted in green, with markers placed every 1 [s]. The current velocities of the GRT and target vehicle are displayed next to the respective object. The nominal trajectory is depicted in dashed grey. Finally the road boundaries are depicted in solid grey. The inner lines represent the typical driving lane on an access road. These can be crossed by the GRT and are only added to give the reader a better insight in the situation. They do not represent the box constraints. The outer lines represent the full road width including the two advisory bike lanes on both sides. Note that in the visualisation, the target vehicle can enter the constraint partially, as the constraint is formulated with respect to the target vehicle's geometrical center (see Figure 3-8).

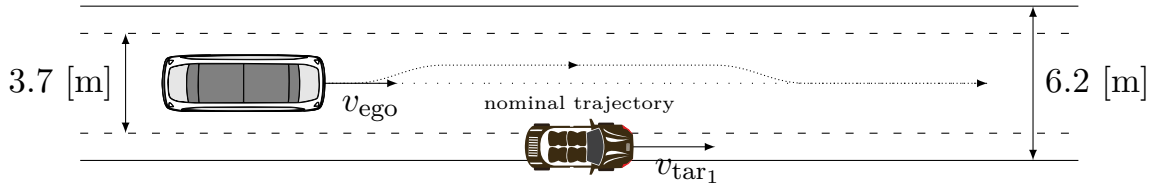


Figure 5-1: Schematic representation of an access road.

All numerical simulations are run on an off-the-shelf 2011 ASUS notebook PC running Windows 10 with an Intel Core i7-2630QM @ 2GHz and 8GB-DDR3. Since the notebook PC used is running multiple tasks in parallel there exists a possibility that strange behaviour in the computation time of the solver is a result from running background tasks instead of being caused by the solver itself. Still, the computation time is a good indicator to predict whether this solver would run faster than the set sampling time on the GRT as well.

5-1 Scenario 1: No overtake, only longitudinal planning

In this first scenario, the longitudinal planner response is tested. Results shown are only plotted for the open-loop planner since results for the open-loop and closed-loop planner are nearly identical. The GRT is initialized at 9 [m/s] with as reference velocity 10 [m/s]. The target vehicle drives 5 [m/s] for the first 5 [s] of the simulation. After the first 5 [s] the target vehicle slows down with constant acceleration to 6 [m/s] over 7 [s]. Afterwards the target vehicle accelerates again to 7.5 [m/s] over 6 [s]. This final velocity is held constant to the end of the simulation.

The velocity difference between target and GRT initially is too small to enable an overtake request. Therefore the distance holder constraint is activated and the GRT is in following mode. The GRT is not allowed to overtake during the entire simulation, therefore little to none lateral movement is expected.

General Analysis

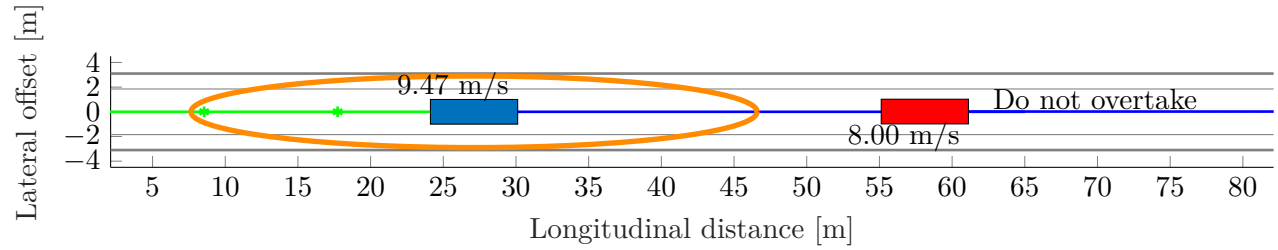
The development of the scenario is depicted in Figure 5-2. From Figure 5-3f and 5-3h it can be seen that the GRT is still accelerating at $T = 3$ [s] in order to drive at the desired speed of 10 [m/s]. A snapshot of this time instance is depicted in Figure 5-2a. Then between $T = 6$ [s] and $T = 15$ [s], the GRT decelerates to avoid violating the distance holder constraint. Thereafter it accelerates again until the target vehicle reaches its final velocity in Figure 5-2f. The distance to the target vehicle and the constraint values are depicted in Figure 5-3c, from which it is clear that the inter-vehicle distances are exceeding the distance holder size.

Comfort Analysis

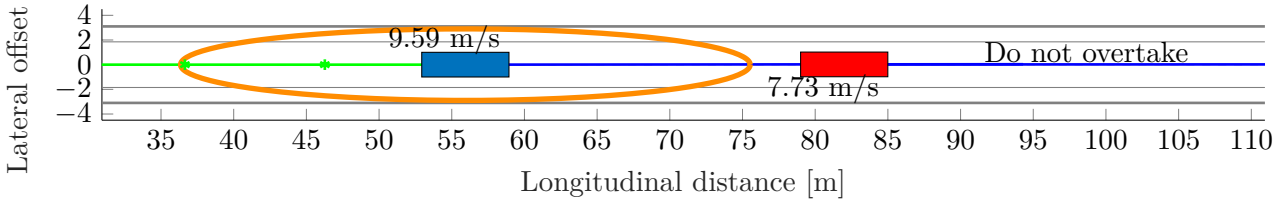
As expected, only a small lateral deviation from the nominal trajectory is present. This offset is best visible in the snapshots in Figure 5-2d and the state evolution in Figure 5-3a. Despite the fact that this lateral offset is present, values are so small that the offset is likely to be unnoticeable for a passenger. The same holds for the lateral acceleration and jerk as depicted in Figure 5-3g and Figure 5-3i respectively.

Real-time performance Analysis

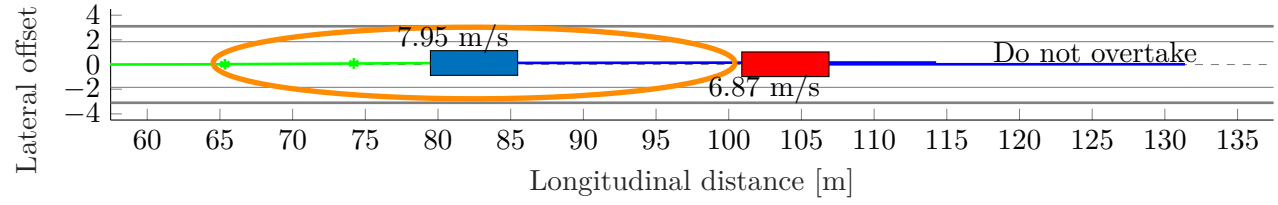
Computation times are well under the desired sampling time for both the lateral and the longitudinal planner. Moreover Figure 5-3j shows KKT values near 0 for the lateral planner which means the solution is near optimal. The increase in the KKT value just after $T = 5$ [s] for the longitudinal planner is likely to be caused by the distance holder constraint that is added to the QP solver's active set from this moment on.



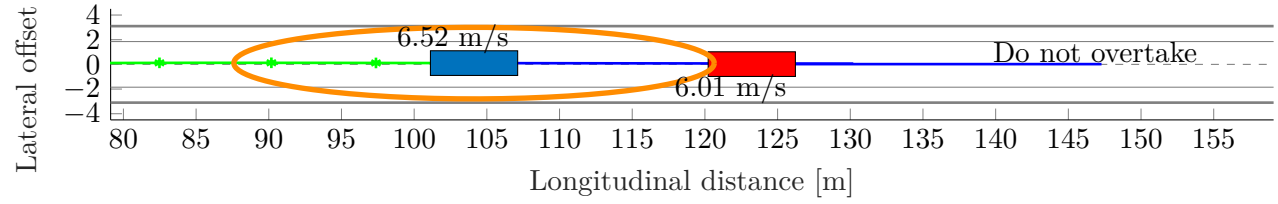
(a) Snapshot at $T = 3$ [s]



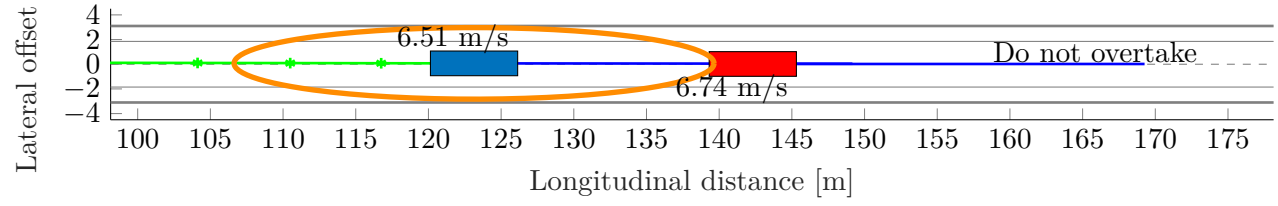
(b) Snapshot at $T = 6$ [s]



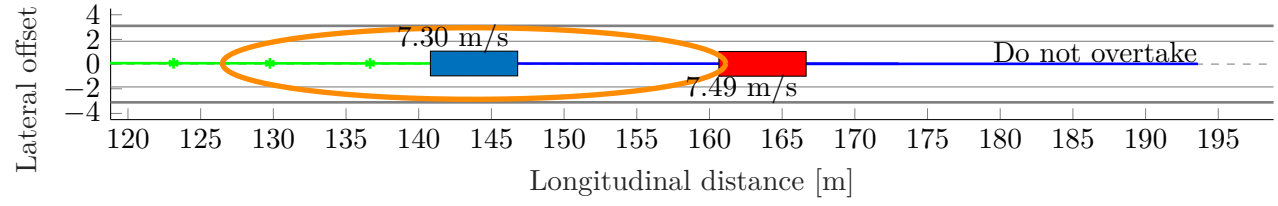
(c) Snapshot at $T = 9$ [s]



(d) Snapshot at $T = 12$ [s]

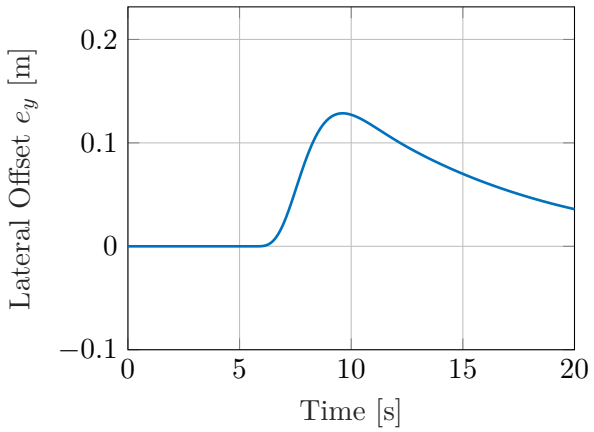


(e) Snapshot at $T = 15$ [s]

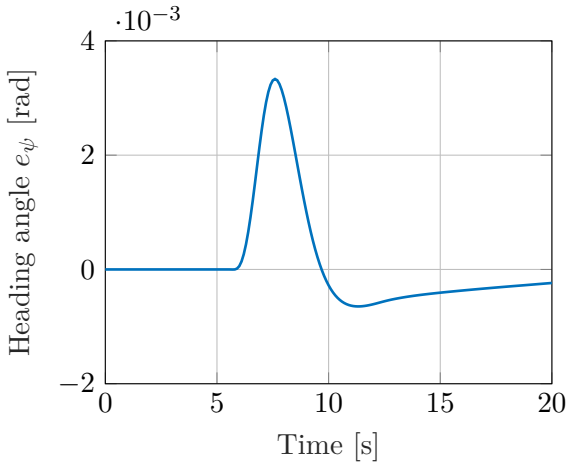


(f) Snapshot at $T = 18$ [s]

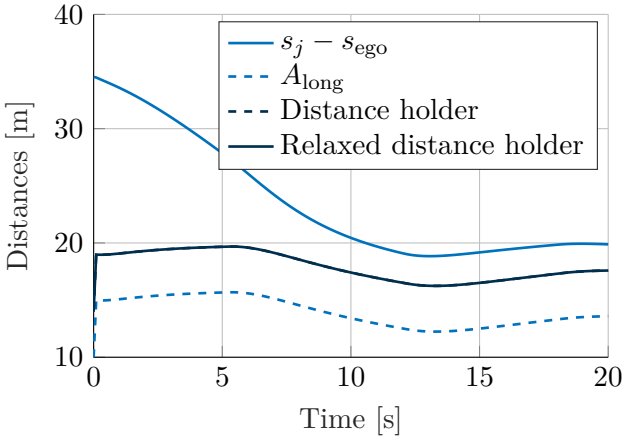
Figure 5-2: Snapshots of the target vehicle following scenario, showing a successful following action of the GRT's open-loop planner.



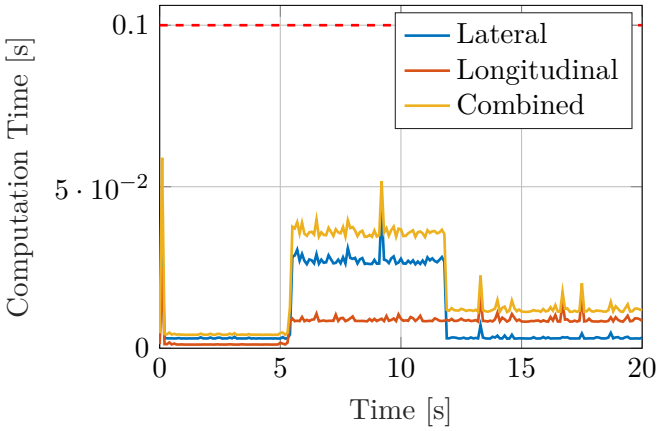
(a) Trajectory driven by the GRT



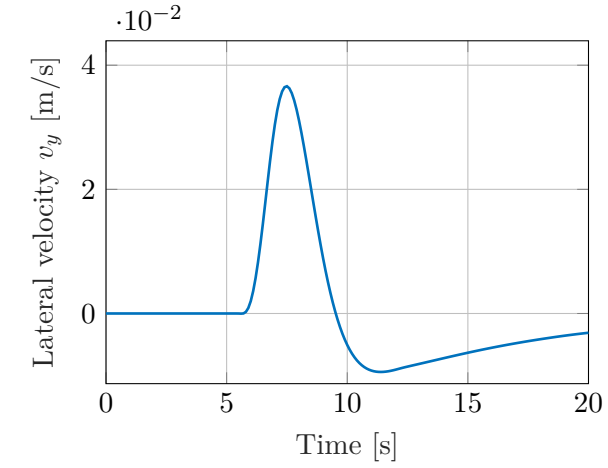
(b) Heading driven by the GRT



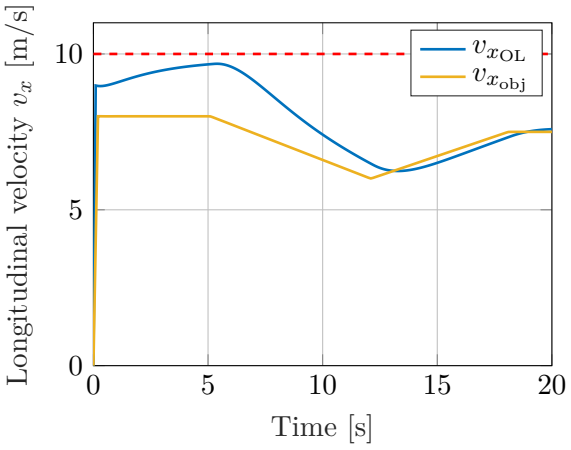
(c) Open-loop distances and constraint values



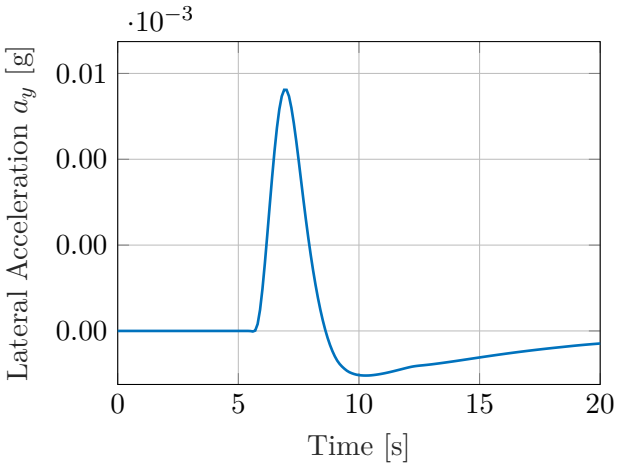
(d) Solution times for the two planning steps for the open-loop planner



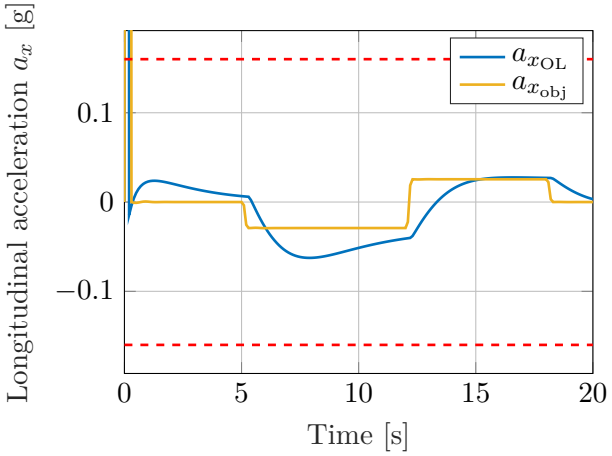
(e) Lateral velocity of the GRT



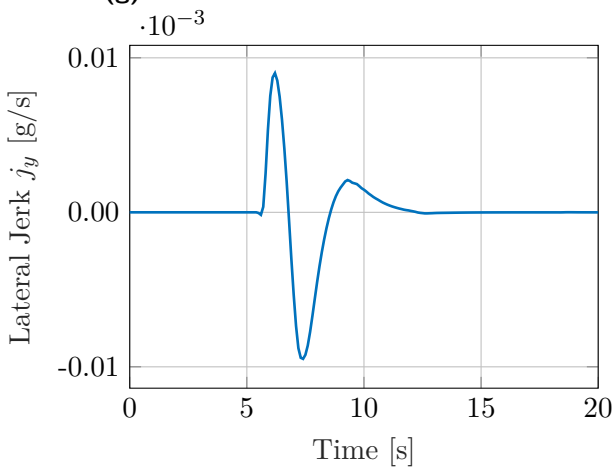
(f) Longitudinal velocity of the GRT



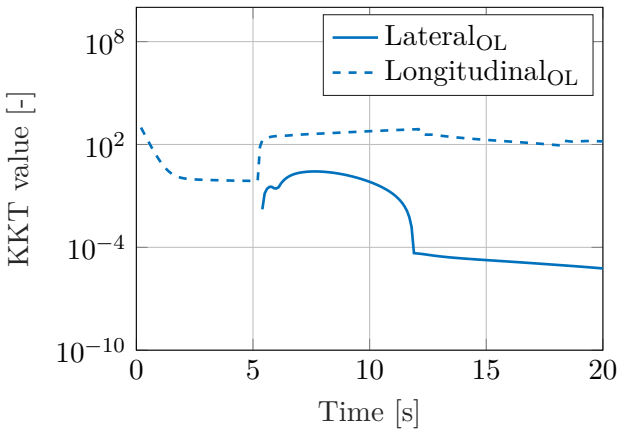
(g) Lateral acceleration of the GRT



(h) Longitudinal acceleration of the GRT



(i) Lateral jerk of the GRT



(j) Development of the KKT value

Figure 5-3: Closed-loop simulation results of the target vehicle following scenario.

5-2 Scenario 2: Overtake moving obstacle on the side of the road

The second scenario is depending on both the lateral and longitudinal planner. The target vehicle is driving 5 [m/s] all simulation long. The initial lateral offset between the target vehicle and the ego vehicle equals -2.18 [m]. The initial longitudinal gap equals 34.53 [m]. At $T = 0$ [s] the GRT has an initial velocity of 6 [m/s]. The GRT's reference velocity equals 10 [m/s].

The target vehicle is driving on the advisory bike lane but too slow to follow. Moreover, the remaining driving corridor is wide enough to fit the GRT. Therefore a new collision-free trajectory is generated and driven. Overtaking is allowed at all times during this simulation. The distance holder constraint is disabled all simulation long. Only the ellipse shaped safety zones for lateral and longitudinal planner are enabled.

The development of the scenario is depicted in Figure 5-4. In Figure 5-4a both systems reached a velocity of approximately 8 [m/s]. A new trajectory has not been planned since the elliptical constraint did not encounter the target vehicle while following the nominal trajectory yet. Around $T = 5$ [s] both systems start deviating from their nominal trajectories. The planned and executed overtake steps are depicted in Figure 5-4b until Figure 5-4f. Both planners show a similar behaviour and return to their nominal trajectories in Figure 5-4f.

General Analysis

The time-parametrized states are depicted in Figure 5-5. From Figure 5-5h and Figure 5-5j it can be seen that the GRT for both both planners keeps persistently accelerating. This is similar to human planning, as there is a clear line of sight along the target vehicle. The lateral planner results in different behaviour for the open-loop and closed-loop planner as can be seen from Figure 5-5a, 5-5b and Figure 5-5g. The open-loop planner results in more crabbing behaviour utilizing the four wheel steering capabilities whereas the closed-loop planner uses a different approach, resulting in a larger rotational offset to the nominal trajectory. From Figure 5-5c and 5-5d it can be seen that the open-loop planner needs larger steering angles to complete the same manoeuvre compared to the closed-loop planner.

Note that although systems consist of the same internal states, both planners are completely different dynamical systems and thus different behaviour is expected. In case different results are desired, the penalties listed in Table 5-2 can be re-tuned.

Comfort Analysis

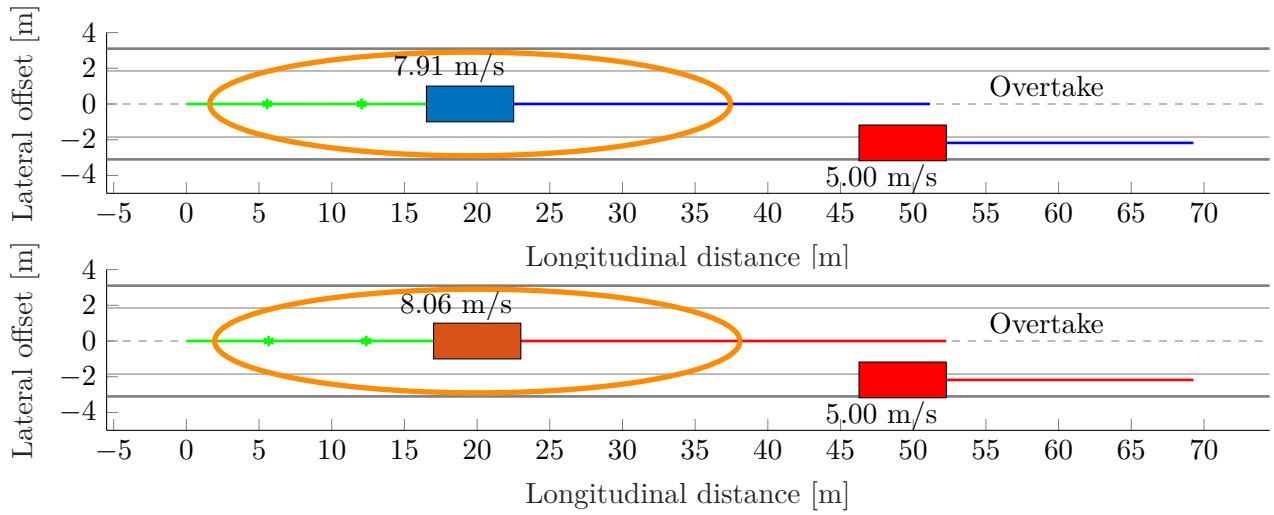
A striking result is depicted in Figure 5-5k, where the lateral jerk for both the open-loop and closed-loop planner is depicted. Results for the open-loop planner are quite smooth and stay perfectly between the box constraints set. The results for the closed-loop planner on the other hand are non-smooth and jerky, especially after $T \approx 9$ [s]. The scenario at this time instance is depicted in Figure 5-4c. As can be seen from the scenario snapshot, this time instance matches the moment the GRT is close to passing the target vehicle. From Figure 5-5b it can be seen that the GRT starts rotating back in order to return to the lane center. It is likely that this jerky lateral acceleration is caused by the discontinuous, first order steering angle

approximation as computed based on the feedforward/feedback steering rules introduced in Chapter 3 combined with the tire relaxation and high sampling rates of the high fidelity PreScan model. Improvements for future work regarding this problem will be discussed in Chapter 6. For the rest of the scenarios, the lateral jerk of the closed-loop planner will not be included in the planner and therefore also not shown in the results.

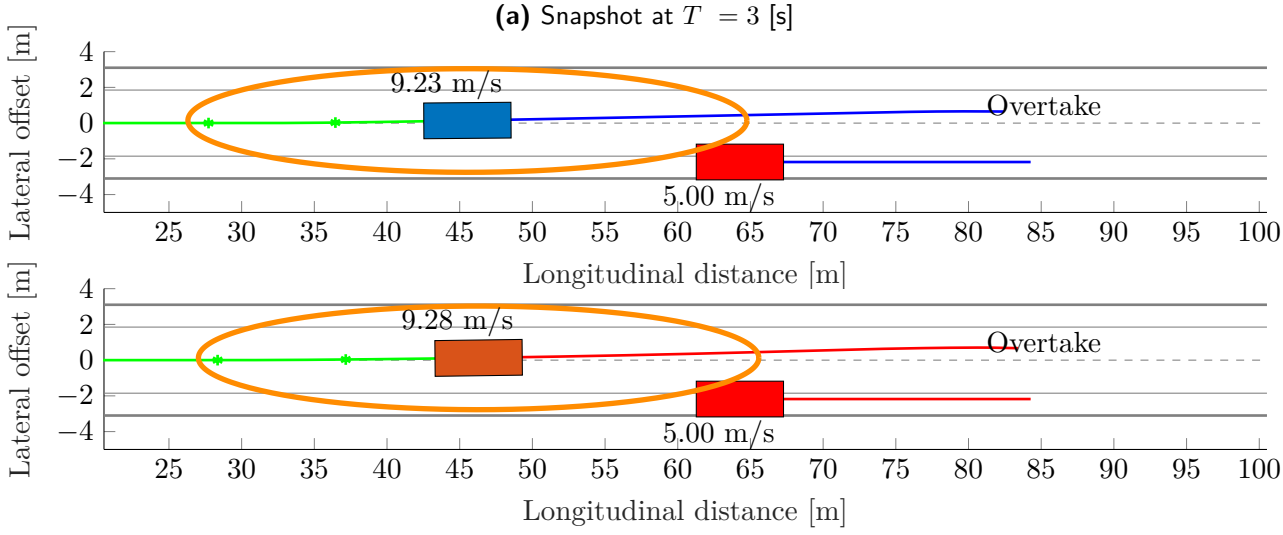
Real-time performance Analysis

From Figure 5-5e it can be concluded that the open-loop trajectory generator is real-time feasible. With no collision avoidance constraint active, the lateral open-loop planner runs at approximately 4 [ms]. With active constraints, the lateral open-loop planner runs at approximately 30 [ms] which is below the upper-bound set on the computation time. From Figure 5-5f it can be concluded that the closed-loop planner on the other hand is not real-time feasible as computation times measure approximately 450 [ms] on average, which exceeds the upper bound by more than four times. This difference in computation times between the open-loop and closed-loop planner is likely to be caused by the difference in control input size. Where the open-loop only measures 3 control inputs, the closed-loop measures 7 resulting in an optimization problem 2.5 times as big as the open-loop problem. It is likely that the computation times can be reduced for this closed-loop planner by using a structure exploiting solver such as `qpDUNES`.

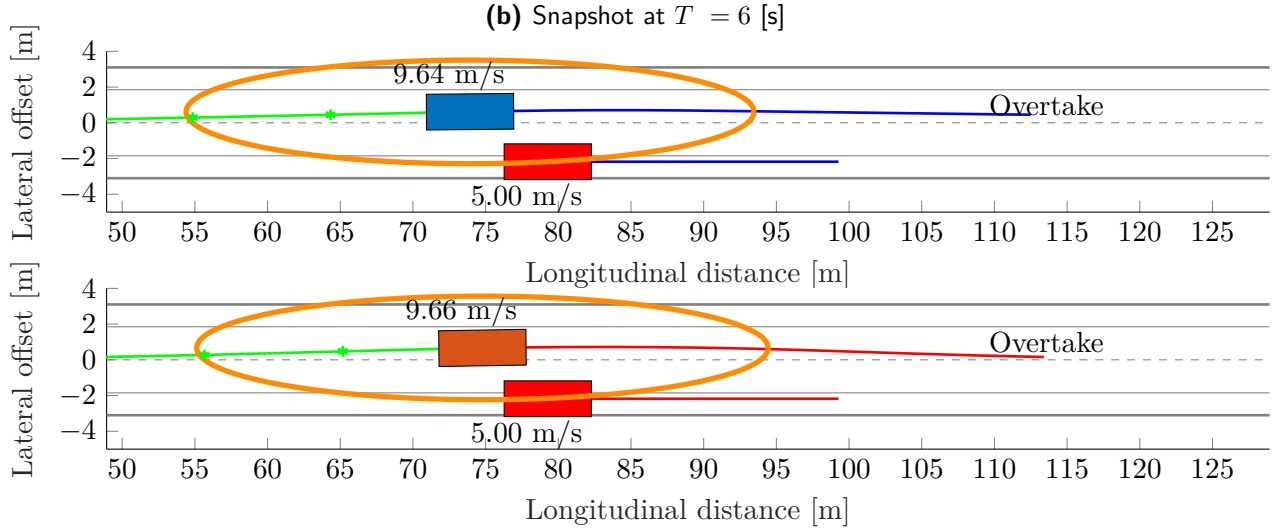
Except for the discussion points handled above, the results from both planners look promising. Both generated trajectories are both followable and collision-free. Therefore the following two scenarios have been designed to challenge the planner in re-routing in scenarios where other implementations with standard ellipse shaped or linear collision avoidance constraints might have failed.



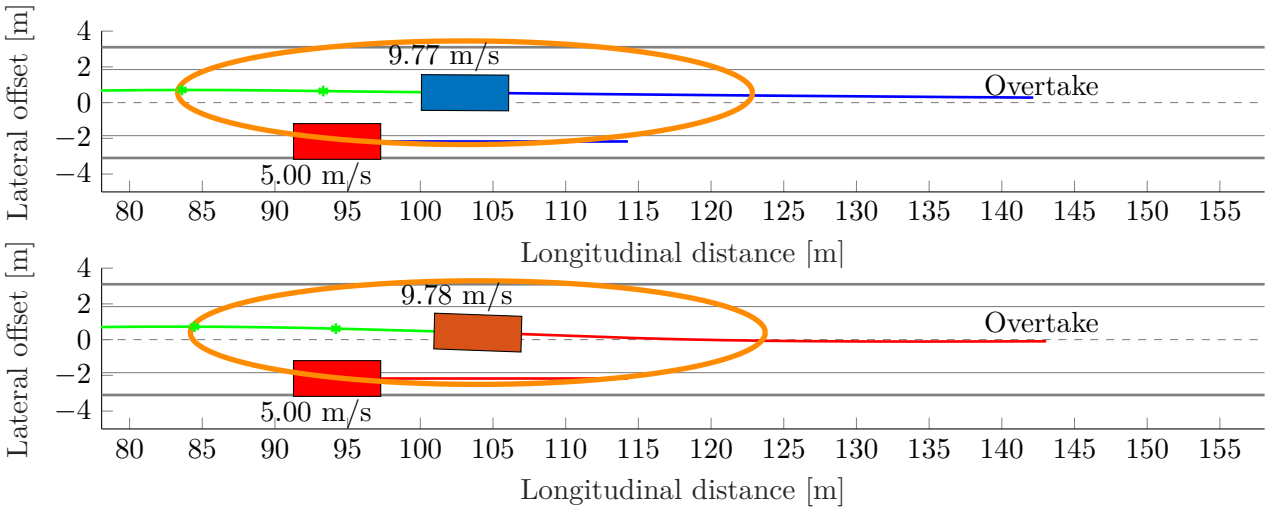
(a) Snapshot at $T = 3$ [s]



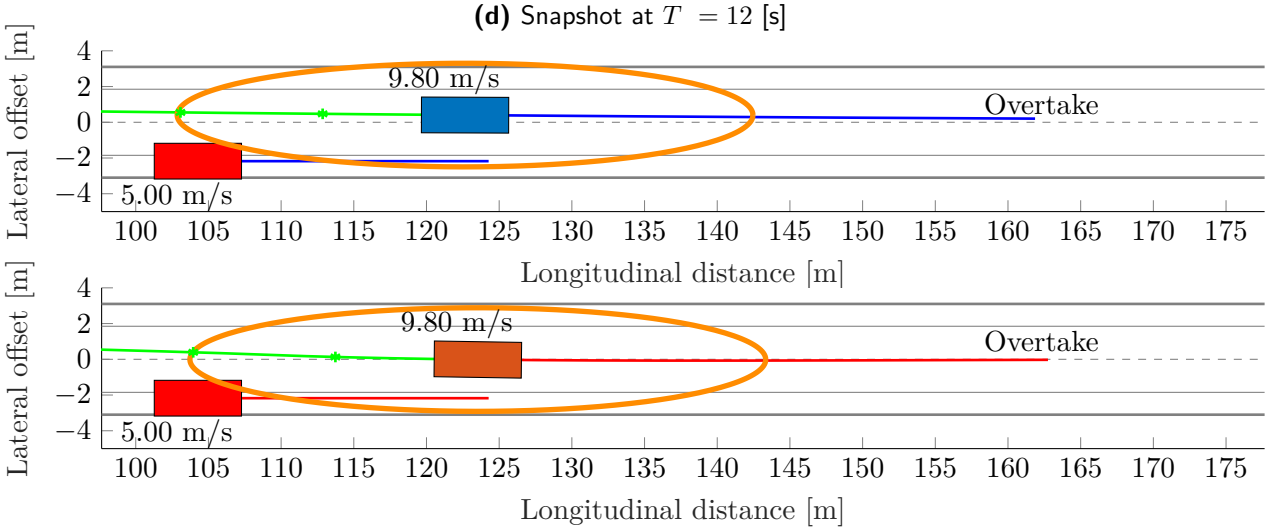
(b) Snapshot at $T = 6$ [s]



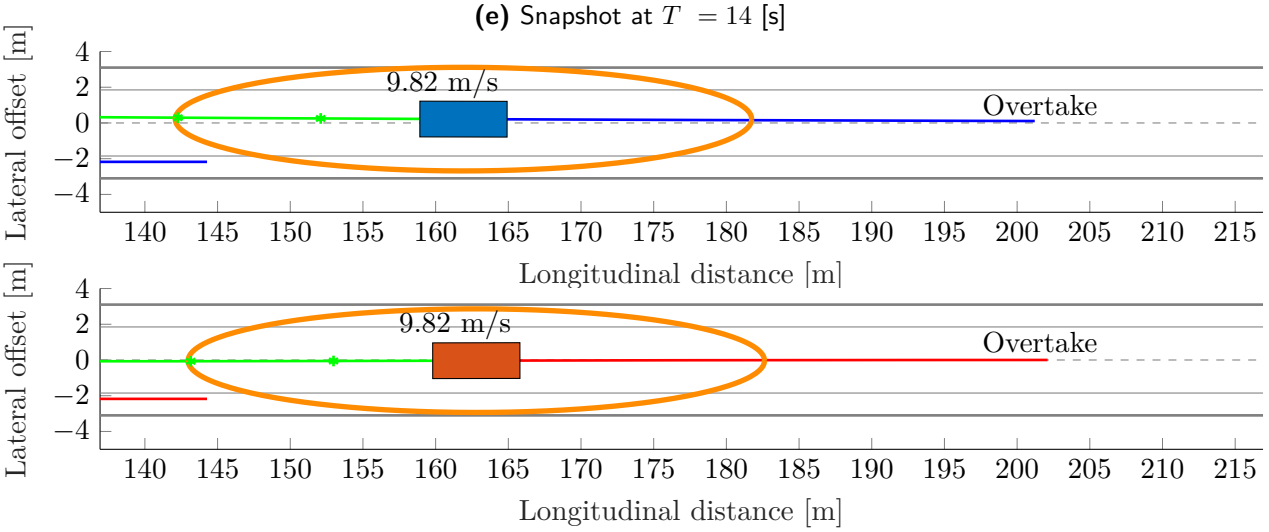
(c) Snapshot at $T = 9$ [s]



(d) Snapshot at $T = 12$ [s]

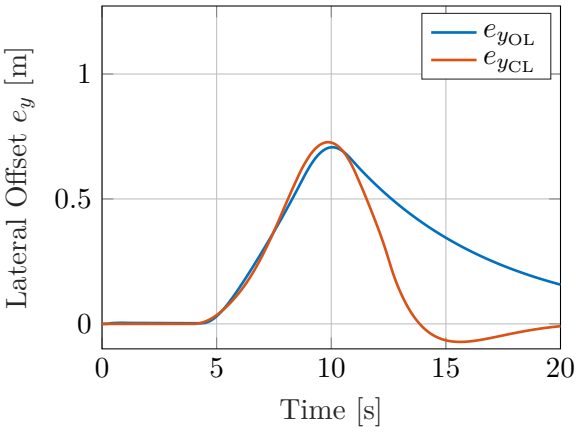


(e) Snapshot at $T = 14$ [s]

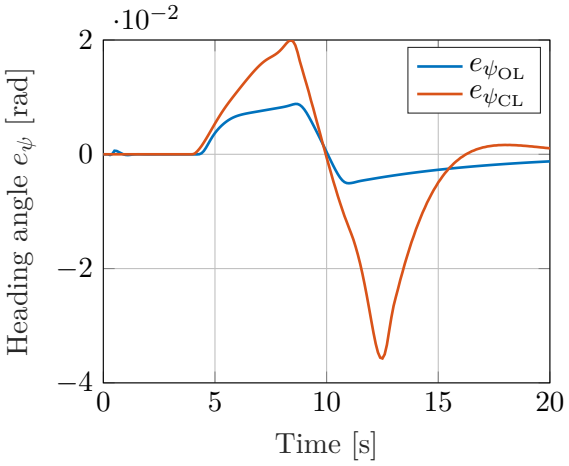


(f) Snapshot at $T = 18$ [s]

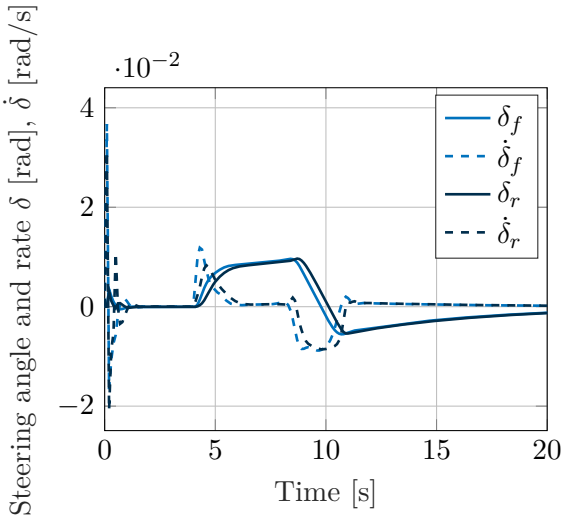
Figure 5-4: Snapshots of the simple overtake scenario, showing a successful overtake by both the GRT open-loop planner (blue) and closed-loop planner (dark orange).



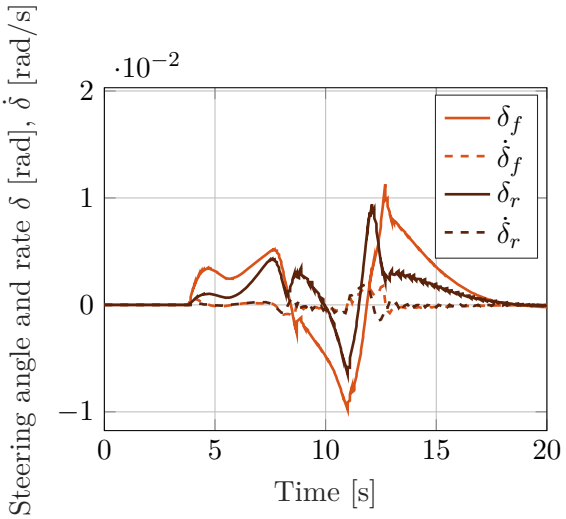
(a) Trajectory driven by the GRT



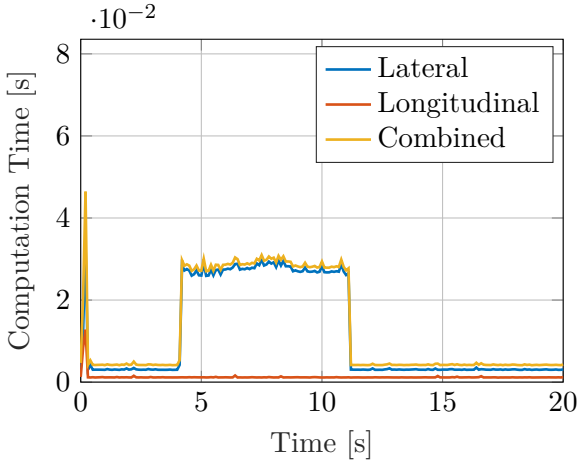
(b) Heading driven by the GRT



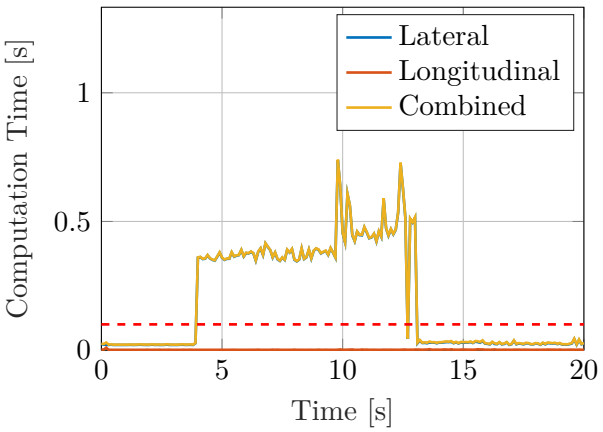
(c) Open-loop steering angles and rates



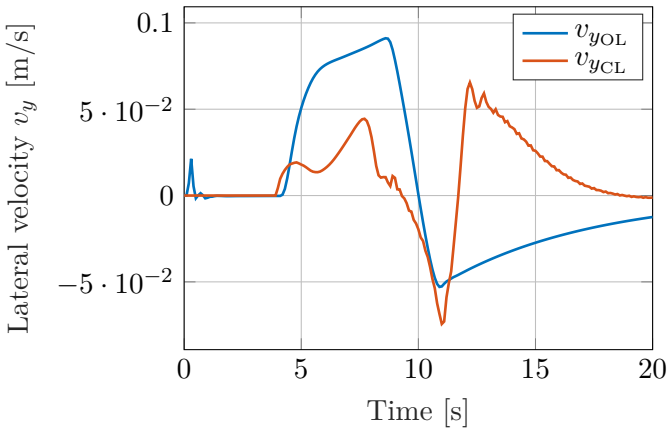
(d) Closed-loop steering angles and rates



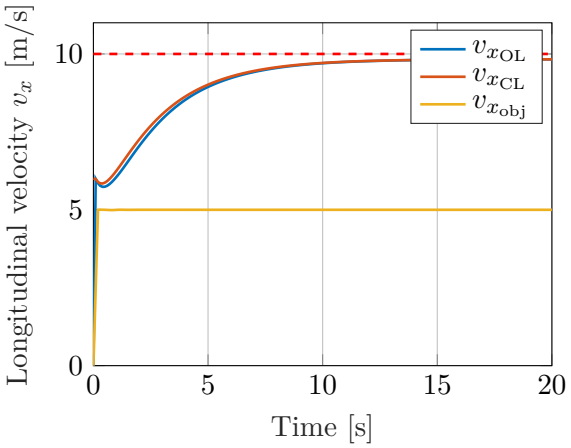
(e) Solution times for the two planning steps for the open-loop planner



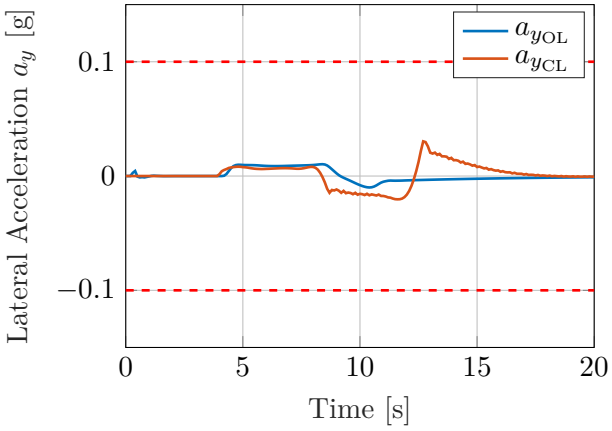
(f) Solution times for the two planning steps for the closed-loop planner



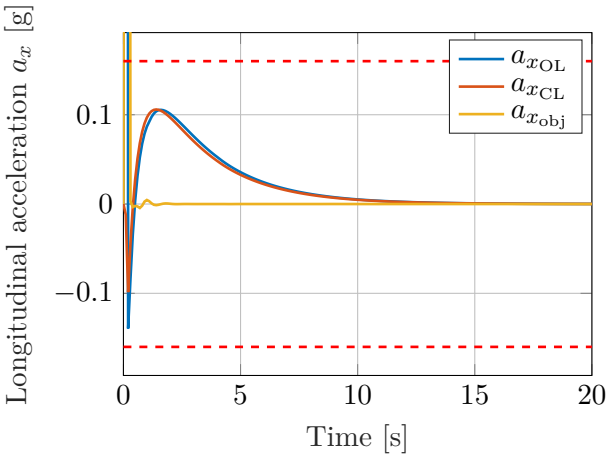
(g) Lateral velocity of the GRT



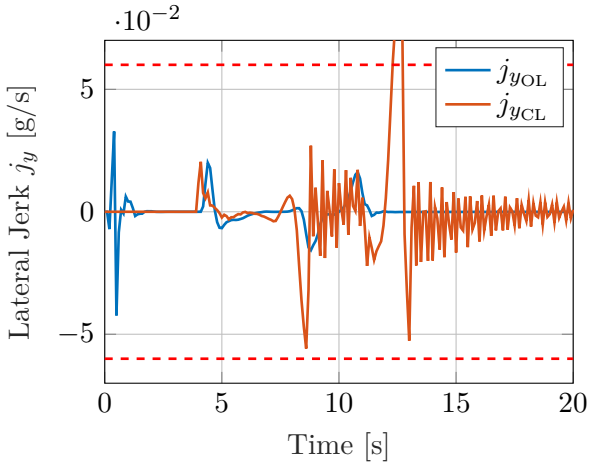
(h) Longitudinal velocity of the GRT



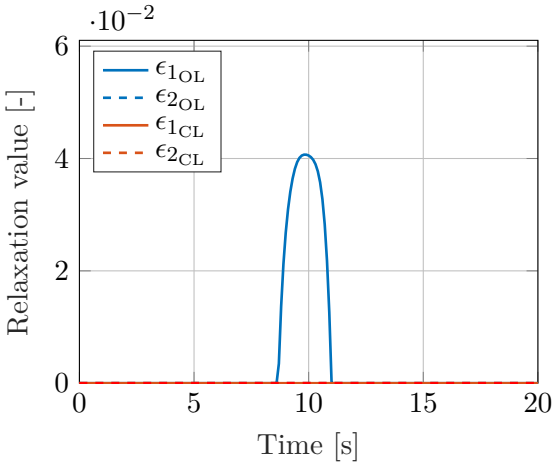
(i) Lateral acceleration of the GRT



(j) Longitudinal acceleration of the GRT



(k) Lateral jerk of the GRT



(l) Relaxation value of constraints

Figure 5-5: Closed-loop simulation results of the simple overtake scenario.

5-3 Scenario 3: Overtake moving obstacle in middle of the road

Scenario 3 is an extension of scenario 2 where the target vehicle has been moved to start straight ahead of the GRT. Solving such a scenario using the standard ellipse avoidance constraint without the flexible shell, would not be possible. The target vehicle is again driving 5 [m/s] all simulation long. The initial longitudinal gap is set to 34.53 [m]. Like in scenario 2, the GRT is initialized at 6 [m/s] with as reference velocity 10 [m/s].

For this scenario, both the GRT and obstacle are shifted -1.75 [m] in lateral initial position with respect to the other simulations. This way it is made sure that there is a collision-free corridor of sufficient width for overtaking on the left on the target vehicle, while staying within the box constraints. As in scenario 2, the target vehicle is driving to slow to follow. Therefore a new collision-free trajectory is generated and driven.

General Analysis

The development of the scenario is depicted in Figure 5-6. Like in scenario 2, initially no re-routing is necessary. Only after $T = 3$ [s] planners start deviating from the nominal trajectory. For both planners Figure 5-7j depicts a reduction in acceleration, while the difference in velocities between the GRT and the target vehicle stays large enough to remain in overtake mode. When increasing the lateral offset to the nominal trajectory, the velocity starts to increase again. This behaviour is similar to human driving as one only starts accelerating when having a clear line of sight along the target vehicle.

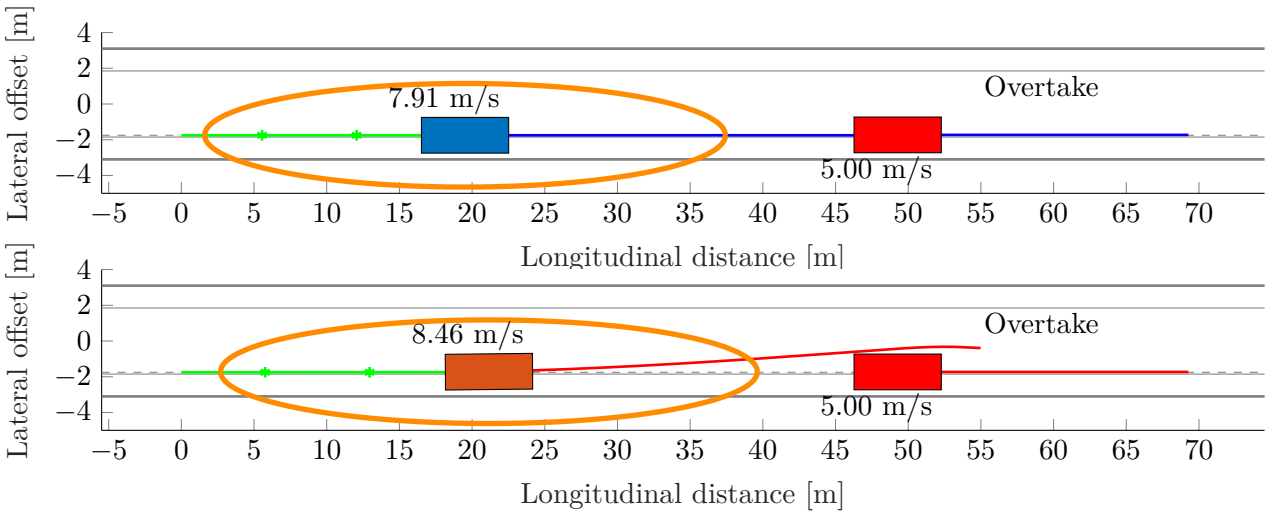
There are a lot of similarities between scenario 2 and 3. Also in scenario 3, the open-loop planner makes more use of its crabbing capabilities as can be seen from Figure 5-7b and 5-7g. Figure 5-7l shows the open-loop slack variable ϵ_1. It can be seen that while executing the left overtake on the target vehicle the lateral open-loop collision avoidance constraint is slightly relaxed. This does not compromise the GRT safety, as the safety margin is based on a fully relaxed constraint.

Comfort Analysis

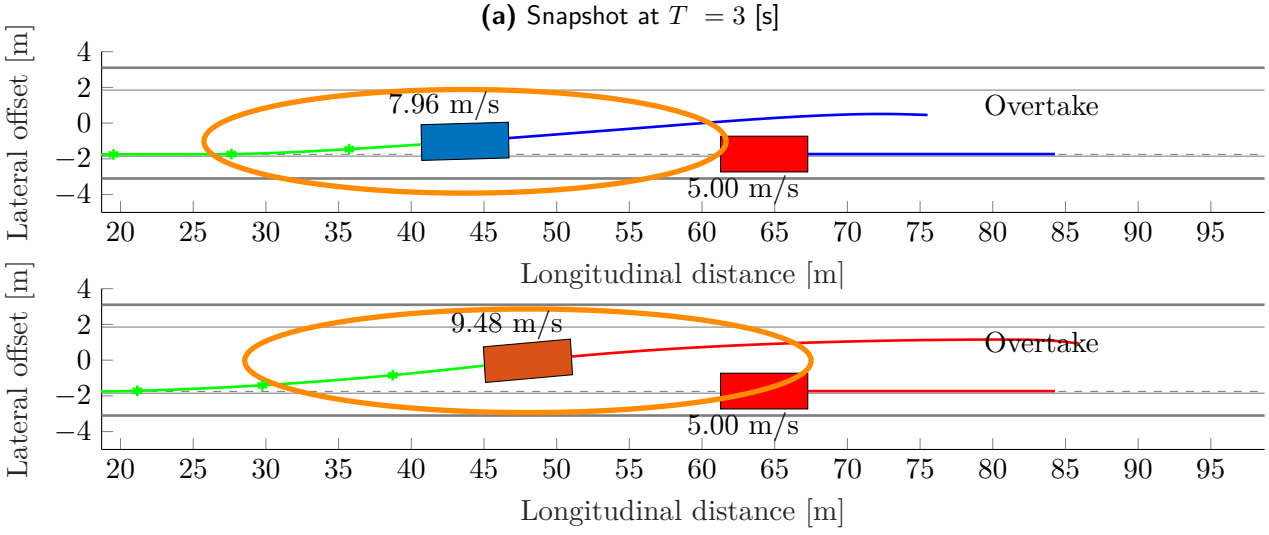
Also for the more challenging scenario 3, the comfort constraints are satisfied by both the open-loop and closed-loop planner. For the open-loop planner, the lateral jerk is smooth and remains within the comfort bounds set. The resulting trajectory is depicted in Figure 5-7a. It is sufficiently smooth and leaves enough lateral spacing between the GRT and target vehicle in order to comply with the collision avoidance constraint.

Real-time performance Analysis

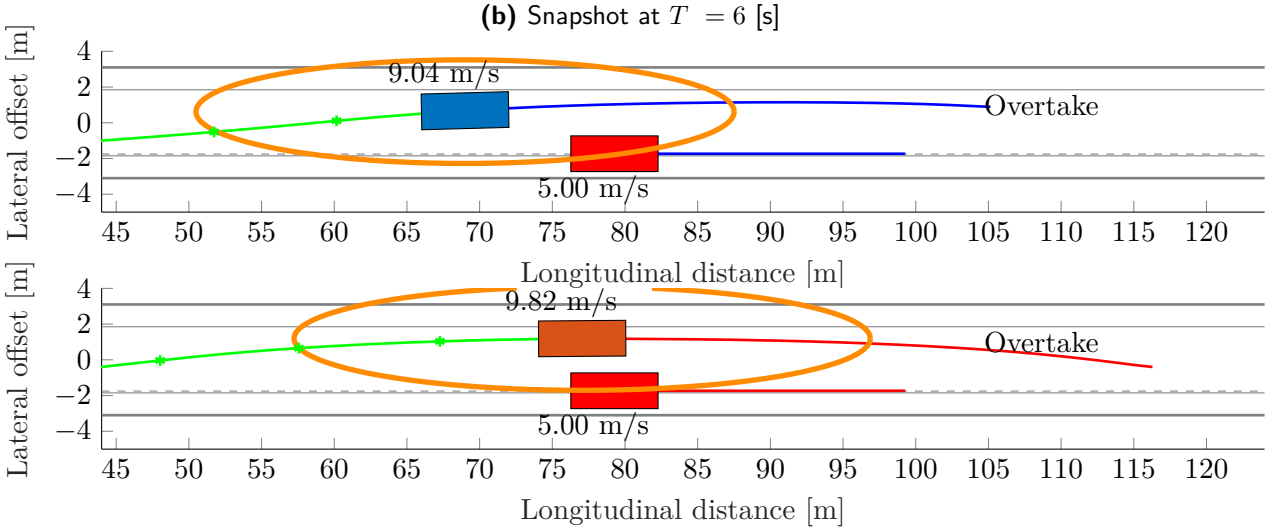
Although the free manoeuvring space is smaller for this scenario compared to the latter one, the computation times for solving the trajectory generation problem are only slightly worse than in the much simpler scenario 2 in Section 5-2. This performance reduction is mainly caused by the increase of computational time needed for the longitudinal planner. Still, computation times for the open-loop planner are well under the set limit of 100 [ms].



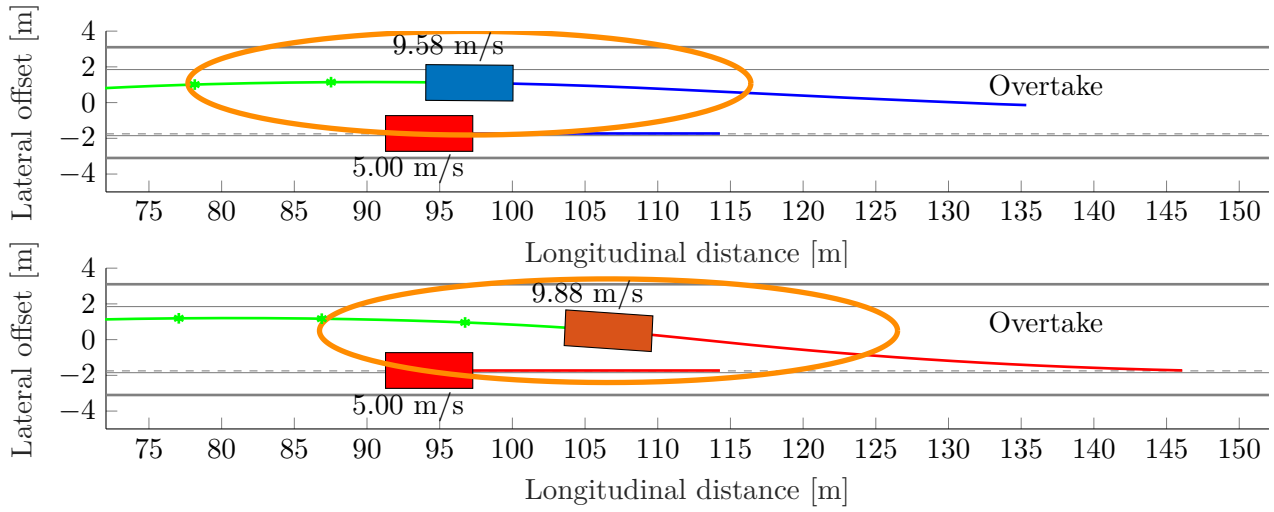
(a) Snapshot at $T = 3$ [s]



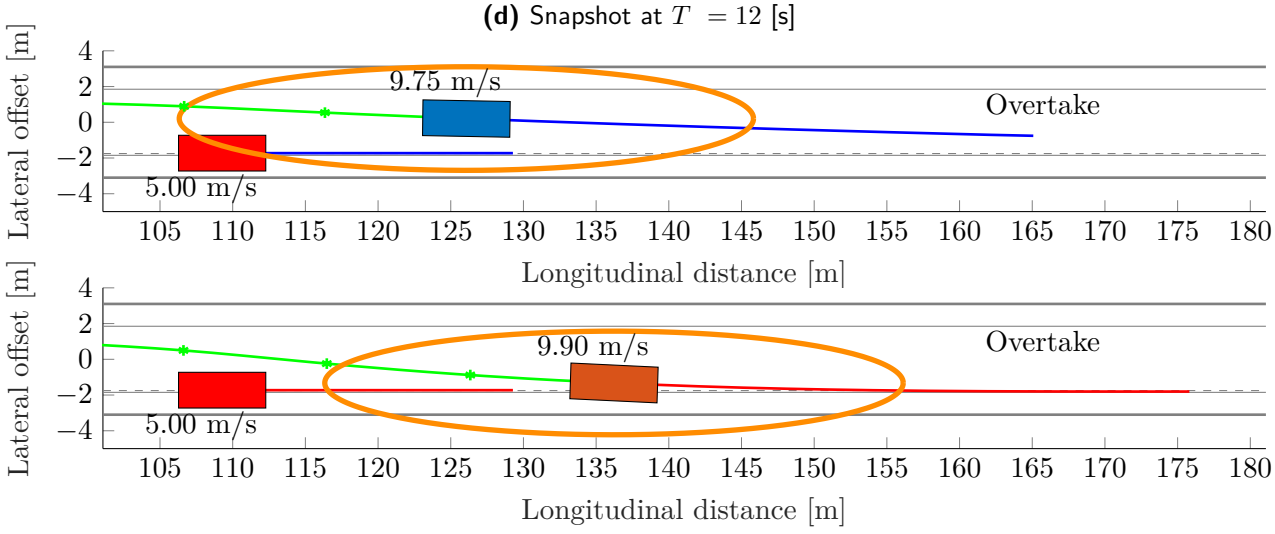
(b) Snapshot at $T = 6$ [s]



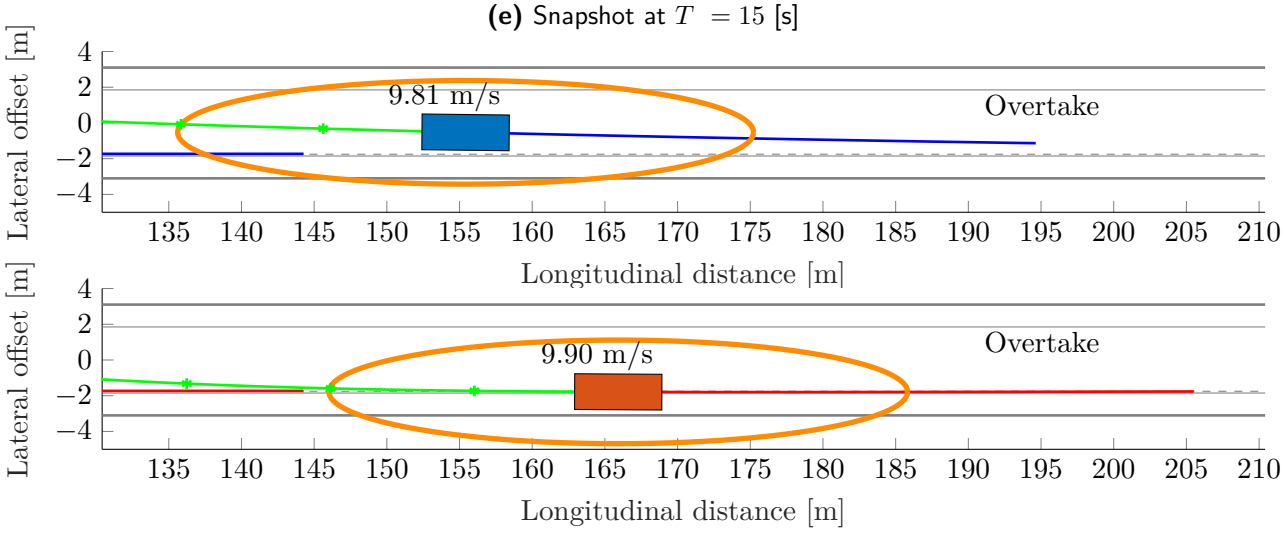
(c) Snapshot at $T = 9$ [s]



(d) Snapshot at $T = 12$ [s]

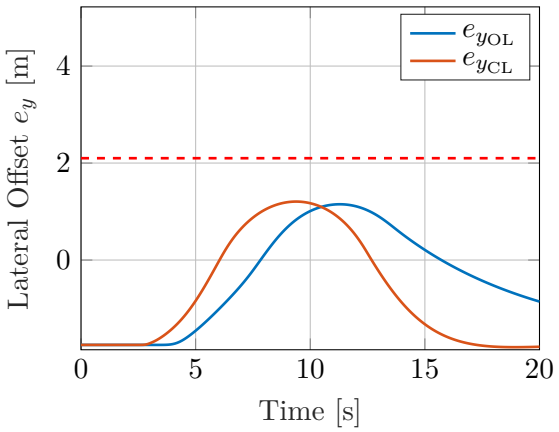


(e) Snapshot at $T = 15$ [s]

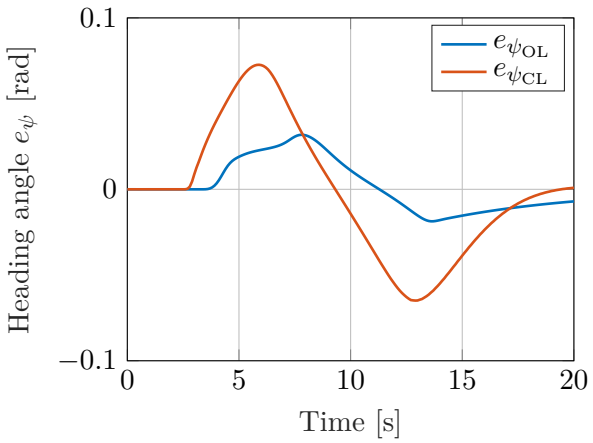


(f) Snapshot at $T = 18$ [s]

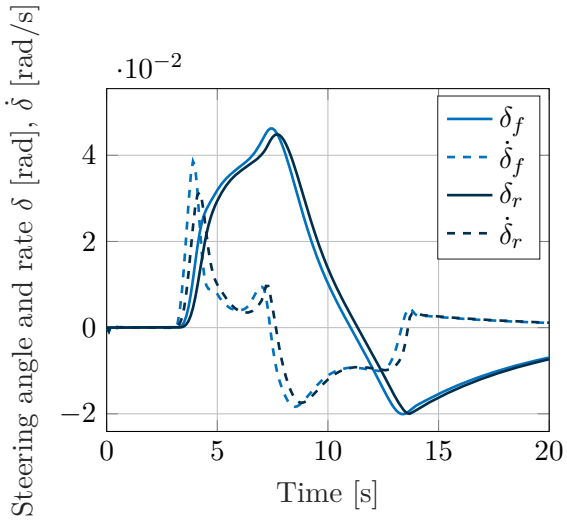
Figure 5-6: Snapshots of the complex overtake scenario, showing a successful overtake by both the GRT open-loop planner (blue) and closed-loop planner (dark orange).



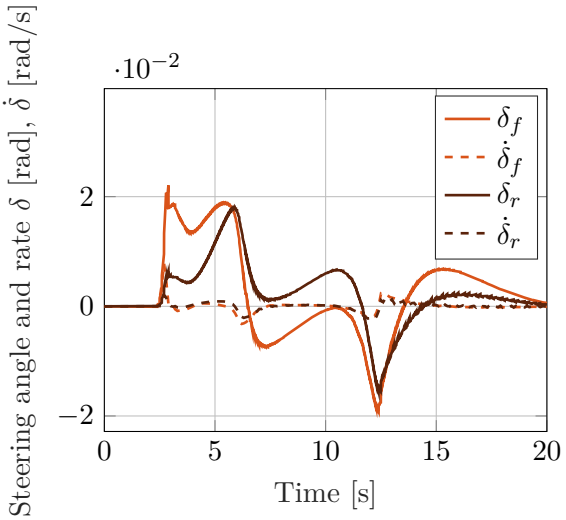
(a) Trajectory driven by the GRT



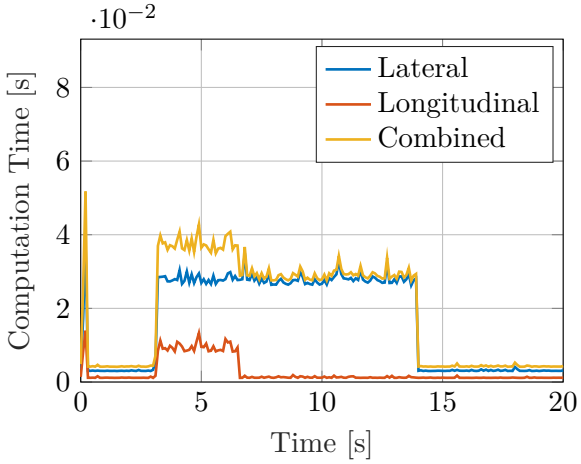
(b) Heading driven by the GRT



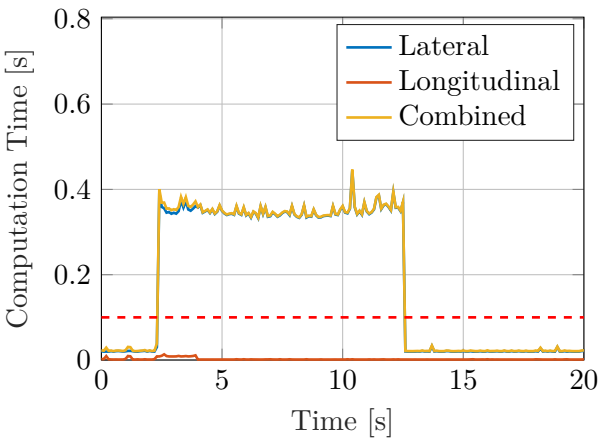
(c) Open-loop steering angles and rates



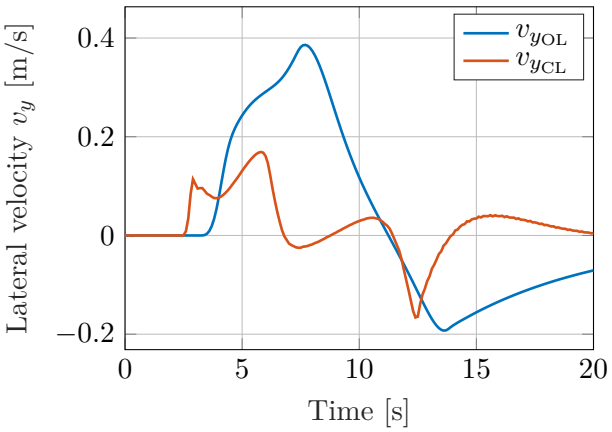
(d) Closed-loop steering angles and rates



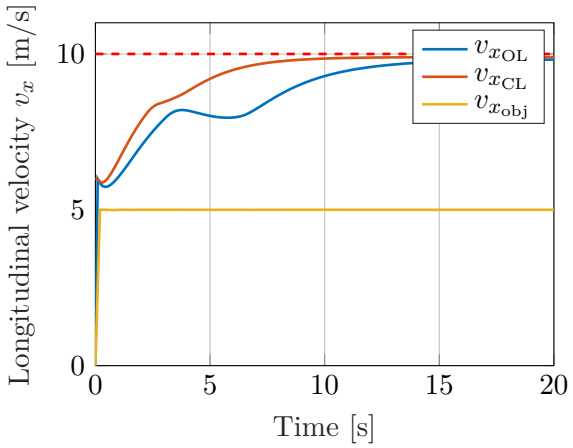
(e) Solution times for the two planning steps for the open-loop planner



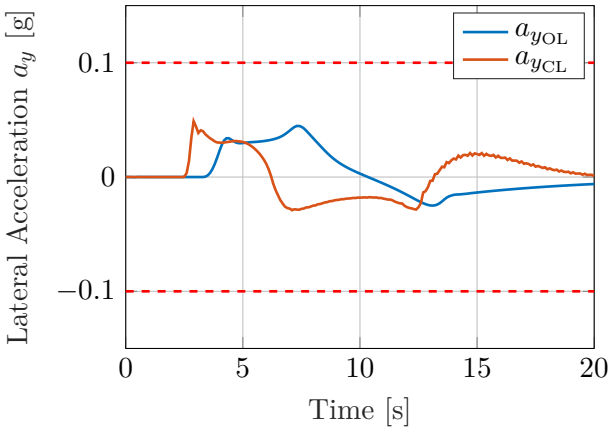
(f) Solution times for the two planning steps for the closed-loop planner



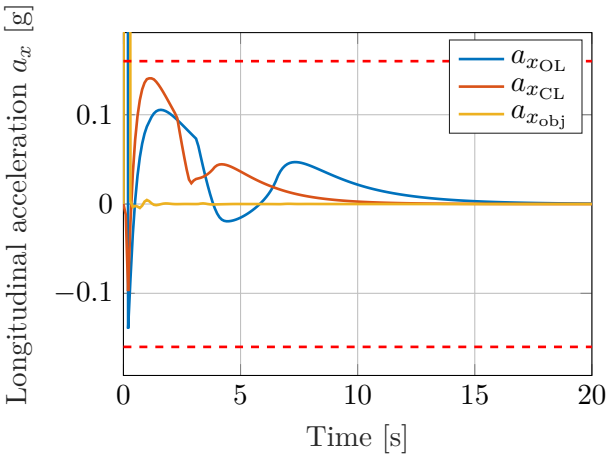
(g) Lateral velocity of the GRT



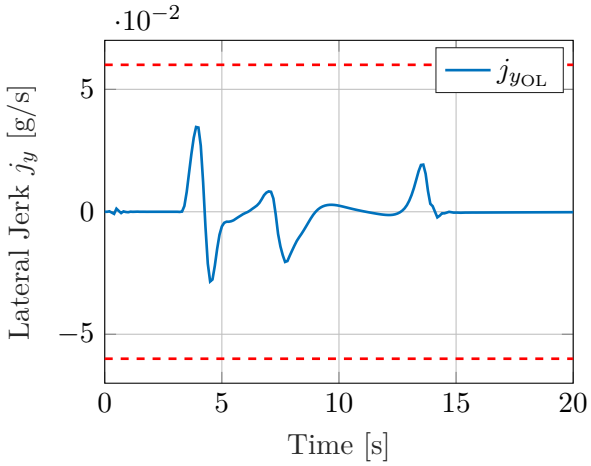
(h) Longitudinal velocity of the GRT



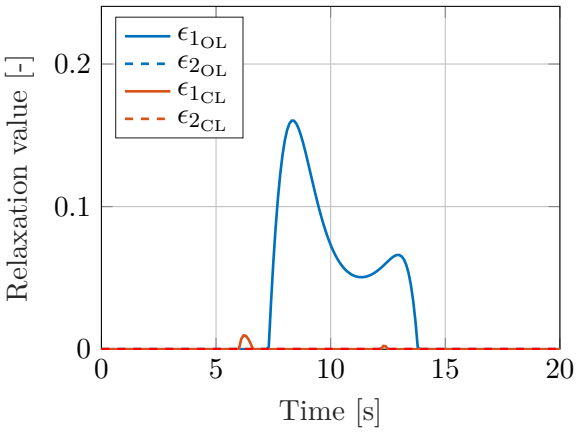
(i) Lateral acceleration of the GRT



(j) Longitudinal acceleration of the GRT



(k) Lateral jerk of the GRT



(l) Relaxation value of constraints

Figure 5-7: Closed-loop simulation results of the complex overtake scenario.

5-4 Scenario 4: Abort overtake due to lateral movement obstacle

In scenario 4 a general overtaking scenario is considered of a target vehicle having an arbitrary longitudinal velocity. Initially scenario 4 is similar to scenario 2 of Section 5-2. Only for scenario 4, the target vehicle starts moving to the left, merging in front of the GRT. Due to this lateral movement, the simple decision making module aborts the overtake manueovre and the GRT has to return to a obstacle-following situation similar to scenario 1.

The target vehicle is initially at standstill, standing 49.53 [m] in front of the GRT. The initial lateral offset equals -2.18 [m], as if the target vehicle is parked on the advisory bicycle lane. The first 7 [s], the target accelerates up to 7 [m/s]. After a travelled distance of 35 [m], the target vehicle initiates its merging manoeuvre. During this manoeuvre the target velocity is kept constant. Due to this merger the target vehicle cuts off the GRT's planned overtake and ends up straight ahead of the GRT. After finishing this cut-off manoeuvre at $T = 13.5$ [s] the target accelerates with a constant acceleration of 0.5 [m/s^2] until it reaches 10 [m/s] at $T = 19.5$ [s]. This velocity is then kept constant during the rest of the simulation.

Initially the velocity gap between GRT and target vehicle is large enough to enable an overtake manoeuvre. However, once the target vehicle initiates its merging manoeuvre this lateral movement is sensed by the GRT and the planned overtake is aborted. The distance holder constraint is enabled and the GRT returns to the nominal trajectory, following the target vehicle.

General Analysis

The development of the scenario is depicted in Figure 5-8. As in scenario 2 and 3, initially no re-routing is necessary for scenario 4. As the gap between the target vehicle and the GRT closes, re-routing is necessary to maintain a collision-free trajectory. From Figure 5-7a it is clear that this re-routing starts at $T \approx 4$ [s]. At $T = 6$ [s] overtaking is still allowed and the GRT already developed a lateral offset with respect to the nominal trajectory. Figure 5-8c depicts a snapshot just after the overtake abort has been triggered and the distance holder constraint has been enabled. From this Figure it is clear that for both the open-loop and closed-loop case, the distance holder constraint has to be relaxed as the distance between the GRT and target vehicle is initially smaller than the constraint size. The merging target vehicle is depicted in Figure 5-8d and Figure 5-8e. From Figure 5-8f and on the scenario develops as an obstacle following scenario as in Section 5-1.

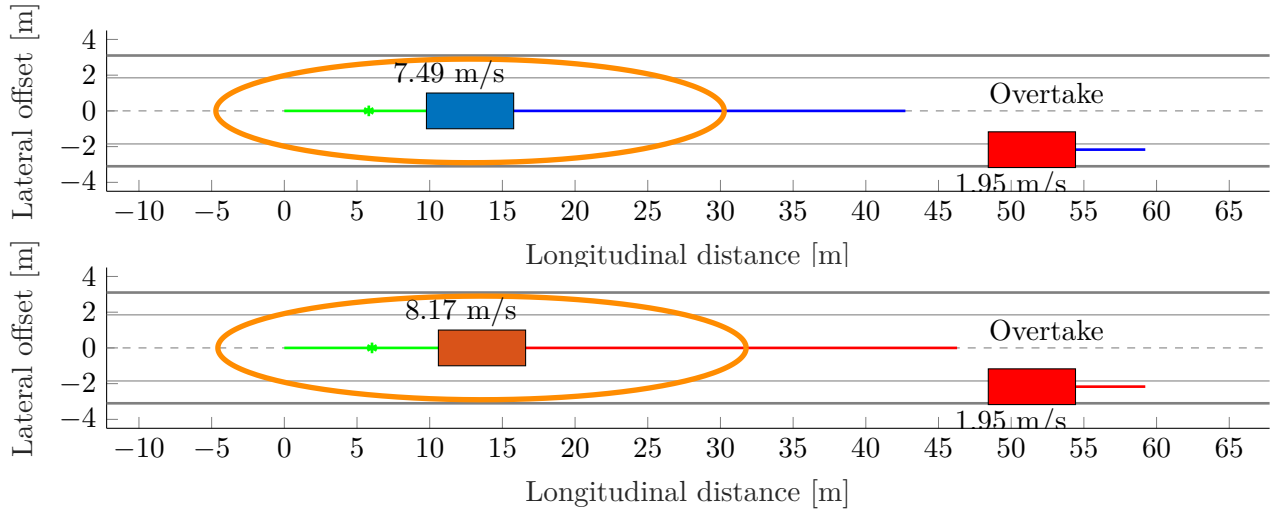
The moment the lateral movement of the target is detected is visible in Figure 5-9k and 5-9l. With the overtake being aborted, the distance holder constraint is enabled. Since the overtake manoeuvre has already been initiated, this distance holder constraint has to be relaxed quite drastically in order not to violate the constraint. The relaxation of the distance holder constraint is depicted in Figure 5-9c and 5-9d for the open-loop and closed loop case respectively. The value of the relaxation variable itself is shown in Figure 5-9k. This step is quite aggressive and thus not ideal for the RTI algorithm, which assumes a close to optimal hot-start. This decrease in optimality is also clearly visible in Figure 5-9l. In the next SQP iterations, the KKT value improves again.

Comfort Analysis

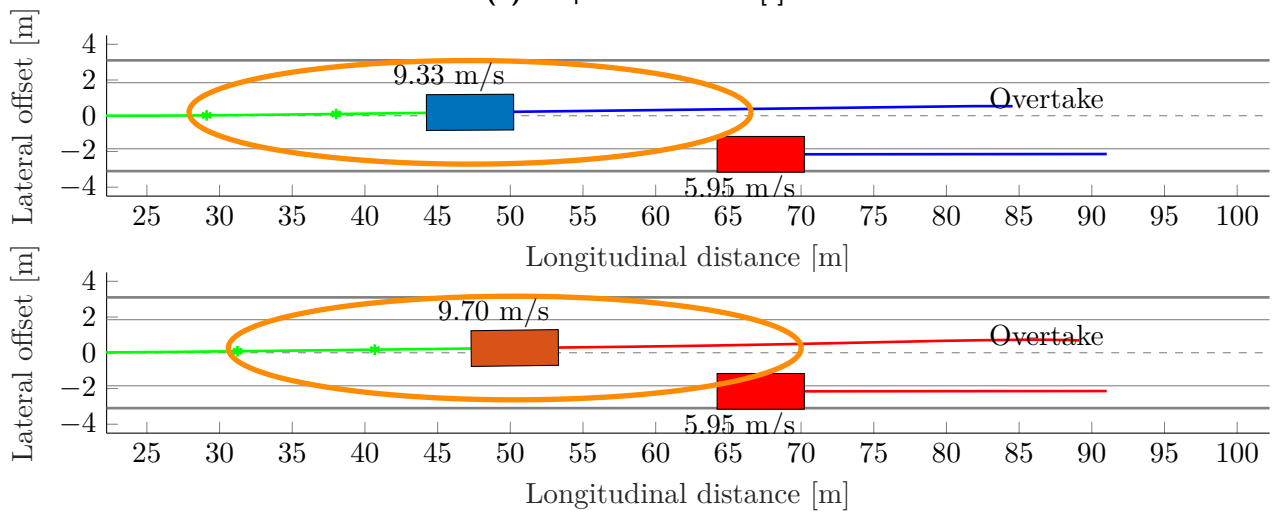
Other than the suboptimalities in the optimization, the solution of the QP problem is feasible during the entire simulation and thus all comfort and safety constraints are met, bringing the GRT both comfortably and safely back to a following scenario. In Figure 5-9a around $T = 10$ [s] it can be seen that, while aborting the overtake, the closed-loop planner even moves the GRT more to the left avoiding a collision with the merging target vehicle.

Real-time performance Analysis

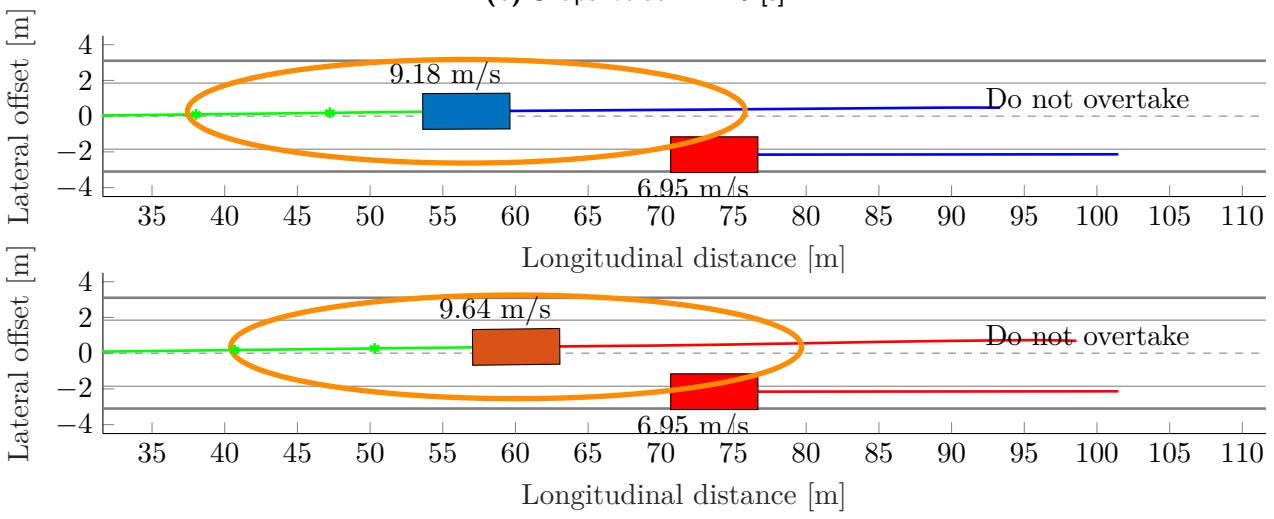
Due to the large optimality gap between initialization and result of the QP solver at the instance the distance holder constraint is activated, lots of QP iterations are needed before termination, resulting in the spike in computation time in Figure 5-9e. Although for this scenario the resulting total computation time of the open-loop planner remains well below the set upper-limit of 100 [ms], this is only an indication of real-time performance as results cannot directly be generalized.



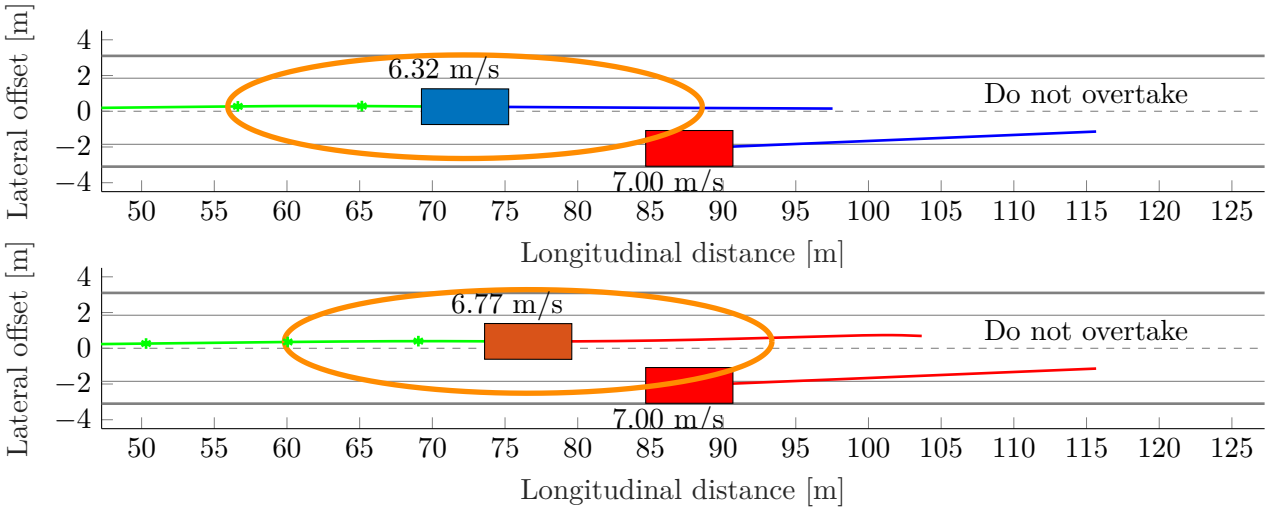
(a) Snapshot at $T = 2$ [s]



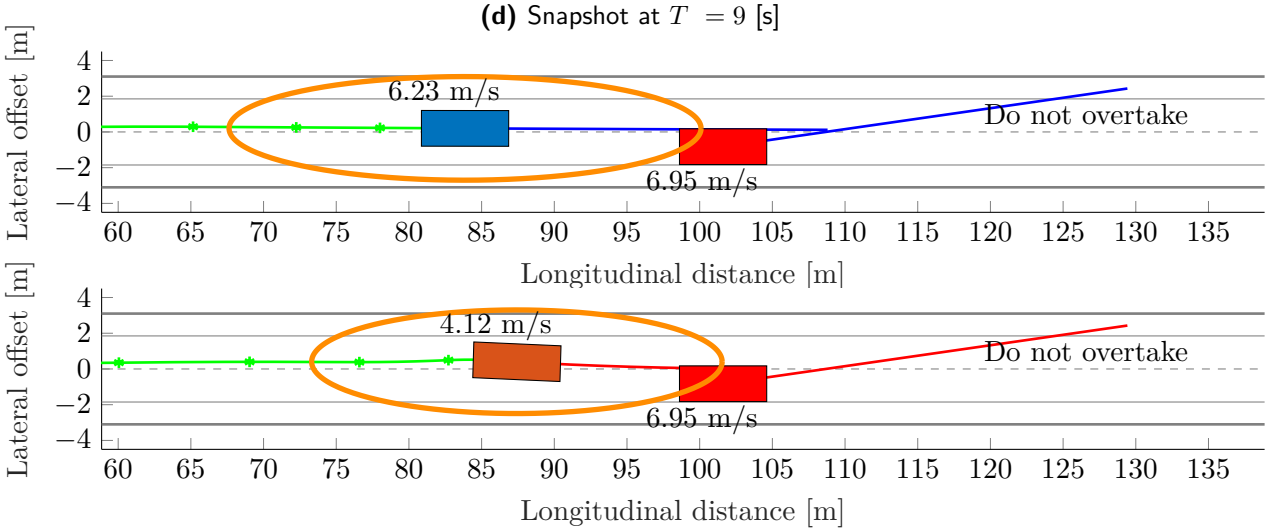
(b) Snapshot at $T = 6$ [s]



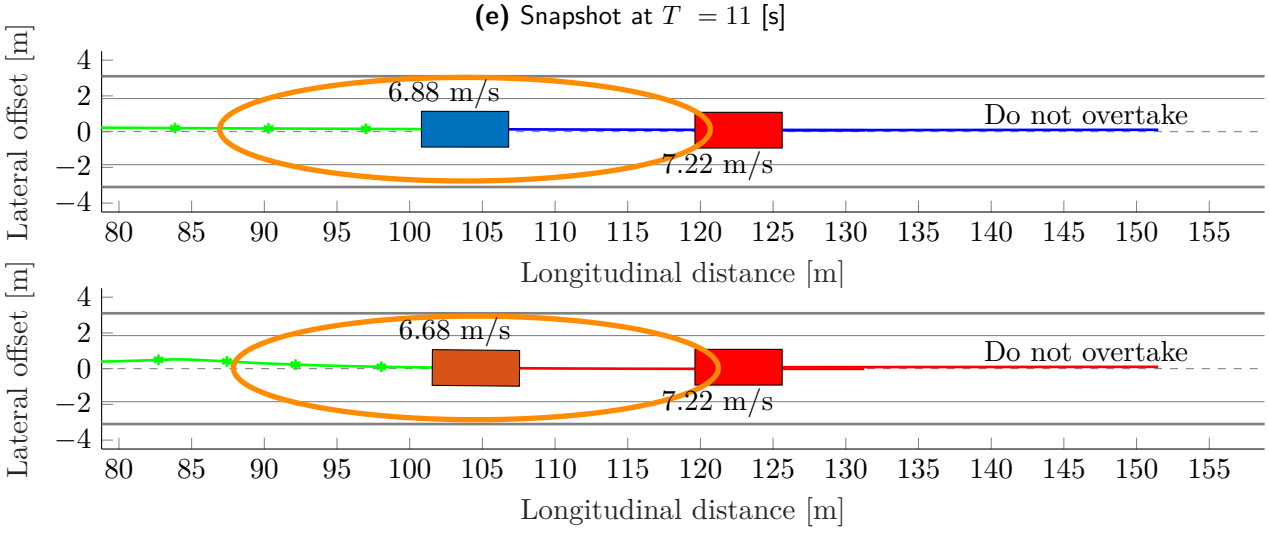
(c) Snapshot at $T = 7$ [s]



(d) Snapshot at $T = 9$ [s]

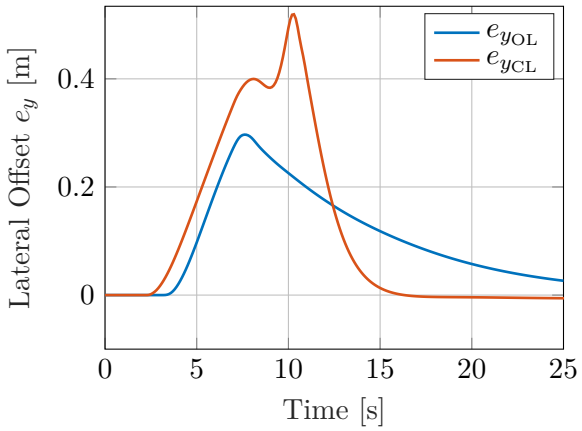


(e) Snapshot at $T = 11$ [s]

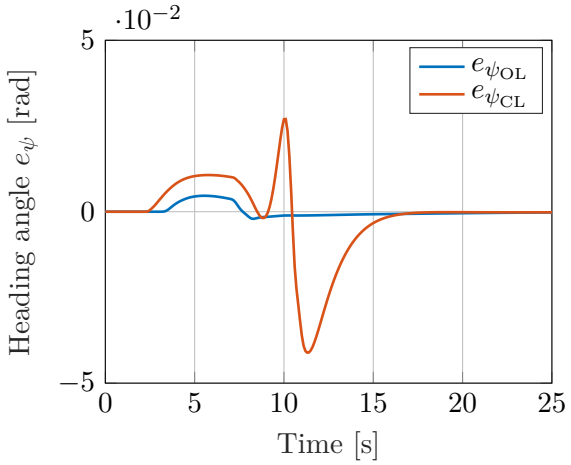


(f) Snapshot at $T = 14$ [s]

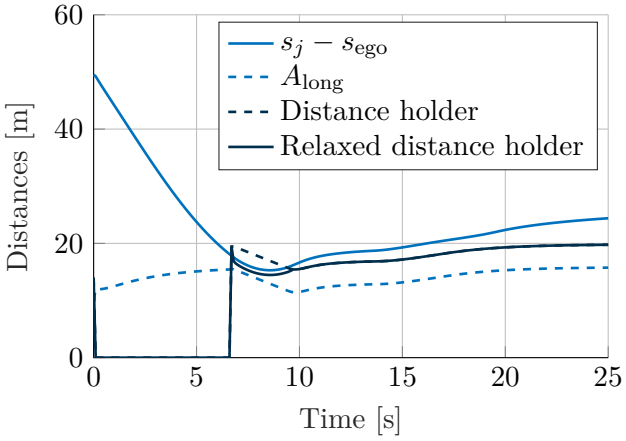
Figure 5-8: Snapshots of an overtake-abort scenario, showing a successful abort of a planned overtake by both the GRT open-loop planner (blue) and closed-loop planner (dark orange).



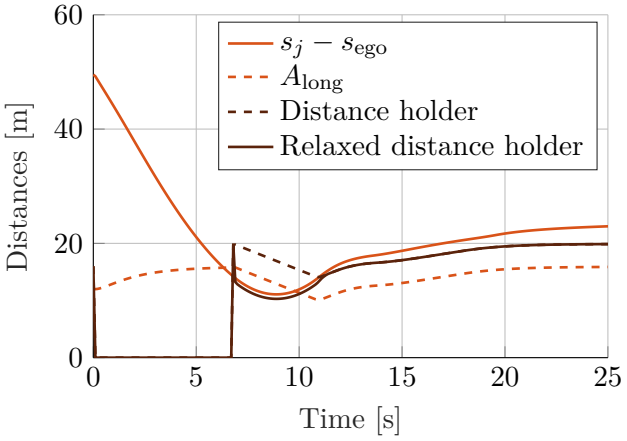
(a) Trajectory driven by the GRT



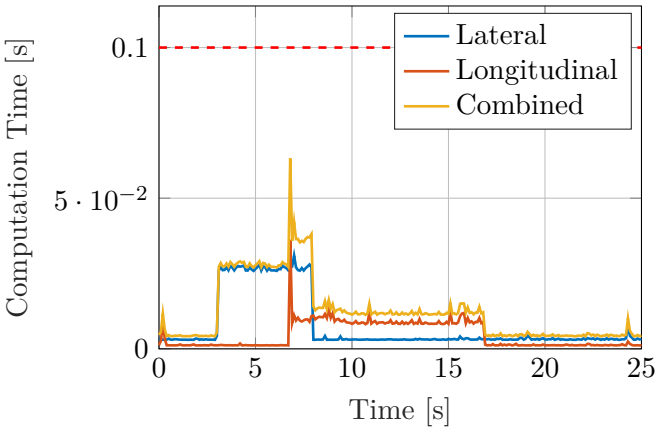
(b) Heading driven by the GRT



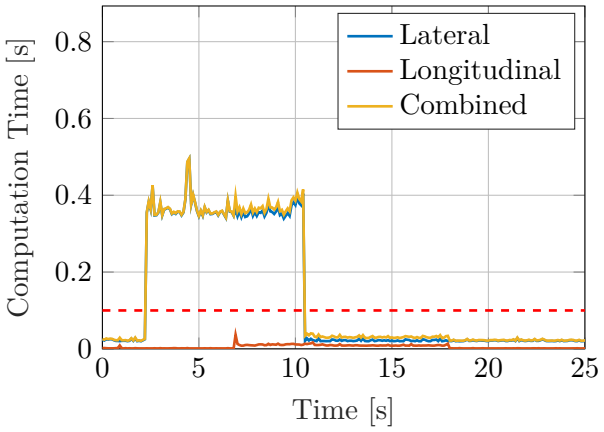
(c) Open-loop distances and constraint values



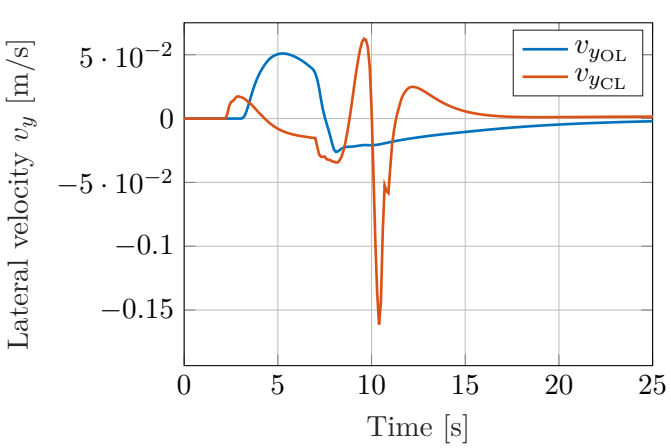
(d) Closed-loop distances and constraint values



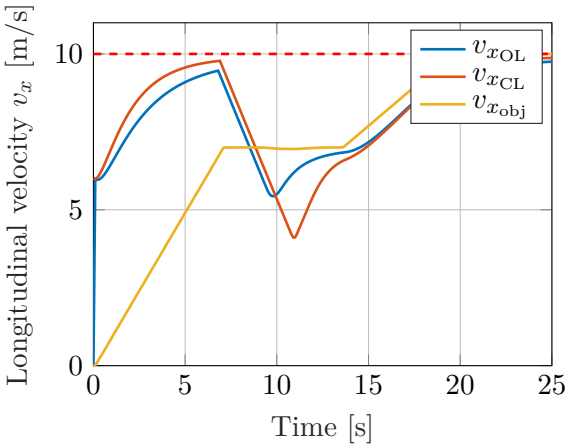
(e) Solution times for the two planning steps for the open-loop planner



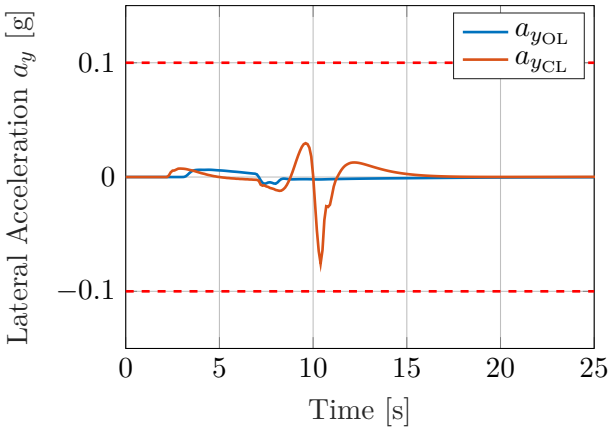
(f) Solution times for the two planning steps for the closed-loop planner



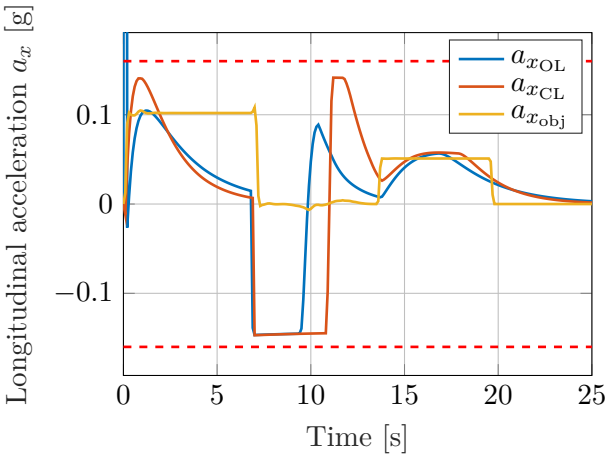
(g) Lateral velocity of the GRT



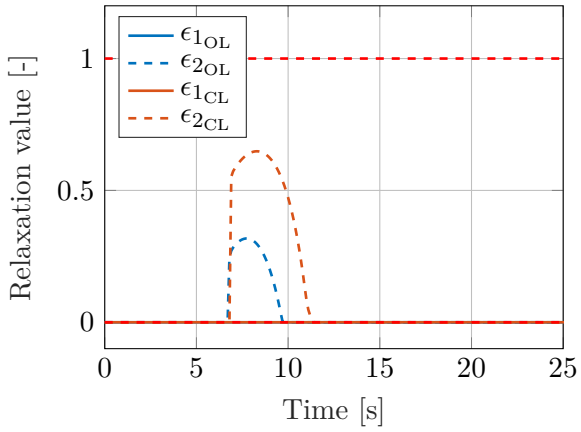
(h) Longitudinal velocity of the GRT



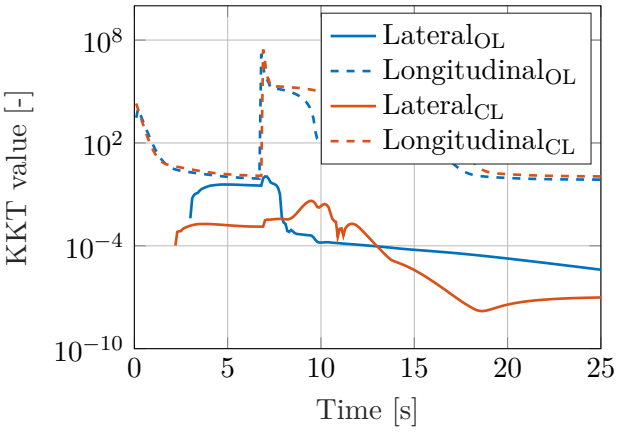
(i) Lateral acceleration of the GRT



(j) Longitudinal acceleration of the GRT



(k) Relaxation value of constraints



(l) Development of the KKT value

Figure 5-9: Closed-loop simulation results of an overtake-abort scenario.

5-5 Summary

This chapter showed and discussed the results of four numerical simulations on the developed trajectory generator. The results of the simulations have been verified to meet the constraints and requirements imposed on the trajectory generator.

It is shown that the open-loop trajectory generator can deal with following scenarios as well as overtaking scenarios. Furthermore there exists a possibility of aborting an overtake. All results have showed that the trajectory generator can deal with those simulations in real-time while meeting the comfort and safety constraints developed for the planner. Although, results look promising, spikes in computation time are observed when aborting simulations, possibly leading to computation times exceeding the set upper-bound of 100 [ms].

It is shown that the closed-loop trajectory generator does not meet all requirements set. Both the real-time aspect of the requirements and the comfort jerk constraints have been shown to be violated. The real-time issue might be reduced by setting up the Quadratic Programming problem for a sparse QP solver. It is argued that the violation of the jerk constraint is most likely to be caused by the non-smoothness of the applied steering angles and therefore is likely to be a simulation issue mostly.

The next, and final, chapter discusses the main conclusions from this thesis. A short summary of the work is given and opportunities for future work are listed.

Chapter 6

Conclusions and Future Work

This thesis presents a real-time implementation of a dynamic re-routing procedure for 2getthere's GRT. An Optimal Control Problem (OCP) has been defined for both an open-loop and closed-loop planning case. The resulting Nonlinear Program (NLP) is solved in the `ACADO toolkit` modeling environment connected to the `qpOASES` Quadratic Program (QP) solver. Numerical simulations show for several scenarios that the Real-Time Iteration (RTI) solution scheme provides updated trajectories successfully every 100 [ms] for the open-loop planner case. Moreover, the generated trajectories are feasible and comfort requirements are met. However, for the closed-loop planner case, the real-time requirement is not satisfied.

The main contributions of this thesis work can be summarized as follows:

- The extension of the elliptical collision avoidance constraint by a flexible shell, creating a flexible safety zone surrounding the constraint such that the constraint can be applied to Sequential Quadratic Program (SQP) solving techniques.
- The development and real-time implementation of an OCP formulation tailored to the autonomous re-routing of a GRT in a standard driving scenario.

The latter chapter depicted results and discussed the performance of both designed planners. The remainder of this chapter provides a general summary and conclusions of this work in Section 6-1 and elaborates on the critical points in this thesis open for future work in Section 6-2.

6-1 Summary and Conclusions

In Chapter 2 several options for re-routing have been discussed. It is concluded that Model Predictive Control (MPC) is an appropriate modeling framework for setting up the trajectory generation problem for this thesis, due to its advantages in extending the problem and the possibility of fine-tuning resulting trajectories on comfort and safety.

In Chapter 3 an analysis of the open-loop and closed-loop dynamical systems results in lateral comfort constraints. Additionally, it is concluded that a standard ellipse shaped collision avoidance constraint does not fit a general SQP optimization routine, therefore a relaxed version with a flexible shell is proposed. Three different OCPs have been formulated to facilitate the lateral open-loop, lateral closed-loop and longitudinal planning.

In Chapter 4 it is concluded that the `ACADO toolkit` is an appropriate modeling environment for implementation the OCPs. The system size for the open-loop system should be small enough to solve the QP subproblems of the SQP routine using the `qpOASES` QP solver. It is concluded that running this optimization routine at $T_s = 0.1$ [s] with a prediction horizon of $N_p = 40$ steps is sufficient for the scenarios a GRT will encounter. The PreScan simulation package is used, as it is an effective tool to set a priori known target vehicle trajectories. Furthermore, it is noted that the trajectory generator is compiled into a C MEX s-function, which enables testing the planner on a high-fidelity vehicle model.

In Chapter 5 simulations were run on an off-the-shelf 2011 ASUS notebook PC running Windows 10 with an Intel Core i7-2630QM @ 2GHz and 8GB-DDR3. Average sampling times for the open-loop planner lie around approximately 13 [ms] on average, never exceeding 64 [ms], even for the most complex scenario. Therefore, this open-loop planner is suitable for real-time implementation. It has been concluded that the same is not true for the closed-loop planner. Even though the planner runs at approximately 25 [ms] on average in idle mode, re-routing takes up approximately 370 [ms] on average with active constraints with maximum computation times reaching approximately 500 [ms].

Both planners generate fully collision-free trajectories for all simulations carried out for this thesis. For the open-loop planner additionally, the comfort constraints are met as well. For the closed-loop planner, only lateral jerk comfort constraints are not met. As discussed in the latter chapter, this is likely to be caused by the steering interface between planner and dynamical model. Improvements in this interface will also improve the closed-loop planner comfort performance. Furthermore, it is clear from the simulation results that trajectories are followable. However, no guarantees for real-world performance can be deduced from these numerical simulations. Running similar scenarios with 2getthere's low-level controller in the loop will already provide a better estimate of the real-world performance, but there are more points that can be improved in future work. These will be discussed in the next section.

6-2 Recommendations for Future Work

Although the presented work is a good initial basis for the dynamic re-routing of a GRT, there are limitations present. Overcoming those limitations would take the trajectory generation for the GRT to an even higher level, bringing it closer to on-road implementation. The upcoming subsections will elaborate on some of the recommendations for future work. In case no explicit distinction is made, discussed topics are applicable to both the open-loop and closed-loop planning case.

6-2-1 Design for robust lateral planning

The current trajectory planners are neither designed, nor tested for robust lateral planning. However, planners might encounter several uncertainties that the planner has to be robust

to. Applications where robustness is necessary are e.g. planning in crosswind applications or planning with uncertain vehicle parameters. Although the actual weight and rotational inertia can be approximated using onboard sensors, a delta with the actual parameter value will always exist. Therefore the planner has to be robust to this prediction error.

Additionally running the numerical simulation on an even more detailed vehicle model might be beneficial for evaluating the robustness of the lateral planner. The CarSim simulation package provides highly accurate vehicle state simulations [39] and can be integrated into a PreScan environment. This way the simulation vehicle model order is increased and additional dynamical phenomena such as tire non-linearities and load transfer can be simulated. Furthermore, the CarSim vehicle model is implemented in a 3D coordinate frame enabling additional scenarios incorporating e.g. road banking.

6-2-2 Accurate modeling of longitudinal dynamics

Although the current planners already incorporate quite some details on the vehicle's longitudinal dynamics, predictions will even be more accurate when modeling additional variables such as the round inclination. Moreover, the longitudinal planning can be tailored even more to the GRT's longitudinal control when additionally the longitudinal tire-slip is incorporated in the longitudinal planning module, computing a torque set-point for the electrical engines to provide.

6-2-3 Model-based surrounding traffic

In this work, surrounding traffic was assumed to follow a priori known trajectories. No interactions between surrounding traffic and the GRT were included in the scenarios tested. In real-world scenarios, an interaction would exist between the GRT and the target vehicle. In the takeover-abort scenario of Section 5-4, one can imagine that the target vehicle would stop merging into the GRT's lane as soon as the presence of the GRT is noticed. Modeling these interactions can be achieved e.g. by implementing (intelligent) driver models for every target vehicle as is used in the field of microscopic traffic simulation.

6-2-4 Imperfect sensing of the road and surrounding traffic

The current a priori known trajectories are fed to the trajectory generator using an Actor Information Receiver (AIR) sensor. This sensor provides the information on target vehicles without uncertainty or measurement noise. Work needs to be done to test the planner's robustness to inaccurate or noisy target information. A different sensor modeling approach, and possibly the implementation of sensor fusion methods would provide target information in a more natural way. Alternatively, simulations can be run on processed data recorded in real-world scenarios.

6-2-5 Closed-loop planner performance

It has been shown in the results that the closed-loop planner suffers from two major performance issues. The issue of not reaching the real-time target might be reduced by incorporating

different solving techniques. As the partially non-linear system can be fully linearized such that the resulting system is a Linear Parameter Varying (LPV) system. Structure exploiting QP solvers like `qpDUNES` might increase performance over the parametric active set approach of `qpOASES`.

Smoothness of the driven trajectory of the closed-loop planner might be increased when implementing a different steering interface in between the planner and the PreScan high-fidelity model. Currently, it is likely that the used steering interface is the main reason why comfort constraints are not satisfied for the closed-loop planner. It is proposed to implement a copy of the GRT's steering interface in the simulation environment for Software in the Loop (SIL) testing.

6-2-6 Test planner on real vehicle software

The ultimate goal is to implement the planner on the GRT, enabling dynamic re-routing as part of the Dynamic Driving Task (DDT). Initial future work should scope on testing the designed planners in a SIL environment including the 2getthere low-level controller. This creates a safe environment to perform initial tests on real-vehicle software. If acceptance of trajectories and the planner's robustness to delays have been proven to work in this environment, further work can be done on implementing the real-time planner on the GRT itself.

Appendix A

Mathematical derivations

A-1 Condensing

The condensing of an optimal control problem has first been introduced by H.G. Bock. and K.J. Plitt [33] and has been used in the `qpOASES` solver [36] which is the quadratic program solver used in this work. This section summarizes the main working principle and serves as a backup for the explanation of the cause of the infeasibility. Let the discretised linear open-loop optimal control problem be presented in the following form:

$$\min_{\substack{x_{k_0},\dots,x_{k_0+n_p} \\ y_{k_0},\dots,y_{k_0+n_p} \\ u_{k_0},\dots,u_{k_0+n_p}}} \frac{1}{2} \sum_{k=k_0}^{k_0+n_p-1} (y_k - y_{\text{ref}})' Q (y_k - y_{\text{ref}}) + (u_k - u_{\text{ref}})' R (u_k - u_{\text{ref}}) + \frac{1}{2} \left(y_{k+n_p} - y_{\text{ref}}\right)' P \left(y_{k+n_p} - y_{\text{ref}}\right) \tag{A-1a}$$

subject to

$$x_{k_0} = \tilde{x}_0 \tag{A-1b}$$
$$x_{k+1} = Ax_k + Bu_k \tag{A-1c}$$
$$y_k = Cx_k \tag{A-1d}$$
$$l \leq My_k + Nu_k \tag{A-1e}$$

The transformation of the parametric quadratic optimal control problem of (A-1), to an OCP of a much smaller scale is called condensing. The evolution of states for the N_p prediction steps can be expressed by evolving (A-1c) over all N_p steps. This then looks as follows:

$$x_{k+1} = Ax_k + Bu_k \tag{A-2a}$$
$$x_{k+2} = Ax_{k+1} + Bu_{k+1} = A^2x_k + ABu_k + Bu_{k+1} \tag{A-2b}$$
$$\vdots$$
$$x_{k+N_p} = A^{N_p}x_k + \sum_{i=0}^{N_p-1} A^{N_p-1-i}Bu_{k+i} \tag{A-2c}$$

From (A-2) a structure becomes visible. This structure can be summarized in the following augmented quantities:

$$\tilde{x} \overset{\text{def}}{=} \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+N_p} \end{bmatrix}, \qquad \tilde{u} \overset{\text{def}}{=} \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N_p-1} \end{bmatrix},$$

$$\tilde{Q} \overset{\text{def}}{=} \begin{bmatrix} Q & & & & \\ & Q & & & \\ & & \ddots & & \\ & & & Q & \\ & & & & P \end{bmatrix}, \qquad \tilde{R} \overset{\text{def}}{=} \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix},$$

$$\tilde{A} \overset{\text{def}}{=} \begin{bmatrix} Id \\ A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}, \qquad \tilde{B} \overset{\text{def}}{=} \begin{bmatrix} 0 & & & \\ B & & & \\ AB & B & & \\ \vdots & \ddots & \ddots & \\ A^{N_p-1}B & \cdots & AB & B \end{bmatrix},$$

$$\tilde{M} \overset{\text{def}}{=} \begin{bmatrix} M & & & & 0 \\ & M & & & 0 \\ & & \ddots & & \vdots \\ & & & M & 0 \end{bmatrix}, \qquad \tilde{N} \overset{\text{def}}{=} \begin{bmatrix} N & & & \\ & N & & \\ & & \ddots & \\ & & & N \end{bmatrix}, \qquad \tilde{l} \overset{\text{def}}{=} \begin{bmatrix} l \\ l \\ \vdots \\ l \end{bmatrix}$$

Which are of the following dimensions: $\tilde{x} \in \mathbb{R}^{(N_p+1)\cdot n_x}$, $\tilde{u} \in \mathbb{R}^{N_p \cdot n_x}$, $\tilde{Q} \in \mathbb{R}^{(N_p+1)\cdot n_x \times (N_p+1)\cdot n_x}$, $\tilde{R} \in \mathbb{R}^{N_p \cdot n_u \times N_p \cdot n_u}$, $\tilde{A} \in \mathbb{R}^{(N_p+1)\cdot n_x \times n_x}$, $\tilde{B} \in \mathbb{R}^{(N_p+1)\cdot n_x \times N_p \cdot n_u}$, $\tilde{M} \in \mathbb{R}^{N_p \cdot n_c \times (N_p+1)\cdot n_x}$, $\tilde{N} \in \mathbb{R}^{N_p \cdot n_c \times N_p \cdot n_u}$, $\tilde{l} \in \mathbb{R}^{N_p \cdot n_c}$. Using these augmented quantities, the linear parametric Quadratic Program can be reformulated to the following form:

$$\min_{\tilde{u}} \quad \frac{1}{2}\tilde{u}'\tilde{H}\tilde{u} + \tilde{u}'\tilde{F}\tilde{x}_0 \tag{A-4a}$$
$$\text{subject to} \quad \tilde{G}\tilde{u} \geq \tilde{l} - \tilde{E}\tilde{x}_0 \tag{A-4b}$$

In which $\tilde{H} \overset{\text{def}}{=} \tilde{B}'\tilde{Q}\tilde{B} + \tilde{R} \in \mathbb{R}^{N_p \cdot n_u \times N_p \cdot n_u}$, $\tilde{F} \overset{\text{def}}{=} \tilde{B}'\tilde{Q}\tilde{A} \in \mathbb{R}^{N_p \cdot n_u \times n_x}$, $\tilde{G} \overset{\text{def}}{=} \tilde{M}\tilde{B} + \tilde{N} \in \mathbb{R}^{N_p \cdot n_c \times N_p \cdot n_u}$, $\tilde{E} \overset{\text{def}}{=} \tilde{M}\tilde{A} \in \mathbb{R}^{N_p \cdot n_c \times n_x}$ The problem as described by (A-4) is then called the condensed form of the original QP problem of (4-1).

Bibliography

[1] *Vehicle Dynamics Terminology. SAE-J670-2.* Society of Automotive Engineers, 2008. [Online]. Available: https://doi.org/10.4271/J670_200801

[2] R. Quirynen, "Numerical Simulation Methods for Embedded Optimization," PhD thesis, KU Leuven, 2017.

[3] T. Tjin-A-Tsoi, "Tijdwinst in spitsverkeer van en naar de grote steden," in *CBS, Transport en mobiliteit 2015*, 1st ed. Den Haag: Centraal Bureau voor de Statistiek, 2016, ch. 11, pp. 136–145.

[4] W. Mitchell, C. Borroni-Bird, and L. Burns, *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*, 1st ed. Cambridge: The MIT Press, 2010.

[5] E. Guerra, "Planning for Cars That Drive Themselves: Metropolitan Planning Organizations, Regional Transportation Plans, and Autonomous Vehicles," *Journal of Planning Education and Research*, vol. 36, no. 2, pp. 210–224, 2016.

[6] "Automated Transit Systems | 2getthere." [Online]. Available: https://www.2getthere.eu/

[7] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C*, vol. 60, pp. 416–442, 2015.

[8] *Automated People Mover Standards. ANSI/ASCE/T&DI 21-13.* American Society of Civil Engineers, 2013. [Online]. Available: https://doi.org/10.1061/9780784412985

[9] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. SAE-J3016.* Society of Automotive Engineers, 2016. [Online]. Available: https://doi.org/10.4271/J3016_201609

[10] N. J. Brouwer, "Collision-free predictive trajectory generation for four-wheel-steered mini-buses," Literature survey, TU Delft, 2017.

[11] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[12] E. Masehian and D. Sedighizadeh, "Classic and Heuristic Approaches in Robot Motion Planning - A Chronological Review," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 1, no. 5, pp. 228–233, 2007.

[13] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[14] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," *IEEE International Conference on Robotics and Automation*, pp. 2061–2067, 2012.

[15] D. González, J. Pérez, R. Lattarulo, V. Milanés, and F. Nashashibi, "Continuous curvature planning with obstacle avoidance capabilities in urban scenarios," in *IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 1430–1435.

[16] K. Maček, M. Becker, and R. Siegwart, "Motion planning for car-like vehicles in dynamic urban scenarios," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4375–4380, 2006.

[17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[18] V. Jain and T. Weiskircher, "Prediction-based hierarchical control framework for autonomous vehicles," *International Journal of Vehicle Autonomous Systems*, vol. 12, no. 4, pp. 307–333, 2014.

[19] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*, 1st ed. Cambridge University Press, 2017.

[20] S. Arrigoni and F. Cheli, "MPC-based framework for autonomous ground vehicles in a complex environment," in *The Dynamics of Vehicles on Roads and Tracks*, 2016, pp. 669–678.

[21] J. Frasch, A. Gray, M. Zanon, H. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," *European Control Conference*, pp. 4136–4141, 2013.

[22] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, p. 1, 2014.

[23] M. Mitschke, "Das Einspurmodell von Riekert-Schunck," *Atz*, vol. 107, no. 11, pp. 1030–1031, 2005.

[24] B. Heißing and M. Ersoy, *Chassis Handbook*. Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, 2011.

[25] M. Mitschke and H. Wallentowitz, *Dynamik der Kraftfahrzeuge*. Springer Fachmedien Wiesbaden, 2014. [Online]. Available: http://doi.org/10.1007/978-3-662-06802-1

[26] R. Rajamani, *Vehicle Dynamics and Control*. Springer, Boston, MA, 2012. [Online]. Available: https://doi.org/10.1007/978-1-4614-1433-9

[27] A. Schmeitz, J. Zegers, J. Ploeg, and M. Alirezaei, "Towards a generic lateral control concept for cooperative automated driving theoretical and experimental evaluation," *IEEE International Conference on Models and Technologies for Intelligent Transportation Systems*, pp. 134–139, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8005653/

[28] T. Weiskircher and B. Ayalew, "Predictive trajectory guidance for (semi-)autonomous vehicles in public traffic," *Proceedings of the American Control Conference*, pp. 3328–3333, 2015.

[29] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Parallel Autonomy in Automated Vehicles: Safe Motion Generation with Minimal Intervention," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 1928–1935.

[30] H. J. Ferreau, "Model Predictive Control Algorithms for Applications with Millisecond Timescales," PhD thesis, KU Leuven, 2011.

[31] B. Houska, H. Ferreau, M. Vukov, and R. Quirynen, "ACADO Toolkit User's Manual." [Online]. Available: http://www.acadotoolkit.org

[32] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.

[33] G. H. Bock and J. K. Pitt, "A Multiple Shooting Algorithm for Direct Solution of Optimal control Problems," *Ninth IFAC World Congress*, vol. 17, no. 2, pp. 1603–1608, 1984. [Online]. Available: http://dx.doi.org/10.1016/S1474-6670(17)61205-9

[34] A. Domahidi and J. Jerez, "FORCES Professional. embotech GmbH," 2014. [Online]. Available: http://embotech.com/FORCES-Pro

[35] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[36] H. J. Ferreau, "An Online Active Set Strategy for Fast Solution of Parametric Quadratic Programs with Applications to Predictive Engine Control," MSc thesis, Ruprecht-Karls-Universität Heidelberg, 2006.

[37] D. Kouzoupis, R. Quirynen, J. V. Frasch, and M. Diehl, "Block Condensing for Fast Nonlinear MPC with the Dual Newton Strategy," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 26–31, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ifacol.2015.11.258

[38] "PreScan | TASS International." [Online]. Available: https://www.tassinternational.com/prescan

[39] "CarSim | Mechanical Simulation Corporation." [Online]. Available: https://www.carsim.com/

Glossary

List of Acronyms

2GT	2getthere
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
ADS	Automated Driving System
ADS-DV	ADS-Dedicated Vehicle
AIR	Actor Information Receiver
APM	Automated People Mover
ASCE	American Society of Civil Engineers
CoG	Center of Gravity
DDT	Dynamic Driving Task
GRT	Group Rapid Transport
KKT	Karush-Kuhn-Tucker
LP	Linear Program
LPV	Linear Parameter Varying
MDP	Markov Decision Processes
MPC	Model Predictive Control
NLP	Nonlinear Program
OCP	Optimal Control Problem
QP	Quadratic Program

RHC	Receding Horizon Control
RRT	Rapidly-Expanding Random Tree
RT	real-time
RTI	Real-Time Iteration
SAE	Society of Automotive Engineers
SIL	Software in the Loop
SQP	Sequential Quadratic Program

List of Symbols

Optimal Control Problem (OCP) Symbols

J	OCP cost.
$K_\star$	Individual penalty on state $\star$.
N_p	Prediction steps.
T_p	Prediction time.
T_s	Sampling time.
$S_\star$	Ellipse semi-major axis for longitudinal ($_{\text{long}}$) and lateral ($_{\text{lat}}$) planner.
$E_\star$	Ellipse semi-minor axis for longitudinal ($_{\text{long}}$) and lateral ($_{\text{lat}}$) planner.
$L_\star$	Length of ego ($_{\text{ego}}$) or object ($_{\text{obj}}$).
$W_\star$	Width of ego ($_{\text{ego}}$) or object ($_{\text{obj}}$).
x	OCP state.
u	OCP input.
y	OCP output.

Group Rapid Transport (GRT) dynamic Symbols

β	The side slip or crab angle of the GRT.
$\delta_\star$	Wheel steering angle for front (f) and rear (r) wheel.
$\dot{\delta}_\star$	Wheel steering rate for front (f) and rear (r) wheel.
$\dot{\psi}$	The yaw rate of the GRT.
κ	Road curvature.
ρ	Cornering radius.
$C_\star$	Tire stiffness of the front (f) and rear (r) tire.
I_z	Vehicle rotational inertia.
L	Wheelbase; length from front to rear wheel.
U	Understeer/Oversteer gradient.
a_x	Acceleration in x direction of the vehicle reference frame, measured with respect to the global reference frame.

a_y	Acceleration in y direction of the vehicle reference frame, measured with respect to the global reference frame.
e_ψ	The yaw angle of the GRT with respect to the road reference frame.
e_y	Lateral offset of the GRT with respect to the road reference frame.
j_x	Jerk in x direction of the vehicle reference frame, measured with respect to the global reference frame.
j_y	Jerk in y direction of the vehicle reference frame, measured with respect to the global reference frame.
$l_\star$	Length from CoG to front (f) and rear (r) wheel.
m	Vehicle mass.
s	Travelled distance along the road.
v_x	Velocity in x direction of the vehicle reference frame, measured with respect to the global reference frame.
v_y	Velocity in y direction of the vehicle reference frame, measured with respect to the global reference frame.

Other Symbols

$\overline{\star}$	Upper bound on state.
$\underline{\star}$	Lower bound on state.
$\vec{e}^{\star}$	Frenet coordinate frame for the global (G), road (s), vehicle (v) and movement (m) of a GRT.