

Real-time receding horizon trajectory generation for long heavy vehicle combinations on highways

N.J. van Duijkeren



Real-time receding horizon trajectory generation for long heavy vehicle combinations on highways

MASTER OF SCIENCE THESIS

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

N.J. van Duijkeren

August 24, 2014

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

VOLVO

The work in this thesis was supported by Volvo Group Trucks Technology. Their cooperation is hereby gratefully acknowledged.



DCSC

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

Abstract

So-called long heavy vehicle combinations (LHVCs) are trucks up to 32 meters in length and 80 tonnes heavy. They have the potential to reduce greenhouse gas emissions, traffic congestions and transportation costs for road freight transport. LHVCs are widely used in Canada and Australia today, and their abundance on European roads is expected to increase in the near future. However, an undesired effect of the added towed units is the increase in difficulty to maneuver such trucks on roads and in busy traffic. The increasing complexity for truck drivers to handle trivial tasks, like changing lane, call for advanced assistance functions. The development of advanced driver assistance systems or potentially autonomous functioning trucks can improve traffic safety. This allows a further increase in use of long combination trucks. One crucial component of such a driver assistance system is the ability to plan a safe and smooth trajectory in real-time.

In this thesis, a nonlinear receding horizon trajectory generator is proposed for highway driving of an A-double combination type of LHVC. An optimal control problem (OCP) is formulated to define the open-loop constrained optimal trajectories. Optimality is defined as a trade-off between three main components. Firstly, the jerk levels perceived by the driver is to be minimized. Secondly, the lane center ought to be tracked and the velocity of the traffic flow should be followed. And finally, depending on the detected scenario, distance is maintained from a set of fellow road users. Hard constraints are imposed for the actuator limitations, to prohibit the truck to leave the lane boundaries and to limit lateral acceleration levels. Actuation signals are generated for the low-level steering control and a longitudinal velocity tracker. The prediction of the vehicle states is performed using a nonlinear single-track model of the A-double combination with the linear tire slip assumption.

The prediction horizon of the optimal trajectory is defined in the traveled distance along the lane center. This allows a natural definition of the road curvature, progressing linear in the prediction. A reformulation of the vehicle prediction model is executed for its incorporation in the OCP. Surrounding vehicles are assumed to drive constant velocity, a spatial prediction model is defined for the trajectories of other road users.

A direct multiple-shooting solution technique with a piecewise constant control parameterization is used to obtain a nonlinear program (NLP). Using the **ACADO Toolkit**, the so-called

Real-Time Iteration (RTI) scheme is implemented exploiting a constrained Gauss-Newton solution algorithm. Instead of solving the entire NLP each control interval, a sequential quadratic programming (SQP) technique is synchronized with the sampling of the controller. Each intermediate solution iterate of the SQP is used as a control reference to the vehicle. The solution strategy is separated into two distinct phases, a preparation step and a feedback step. Only the feedback step requires knowledge of the most recent state, a short duration of this step assures minimal feedback delay.

The solution strategy to the NLP is implemented in efficient stand-alone C/C++ code, interfaced with SIMULINK and the motion simulator of the Swedish National Road and Transport Research Institute (VTI) in Göteborg, Sweden. All routines are executed based on a clearly defined set of measurements on the vehicle, the road and the surrounding traffic.

The overall control algorithm is tested in closed-loop simulations on two A-double combination models. The vehicle prediction model (also employed in the NLP) and a high-fidelity vehicle model, provided and validated by Volvo Group Truck Technologies. Results of simulations are presented for lane changes, merging actions and evasive maneuvers on low-curvature highways. The trajectory generator successfully controls both plant models for the intended scenarios. Slight model-mismatch is observed between the prediction model and the high-fidelity plant, which has a limited deteriorating effect on the control performance. Execution times of the NLP solution strategy show that the trajectory generator implementation is real-time feasible.

In general we conclude that this work successfully demonstrates the applicability of the RTI-algorithm to control the A-double combination for highway maneuvering.

Table of Contents

Acknowledgements	xi
1 Introduction	1
1-1 Research goals and motivation	2
1-2 Thesis contributions	2
1-3 Structure of the thesis	3
2 Modeling of the vehicle, the road and surrounding traffic	5
2-1 General A-double combination vehicle dynamics characteristics	6
2-2 Nonlinear single track model of the A-double combination	7
2-3 Planar kinematic relations in the multi-body system	12
2-3-1 Exact kinematics	15
2-3-2 Simplified kinematics	16
2-4 Modeling of the vehicle position with respect to the road geometry	17
2-5 Spatial reformulation of the vehicle dynamics along the road geometry	18
2-6 Modeling of surrounding highway users	20
2-7 Description of the high-fidelity vehicle model	20
3 Formulation of the trajectory generation problem as an optimal control problem	21
3-1 Comfortable driving, safety considerations and actuator constraints	22
3-2 Definition of an optimal trajectory	23
3-2-1 Approximation of the lateral jerk	25
3-2-2 Lateral acceleration bounds	25
3-2-3 Lateral distance off-set bounds	26
3-2-4 Longitudinal velocity bounds	26
3-2-5 Steering actuation bounds	27
3-3 Alternative obstacle avoidance formulations	27

3-3-1	Naming convention of the obstacles	27
3-3-2	Prediction model for the distance to surrounding traffic	28
3-3-3	Collision avoidance keeping lane	29
3-3-4	Collision avoidance in a lane change	29
3-4	Formulation of the resulting optimal control problem	31
3-4-1	Prediction horizon	32
3-4-2	Cost function	33
3-4-3	Inequality constraints	33
3-4-4	Prediction model equality constraints	34
3-5	Summary	35
4	Direct multiple-shooting solution strategy of the optimal control problem	37
4-1	Introduction to the Real-Time Iteration scheme	38
4-2	Problem discretization and Nonlinear Program formulation	39
4-3	Preparation step in the Real-Time Iteration scheme	42
4-3-1	Definition of the road geometry	42
4-3-2	Shift previous SQP solution	43
4-3-3	Constrained Gauss-Newton algorithm	44
4-3-4	Problem condensing	45
4-4	Feedback step in the Real-Time Iteration scheme	45
4-4-1	Initial value embedding of state measurements	45
4-4-2	Quadratic programming and problem expansion	46
4-4-3	Assessment of solution quality	46
4-5	Summary	47
5	Real-time implementation of the trajectory generator	49
5-1	Simulation environments	50
5-2	Control hierarchy	52
5-3	Control scheduling and handling of delay	53
5-3-1	Selection of sampling rate	53
5-3-2	Scheduling	54
5-4	Reference calculation	55
5-4-1	Reference calculation for the longitudinal velocity	55
5-4-2	Reference calculation for the Lateral position	57
5-5	Code generation using the ACADO Toolkit	58
5-6	Implementation details	59
5-6-1	Decision making	59
5-6-2	Handling infeasibility	60
5-6-3	UDP communication	60
5-6-4	Considerations for embedded hardware application	61
5-7	Summary	61

6 Simulation results and discussion	63
6-1 Lane change maneuver	66
6-1-1 Simulations with single-track model	66
6-1-2 Simulations with high-fidelity plant	73
6-1-3 Effect of cost function tuning on lane change maneuver	79
6-2 Merging maneuver	81
6-2-1 Simulation with high-fidelity plant	81
6-3 Evasive return to lane maneuver	87
6-3-1 Simulation with the single-track model on a curved road	87
6-3-2 Simulation with the high-fidelity plant on a straight road	92
6-4 Lane keeping on increasing curvature road	96
6-5 Summary	99
7 Conclusions and recommendations for future work	101
7-1 Summary of the conclusions	101
7-2 Recommendations for future work	102
A Mathematical derivations	105
A-1 Lateral reference calculation	105
Bibliography	111
Glossary	115
List of Acronyms	115
List of Symbols	116

List of Figures

1-1	LHVC in the A-double configuration (tractor - semi-trailer - dolly - semi-trailer).	1
2-1	The A-double combination is a truck that consists of four distinct units. The tractor, a semi-trailer, the converter dolly and a second semi-trailer. The orientation of the A-double combination in the picture clearly shows how the four units are interconnected.	5
2-2	Illustration of high speed off-tracking for an A-double combination. The swept path of the tractor in a curve is different from the swept path of the other units. Note that at low speeds, inward off-tracking occurs.	7
2-3	Parameters of the A-double dynamic model.	8
2-4	Illustration of the physical tire behavior for the lateral force as a function of the slip angle and a linear approximation of this curve.	9
2-5	States of the A-double dynamic model.	10
2-6	Kinematics for the A-double combination. The green arrows indicate the vectors from the origin of the global coordinate system to the axles of the A-double combination. The orange arrows represent the same coordinates relative to the COM of the tractor.	14
2-7	A visualization of the Frenét frames constructed at the front axle of the tractor and the rear axle of the second semi-trailer.	18
3-1	Illustration of the lateral distance bounds for the axles, corrected for the width of the truck. The center line of the road is equal to zero lateral distance offset.	26
3-2	Indication of the prohibited regions (indicated in red) in a lane keeping scenario or when no lane change is possible. A hyperplane, at distance $\Delta s_{o,2}$ moves with the obstacle 2 over the prediction horizon. Note that the left road boundary constraint is not visualized.	29
3-3	In the rectangular formulation of the predicted collision-free corridor the feasible area is bounded by obstacle 2, 3 and 4, which each introduce a prohibited half space. The lateral distance constraint is temporarily relaxed to allow the truck to reside in two lanes.	30
3-4	Half-spaces shaped by sigmoid support functions.	31
4-1	The RTI is designed to provide feedback from the OCP as soon as possible after a measurement is taken from the dynamic system.	39

4-2	An illustration of the scenario when a the distance between the Frenét frames for the front axle of the tractor and the rear axle of the second semi-trailer are not a truck length apart.	43
5-1	Graphical overview of SIMULINK implementation of the trajectory generator. The nonlinear model predictive control (NMPC), written in C/C++, is interfaced with the SIMULINK models of the A-double combination using am S-function wrapper.	50
5-2	Graphical overview of the implementation at the VTI motion simulator. The NMPC is interfaced with the motion support and A-double combination simulation platform via a UDP connection. The controller implementation runs as a stand-alone application on a dedicated PC.	51
5-3	Control hierarchy of the trajectory generator in closed-loop with the A-double combination.	52
5-4	The control step is applied asynchronously from the measurement steps. For measurement step s_k , an optimal trajectory is computed. Δs_{fb} later, the time/space consumed by the feedback step of the solution algorithm, the control action is applied.	54
5-5	Illustration of the sensitivity areas to actions of the car that is being followed. At each time instant, the depicted figure represents a decision diagram. The velocity reference is defined by the projection parallel to velocity reference axis of the coordinate of by the current velocity and distance with the leading vehicle onto the graph. In the invariant region, the velocity reference is kept the same as the current velocity.	57
6-1	Architectural diagrams of signals for the two vehicle models in SIMULINK.	65
6-2	Results from a lane change on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	68
6-3	Results from a lane change on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	69
6-4	Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	71
6-5	Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	72
6-6	Results from a lane change on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	74
6-7	Results from a lane change on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	75
6-8	Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	77
6-9	Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	78
6-10	A comparison of a lane change maneuver with four different cost function settings. The results are from closed-loop simulations with the single-track model.	80

6-11 Results from a merging scenario to the right, the trajectory generator is in closed-loop with the high-fidelity model.	84
6-12 Results from a merging scenario to the right, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	85
6-13 Results from a merging scenario to the right, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	86
6-14 Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	89
6-15 Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	90
6-16 Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.	91
6-17 Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	93
6-18 Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	94
6-19 Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.	95
6-20 Illustration of the road profile for the simulation with the high-fidelity model on an increasing curvature road. The arrow indicates the driving direction.	96
6-21 Results from lane keeping on a constantly increasing road curvature. Trajectory generator is in closed-loop with high-fidelity plant model.	97
6-22 Results from lane keeping on a constantly increasing road curvature. Trajectory generator is in closed-loop with high-fidelity plant model.	98

Acknowledgements

First and foremost, I want to thank my supervisor Dr. Tamás Keviczky for his assistance during this thesis work. I was privileged to receive his guidance and support in regular meetings and discussions. There was never a lack of critical review of intermediate work in which no minor detail was left unnoticed. The dedication with which he supervised the work has set an example for my future career.

This thesis is conducted in affiliation with Volvo Group Truck Technologies. I wish to express my gratitude towards my daily supervisors from Volvo. The continuous support and trust from Dr. Leo Laine and Peter Nilsson have had a great contributions to this work and this is hereby kindly acknowledged. I eagerly used their willingness to provide help and to review my work on multiple occasions. Additionally I would like to thank Dr. Peter Lindroth, he always found the time to help me when I had yet another *optimization*-related question.

Additionally, I would like to thank Bruno Augusto, employee at the Swedish National Road and Transport Research Institute (VTI). He joined many regular meetings far before the hardware implementation became relevant and has provided useful input. During the controller implementation at VTI, working with him felt as a streamlined process I greatly enjoyed. His skills in control engineering and programming are to be attributed for that.

Finally, I would like to acknowledge the direct and indirect input from the *Chassis Strategy & Vehicle Analysis* group at Volvo. I have taken full advantage of the facilities that were available at Volvo Group Truck Technologies, it is a luxury to be able to take accurate models of dynamic systems for granted.

Delft, University of Technology
August 24, 2014

N.J. van Duijkeren

“For every complex problem there is an answer that is clear, simple and wrong.”

— *H.L. Mencken*

Chapter 1

Introduction

The so-called long heavy vehicle combinations (LHVCs) are a new generation of trucks that are widely used in Canada and Australia today, and will be more abundant on the road also in Europe in the near future [1, 2]. They have the potential to decrease road transport cost, traffic congestions and generate lower emissions than current road freight transport. However, an undesired effect of the added towed units is the increase in difficulty to maneuver these trucks on roads and in busy traffic. The increasing complexity for truck drivers to handle trivial tasks, like changing lane, call for advanced assistance functions. The development of advanced driver assistance systems, or potentially autonomous functioning trucks, can improve traffic safety, allowing a further increase in use of long combination trucks.

The main goal of this thesis work is to develop, implement and evaluate a real-time optimization-based trajectory planning algorithm for the A-double combination type of LHVCs. In Figure 1-1, an illustration of such a truck is depicted. A-double combinations are trucks of 32[m] in length and can have a total weight of 80[t]. The trajectories for the A-double combination must be generated such that the vehicles operate safely and maximizes for driver comfort. The trajectory generator should be able to perform basic highway maneuvers on medium curvature high friction roads. Among these basic maneuvers are, staying inside the lane, changing lane, but also to perform emergency maneuvers.

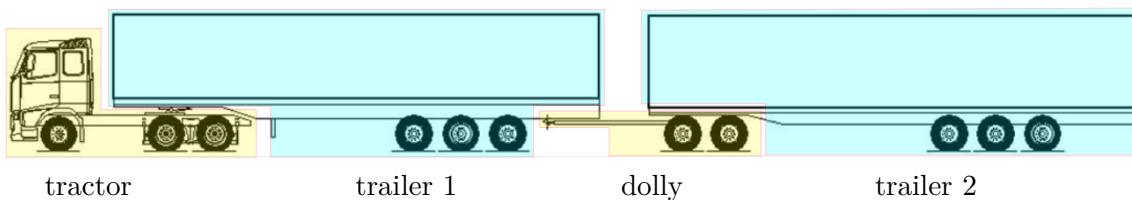


Figure 1-1: LHVC in the A-double configuration (tractor - semi-trailer - dolly - semi-trailer).

1-1 Research goals and motivation

From a safety perspective, there are concerns regarding an increased abundance of LHVCs in traffic. The longer vehicle combinations are more difficult to handle than conventional truck combinations. Also, a relation is found between longer vehicles and an increased involvement rate of these vehicles in highway lane change collisions[3]. The increased handling difficulty can be attributed to several lateral vehicle characteristics. Among which most prominently the phenomena of off-tracking and rearward amplification (RWA) [4]. Off-tracking is the effect where the towing and the towed units do not necessarily travel the same paths. At low speeds the rearmost unit tends to cut the corner, whereas at high speeds to rear units swing out. Unsurprisingly, this can lead to dangerous scenarios for both the truck and the surrounding traffic. Secondly, RWA is the effect where lateral accelerations amplify through the towed units. Strong RWA increases the risk of hitting other objects, or (more severely) that one of the units starts to roll over.

Previous work has shown promising results for path planning of long heavy vehicle combination (LHVC) using optimization-based receding horizon control [5]. Efficiently formulating the trajectory generation problem as a nonlinear program (NLP) and using state-of-art solvers can solve the trajectory generation problem in real-time. However, it also showed that a too low complexity kinematic vehicle model can obstruct stabilizing performance when applied to a high fidelity plant. Especially in high speed scenarios, the prediction of the kinematic vehicle model insufficiently represented the dynamic behavior. External disturbances, such as bad road conditions and measurement noise present in real-life, will not help. This motivated to perform additional research to improving closed-loop performance of a receding horizon controller for the trajectory generation problem.

Recently, great progress in receding horizon control has taken place in real-time applicability. Especially, promising results in the application of the Real-Time Iteration (RTI) algorithm [6] bring real-time optimization based trajectory generation a step closer to reality [7, 8]. This thesis work seeks a great part of the real-time performance in efficient mathematical solution strategies and the advantages of code-generation techniques. This allows the use of a non-linear dynamic prediction model of the LHVC, including a model for linear tire slip [9].

In this thesis work the applicability of the RTI-algorithm and code generation is researched in application to trajectory generation for A-double combinations on highways. The code generation functionality of the **ACADO Toolkit** [10] facilitates the streamlined implementation of the solution scheme to the optimal control problem (OCP). The goal of the research is to find whether real-time performance can be achieved and to assess closed-loop behavior on the high-fidelity model of the A-double combination developed by Volvo Group Truck Technologies (Volvo GTT). A suitable formulation for the OCP is designed to incorporate collision avoidance. Perfect knowledge of the vehicle state, surrounding traffic and the road geometry are assumed to be available and is used accordingly.

1-2 Thesis contributions

The application of model predictive control (MPC) control techniques in fast mechanical systems is, despite advances in computational abilities of computer systems, limited because of

real-time performance issues. The work in this thesis applies an existing state-of-art optimization algorithm to a complex system like the A-double combination, while assuring real-time applicability of the overall control algorithm. A receding horizon trajectory generator with a medium length prediction length ($\sim 100[m]$) is shown to be able to stabilize a high-fidelity model of the A-double combination in typical maneuvers on low curvature highways.

Collision avoidance is incorporated in the OCP by formulating a predicted collision-free corridor and by applying an incentive on distance keeping for surrounding vehicles in the optimal trajectory.

A real-time implementation of the trajectory generator at the Swedish National Road and Transport Research Institute (VTI) motion simulator enables near future research in driver acceptance of this type of advanced drivers assistance systems. Input from professional drivers can be used to place the results in perspective with the overall complexity of driving a truck. It can be researched if this type of driver assistance systems meet the expectations and improves handling for truck drivers of LHVCs.

This work presents a proof-of-concept for the application of the RTI to maneuver LHVCs on medium curvature highways.

1-3 Structure of the thesis

In Chapter 2 modeling of the vehicle, road and surrounding traffic is introduced. The derivation of the nonlinear single track model of the A-double combination is summarized from [9]. The spatial reformulation of the truck dynamics are described and one finds details on approach for modeling surrounding traffic.

Chapter 3 focuses on the definition of an optimal trajectory, desired driving behavior and considerations regarding constraints. Ultimately, the full-sized OCP used in this work is described.

Chapter 4 describes the solution strategy to the OCP. This is of particular interest to reader to understand the merits and drawbacks of the RTI algorithm and how they may affect the results.

Chapter 5 presents a brief description on the implementation details of the controller. The simulation environments are described and the main algorithmic components of the overall control algorithm are introduced.

Chapter 6 presents the numerical results of the closed-loop performance of the trajectory generator on both the single-track model as the high-fidelity plant of the LHVC. The main functionalities of the truck, to be able to keep lane, change lane and to perform evasive maneuvers are tested and discussed.

Finally, in Chapter 7 the conclusions and contributions of the thesis are summarized and recommendations for future work are listed.

Chapter 2

Modeling of the vehicle, the road and surrounding traffic

Long heavy vehicle combinations are trucks that consist of a chain of trailers. Several variants are actively in use today in multiple countries in Europe, among which in Sweden and the Netherlands. Despite efforts on harmonization of regulations, each country has different size and weight limitations [1]. Shorter combinations are in the range of 19[m] long, whereas the longest double trailer trucks are up to 32[m] in length. This thesis work is focused on the implementation of a trajectory generator for the A-double combination. This long heavy vehicle combination (LHVC) is the longest for use in the near future, with a total length of 32[m] and with a maximum mass of 80[t]. It consists of four separate units, the tractor, a semi-trailer, a converter dolly and a second semi-trailer. In Figure 2-1 one finds a picture of such a vehicle in which the degrees of freedom in the chain are clearly visible. In this chapter, the A-double combination is introduced in more detail.



Figure 2-1: The A-double combination is a truck that consists of four distinct units. The tractor, a semi-trailer, the converter dolly and a second semi-trailer. The orientation of the A-double combination in the picture clearly shows how the four units are interconnected.

In Section 2-1 general vehicle dynamics properties of the A-double combination are reviewed. The purpose is to give the reader a broad idea on phenomena such as off-tracking and rearward amplification (RWA).

Secondly, the trajectory generator developed in this thesis work is a model-based control technique. In Section 2-2, the model that is used in this control technique is introduced. This model is derived in previous work at Volvo Group Truck Technologies (Volvo GTT) [9], for a full detailed derivation one is referred to that technical report. In the section, the derivation steps are briefly outlined and simplifying assumptions are summarized.

In Section 2-3, a set of kinematic relations in the vehicle are derived. Velocity vectors and acceleration vectors in arbitrary points on the truck can be computed as a function of the vehicle states. Simplifications are applied to decrease complexity of the expressions, maintaining accuracy when applied to highway driving conditions.

Section 2-4 is focused on the modeling of the road. New states are introduced to the vehicle model that define the position of the truck with respect to the road.

In Section 2-5 the vehicle dynamics of the single-track model are reformulated from dynamics in time to dynamics to evolve with the traveled distance along the lane center. The need for this reformulation is discussed in more detail in Chapter 3.

There are a few important remarks regarding the notation in this chapter:

- The body-fixed coordinate system is defined according to the ISO-8850 norm. A definition of the coordinates and states are introduced in this chapter.
- A state $*$ in the global earth-fixed reference frame is denoted as G* .
- In this thesis work, five local reference frames are used. Consider an arbitrary state $*$, it can be defined in the body-fixed reference frame in:
 - The center of mass (COM) of the tractor, L_1* .
 - The center of the front axle of the tractor, ${}^L_{1f}*$.
 - The center of the rear axle of the first semi-trailer, L_2* .
 - The center of the rear axle of the dolly, L_3* .
 - The center of the rear axle of the second semi-trailer, L_4* .
- All time derivatives $d*/dt$ are performed in the reference frame in which a state is defined. Note that this has important effects on the physical meaning of a time derivative in a non-inertial reference frame, such as the body-fixed coordinate system in the truck.

2-1 General A-double combination vehicle dynamics characteristics

The A-double combination has specific characteristics in the dynamic behavior. These specific phenomena are introduced as performance characteristics in [4]. The report discusses longitudinal and lateral performance characteristics. Let us first discuss the lateral characteristics.

- **Off-tracking:** Off-tracking is the lateral offset between the paths of the center of the front axle of the tractor and the other axles in the combination. There is a distinction between high-speed off-tracking and low-speed off-tracking. In Figure 2-2 high-speed

off-tracking is illustrated, the swept path of the rear axle of second semi-trailer is wider than the path of the tractor. At low speeds, typically below $40[km/h]$ inward off-tracking occurs. In case of transient lateral motion, such as a lane change, transient off-tracking takes place. Due to swing, the swept path of the second semi-trailer can be different from the tractor its path.

- **Rearward Amplification:** RWA is the ratio between the maximum lateral acceleration in the towed units and the lateral acceleration in the tractor during a specific maneuver, such as a lane change. This ratio increases towards the rear of the truck. For gentle maneuvering, this RWA is typically very low. However, previous work [1] reports values up to 1.74 for RWA in the A-double combination can be expected.

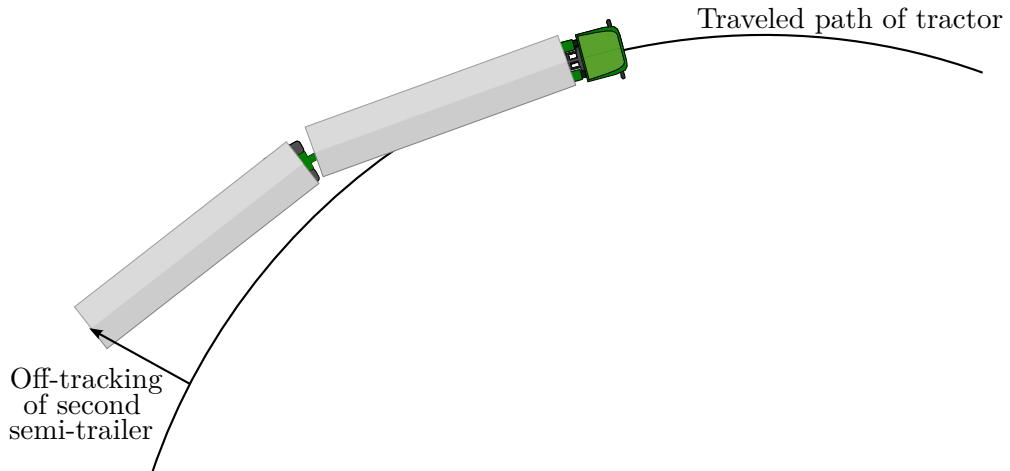


Figure 2-2: Illustration of high speed off-tracking for an A-double combination. The swept path of the tractor in a curve is different from the swept path of the other units. Note that at low speeds, inward off-tracking occurs.

The A-double combination usually has a powerful towing unit. Nevertheless, the longitudinal acceleration of a fully loaded truck is very limited. On a flat road, the acceleration is not higher than $0.25m/s^2$. This is especially relevant in case the truck drives up a hill. The maximum velocity is then limited by the slope of hill that is climbed. In this work, only a flat road is considered. Hence, this issue is not dealt with.

2-2 Nonlinear single track model of the A-double combination

The basis for the prediction model for the model-based control strategy is a single-track vehicle model for the lateral dynamics of the A-double combination. In previous work at Volvo Trucks, such a model has been derived and identified. In this thesis, for the sake of completeness, the derivation of the gray-box model and the parametric identification steps are briefly outlined. For a thorough documentation, one is referred to technical report in [9].

Alternative dynamic vehicle models are for example two-track models [11] and black-box models obtained using system identification techniques. A single-track model is expected to provide a good trade-off in complexity and accuracy for highway driving. A gray-box model is ideally suitable for different scenarios of e.g. the payload that is transported. A kinematic model is considered unsuitable for the use-case of highway driving because of the significant lateral tire slip and inertial effects that are neglected in the derivation of the kinematic vehicle model.

In the field of vehicle dynamics, a single-track model (also known as a bicycle model) refers to a special type of vehicle model. The basic properties of the single-track model are:

- The vehicle model is defined in a top-view 2D Euclidean space. 3D effects, such as roll-over are not modeled.
- The wheels of the vehicle are assumed to be lumped into the center of each axle.
- Each unit of the vehicle is a rigid body mass.
- No suspension, chassis and cabin dynamics are modeled.

Figure 2-3 depicts an illustration of the single track model of the A-double combination and the geometric properties.

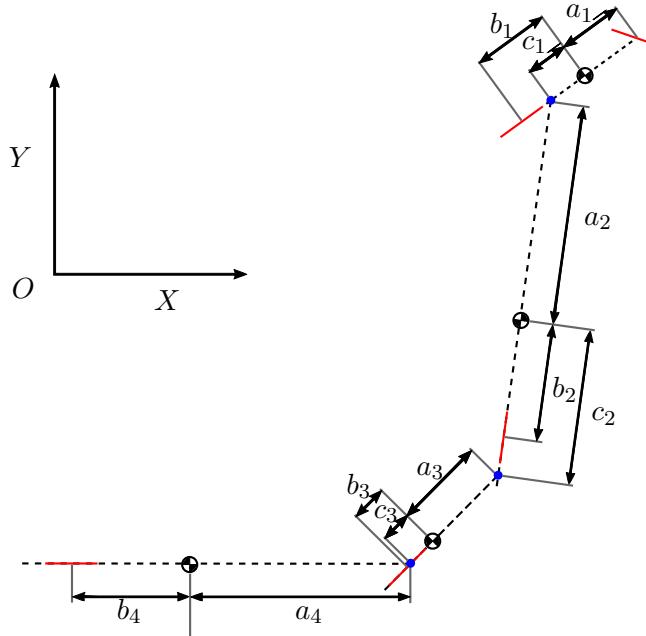


Figure 2-3: Parameters of the A-double dynamic model.

Tire properties have great influence on the vehicle dynamics. In fact, by changing the tire properties one can render a stable vehicle unstable. Modeling of tires is not a trivial task, lots of research is spent on an accurate representation of physical tires [12]. Fortunately, when tires are used conservatively (i.e. far within their performance bounds) their steady-state

lateral force behavior can be modeled as linear springs in the slip angle of the tire. Where the slip angle is the difference between the orientation of the tire and its velocity vector, i.e.

$$\text{slip angle} = \arctan \left(\frac{\text{lateral velocity of tire}}{\text{longitudinal velocity of tire}} \right) \quad (2-1)$$

In Figure 2-4 an illustration is given of the linear tire model approximation. Note that physical tire behavior heavily depends on the normal force on the tire, the temperature of the tire, the suspension geometry, the road conditions, etc. Because of the large influence of tires, this affects the validity of the resulting vehicle dynamics model.

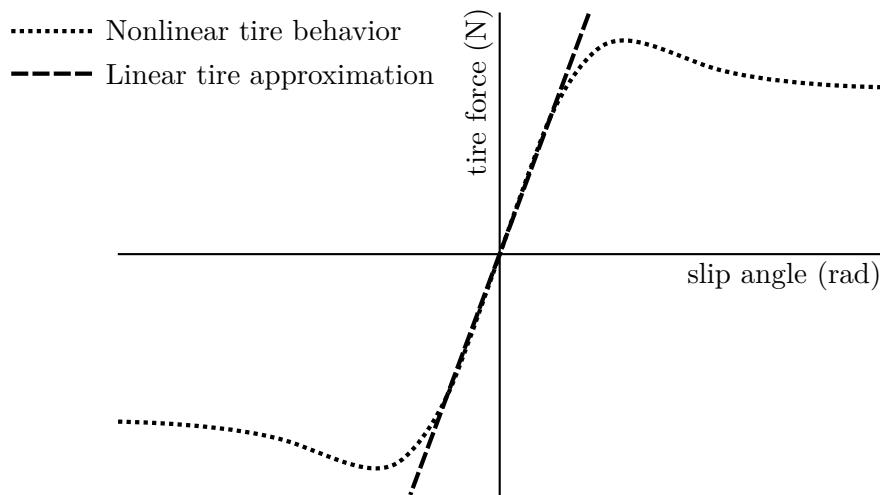


Figure 2-4: Illustration of the physical tire behavior for the lateral force as a function of the slip angle and a linear approximation of this curve.

The gray box model is derived using an Euler-Lagrange approach. The equations of motion are derived using a description of the kinetic energy \mathcal{T} , the potential energy \mathcal{V} and the generalized forces \mathcal{Q} in a chosen set of generalized coordinates r .

$$\frac{d}{dt} \frac{\partial(\mathcal{T} - \mathcal{V})}{\partial \dot{r}_i} - \frac{\partial(\mathcal{T} - \mathcal{V})}{\partial r_i} = \mathcal{Q}_i \quad (2-2)$$

$\forall i$, with r_i the i 'th element from the set of generalized coordinates:

$$r = (X \ Y \ \theta_0 \ \theta_1 \ \theta_2 \ \theta_3) \quad (2-3)$$

An illustration of the generalized coordinates is depicted in Figure 2-5.

The generalized forces entail the effect of the tire forces. No friction and air drag is modeled, neither is the effect of tilt or a slope in the road. Hence, no potential energy is present in the system, i.e. $\mathcal{V} = 0$. The Euler-Lagrange equations simplify to:

$$\frac{d}{dt} \frac{\partial \mathcal{T}}{\partial \dot{r}_i} - \frac{\partial \mathcal{T}}{\partial r_i} = \mathcal{Q}_i \quad (2-4)$$

The generalized coordinates are written in the global coordinates system (X, Y) . It is desired to obtain a model description in the local body-fixed vehicle frame in the COM of the tractor,

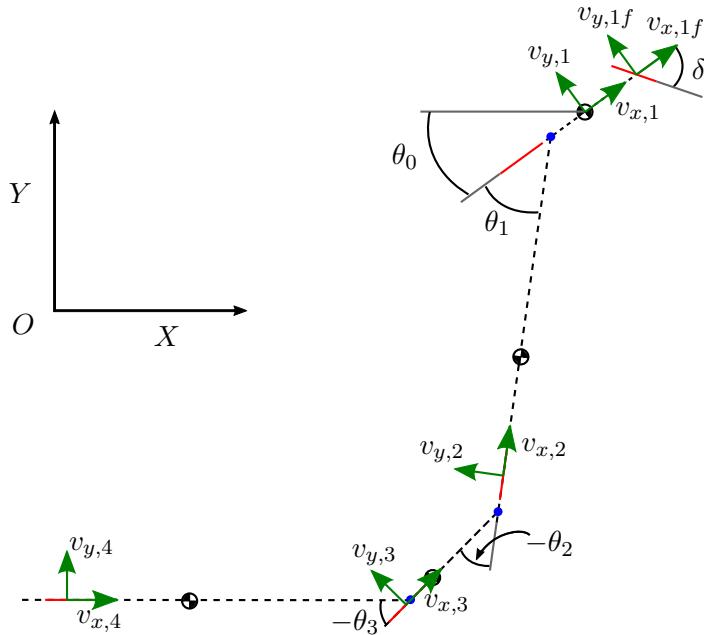


Figure 2-5: States of the A-double dynamic model.

independent from the yaw angle θ_0 and the position (X, Y) . For the three generalized coordinates $(X \ Y \ \theta_0)$, the following relation is used for the partial derivatives of the kinetic energy:

$$\frac{\partial \mathcal{T}}{\partial \dot{X}_G} = \frac{\partial \mathcal{T}}{\partial v_{x,1}} \cos(\theta_0) - \frac{\partial \mathcal{T}}{\partial v_{y,1}} \sin(\theta_0) \quad (2-5a)$$

$$\frac{\partial \mathcal{T}}{\partial \dot{Y}_G} = \frac{\partial \mathcal{T}}{\partial v_{x,1}} \sin(\theta_0) + \frac{\partial \mathcal{T}}{\partial v_{y,1}} \cos(\theta_0) \quad (2-5b)$$

$$\frac{\partial \mathcal{T}}{\partial X} = \frac{\partial \mathcal{T}}{\partial G} = 0 \quad (2-5c)$$

$$\frac{\partial \mathcal{T}}{\partial \theta_0} = \frac{\partial \mathcal{T}}{\partial v_{x,1}} v_{y,1} - \frac{\partial \mathcal{T}}{\partial v_{y,1}} v_{x,1} \quad (2-5d)$$

Solving the system of equations in Eq. (2-4) and Eq. (2-5) leads to a highly non-linear model in nine states and the six inputs for the net longitudinal force on each axle. To simplify the obtained vehicle dynamics model from the Euler-Lagrange method, a set of assumptions is introduced.

- Small angle assumptions, i.e. small steering angle, small tire-slip angles, small articulation angles between subsequent units.
- The longitudinal acceleration is assumed to be zero.
- All longitudinal forces are zero. No aerodynamics are modeled, neither are brakes and propulsion systems.

These assumptions simplify the model greatly. In fact, the lateral dynamics turn linear for constant longitudinal velocity. And a parametric model for the lateral dynamics of the A-double combination is obtained in the following state vector ξ_{lat} and control input vector u_{lat} :

$$\xi_{\text{lat}} = \begin{pmatrix} v_{y,1} & \dot{\theta}_0 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 \end{pmatrix}^T \quad (2-6a)$$

$$u_{\text{lat}} = \delta \quad (2-6b)$$

The equations of motion for the lateral dynamics, written using the conventional state-space A , B matrices, are:

$$\dot{\xi}_{\text{lat}} = A(v_{x,1})\xi_{\text{lat}} + Bu_{\text{lat}} \quad (2-7)$$

The free parameters that remain in the model are the geometric, inertial and tire properties of the truck. The geometric and inertial properties are straightforward to obtain from the truck configuration details. The configuration mainly depends on the payload of the truck and the way it is distributed in the trailers.

The truck in this thesis work is considered to be fully loaded, casting up to a total mass of 80[t]. All details of the vehicle configuration properties are listed in table Table 2-2. The table of vehicle properties also lists values for the cornering stiffness of each axle in the vehicle model. These values cannot be computed without very precise knowledge of the properties of the tire. In the work of [9] the cornering stiffnesses are approximated using a system identification experiment. The identification technique is briefly described.

The measurements are made in an experiment with the high-fidelity A-double combination model described later in Section 2-7 of this chapter. The steering input in the experiment was a white noise signal. And the truck drove with a constant longitudinal velocity of 70[km/h]. The identification objective was to minimize the least-squares error of the yaw rate gain of each unit in the A-double combination on the interval of 0 – 2[Hz]. The yaw rate gain for vehicle unit i is defined as the yaw rate of unit i divided by the front axle steering wheel angle, i.e. $\sum_{j=0}^i \dot{\theta}_j / \delta$.

Table 2-1: Eigenvalues of the linearized single-track A-double combination model at different longitudinal velocities. The table is sorted per column on the real part of the eigenvalues.

$v_{x,1} = 8 \text{ [m/s]}$	$v_{x,1} = 12 \text{ [m/s]}$	$v_{x,1} = 16 \text{ [m/s]}$	$v_{x,1} = 20 \text{ [m/s]}$
$-13.68 \pm j2.50$	$-9.12 \pm j4.11$	$-6.84 \pm j4.54$	$-5.47 \pm j4.71$
-9.33	$-4.39 \pm j2.70$	$-3.2 \pm j3.62$	$-2.57 \pm j4.00$
$-6.86 \pm j0.69$	$-2.87 \pm j0.56$	$-2.13 \pm j1.96$	$-1.70 \pm j2.33$
$-1.46 \pm j0.34$	$-1.78 \pm j1.40$	$-1.44 \pm j1.98$	$-1.15 \pm j2.22$
-1.15			

System Eq. (2-7) its eigenvalues are outlined in Table 2-1 for different longitudinal velocities. We can conclude that the linearized system is stable for the selected set of longitudinal velocities. Also, we observe that the eigenvalues change for velocities. Take for example the fastest eigenvalue-pair from the columns of 8 [m/s] and 20 [m/s]. These are $-13.68 \pm j2.50$ [rad/s] and $-5.47 \pm j4.71$ [rad/s] respectively. This indicates that the velocity is an important parameter

in the model and motivates to keep the dynamics velocity dependent in the prediction model of the model-based controller. The system is characterized to be non-stiff, the eigenvalues are at most a factor 13 apart. No special regard to numeric stability for discrete integration techniques is expected to be necessary.

We desire to obtain a model that includes a simple description for the longitudinal dynamics. More precisely, we want to obtain a simple model for the closed-loop cruise controller dynamics in the A-double combination. The longitudinal closed-loop dynamics are approximated with a first order differential equation on the longitudinal acceleration of the truck,

$$\frac{d^2 v_{x,1}}{dt^2} = \frac{a_{x,1,\text{des}} - a_{x,1}}{\tau_{\text{cc}}} \quad (2-8)$$

The time constant τ_{cc} relates the longitudinal acceleration $a_{x,1}$ of the truck with the desired acceleration input $a_{x,1,\text{des}}$. The extended state and control input vectors include the cruise controller:

$$\xi_{\text{lhvc}} = \begin{pmatrix} v_{y,1} & \dot{\theta}_0 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & v_{x,1} & a_{x,1} \end{pmatrix}^T \quad (2-9a)$$

$$u_{\text{lhvc}} = \begin{pmatrix} \delta & a_{x,1,\text{des}} \end{pmatrix}^T \quad (2-9b)$$

To conclude this section, the author of [9] gives a clear specification for the intended validity region for the single-track model. The model is designed and identified for the velocity range of $30 - 80[\text{km}/\text{h}]$ and amplitudes below $1.5[\text{m}/\text{s}^2]$ for the lateral acceleration. No weight shift is modeled and effects of combined slip are neglected. A model discrepancy is expected in more aggressive maneuvers than specified above.

2-3 Planar kinematic relations in the multi-body system

The dynamic model of the A-double combination that is introduced in Section 2-2 allows us to predict the motion of the truck combination over time. The state-vector of the single-track model, can be used to find a large variety of measures in the chain of vehicle units. By deriving the kinematic relations in the truck, we can find e.g. the acceleration vector in any point of the truck. Although the optimal control problem (OCP) is not introduced yet, it is natural that, among others, the following measures valuable to know:

- The velocity vectors in the axles of the A-double combination.
- The acceleration vectors in the COM of the tractor and the axles of the A-double combination.

In this section, the steps to obtain these measures are briefly outlined. Since the expressions for the kinematic relations are very long, only position vectors are written down explicitly. Let us start with the introduction of the notation that is used in the kinematics. The vectors $\vec{r}_* \in \mathbb{R}^2$ denote the position in the global inertial reference frame, $\vec{r}'_* \in \mathbb{R}^2$ represent the position vectors with respect to the COM of the tractor. Where $* \in \{1, 1f, 2, 3, 4\}$ represents the so-called positions of interest. The points of interest are:

Table 2-2: Vehicle parameters for the configuration of the A-double combination that is used for the single-track model identification executed in [9], including the resulting cornering stiffnesses.

Parameter	Symbol	Value	Unit
Mass, unit 1	m_1	9841	[kg]
Mass, unit 2	m_2	33601	[kg]
Mass, unit 3	m_3	2700	[kg]
Mass, unit 4	m_4	33801	[kg]
Yaw moment of inertia, unit 1	J_1	$20 \cdot 10^3$	[kgm ² /rad]
Yaw moment of inertia, unit 2	J_2	$543 \cdot 10^3$	[kgm ² /rad]
Yaw moment of inertia, unit 3	J_3	$2 \cdot 10^3$	[kgm ² /rad]
Yaw moment of inertia, unit 4	J_4	$546 \cdot 10^3$	[kgm ² /rad]
Distance from COM to front axle, unit 1	a_1	1.45	[m]
Distance from front connection point to COM, unit 2	a_2	4.43	[m]
Distance from front connection point to COM, unit 3	a_3	4.55	[m]
Distance from front connection point to COM, unit 4	a_4	4.65	[m]
Distance COM to rear axle, unit 1	b_1	2.23	[m]
Distance COM to rear axle, unit 2	b_2	3.27	[m]
Distance COM to rear axle, unit 3	b_3	0.65	[m]
Distance COM to rear axle, unit 4	b_4	3.05	[m]
Distance COM to rear connection point, unit 1	c_1	1.95	[m]
Distance COM to rear connection point, unit 2	c_2	5.97	[m]
Distance COM to rear connection point, unit 3	c_3	0.00	[m]
Front axle cornering stiffness, unit 1	C_{1f}	$4.07 \cdot 10^5$	[N/rad]
Rear axle cornering stiffness, unit 1	C_{1r}	$2.07 \cdot 10^6$	[N/rad]
Rear axle cornering stiffness, unit 2	C_{2r}	$1.24 \cdot 10^6$	[N/rad]
Rear axle cornering stiffness, unit 3	C_{3r}	$1.17 \cdot 10^6$	[N/rad]
Rear axle cornering stiffness, unit 4	C_{4r}	$1.42 \cdot 10^6$	[N/rad]

- The COM of the tractor, indicated with $* = 1$
- The front axle of tractor, indicated with $* = 1f$.
- The rear axle of the first semi-trailer, indicated with $* = 2$.
- The rear axle of the dolly, indicated with $* = 3$.
- The rear axle of the second semi-trailer, indicated with $* = 4$.

See Figure 2-6 for an illustration of these points.

Note that these positions are chosen, such that we have a representative measure for the velocity and acceleration in all main units of the truck combination. In the remainder of the thesis, the notation $\{1, 1f, 2, 3, 4\}$ for the positions of interest is repeatedly used.

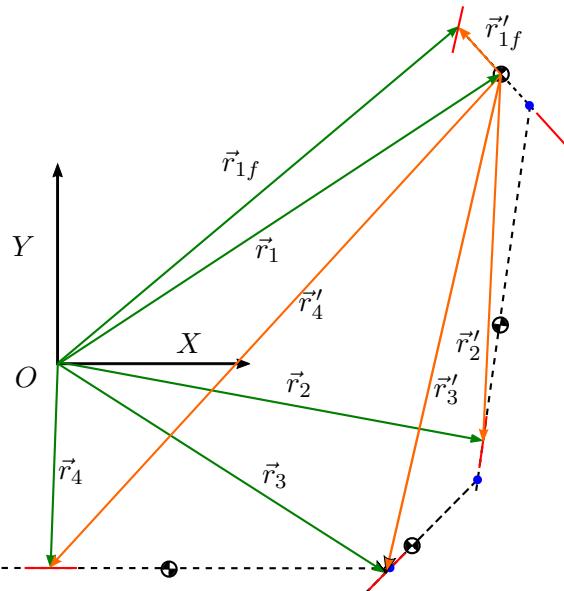


Figure 2-6: Kinematics for the A-double combination. The green arrows indicate the vectors from the origin of the global coordinate system to the axles of the A-double combination. The orange arrows represent the same coordinates relative to the COM of the tractor.

Since the yaw of each unit with respect to the global reference frame is highly relevant in the context of the kinematics, we define a short-hand notation of these states. To avoid discrepancy in the notation of the yaw angle of the different units, the $\phi_1 = \theta_0$ is introduced also.

$$\begin{aligned}\phi_1 &= \theta_0 \\ \phi_2 &= \theta_0 + \theta_1 \\ \phi_3 &= \theta_0 + \theta_1 + \theta_2 \\ \phi_4 &= \theta_0 + \theta_1 + \theta_2 + \theta_3\end{aligned}\tag{2-10}$$

Let us first start with the derivation of the exact kinematic relations for the points of interest. The expressions are general and valid for any arbitrary orientation of the truck.

2-3-1 Exact kinematics

The kinematic relations are derived using symbolic computation tool MATHEMATICA. The basis for the derivation of the velocity and acceleration vectors in certain points of the truck are the position vectors. The vehicle dynamics are derived in the COM of the tractor, we need to define the position vectors relative to that point. In Figure 2-6 the position vectors are illustrated, mathematically we write:

$$\begin{aligned}
 {}^G\vec{r}_{1f}(t) &= {}^G\vec{r}_1(t) + {}^G\vec{r}'_{1f}(t) = {}^G\vec{r}_1(t) + R(\phi_1) \begin{pmatrix} a_1 \\ 0 \end{pmatrix} \\
 {}^G\vec{r}_2(t) &= {}^G\vec{r}_1(t) + {}^G\vec{r}'_2(t) = {}^G\vec{r}_1(t) - R(\phi_1) \begin{pmatrix} c_1 \\ 0 \end{pmatrix} - R(\phi_2) \begin{pmatrix} a_2 + b_2 \\ 0 \end{pmatrix} \\
 {}^G\vec{r}_3(t) &= {}^G\vec{r}_1(t) + {}^G\vec{r}'_3(t) = {}^G\vec{r}_1(t) + R(\phi_1) \begin{pmatrix} c_1 \\ 0 \end{pmatrix} - R(\phi_2) \begin{pmatrix} a_2 + c_2 \\ 0 \end{pmatrix} + \\
 &\quad - R(\phi_3) \begin{pmatrix} a_3 + b_3 \\ 0 \end{pmatrix} \\
 {}^G\vec{r}_4(t) &= {}^G\vec{r}_1(t) + {}^G\vec{r}'_4(t) = {}^G\vec{r}_1(t) + R(\phi_1) \begin{pmatrix} c_1 \\ 0 \end{pmatrix} - R(\phi_2) \begin{pmatrix} a_2 + c_2 \\ 0 \end{pmatrix} + \\
 &\quad - R(\phi_3) \begin{pmatrix} a_3 + c_3 \\ 0 \end{pmatrix} - R(\phi_4) \begin{pmatrix} a_4 + b_4 \\ 0 \end{pmatrix}
 \end{aligned} \tag{2-11}$$

where $R(*)$ denotes the rotation matrix in 2D Euclidean space:

$$R(*) = \begin{pmatrix} \cos(*) & -\sin(*) \\ \sin(*) & \cos(*) \end{pmatrix} \tag{2-12}$$

The velocity vectors in the same points can be obtained using the knowledge of the local velocity vector in the COM of the tractor.

$$\begin{aligned}
 {}^G\vec{v}_{1f}(t) &= R(\phi_1) {}^L_i\vec{v}_1(t) + \frac{d}{dt} {}^G\vec{r}'_{1f}(t) \\
 {}^G\vec{v}_2(t) &= R(\phi_1) {}^L_i\vec{v}_1(t) + \frac{d}{dt} {}^G\vec{r}'_2(t) \\
 {}^G\vec{v}_3(t) &= R(\phi_1) {}^L_i\vec{v}_1(t) + \frac{d}{dt} {}^G\vec{r}'_3(t) \\
 {}^G\vec{v}_4(t) &= R(\phi_1) {}^L_i\vec{v}_1(t) + \frac{d}{dt} {}^G\vec{r}'_4(t)
 \end{aligned} \tag{2-13}$$

Where L_i stands for the body-fixed reference frame in the center of mass of the tractor. And by differentiating this expression to time we obtain expressions for the acceleration vectors in

each unit in the global inertial reference frame.

$$\begin{aligned} {}^G\vec{a}_{1f}(t) &= \frac{d}{dt} {}^G\vec{v}_{1f}(t) = \frac{d}{dt} [R(\phi_1)] {}^L\vec{v}_1(t) + R(\phi_1) \frac{d}{dt} {}^L\vec{v}_1(t) + \frac{d^2}{dt^2} {}^G\vec{r}'_{1f}(t) \\ {}^G\vec{a}_2(t) &= \frac{d}{dt} {}^G\vec{v}_2(t) = \frac{d}{dt} [R(\phi_1)] {}^L\vec{v}_1(t) + R(\phi_1) \frac{d}{dt} {}^L\vec{v}_1(t) + \frac{d^2}{dt^2} {}^G\vec{r}'_2(t) \\ {}^G\vec{a}_3(t) &= \frac{d}{dt} {}^G\vec{v}_3(t) = \frac{d}{dt} [R(\phi_1)] {}^L\vec{v}_1(t) + R(\phi_1) \frac{d}{dt} {}^L\vec{v}_1(t) + \frac{d^2}{dt^2} {}^G\vec{r}'_3(t) \\ {}^G\vec{a}_4(t) &= \frac{d}{dt} {}^G\vec{v}_4(t) = \frac{d}{dt} [R(\phi_1)] {}^L\vec{v}_1(t) + R(\phi_1) \frac{d}{dt} {}^L\vec{v}_1(t) + \frac{d^2}{dt^2} {}^G\vec{r}'_4(t) \end{aligned} \quad (2-14)$$

For the acceleration vectors in the local coordinate system of each unit, one needs to rotate the vectors according to the yaw angle of the unit. For example, the relevant acceleration vectors from Figure 2-6 are now written as:

$$\vec{a}_{1f}(t) = {}^L\vec{a}_{1f}(t) = R(\phi_1)^{-1} {}^G\vec{a}_{1f}(t) \quad (2-15a)$$

$$\vec{a}_2(t) = {}^L\vec{a}_2(t) = R(\phi_2)^{-1} {}^G\vec{a}_2(t) \quad (2-15b)$$

$$\vec{a}_3(t) = {}^L\vec{a}_3(t) = R(\phi_3)^{-1} {}^G\vec{a}_3(t) \quad (2-15c)$$

$$\vec{a}_4(t) = {}^L\vec{a}_4(t) = R(\phi_4)^{-1} {}^G\vec{a}_4(t) \quad (2-15d)$$

Henceforth, when the text refers to the acceleration in a certain position of the truck, it denotes the acceleration aligned with the relevant axle (including the Coriolis acceleration terms).

2-3-2 Simplified kinematics

The kinematic relations derived in section Section 2-3-1 have a few drawbacks in case they are unaltered. Firstly, the equations are unnecessarily precise for highway driving of the truck. Secondly, the functions for the acceleration vectors require knowledge of the time derivative of the lateral velocity in the COM of the tractor in the body-fixed reference frame.

Three assumptions are made to simplify the kinematic relations:

- The articulation angles and their mutual sums are small.

$$\cos\left(\sum_i^j \theta_i\right) = 1, \quad \sin\left(\sum_i^j \theta_i\right) = \sum_{i=1}^j \theta_i \quad \forall i, j \in \{1, 2, 3\}, \quad i \leq j \quad (2-16)$$

- The mutual products of the following states are negligible:

$$\theta_1, \quad \theta_2, \quad \theta_3, \quad \dot{\theta}_0, \quad \dot{\theta}_1, \quad \dot{\theta}_2, \quad \dot{\theta}_3, \quad \ddot{\theta}_0, \quad \ddot{\theta}_1, \quad \ddot{\theta}_2, \quad \ddot{\theta}_3, \quad {}^L\vec{v}_{y,1}, \quad \frac{d}{dt} {}^L\vec{v}_{y,1}, \quad \frac{d}{dt} {}^L\vec{v}_{x,1} \quad (2-17)$$

- The states $d/dt {}^L\vec{v}_{x,1}$, $d/dt {}^L\vec{v}_{y,1}$, $\ddot{\theta}_0, \ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3$ are defined by the single-track equations of motion for the A-double combination.

2-4 Modeling of the vehicle position with respect to the road geometry

Consider that we define a road profile in the global inertial frame G by the function:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = f_{\text{road}}(s) \quad (2-18)$$

where s denotes the arc length of f_{road} from some arbitrary initial point $s = 0$. The position of a certain point in the truck, projected on the road is defined by the coordinate s and the perpendicular distance from this road geometry d . The position (s, d) is not necessarily unique for all coordinates (X, Y) and there may not exist an (s, d) -coordinate for all (X, Y) . In any position s along the road geometry we can construct a Frenét frame [13], with the normalized basis vectors $\hat{\vec{T}}(s)$ and $\hat{\vec{N}}(s)$.

$$\begin{aligned} \vec{T}(s) &= \frac{\partial f_{\text{road}}(s)}{\partial s} & \hat{\vec{T}}(s) &= \frac{\vec{T}(s)}{\|\vec{T}(s)\|_2} \\ \vec{N}(s) &= \frac{\partial^2 f_{\text{road}}(s)}{\partial s^2} & \hat{\vec{N}}(s) &= \frac{\vec{N}(s)}{\|\vec{N}(s)\|_2} \end{aligned} \quad (2-19)$$

The heading $\phi_R(s)$ of the road is defined as the angle between the global X axis and the vector $\vec{T}(s)$, we obtain:

$$\phi_R(s) = \arccos \left(\left\langle \hat{\vec{T}}(s), \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\rangle \right) \quad (2-20)$$

And the road-curvature $\kappa_R(s)$ is defined by:

$$\kappa_R(s) = \|\vec{N}(s)\|_2 \quad (2-21)$$

In Figure 2-7 it is illustrated how two Frenét frames are constructed, one in the front axle of the tractor and one in the rear axle of the second semi-trailer. We name the basis vectors of the two Frenét frames $\hat{\vec{T}}_{1f} = \hat{\vec{T}}(s_{1f})$, $\hat{\vec{N}}_{1f} = \hat{\vec{N}}(s_{1f})$, $\hat{\vec{T}}_4 = \hat{\vec{T}}(s_4)$, $\hat{\vec{N}}_4 = \hat{\vec{N}}(s_4)$. Where s_{1f} denotes the position of the front axle of the tractor on the road geometry and s_4 the equivalent position of the rear axle of the second semi-trailer.

$$\begin{pmatrix} X_{1f} \\ Y_{1f} \end{pmatrix} = f_{\text{road}}(s_{1f}) + \hat{\vec{N}}_{1f}d_{1f}, \quad \begin{pmatrix} X_4 \\ Y_4 \end{pmatrix} = f_{\text{road}}(s_4) + \hat{\vec{N}}_4d_4 \quad (2-22)$$

Where d_{1f} and d_4 denote the orthogonal distance of the front axle of the tractor and the rear axle of the second semi-trailer normal to the road geometry respectively.

The curvature and the heading at s_{1f} and s_4 are written in short as:

$$\begin{aligned} \phi_{R,1f} &= \phi_R(s_{1f}), & \phi_{R,4} &= \phi_R(s_4) \\ \kappa_{R,1f} &= \kappa_R(s_{1f}), & \kappa_{R,4} &= \kappa_R(s_4) \end{aligned} \quad (2-23)$$

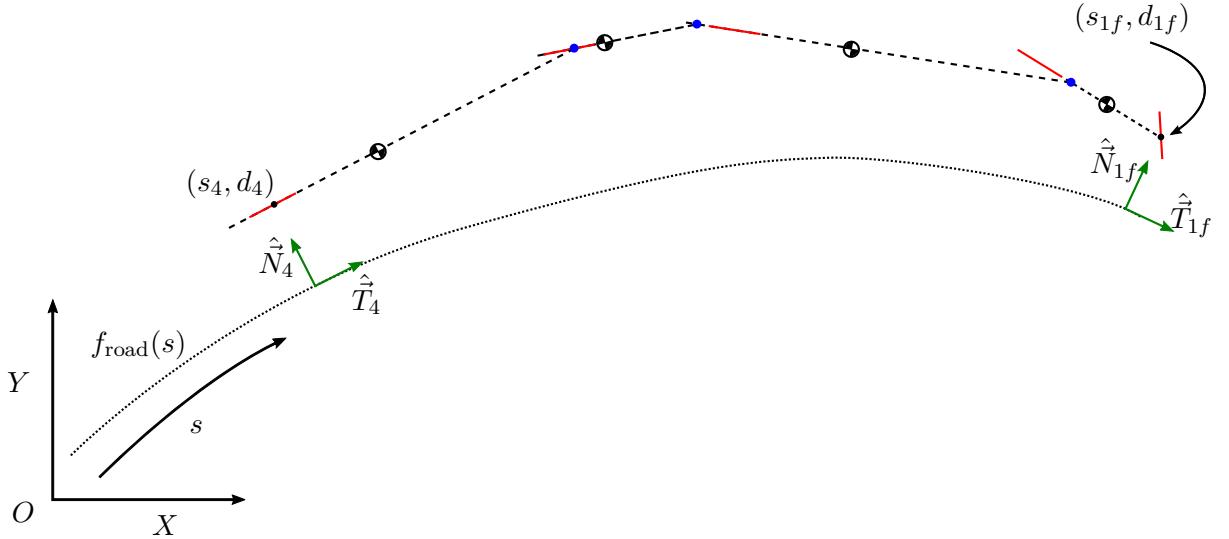


Figure 2-7: A visualization of the Frenét frames constructed at the front axle of the tractor and the rear axle of the second semi-trailer.

In order to model the position of the truck with respect to the road geometry. We construct the differential equations for \dot{s} and \dot{d} in all points we desire to track.

$$\begin{aligned}\dot{s}_{1f} &= \frac{\overset{L}{v}_{x,1f} \cos(\phi_1 - \phi_{R,1f}) - \overset{L}{v}_{y,1f} \sin(\phi_1 - \phi_{R,1f})}{1 - d_{1f} \kappa_{R,1f}} \\ \dot{d}_{1f} &= \overset{L}{v}_{x,1f} \sin(\phi_1 - \phi_{R,1f}) + \overset{L}{v}_{y,1f} \cos(\phi_1 - \phi_{R,1f}) \\ \dot{s}_4 &= \frac{\overset{L}{v}_{x,4} \cos(\phi_1 - \phi_{R,4}) - \overset{L}{v}_{y,4} \sin(\phi_1 - \phi_{R,4})}{1 - d_4 \kappa_{R,4}} \\ \dot{d}_4 &= \overset{L}{v}_{x,4} \sin(\phi_4 - \phi_{R,4}) + \overset{L}{v}_{y,4} \cos(\phi_4 - \phi_{R,4})\end{aligned}\tag{2-24}$$

To model the road information, one integrates this set of first order differential equations. The position of the other axles with respect to the road geometry can be obtained using the same procedure. Naturally, to apply this method to the single-track model the knowledge of $\overset{L}{v}_{1f}$ and $\overset{L}{v}_4$ is required. Expressions of these states as a function of the single-track model states have been described already. We append the newly obtained road position states to the single-track A-double combination model.

$$\xi_{lhvc,road} = \begin{pmatrix} v_{y,1} & \dot{\theta}_0 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & v_{x,1} & a_{x,1} & s_{1f} & d_{1f} & s_4 & d_4 \end{pmatrix}^T \tag{2-25a}$$

$$u_{lhvc,road} = \begin{pmatrix} \delta & a_{x,1,des} \end{pmatrix}^T \tag{2-25b}$$

2-5 Spatial reformulation of the vehicle dynamics along the road geometry

In the eventual trajectory generator, a finite horizon OCP is solved in receding horizon fashion. Each time step a predicted optimal trajectory is generated. To predict the position of the truck its units with respect to the road in time, a parameterization of the road curvature

and heading are required. For example, an analytic expression of the function f_{road} or a polynomial approximation.

An alternative is to reformulate the vehicle dynamics from the original temporal domain to a spatial domain[14]. More precisely, instead of using time as independent variable, one can use the progress of the truck along the road geometry. The greatest advantage of this technique is that the road information is then exactly available over the prediction horizon. It would seem natural to express this spatial model in the position of the front axle of the tractor s_{1f} . As discussed earlier, this is a very well-defined point that is independent from the truck configuration. But since the spatial reformulation is targeted to the OCP-formulation introduced later in this work, the position of the COM of the tractor (s_1) is preferred. The velocity vector in that position is available without the need for any kinematics.

We write the temporal prediction model for the vehicle dynamics and the road position.

$$\frac{d}{dt} \xi_{\text{pred}} = f_{\text{pred}}(\xi_{\text{pred}}, u_{\text{pred}}) \quad (2-26a)$$

$$\xi_{\text{pred}} = \begin{pmatrix} v_{y,1} & \dot{\theta}_0 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & v_{x,1} & a_{x,1} & d_1 & d_4 \end{pmatrix}^T \quad (2-26b)$$

$$u_{\text{pred}} = \begin{pmatrix} \delta & a_{x,1,\text{des}} \end{pmatrix}^T \quad (2-26c)$$

We rewrite the system dynamics in Eq. (2-26a) using the velocity of the COM along the road.

$$\frac{d\xi_{\text{pred}}}{ds_1} = \frac{d\xi_{\text{pred}}}{dt} \frac{dt}{ds_1} \quad (2-27)$$

We obtain,

$$\frac{d\xi_{\text{pred}}}{ds_1} = \frac{f_{\text{pred}}(\xi_{\text{pred}}, u_{\text{pred}})}{v_{x,1} \cos(\phi_1 - \phi_{R,1}) - v_{y,1} \sin(\phi_1 - \phi_{R,1})} (1 - d_1 \kappa_{R,1}) \quad (2-28)$$

Since s_1 is the independent variable, this state is removed from the state vector. We assume that for a low curvature road: $(1 - d_1 \kappa_{R,1}) = 1$. Note that when the truck remains in its lane, the term d_1 is bounded by half the lane width. The term $(1 - d_{1f} \kappa_{R,1})$ can be removed from the spatial formulation of the prediction model. We write,

$$\frac{d\xi_{\text{pred}}}{ds_1} \approx \frac{f_{\text{pred}}(\xi_{\text{pred}}, u_{\text{pred}})}{v_{x,1} \cos(\phi_1 - \phi_{R,1}) - v_{y,1} \sin(\phi_1 - \phi_{R,1})} \quad (2-29)$$

The first order differential equation of the newly introduced state d_4 requires the knowledge of the velocity vector of the rear axle of the second semi-trailer. Using results from Section 2-3-1 and Section 2-3-2 we obtain a compact expression for $d/dt d_4$ in the states of the single-track A-double combination model.

$$\begin{aligned} \frac{d}{dt} d_4 = & \cos(\phi_{R4} - \phi_4) \left(-(a_2 + a_3 + c_1 + c_2 + c_3 + a_4 + b_4) \dot{\theta}_0 \right. \\ & - (a_2 + a_3 + c_2 + c_3 + a_4 + b_4) \dot{\theta}_1 - (a_3 + c_3 + a_4 + b_4) \dot{\theta}_2 - \theta_1 v_{x,1} \\ & \left. - \theta_2(t) v_{x,1} - \theta_3(t) v_{x,1} + v_{y,1} - (a_4 + b_4) \dot{\theta}_3 \right) - v_{x,1} \sin(\phi_{R,4} - \phi_4) \end{aligned} \quad (2-30)$$

2-6 Modeling of surrounding highway users

Modeling of the surrounding traffic is an entire field of study on its own. In the literature survey for this thesis work [15] several types of models are introduced. Parametric models for human car following behavior, lane change maneuvering and decision-making are all addressed. It is urged that in the remainder of this thesis none of these models are used. It is assumed that for all relevant vehicles a complete acceleration-profile is available. Hence, no vehicle interaction is modeled.

Each obstacle $*$ has three states:

- The position on the road geometry: $s_{o,*}(t)$.
- The velocity tangent to the road geometry: $\dot{s}_{o,*}$.
- The lane in which the vehicle resides: $l_{o,*}$.

And because the vehicles do not change lane, this is entirely defined by a predetermined acceleration profile tangent to the road geometry $\ddot{s}_{o,*}(t)$ as a function of time. The differential equations for the resulting point-mass model of the surrounding vehicle $*$ is:

$$\frac{d}{dt} \begin{pmatrix} s_{o,*} \\ \dot{s}_{o,*} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} s_{o,*} \\ \dot{s}_{o,*} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \ddot{s}_{o,*} \quad (2-31)$$

2-7 Description of the high-fidelity vehicle model

The trajectory generator that is developed in this thesis work is tested on a high fidelity model of the A-double combination internally developed at Volvo GTT. The high fidelity model is occasionally referred to as the VTM plant. Since it is part of a library of high-fidelity truck models in the Volvo Truck Model. This state-of-art vehicle model is confidential, developed and validated by Volvo GTT. Details of the abilities are highlighted.

The high-fidelity plant includes detailed sub-models of the vehicle chassis (including frame compliance), cab suspension, steering system, power train, and brakes. The Magic tire formula [12] with combined slip, dynamic relaxation, tire normal load dependency, rolling resistance and suspension geometry is used as the tire model for all 22 wheels in the vehicle combination. The trajectory generation controller interfaces in MATLAB/SIMULINK to the high fidelity A-double combination model with a lower-level controller that coordinates the propulsion and steering systems. In all axles, and other subsystems, of A-double combination a virtual sensor can be read that provides full information on accelerations, yaw rates, etc.

The configuration of the high-fidelity plant is identical to the parameters used in the single-track model derived in Section 2-2.

Chapter 3

Formulation of the trajectory generation problem as an optimal control problem

The general task of driving a vehicle on a highway and maneuver between other traffic comes natural to humans. After a certain period of learning to drive, the trajectories they take result from unconscious handling of the accelerator pedal, brake and steering wheel. Some persons drive more aggressive, while others tend to be more gentle controlling the vehicle. On the other hand, we all know that cornering at too high speed or steering too quickly may result in an uncontrollable vehicle.

An A-double combination is not exactly like the *usual* passenger car. Other aspects need to be taken into consideration that one would not need to do in a car. One significant aspect is to prevent the truck to roll over due to excessive acceleration levels. Another aspects is that the size and weight of the truck limits its agility. In this chapter, an attempt is made to mathematically formulate comfortable driving behavior and to incorporate the limits on agility and safety effectively.

The fellow road users are important ingredients to an optimal trajectory. The social task of driving is hard to formulate mathematically and rule-based methods are usually applied [16] to model it. Several aspects can nevertheless be captured in an optimal control problem (OCP). Most notably keeping distance from the vehicle while following the lane, or adapting the velocity to the vehicles in the adjacent lane in preparation to a lane change. In this chapter, alternative methods are introduced and discussed to incorporate obstacles in the OCP.

One of the main goals of the trajectory generator is to ensure real-time (RT) performance. To achieve this, an appropriate formulation of the OCP is important. The optimization should ideally be solved in the lowest number of dimensions one can afford. As discussed later in Chapter 4, the optimization strategy is already focused at achieving that. Nevertheless, we try to keep the number of so-called decision variables as low as possible. Apart from the

discretization practicalities, the main influencing factor is the number of chosen control input signals, slack variables for the optimization and the horizon of the OCP.

The real-time solution strategy that is introduced in Chapter 4 requires the OCP to be formulated with a least-squares objective. Although one could argue that other norms in the objective can be desired, the 2-norm formulation is a given.

Let us start with the discussion on what exactly safe and comfortable driving means.

3-1 Comfortable driving, safety considerations and actuator constraints

Desired driving behavior is a heuristic that we would like to achieve by minimizing a set of measures. A great deal in generating smooth driving behavior is achieved by tuning the controller. Crucial in finding a generally well-tuned controller is the choice of cost function terms. This section is focused to relate relevant vehicle states for the perception of comfortable driving, the safety considerations and the physical constraints. Exact measures and the bounds used in the OCP are introduced later in this chapter.

Acceleration Acceleration is in particular related to the safety considerations. Especially the effect of RWA, lateral acceleration levels in all units need to be limited for safe driving. The payload of the truck causes a high center of mass (COM), which introduces a risk of rollover. Rollover thresholds highly depend on the payload, they can be as low as $3.9[m/s^2]$. The longitudinal acceleration of the truck is limited by the propulsion system and the tire limits. A fully loaded A-double combination has longitudinal acceleration capabilities that are in the order of $0.25[m/s^2]$, whereas braking faster than $3[m/s^2]$ can be considered as an evasive maneuver.

Jerk Jerk is defined as the first time derivative of acceleration. Jerk is highly influential on comfortable driving. Maneuvers exciting high jerk levels are perceived as unpleasant to passengers of a vehicle. High jerk is in general caused by sudden braking and aggressive steering motion. The truck driver is located in a cabin, suspended in the tractor about 1-2 meters above the front axle. In very smooth driving, jerk levels are typically far below $1[m/s^3]$.

Steering The steering input is very directly perceived by the driver. Apart from the effect of the steering angle and the steering angle rate on the vehicle dynamics, nervous behavior of both measures are undesired. Additionally, the steering geometry limits the maximum steering wheel angle and the actuator for the angle has limitations to the maximum rate it can achieve. Steering compliance limits the angle to approximately $1[rad]$, in highway driving these values are typically much lower.

Distance offset from lane center The distance from the lane center may be the most important factor in the optimization. Staying near the center of the lane is a great part of the maneuvering capability of the controller. A lane change is defined as a gradual change of

lane-center, which shifts to the adjacent lane of preference. At the same time, the distance from the lane center, has a great influence on the safe operation of the closed-loop system. Consider the highway driving scenario of the long heavy vehicle combination (LHVC), keeping all units within the lane in which the truck currently resides is important. If this condition is not met, the truck may collide with vehicles in other lanes or wheels may abandon the road. The width of the truck is approximately $2.5[m]$, whereas the width of a highway lane typically lies between $2.7 - 3.7[m]$.

Longitudinal distance from surrounding traffic The overall problem of collision avoidance with surrounding traffic is to avoid both lateral and longitudinal collisions. However, the structure of highway driving allows us to decompose this overall problem to staying within the lane and not having any longitudinal collisions. Safe driving of the truck in relation to obstacle avoidance is mainly directed to keeping an appropriate distance with preceding vehicles. In case of lane changing, this problem is extended to also consider the vehicles in the adjacent lane. The significance of each surrounding object in relation to the trajectory generation problem changes when a lane change is considered or not. For obvious reasons, distance keeping from other vehicles also has an effect on driver comfort.

With these main measures and limitations in mind, we can formulate a definition for an optimal trajectory. As mentioned before, tuning the controller to mimic the driving behavior is an important step. Let us attempt to formulate the optimal trajectory, in absence of surrounding traffic.

3-2 Definition of an optimal trajectory

Before we write the final OCP, we first describe an optimal trajectory mathematically without simplifications. It is assumed that on any time instant of the prediction we have perfect knowledge on the vehicle state. Naturally, in practice this requires a vehicle prediction model as introduced in Chapter 2. In this section, the elements of the optimal trajectory are discussed and simplifications are proposed. So that we can, later, arrive at the OCP that will have the desired objective formulation and the minimum set of constraints that are required.

First, consider the optimal trajectory. Let us disregard the effect of the surrounding traffic and focus on the problem of maneuvering the truck without obstacles. The inclusion of obstacles is introduced and discussed in a later part of the thesis. We say in words:

The optimal trajectory for the A-double combination in highway maneuvering is defined by the trajectory that minimizes the finite spatial horizon integral of the weighted two-norm of the lateral and longitudinal jerk, the distance offset of the tractor from the lane-center, the offset from the velocity reference profile. While at the same time, the lateral acceleration in all units stay in safe bounds, all units stay within the lane boundaries and the actuator constraints are complied with.

We can translate the objective for the optimal trajectory into a mathematical problem.

$$\begin{aligned}
 & \min_{\substack{s_f \\ s=0}} \int_{s=0}^{s_f} \left(K_{j_x,1} j_{x,1}^2 + K_{j_y,1} j_{y,1}^2 + K_{d_1} (d_1 - d_{1,\text{ref}})^2 + K_{v_{x,1}} (v_{x,1} - v_{x,1,\text{ref}})^2 \right) d\sigma \\
 \text{s.t.} \quad & \underline{a}_y \leq a_{y,i}(s) \leq \bar{a}_y \quad \forall i \in \{1f, 2, 3, 4\} \\
 & \underline{d} \leq d_i(s) \leq \bar{d} \quad \forall i \in \{1f, 2, 3, 4\} \\
 & \underline{a}_{x,1,\text{des}} \leq a_{x,1,\text{des}}(s) \leq \bar{a}_{x,1,\text{des}} \\
 & \underline{v}_{x,1} \leq v_{x,1}(s) \leq \bar{v}_{x,1} \\
 & \underline{\delta} \leq \delta(s) \leq \bar{\delta} \\
 & \underline{d\delta/dt} \leq d\delta/dt(s) \leq \bar{d\delta/dt}
 \end{aligned} \tag{3-1}$$

The weighting factors K_* scale the penalty on the separate contributing terms in the cost function. Notice that all cross terms (e.g. $j_{x,1} j_{y,1}$) have been disregarded. Cross-terms can be selectively used to fine-tune the controller, simultaneously it increases the difficulty tuning the controller. In order to have all elements in the cost-function to intuitively relate to physical phenomena, the cross-terms are removed. No end-state cost term is present in the formulation either. In case of the trajectory generation, there is no obvious motivation to penalize the end state any different from the intermediate states. We have not necessarily completed a maneuver by the end of prediction. End-state cost terms are often used in model predictive control (MPC) applications to ensure closed-loop stability [17]. Since the A-double combination is stable in normal operation this is not a motivation to include an end-state cost term.

Notice that integral in the cost function is not defined over time. Instead, the integration takes place over the distance traveled by the COM of the tractor along the road geometry, between $[0, s_f]$. In this formulation there is no need to parameterize the heading of the road, required to model the position of the truck with respect to the lane center. The road profile in the COM is defined naturally over the prediction $[0, s_f]$. This trick does however require the vehicle dynamics to evolve over the traveled distance along the road too. This requires a so-called spatial reformulation, which has already been introduced in Section 2-5. It has a few notable effects to be discussed:

- The time-horizon of the prediction is dependent on the velocity of the truck. At high speeds, the prediction in time is much shorter compared to a scenario where the truck drives slow. It takes a longer time, to travel the distance s_f when driving slow. Considering the highway driving scenario, the velocity of the truck may vary between $8[m/s]$ and $23[m/s]$. For a prediction horizon of $100[m]$ this results in horizon of about $4.3[s]$ to $12.5[s]$ respectively. Of course, it would be possible to shorten the prediction horizon for lower velocities. This switching logic is not implemented in this thesis work.
- To make a small leap forward in the thesis work. A discrete integration method in the solution technique will have a spatial step size. Thus, it depends on the velocity of the truck what the temporal step size will be. A wide accepted velocity range introduces difficulty optimizing the sampling rate.

- Whereas a spatial formulation facilitates an straightforward parameterization of the road. It naturally has the opposite effect on explicit time-varying entities. An example of such entity, is the trajectory prediction of surrounding traffic. Interestingly, as we can read in [8] a spatial formulation can have advantages in case of static obstacles.

The cost function of the optimal trajectory is kept simple and concise. We desire to minimize jerk, track the lane center and velocity reference. A set of constraints are introduced to assure safe driving and to impose actuator constraints. Let us move forward and discuss possible simplification steps.

3-2-1 Approximation of the lateral jerk

The lateral jerk observed by the COM of the tractor is analytically defined by:

$${}_1^L j_{y,1} = 2\dot{\theta}_0 \frac{d}{dt} {}_1^L v_{x,1} + \ddot{\theta}_0 {}_1^L v_{x,1} - \dot{\theta}_0^2 {}_1^L v_{y,1} + \frac{d^2}{dt^2} {}_1^L v_{y,1} \quad (3-2)$$

One problem arises in this description of the lateral jerk. It requires the knowledge of $\frac{d^2}{dt^2} {}_1^L v_{y,1}$, which is not directly available. In order to minimize for the jerk in the optimal trajectory, it is not necessary to have the most accurate measure for the jerk itself. Additionally, the trajectory generator is designed for highway maneuvering. Because the longitudinal velocity is generally high, we can safely assume that the greatest contributing term to the lateral jerk is $\dot{\theta}_0 {}_1^L v_{x,1}$. The approximation for the lateral jerk that is proposed neglects $\frac{d^2}{dt^2} {}_1^L v_{y,1}$. Like in the expressions for auxiliary acceleration vectors, $\ddot{\theta}_0$ is not a state which directly measured, we do however have an expression for this measure available as a function of our state vector in the single-track vehicle model.

3-2-2 Lateral acceleration bounds

The lateral acceleration of all units ought to stay in bounds at each point of the truck. However, the lateral acceleration highly depends on where it is measured. It must be assured that the lateral acceleration in the COM of each unit stays in bounds. The COM is highly dependent on the payload and thus not practical location to measure the acceleration. It is possible to formulate unsafe lateral acceleration levels at the wheels. Admittedly, conservative bounds are then required.

In practice, the A-double combination has active safety systems that mitigate the rollover possibility. By various techniques, such as applying brakes on selective axles, it is attempted to make the truck slide in prevention of rolling over. Any automatic trajectory generator should avoid to plan a path near such regions. This is achieved in the optimal trajectory formulation by introducing hard constraint on the lateral acceleration. Considering that active rollover mitigation systems kick in at about $0.4G$, the bounds are suggested conservatively in the region of $2.5 - 3[m/s^2]$.

The effect of rearward amplification (RWA) motivates prioritizing the limit on the lateral acceleration in the rearmost unit, since it is likely that the maximum accelerations will occur there. However, the chained configuration of the truck also imposes a delay on the motion.

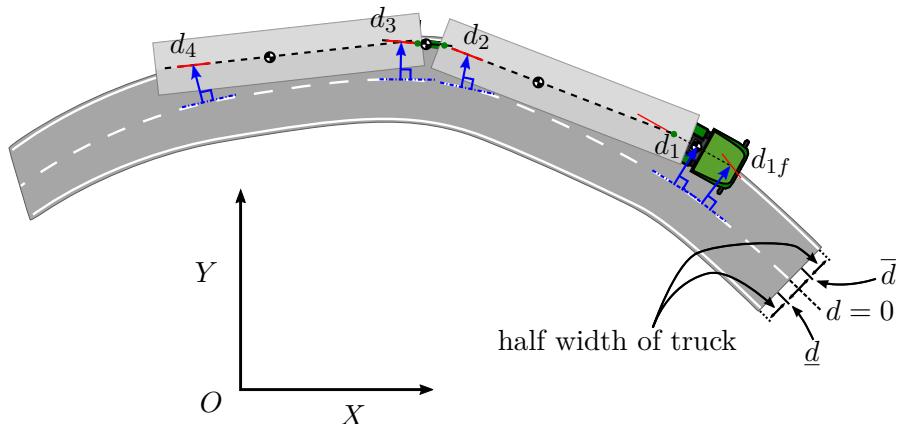


Figure 3-1: Illustration of the lateral distance bounds for the axles, corrected for the width of the truck. The center line of the road is equal to zero lateral distance offset.

The prediction horizon should be long enough to account for the delay in the acceleration level at the rear of the truck. This motivates priority to bound on the lateral acceleration on the first unit too. It is proposed to only check the lateral acceleration bounds on the tractor and in the rearmost unit.

3-2-3 Lateral distance off-set bounds

The lateral distance of each unit is defined as the perpendicular distance of the axles from the center of the lane. In Figure 3-1 we see the geometric meaning of this measure on a curved road. The lateral distance of the truck is measured in the center of the axle. The bounds d and \bar{d} are determined from the width of the lane, corrected by the width of the truck itself.

Evidently, all wheels of the A-double combination need to stay within the lane boundaries to avoid dangerous scenarios. Hence, in general it is appropriate, like formulated in Eq. (3-1), to apply constraints on all axles. However, the scenario for which the controller is developed allows to reduce the number of constraints. If we find ourselves in the situation that the rear axle of the second semi-trailer and the front axle of the tractor are both inside the lane boundaries, it is very probable that the other axles are as well. Circumstances in which this may not be the case are, when the road curvature is very high, the truck is in some unnatural orientation to driving conditions or the road width is highly variable. A design choice is made to only consider the lateral distance off-set in the COM of the tractor d_1 and in the rear axle of the second semi-trailer d_4 . The assumption is implicitly made that when the trajectories of both these axles are within the road boundaries, the complete truck stays in the lane.

3-2-4 Longitudinal velocity bounds

In the most naive sense, the velocity bounds $\underline{v}_{x,1}$ and $\bar{v}_{x,1}$ would reflect the actuator constraints and the legal speed limitations. Setting the lower velocity bound to a small value does not come without implications though. As discussed in the beginning of the section, a low velocity causes a long temporal prediction horizon and a large (temporal equivalent)

integration step.

Meanwhile, the intended validity of the vehicle prediction model is between $8 - 23[m/s]$. Based solely on this given, the velocity bounds are suggested to span the entire vehicle model validity region.

3-2-5 Steering actuation bounds

The steering actuation bounds apply to the road wheel angle δ and the road wheel angular rate $d\delta/dt$. Notice that the constraints thus apply to the angle of the wheels with the chassis. The steering wheel angle in the cabin of the tractor is connected via a gearing. The maximum road wheel angle is limited by steering geometry, which is designed to cope with all scenarios including parking maneuvers. There is no use to allow the truck to make the maximum steering angles imposed by the steering compliance. In high speed scenarios the maximum steering angles would cause rollover, or understeer in case of a low friction road surface. The road wheel angle is limited to $0.1[rad]$. The angular rate of the steering wheel angle, that is limited by the steering wheel actuator is limited to $0.05[rad/s]$.

3-3 Alternative obstacle avoidance formulations

The optimal trajectory that is formulated up to this point, does not incorporate surrounding traffic. Hence, in the scenario that the truck is ought to keep lane, it is possible that we have a frontal collision with our preceding fellow road user. In case a lane change scenario is considered, vehicles in the adjacent lane are not considered, and a lateral collision may occur. The goal of this section is to formulate a parameterization of a predicted collision-free corridor and introducing a distance-keeping incentive.

The lateral collision avoidance problem is tackled by a decision algorithm that tells in advance whether a lane change is possible or not. Based on this decision, lateral position bounds on the truck are relaxed accordingly to allow the truck to occupy two lanes. This decision algorithm is based on the availability of the desired lane over the prediction horizon. The longitudinal collision avoidance problem is less discrete in nature and is prone to more discussion.

First a naming convention for the obstacles is introduced. This naming convention is used in the remainder of the thesis.

3-3-1 Naming convention of the obstacles

The highway gives space to many road users, only few of them are relevant to our collision avoidance problem. We define the surrounding road user that are of interest to the trajectory generation problem to satisfy very specific criteria and name them accordingly.

- The first road user in the same lane of the truck or the departure lane of a lane change that drives behind the front axle of the tractor is called **obstacle 1**.

- The first road user in the same lane of the truck or the departure lane of a lane change that drives ahead of the front axle of the tractor is called **obstacle 2**.
- The first road user in the goal lane of the lane change that drives behind the front axle of the tractor is called **obstacle 3**.
- The first road user in the goal lane of the lane change that drives ahead of the front axle of the tractor is called **obstacle 4**.

Notice that in case no lane change is desired or that a lane change is not possible, the last two criteria are not relevant.

Although obstacle 1 is named, it is argued only 2, 3 and 4 influence the optimal trajectory. We may expect that obstacle 1 adapts its speed to the A-double combination at all time. The same does not hold for obstacle 3 in case the lane change is in progress.

The optimal trajectory is defined at each time-step in a receding horizon fashion. In order to predict an optimal trajectory for the A-double combination, we require a prediction model for the surrounding traffic. The next subsection will first introduce the modeling technique.

3-3-2 Prediction model for the distance to surrounding traffic

The trajectories of the surrounding traffic depend on the character of the drivers, their current state of mind and their intentions. Predicting their behavior is everything but a trivial task. In fact, the optimal trajectory for the A-double combination itself influences the behavior of other vehicles. This thesis does not give an answer how to predict behavior of traffic participants. However, the thesis does provide a possible interface for an externally predicted trajectory profile in the spatial problem formulation for an optimal trajectory.

Based on the distance with the surrounding traffic that is measured and the predicted velocity profile along the road geometry we have the following differential equations to describe the distance $\Delta s_{o,i}$ with obstacle i .

$$\frac{d\Delta s_{o,i}}{ds_1} = \begin{cases} 1 - \frac{ds_{o,i}}{dt} \frac{dt}{ds_1}, & \text{if obstacle } i \text{ is trailing} \\ \frac{ds_{o,i}}{dt} \frac{dt}{ds_1} - 1, & \text{if obstacle } i \text{ is leading} \end{cases} \quad (3-3)$$

Notice that $\frac{ds_{o,i}}{dt}(t)$ is included as a time varying path. This time-varying path needs to be parameterized in a suitable way. We can do this by defining, e.g., a polynomial fit of the predicted trajectory. Each time step when the OCP is solved, coefficients of the polynomial can be provided. Where the time parameter in this basis function is obtained by integration:

$$t(s_1) = \int_0^{s_1} \frac{dt}{d\sigma} d\sigma = \int_0^{s_1} \frac{1}{v_{x,1} \cos(\phi_1 - \phi_{R,1}) - v_{y,1} \sin(\phi_1 - \phi_{R,1})} d\sigma \quad (3-4)$$

We obtain a possible expression for the velocity of the obstacle at s_1

$$\frac{ds_{o,i}}{dt}(s_1) = \sum_{k=0}^{n_\alpha} \alpha_k t(s_1)^k \quad (3-5)$$

where n_α is the number of coefficients in the polynomial.

During this thesis work no external prediction engine for the motion of surrounding traffic was developed. As described in Section 2-6 from the previous chapter, the obstacles are modeled to move according to a acceleration profile provided in advance of the simulation. Thus, perfect knowledge of the motion of the surrounding traffic in time is available. It is possible to include this knowledge in the control algorithm.

Choice of priorities in the thesis work have led to the implementation of only the trivial polynomial, where $n_\alpha = 0$ and where α_0 is simply assumed to be the measured velocity of the obstacle.

3-3-3 Collision avoidance keeping lane

The simplest case of obstacle avoidance is when the truck combination is keeping lane. As discussed before, the trailing vehicle is not considered in the OCP, neither are vehicles in the adjacent lanes because no lane change is in progress. The collision avoidance is reduced complying with the constraint to stay in the lane and to stay behind the leading vehicle, obstacle 2. In Figure 3-2 we can see an illustration of this scenario.

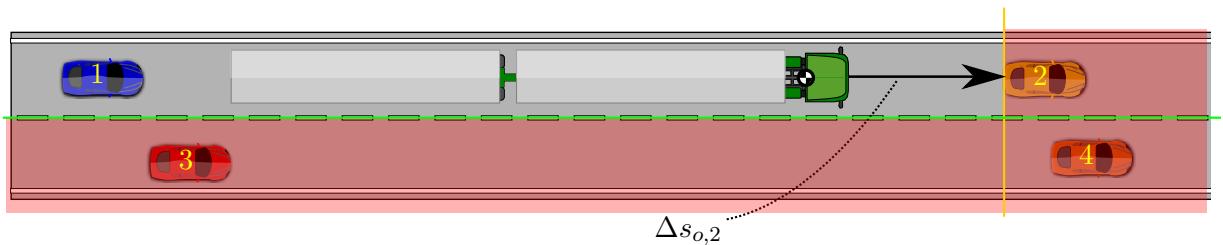


Figure 3-2: Indication of the prohibited regions (indicated in red) in a lane keeping scenario or when no lane change is possible. A hyperplane, at distance $\Delta s_{o,2}$ moves with the obstacle 2 over the prediction horizon. Note that the left road boundary constraint is not visualized.

Distance keeping with a preceding vehicle is achieved based on the temporal gap with the leading car. The temporal gap is defined as,

$$\Delta\tau_{o,i} = \frac{\Delta s_{o,2}}{\dot{s}_1} \quad (3-6)$$

Where \dot{s}_1 is the velocity of the COM along the lane center, that we approximate for simplicity with $v_{x,1}$. The distance keeping incentive in the cost function of the OCP should incorporate some logic. Whenever the leading vehicle is far from the temporal gap we desire, it should not contribute to the cost function. This can be achieved using slack variables, as implemented in [18]. Alternatively, since the OCP allows for non-linear cost terms, a sigmoid function can be used to approximate the switching behavior. The latter technique is used in this thesis work. In Section 3-4 one finds the exact expressions for these sigmoid functions.

3-3-4 Collision avoidance in a lane change

Avoiding collisions with other vehicles becomes more challenging when a lane change is made. In this thesis work, different formulations for the so-called predicted collision-free corridor

have been considered. Two of them are introduced and described in detail in this thesis. It must be noted, that in the results shown in this work, only one parameterization is evaluated. Another parameterization was, at the time of writing, a work in progress.

Rectangular predicted collision-free corridor

The first proposed parameterization of the predicted collision-free corridor is rectangular. Like illustrated in Figure 3-3, we can introduce prohibited half-spaces for each relevant obstacle. Hence, perpendicular to the road geometry, hyperplanes are constructed at obstacles 2, 3, and 4. The distance with each obstacle is bounded to avoid collision. An incentive is introduced in the cost function to divert the optimal trajectory from the obstacles.

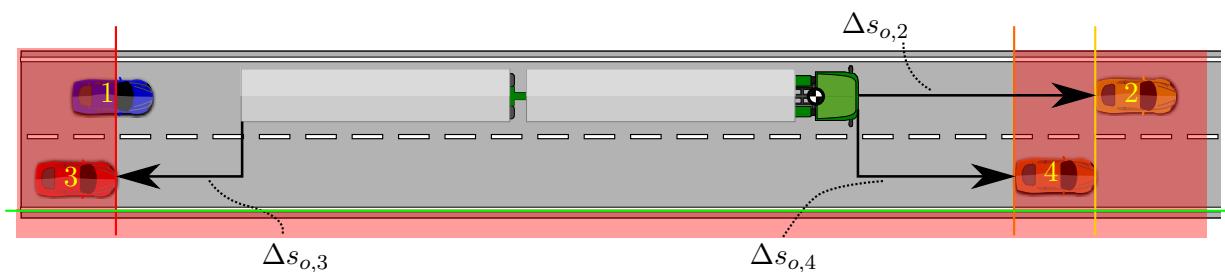


Figure 3-3: In the rectangular formulation of the predicted collision-free corridor the feasible area is bounded by obstacle 2, 3 and 4, which each introduce a prohibited half space. The lateral distance constraint is temporarily relaxed to allow the truck to reside in two lanes.

The main motivation for the rectangular predicted collision-free corridor is its simplicity. And frankly, it works quite well as long as vehicles in the adjacent lane drive at a similar speed as the truck. Or when the traffic in the adjacent lane drives slower than the truck. The problem arises when the vehicles in the adjacent lane drive faster. The asymmetry in the possibility to slow down, but not to be able to speed up lies in the choice for the obstacles of interest.

Consider the following situation, illustrated in Figure 3-3. The truck drives in the left lane and we desire to change lane to the right. In the right lane, the vehicles drive faster. We see that, as long as a lane change is in progress we cannot plan a trajectory that passes obstacle 2. The inability to pass obstacle 2 is not necessarily a problem itself. However, unless we model that obstacle 3 decelerates. A lane change might be infeasible, because the space in the predicted collision free corridor contracts. It is squashed between obstacles 2 and 3.

Two potential solutions to this problem in order to be able to use this formulation are: One, to model deceleration of obstacle 3. Or secondly, not allowing a lane change to be initiated if obstacle 3 is in the near vicinity. Both heavily rely on heuristics, which is not appreciated.

Another solution is to choose a different predicted collision free corridor, let us introduce such an alternative.

Sigmoidal predicted collision-free corridor

An alternative parameterization of the predicted collision-free corridor is illustrated in Figure 3-4. In this case, only two separating hyperplanes are being considered. In contrast the rectangular version, that required three.

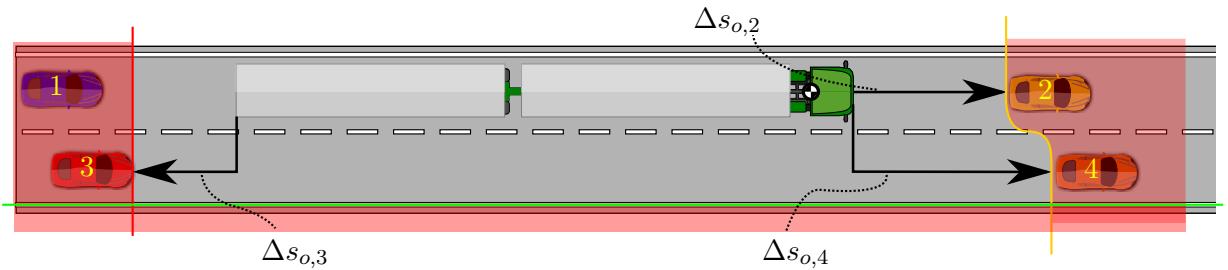


Figure 3-4: Half-spaces shaped by sigmoid support functions.

Like with the distance keeping incentive, logic can be incorporated by using sigmoid functions. Simply put, we shape the predicted collision-free corridor along the lateral position on the lanes.

The distance keeping incentive is very similar to before. However, we only have two cost terms. One for the distance with obstacle 3 and one for the hyperplane ahead that is a function of the position of obstacles 2 and 4.

Although implementation of this method is a limited effort. It would require a numerical analysis on how the solution algorithm, described in the next chapter, behaves with this formulation. The applicability of the rectangular predicted collision-free corridor under conservative decision making has motivated its preference in presence of other tasks that had priority.

In the next section, the complete OCP is introduced. This includes the implementation of the rectangular predicted collision free corridor for obstacle avoidance.

3-4 Formulation of the resulting optimal control problem

All prerequisites for the trajectory generation problem are introduced, we can converge to the OCP formulation. The OCP is based on the optimal trajectory formulation from Section 3-2, rewritten to an initial value problem. The initial state of the state vector ξ_{ocp} is known and the single-track prediction model is incorporated to evaluate the state vector over the prediction horizon. In this section, details of the OCP are described. This involves explicit expressions for all elements of the OCP, including the vehicle model.

$$\begin{aligned}
\min_{s_1=0} \int_{s_1=0}^{s_f} & \left(K_{j_{x,1}} j_{x,1}^2 + K_{j_{y,1}} (2\dot{\theta}_0 \frac{d}{dt} v_{x,1} + \dot{\theta}_0 v_{x,1} - \dot{\theta}_0^2 v_{y,1})^2 + K_{d_1} (d_1 - d_{1,\text{ref}}(\sigma))^2 + \right. \\
& + K_{v_{x,1}} (v_{x,1} - v_{x,1,\text{ref}}(\sigma))^2 + K_{\dot{\delta}} \frac{d\delta^2}{dt} + K_{\Delta s_{o,2}} (f_{\text{dk}}(\Delta s_{o,2}, v_{x,1}))^2 \\
& \left. + K_{\Delta s_{o,3}} (f_{\text{dk}}(\Delta s_{o,3}, v_{x,1}))^2 + K_{\Delta s_{o,4}} (f_{\text{dk}}(\Delta s_{o,4}, v_{x,1}))^2 \right) d\sigma \\
\text{s.t. } & \frac{d\xi_{\text{ocp}}}{ds_1} = g(\xi_{\text{ocp}}, u, s_1) \\
\xi_{\text{ocp}}(0) = & \xi_0 \\
\underline{a}_{y,1} \leq & a_{y,1}(s_1) \leq \bar{a}_{y,1} \\
\underline{a}_{y,4} \leq & a_{y,4}(s_1) \leq \bar{a}_{y,4} \\
\underline{d} \leq & d_1(s_1) \leq \bar{d} \\
\underline{d} \leq & d_4(s_1) \leq \bar{d} \\
\underline{j}_{x,1,\text{des}} \leq & j_{x,1,\text{des}}(s_1) \leq \bar{j}_{x,1,\text{des}} \\
\underline{a}_{x,1,\text{des}} \leq & a_{x,1,\text{des}}(s_1) \leq \bar{a}_{x,1,\text{des}} \\
\underline{v}_{x,1} \leq & v_{x,1}(s_1) \leq \bar{v}_{x,1} \\
\underline{\delta} \leq & \delta(s_1) \leq \bar{\delta} \\
\underline{d\delta/dt} \leq & d\delta/dt(s_1) \leq \overline{d\delta/dt} \\
\underline{\Delta s_{o,2}} \leq & \Delta s_{o,2} \\
\underline{\Delta s_{o,3}} \leq & \Delta s_{o,3} \\
\underline{\Delta s_{o,4}} \leq & \Delta s_{o,4}
\end{aligned} \tag{3-7}$$

The state vector of the prediction model ξ_{ocp} and the control input u are defined by:

$$\xi_{\text{ocp}} = \begin{pmatrix} v_{y,1} & \theta_0 & \dot{\theta}_0 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & v_{x,1} \\ a_{x,1} & a_{x,1,\text{des}} & d_1 & d_4 & \delta & \Delta s_{o,2} & \Delta s_{o,3} & \Delta s_{o,4} \end{pmatrix}^T \tag{3-8a}$$

$$u = \begin{pmatrix} \dot{\delta} & j_{x,1,\text{des}} \end{pmatrix}^T \tag{3-8b}$$

3-4-1 Prediction horizon

The prediction horizon determines how far we look into the future to determine the control action. A longer horizon enables the ability to start adopting for future events early. Whereas it introduces a higher computational load of the overall optimization algorithm.

The prediction horizon of the receding horizon OCP is set to 100[m]. This value is chosen as a trade-off to achieve real-time performance, while maximizing the horizon length. Two main motivations advocate for a long horizon:

- The motion in the rear trailer is delayed significantly with respect to the control input.

- The optimal trajectory must account for corners far ahead, to be able to find a comfortable deceleration profile (if necessary).

The delay of the motion in the second semi-trailer is in the order of the length of the truck. In order to plan the trajectory of this unit, the horizon should be a multiple of the truck its length. A typical smooth lane change at $20m/s$ is about $170 - 200[m]$ long. The prediction horizon needs to cover a significant part of the lane change in order to effectively plan it.

3-4-2 Cost function

The longitudinal jerk $j_{x,1,des}$ is chosen as a decision variable directly. The control input is integrated explicitly in the prediction model to obtain a desired acceleration. Longitudinal jerk minimization is incorporated as such, rather than using a finite differences approach to approximate the jerk in the cost function. Using the same argumentation, the steering angle rate is chosen as decision variable.

The lateral jerk is approximated using the expression from Section 3-2-1. To recapitulate, it is not the goal to approximate the jerk as accurate as possible. The main goal is to capture the trend of the jerk well.

Notice, that one of the new terms in the cost function is $K_{\dot{\delta}} \frac{d\dot{\delta}}{dt}^2$. This term is introduced to avoid cheap control and is necessary for a well-posed problem definition.

Other new terms in the cost function are the incentive to keep distance with surrounding obstacles $* \in \{2, 3, 4\}$. The function f_{dk} is defined as follows:

$$f_{dk}(\Delta s_{o,*}, v_{x,1}) = \frac{\tau_{o,*} - \Delta s_{o,*}/v_{x,1}}{\tau_{o,*}} \frac{1}{1 + e^{-c_{o,*}(\tau_{o,*} - \Delta s_{o,*}/v_{x,1})}} \quad (3-9)$$

with the tuning parameters $\tau_{o,*}$ and $c_{o,*}$. Tuning parameter $\tau_{o,*}$ denotes the time gap at which the sigmoid function switches. And $c_{o,*}$ can be considered as a scaling for the steepness of the sigmoid function.

3-4-3 Inequality constraints

Most of the inequality constraints are box constraints on the predicted states and input. Except for the lateral acceleration constraints. The following two expressions approximate the lateral acceleration in the COM of the tractor and in the rear axle of the second semi-trailer respectively. The kinematic relations have been derived in Section 2-3.

$$a_{y,1} = \dot{\theta}_0 v_{x,1} + \frac{dv_{y,1}}{dt} \quad (3-10a)$$

$$\begin{aligned} a_{y,4} = & -(a_2 + a_3 + c_1 + c_2 + c_3 + a_4 + b_4) \ddot{\theta}_0 - (a_2 + a_3 + c_2 + c_3 + a_4 + b_4) \ddot{\theta}_1 \\ & - (a_3 + c_3 + a_4 + b_4) \ddot{\theta}_2 + \dot{\theta}_0 v_{x,1} + \frac{dv_{y,1}}{dt} - (a_4 + b_4) \ddot{\theta}_3 \end{aligned} \quad (3-10b)$$

where $dv_{y,1}/dt$ is available from the vehicle prediction model as a function of the vehicle states.

3-4-4 Prediction model equality constraints

The OCP uses a non-parametric version of the vehicle model, using the vehicle configuration from Table 2-2. The temporal dynamics of the vehicle prediction model are given in Eq. (3-11). This spatial representation is obtained by dividing each first order derivative by $\dot{v}_{x,1} = v_{x,1} \cos(\phi_1 - \phi_{R,1}) - v_{y,1} \sin(\phi_1 - \phi_{R,1})$.

$$\begin{aligned}
\frac{dv_{y,1}}{dt} &= 45.9558 \delta + 1.9775 \theta_1 + 0.8494 \theta_2 - 0.0022 \theta_3 + 21.9217 \frac{\dot{\theta}_1}{v_{x,1}} + 4.4014 \frac{\dot{\theta}_2}{v_{x,1}} + \\
&\quad - 0.0170 \frac{\dot{\theta}_3}{v_{x,1}} - 70.6191 \frac{v_{y,1}}{v_{x,1}} - \dot{\theta}_0 \frac{v_{x,1}^2 - 9.7314}{v_{x,1}} \\
\frac{d\dot{\theta}_0}{dt} &= 25.0956 \delta - 1.8974 \theta_1 - 0.8150 \theta_2 + 0.0021 \theta_3 - 21.0338 \frac{\dot{\theta}_1}{v_{x,1}} - 4.2231 \frac{\dot{\theta}_2}{v_{x,1}} + \\
&\quad + 0.0164 \frac{\dot{\theta}_3}{v_{x,1}} + 27.5489 \frac{v_{y,1}}{v_{x,1}} - 174.2882 \frac{\dot{\theta}_0}{v_{x,1}} \\
\frac{d\dot{\theta}_1}{dt} &= 2.4818 \theta_2 - 3.9082 \theta_1 - 25.4638 \delta - 0.0065 \theta_3 - 10.5324 \frac{\dot{\theta}_1}{v_{x,1}} + 12.8600 \frac{\dot{\theta}_2}{v_{x,1}} + \\
&\quad - 0.0498 \frac{\dot{\theta}_3}{v_{x,1}} - 36.4048 \frac{v_{y,1}}{v_{x,1}} + 165.4516 \frac{\dot{\theta}_0}{v_{x,1}} \\
\frac{d\dot{\theta}_2}{dt} &= 0.5539 \delta + 2.2622 \theta_1 - 22.9024 \theta_2 - 0.9311 \theta_3 - 170.0741 \frac{\dot{\theta}_1}{v_{x,1}} - 125.6565 \frac{\dot{\theta}_2}{v_{x,1}} + \\
&\quad - 7.1692 \frac{\dot{\theta}_3}{v_{x,1}} + 19.7904 \frac{v_{y,1}}{v_{x,1}} - 216.8786 \frac{\dot{\theta}_0}{v_{x,1}} \\
\frac{d\dot{\theta}_3}{dt} &= 5.0960 \theta_1 - 0.1851 \delta + 22.7324 \theta_2 - 7.0991 \theta_3 + 168.7766 \frac{\dot{\theta}_1}{v_{x,1}} + 68.1597 \frac{\dot{\theta}_2}{v_{x,1}} + \\
&\quad - 54.6629 \frac{\dot{\theta}_3}{v_{x,1}} - 12.4638 \frac{v_{y,1}}{v_{x,1}} + 195.8250 \frac{\dot{\theta}_0}{v_{x,1}} \\
\frac{dd_1}{dt} &= v_{x,1} \sin(\phi_1 - \phi_{R,1}) + v_{y,1} \cos(\phi_1 - \phi_{R,1}) \\
\frac{dd_4}{dt} &= \cos(\phi_{R,4} - \phi_4) \left(-(a_2 + a_3 + c_1 + c_2 + c_3 + a_4 + b_4) \dot{\theta}_0 + \right. \\
&\quad \left. - (a_2 + a_3 + c_2 + c_3 + a_4 + b_4) \dot{\theta}_1 - (a_3 + c_3 + a_4 + b_4) \dot{\theta}_2 - \theta_1 v_{x,1} + \right. \\
&\quad \left. - \theta_2(t) v_{x,1} - \theta_3(t) v_{x,1} + v_{y,1} - (a_4 + b_4) \dot{\theta}_3 \right) - v_{x,1} \sin(\phi_{R,4} - \phi_4) \\
\frac{d\theta_0}{dt} &= \dot{\theta}_0, \quad \frac{d\theta_1}{dt} = \dot{\theta}_1, \quad \frac{d\theta_2}{dt} = \dot{\theta}_2, \quad \frac{d\theta_3}{dt} = \dot{\theta}_3, \\
\frac{d\delta}{dt} &= \dot{\delta}, \quad \frac{dv_{x,1}}{dt} = a_{x,1}, \quad \frac{da_{x,1}}{dt} = \frac{a_{x,1,des} - a_{x,1}}{\tau_{cc}}, \quad \frac{da_{x,1,des}}{dt} = j_{x,1,des}, \\
\frac{\Delta s_{o,2}}{dt} &= v_{o,2} - \frac{ds_1}{dt}, \quad \frac{\Delta s_{o,3}}{dt} = -v_{o,3} + \frac{ds_1}{dt}, \quad \frac{\Delta s_{o,4}}{dt} = v_{o,4} - \frac{ds_1}{dt}
\end{aligned} \tag{3-11}$$

3-5 Summary

In this chapter, the trajectory generation problem to drive a LHVC on a highway is formulated as an OCP. Lateral and longitudinal jerk are minimized to obtain smooth driving behavior, while tracking error on the speed and the lateral position of the truck are minimized to maneuver the vehicle.

Two different formulations for collision avoidance are introduced. Despite its more limited capability to capture highway scenarios compared to the so-called sigmoidal predicted collision free corridor, the so-called rectangular predicted collision-free corridor is chosen for its simplicity. Future research can be directed to evaluate alternatively shaped predicted collision-free corridors.

A full explicit formulation of the OCP is given. All requirements are gathered from the previous chapter and the preceding sections. Based on this formulation we can continue to introduce the solution strategy for this OCP, which is described in detail in the next chapter.

Chapter 4

Direct multiple-shooting solution strategy of the optimal control problem

In the previous chapters we have converged to an infinite dimensional optimal control problem (OCP). We have argued how the solution of the optimization problem describes open-loop optimal trajectories for the A-double combination. At the same time, in Section 3-4 we see that the OCP is formulated as an optimization that involves both a non-linear cost function and non-linear constraints. Different strategies are readily available to solve such problems, among which we have most prominently the so-called direct and indirect methods. In the application of indirect methods, necessary conditions for solution optimality are derived. One tries to solve for these necessary conditions in order to arrive at a optimal solution. Direct methods have a slightly different approach, the solution of the OCP is searched by directly minimizing the cost function numerically. Plenty of literature can be found on the topic of direct and indirect methods for trajectory optimization, see for example [19]. Indirect methods are famous for their ability to find very accurate solutions for the optimal trajectory compared to direct methods. But two main drawbacks of indirect methods lead to the general preference for direct methods for solving OCPs in most engineering applications.

- The region of convergence near the optimal solution of an OCP is generally very small. The initial guess of the optimal path needs to be close to real optimum for stable results.
- The necessary conditions for an optimal trajectory require knowledge on active and inactive inequality constraints in advance. Any switching of inactive and active constraints need to be parameterized up front as well.

In this work, a direct method is used to solve the OCP. More precisely a direct multiple shooting approach is used. A main step in the direct solution methods is to discretize the state and control trajectories, in order to be able to solve a nonlinear program (NLP). The

solution of the NLP is hopefully close to optimal trajectory of the OCP. Two alternatives to direct multiple shooting techniques are, direct collocation methods and direct single shooting techniques. Direct collocation methods parameterize the state and control trajectory using basis functions, such as polynomials. Or as previously applied to trajectory generation for the A-double combination, using B-splines [5]. Direct single shooting methods discretize the control signal over the prediction interval, the state integration is solved as a initial value problem over the entire prediction.

In the direct multiple shooting technique we discretize both the state trajectory and the control input over the receding prediction horizon. The system is separately integrated in each interval between the discretization nodes, based on an initial guess of the initial state at each interval. By introducing continuity constraints, a state prediction is obtained for the OCP. Multiple-shooting algorithms tend to perform better compared to single shooting techniques in numerical stability.

In this chapter the overall solution strategy to the OCP is described and discussed. The algorithm is an implementation of the Real-Time Iteration (RTI) scheme. The algorithm involves details with drastic consequences on both real-time performance and optimality of the intermediate solutions of the OCP at each control step. A trade-off is made between the optimality of the generated trajectories and the solution time. This chapter can be viewed as a review chapter on the RTI scheme, while it is applied specifically to our trajectory generation problem. In order to fully understand the results from this thesis work, one should know the general outline of the RTI. For the interested reader, more detailed work is available on this algorithm in e.g. [6, 20].

For the sake of clarity, contributions in this chapter to the RTI algorithm are explicitly listed. People that are familiar to the RTI algorithm can skip the other sections.

- Section 4-2 treats the discretization of the road geometry and the reference profile.
- In Section 4-4-3 one finds a discussion on the solution quality assessment.
- Finally it is advised to read the summary to find a brief discussion on the consequences of solution strategy on the implementation.

4-1 Introduction to the Real-Time Iteration scheme

About a decade ago, a solution strategy for OCPs was introduced that proposed to exploit the repetitive nature of receding horizon control[21, 6, 20]. It differs from previous methods in the way that it is not attempting to find the solution of the full nonlinear problem at each time step. This idea stands at the basis of a solution technique that is computationally very efficient and allows for real-time implementation of nonlinear model predictive control (MPC). Simply put, the RTI-algorithm is a sequential quadratic programming (SQP) technique implemented in a very specific manner.

SQP [22] is a popular technique to solve nonlinear programs. In SQP the NLP is quadratically approximated in the most recent solution iterate, the step towards a new solution iterate is obtained by solving this intermediate quadratic program (QP). By repeating these steps, one may converge to a solution of the original NLP. In the RTI scheme it is proposed to

synchronize the SQP iterations with the timing scheme of the controller. Hence, in contrast to conventional nonlinear program (NLP) solution schemes, the RTI scheme only performs one QP iteration per sampling instance. The next sampling instance, a new problem is solved initialized by shifted results from the previous iteration and with the newly acquired knowledge about the system from measurements.

The RTI scheme is divided in two phases, a computationally costly *preparation phase*, that is independent of the most recent state measurement. And a *feedback phase* that resembles the steps that need to be taken between the observation of the state measurement and providing the feedback to the controlled plant. In Figure 4-1 the timing scheme of the RTI algorithm is illustrated.

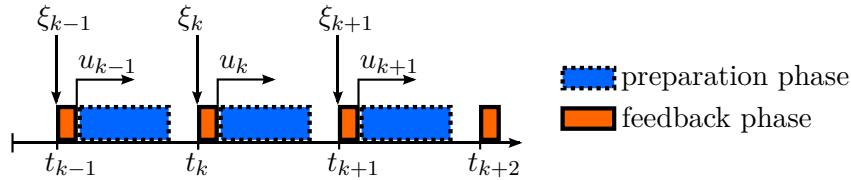


Figure 4-1: The RTI is designed to provide feedback from the OCP as soon as possible after a measurement is taken from the dynamic system.

At time step t_k a measurement is taken, the feedback phase is executed and control u_k is applied immediately thereafter. The preparation phase is entered to prepare a new QP for the next state measurement. Not only do we have an optimization-based control scheme in which we can reach high sampling rates. The need for handling computational delay is avoided by a clever arrangements of the steps of a SQP.

Let us walk through the main steps of the RTI algorithm in the next sections. First the problem needs to be discretized so that we obtain a NLP to solve using the RTI-scheme.

4-2 Problem discretization and Nonlinear Program formulation

A direct solution strategy to an OCP requires a discretized problem. In this section, the discretization steps are described and the parametric NLP is explicitly given.

Consider the general formulation of an optimal control problem, written in the same shape as our OCP in Section 3-4:

$$\begin{aligned}
 & \min_{\xi, u} \int_0^{s_f} \|f(\xi(\sigma), u(\sigma)) - r(\sigma)\|_W^2 d\sigma \\
 & \text{s.t.} \quad \frac{d\xi}{ds} = g(\xi, u, p) \\
 & \quad \xi(0) = \xi_0 \\
 & \quad \underline{h}(s) \leq h(\xi, u, s) \leq \bar{h}(s) \\
 & \quad \underline{u}(s) \leq u(s) \leq \bar{u}(s) \\
 & \quad \forall s \in [0, s_f]
 \end{aligned} \tag{4-1}$$

Here, $\xi \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $p \in \mathbb{R}^l$ are the state vector, the control vector and the parameter vector respectively. $\underline{h} \leq \bar{h}$ are the lower bounds and upper bounds on the set of non-linear

constraints. $\underline{u} \leq \bar{u}$ are upper and lower bounds on the control input, hence box constraints on the controls. ξ_0 is the initial condition for the state vector in the prediction of the OCP. $\mathbb{R}^{N_f} \ni f(\xi(\sigma), u(\sigma)) - r(\sigma)$ is the vector valued objective which we want to minimize in a least-squares sense.

The problem Eq. (4-1) can be discretized over an equidistant spatial grid of $N + 1$ points for the horizon $[0, s_f]$ with step size Δs .

$$0 = s_0 \leq s_1 \leq \dots \leq s_N = s_f \quad (4-2)$$

Where $s \in \mathbb{R}^{N+1}$ denotes the traveled distance of the tractor its center of mass (COM) along the road geometry. The state-vector is discretized so that we obtain a prediction at each node $w \in \mathbb{R}^{nN}$:

$$w = (w_1, \dots, w_N) = (\xi(s_1), \dots, \xi(s_N)) \quad (4-3)$$

And $q \in \mathbb{R}^n$ is the state at the start of the prediction, in the first node.

$$q = w_0 = \xi(s_0) \quad (4-4)$$

The control input is parameterized as a piecewise constant function, constant in each interval $[s_i, s_{i+1}) \forall i \in \{0, \dots, N - 1\}$. The control input parameterization $z \in \mathbb{R}^{mN}$ is defined as.

$$z = (z_0, \dots, z_{N-1}) \quad (4-5)$$

The solution to the differential equation is approximated on each shooting node using the operator Ξ .

$$\xi(s_{i+1}) = \Xi(w_i, z_i, p_i) \quad (4-6)$$

Where $w_i = \xi(s_i)$. Naturally, the operator Ξ can be any explicit update rule of a numerical integration algorithm. The system in our trajectory generator is discretized using a forward-Euler approach. In a forward-Euler integration algorithm, the integration operator $\Xi(w_i, z_i, p_i)$ is defined as:

$$\xi(s_{i+1}) = w_i + \Delta s g(w_i, z_i, p_i) \quad (4-7)$$

where $g(w_i, z_i, p)$ defines the continuous-time spatial derivative of the prediction model and p_i is the parameter vector provided with the differential equation, defined for shooting node i . In the trajectory generation problem described in this thesis, the parameter vector p consists of information about the heading of the road and the parameterization of the predicted collision-free corridor. Other update rules, such as higher order implicit or explicit Runge-Kutta integration algorithms can be considered. To minimize computation time, the computationally light forward-Euler integration algorithm is used.

In this discretized problem, the cost-function is approximated by:

$$\int_0^{s_f} \|f(\xi(\sigma), u(\sigma)) - r(\sigma)\|_W^2 d\sigma \approx \sum_{i=0}^{N-1} \|F(w_i, z_i)\|_W^2 \quad (4-8)$$

The residual of the system dynamics equality constraints in the shooting nodes $\{1, \dots, N\}$ are defined by the following equation.

$$G(w_{i+1}, w_i, z_i) = w_{i+1} - \Xi(w_i, z_i) \quad (4-9)$$

The non-linear inequality constraints and the box constraints on the control input evaluated at each shooting node are:

$$\left. \begin{array}{l} \underline{h}_i \leq H(w_i, z_i) \leq \bar{h}_i \\ \underline{u}_i \leq z_i \leq \bar{u}_i \end{array} \right\} \forall i \in \{0, \dots, N-1\} \quad (4-10)$$

The result is a nonlinear program that approximates the optimal control problem outlined in Eq. (4-1).

$$\begin{aligned} & \min_{w, q, z} \sum_{i=0}^{N-1} \|F(w_i, z_i)\|_W^2 \\ \text{s.t. } & 0 = G(w_{i+1}, w_i, z_i) \\ & q = \xi(0) \\ & \underline{h}_i \leq H(w_i, z_i) \leq \bar{h}_i \\ & \underline{u}_i \leq z_i \leq \bar{u}_i \\ & \forall i \in \{0, \dots, N-1\} \end{aligned} \quad (4-11)$$

Let us rewrite the NLP so that we obtain a general least-squares formulation, an important detail in the RTI-algorithm. We lose the sum in the cost function and obtain one expression for the inequality constraints in all shooting nodes.

$$\begin{aligned} & \min_{w, q, z} \|\mathbf{F}(w, q, z)\|_W^2 \\ \text{s.t. } & 0 = \mathbf{G}(w, q, z) \\ & q = \xi(0) \\ & \underline{h} \leq \mathbf{H}(w, q, z) \leq \bar{h} \\ & \underline{u} \leq z \leq \bar{u} \end{aligned} \quad (4-12)$$

Where the symbols denoted in bold font, $\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{W}$ are defined as the following augmented vectors and matrices:

$$\begin{aligned} \mathbf{F}(w, q, z) &= \begin{pmatrix} F(q, z_0) \\ F(w_1, z_1) \\ \vdots \\ F(w_{N-1}, z_{N-1}) \end{pmatrix} \in \mathbb{R}^{N_f N} & \mathbf{G}(w, q, z) &= \begin{pmatrix} G(w_1, q, z_0) \\ G(w_2, w_1, z_1) \\ \vdots \\ G(w_N, w_{N-1}, z_{N-1}) \end{pmatrix} \in \mathbb{R}^{n N} \\ \mathbf{H}(w, q, z) &= \begin{pmatrix} H(q, z_0) \\ H(w_1, z_1) \\ \vdots \\ H(w_{N-1}, z_{N-1}) \end{pmatrix} \in \mathbb{R}^{N_c N} & \mathbf{W} &= \begin{pmatrix} W & & & \\ & W & & \\ & & \ddots & \\ & & & W \end{pmatrix} \in \mathbb{R}^{N_f N \times N_f N} \end{aligned}$$

An important ingredient to the discretization is the discretization step size Δs . It is however not relevant to the remainder of this chapter, describing the general solution algorithm. In Section 5-3, the discretization step size is selected and motivated. Let us start to discuss the solution algorithm to this NLP, for which we start with the so-called preparation step.

4-3 Preparation step in the Real-Time Iteration scheme

The goal of the preparation step is to prepare the quadratic program in the feedback step as much as possible, without the knowledge of the most-recent state measurement. In this section, all steps in this preparation step are briefly outlined.

In line of the sequential quadratic programming (SQP) methodology, the NLP is locally approximated in latest solution iterate $(w^{k-1}, q^{k-1}, z^{k-1})$ by a quadratic function. Consider the NLP in Eq. (4-12), it can be locally approximated by the QP:

$$\begin{aligned} \min_{\Delta w, \Delta q, \Delta z} & \frac{1}{2} (\Delta w^T \quad \Delta q^T \quad \Delta z^T) \underbrace{\begin{pmatrix} \nabla_{ww}^2 \mathcal{L} & \nabla_{qw}^2 \mathcal{L} & \nabla_{zw}^2 \mathcal{L} \\ \nabla_{wq}^2 \mathcal{L} & \nabla_{qq}^2 \mathcal{L} & \nabla_{zq}^2 \mathcal{L} \\ \nabla_{wz}^2 \mathcal{L} & \nabla_{qw}^2 \mathcal{L} & \nabla_{zz}^2 \mathcal{L} \end{pmatrix}}_{\mathcal{B}} \begin{pmatrix} \Delta w \\ \Delta q \\ \Delta z \end{pmatrix} + \\ & + \underbrace{\begin{pmatrix} \nabla_w^T \mathcal{L} & \nabla_q^T \mathcal{L} & \nabla_z^T \mathcal{L} \end{pmatrix}}_{\mathcal{J}} \begin{pmatrix} \Delta w \\ \Delta q \\ \Delta z \end{pmatrix} \\ \text{s.t. } & \mathcal{H}(\Delta w, \Delta q, \Delta z) \leq 0 \\ & \mathcal{G}(\Delta w, \Delta q, \Delta z) = 0 \end{aligned} \tag{4-13}$$

Here, \mathcal{H} is the vector of inequality constraints and \mathcal{G} the vector of expressions for the equality constraints, $\mathcal{L} = \mathcal{L}(w, q, z)$ is the Lagrangian function of the optimization problem:

$$\mathcal{L}(w, q, z) = \underbrace{\sum_{i=0}^{N-1} \|F(w_i, z_i)\|_W^2}_{L} + \mu^T \mathcal{G} + \nu^T \mathcal{H} \tag{4-14}$$

With μ and ν the vectors of Lagrange multipliers. The biggest concern in sequential quadratic programming is to efficiently obtain (an approximation of) \mathcal{B} . The solution of this QP is used to move towards a local optimum of the NLP provided that the algorithm converges.

In this section, the method to find the approximate quadratic program is outlined. The merits and shortcomings of the algorithm are discussed.

4-3-1 Definition of the road geometry

In preparation of the SQP-step, an updated road definition is provided to initialize the system dynamics equations in $\mathbf{G}(w, q, z) = 0$. The required parameters to define the road geometry are $\phi_{R,1}(s)$ and $\phi_{R,4}(s)$. Since, the prediction horizon of the optimal control problem is defined

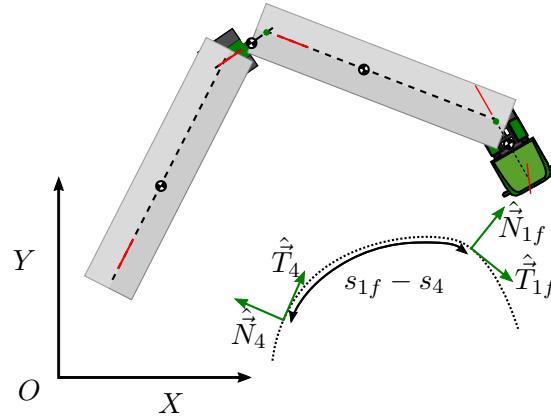


Figure 4-2: An illustration of the scenario when the distance between the Frenét frames for the front axle of the tractor and the rear axle of the second semi-trailer are not a truck length apart.

over the traveled distance of the tractor along the road geometry, providing the road geometry for the tractor is straightforward. At each shooting node of the nonlinear program, the road geometry is known up front. Hence, we obtain a set of road headings relative to the heading at the start of the prediction.

$$(\phi_{R,1,0}, \phi_{R,1,1}, \dots, \phi_{R,1,N}) = (0, \phi_{R,1}(s_1) - \phi_{R,1}(s_0), \dots, \phi_{R,1}(s_N) - \phi_{R,1}(s_0)) \quad (4-15)$$

The road heading at the rear axle of the second semitrailer is less straightforward to define. In fact, we need to provide the expression for the curvature profile and integrate that for the position of the last axle to accurately obtain the heading. This is necessary because we do not know the position of the last-axle at each step in the prediction. However, one can question how much the distance along the road between the front axle of the tractor and the rear-axle of the second semitrailer varies. Especially, considering the low-curvature conditions for which the trajectory generator is designed.

The heading for the rear axle of the second semi-trailer is provided as a known quantity at each shooting node. This avoids the need for online prediction of the rear axle it position and to parameterize the road heading using e.g. a polynomial. To illustrate how this can be an inappropriate assumption, consider the situation in Figure 4-2. The distance between the front axle of the tractor and the rear axle of the second semi-trailer are not a truck length apart in this case.

4-3-2 Shift previous SQP solution

The latest SQP solution contains an open-loop optimal prediction of a great part of the new horizon for the next control interval. In fact, only the last shooting node is not known yet. To initialize the new QP-step, the solution of the previous iterate is shifted to match the shooting nodes of the new discrete horizon. The state prediction at the last shooting node is obtained by integration, assuming that the control action is constant at the end of the horizon.

4-3-3 Constrained Gauss-Newton algorithm

Let us now discuss the method to locally approximate the NLP to obtain a QP. Since the objective is a nonlinear least-squares problem, a Gauss-Newton algorithm can be used. Consider the Lagrange-term L of least-squares form,

$$L(w, q, z) = \|\mathbf{F}(w, q, z)\|_{\mathbf{W}}^2 \quad (4-16)$$

in that case the Hessian can be approximated cheaply. In line with the Gauss-Newton algorithm, consider that we can drop all terms in the exact Hessian that contain the residual $\mathbf{F}(w, q, z)$. The Hessian \mathcal{B} in Eq. (4-13) can approximated by:

$$\mathcal{B} \approx \begin{pmatrix} \nabla_w \mathbf{F} \\ \nabla_q \mathbf{F} \\ \nabla_z \mathbf{F} \end{pmatrix} \mathbf{W} \begin{pmatrix} \nabla_w^T \mathbf{F} & \nabla_q^T \mathbf{F} & \nabla_z^T \mathbf{F} \end{pmatrix} \quad (4-17)$$

The greatest advantage of this approach is that a simple evaluation of the matrix-valued functions $\nabla_w \mathbf{F}, \nabla_q \mathbf{F}, \nabla_z \mathbf{F}$ and a matrix multiplication give us an updated Hessian. However, for the Hessian approximation in the Gauss-Newton algorithm to be effective one or both of the following conditions must hold:

- The objective function of the NLP behaves linear near the optimal solution (w^*, q^*, z^*) , i.e.
 $\nabla^2 \mathbf{F}(w^*, q^*, z^*) \approx 0$.
- The residual of the objective function of the NLP is small near the optimum solution (w^*, q^*, z^*) , i.e. $\mathbf{F}(w^*, q^*, z^*) \approx 0$.

The computation of the Jacobian \mathcal{J} of the objective function in the quadratic program (Eq. (4-13)) is approximated by:

$$\mathcal{J} \approx \begin{pmatrix} \nabla_w^T L & \nabla_q^T L & \nabla_z^T L \end{pmatrix} \quad (4-18)$$

We can now write the complete description of each intermediate QP problem of the constrained Gauss-Newton algorithm.

$$\begin{aligned} \min_{\Delta w, \Delta q, \Delta z} & \left\| \mathbf{F} + (\nabla_w^T \mathbf{F}) \Delta w + (\nabla_q^T \mathbf{F}) \Delta q + (\nabla_z^T \mathbf{F}) \Delta z \right\|_{\mathbf{W}}^2 \\ \text{s.t.} \quad 0 = & \mathbf{G} + (\nabla_w^T \mathbf{G}) \Delta w + (\nabla_q^T \mathbf{G}) \Delta q + (\nabla_z^T \mathbf{G}) \Delta z \\ \xi(0) = & q + \Delta q \\ \underline{h} \leq & \mathbf{H} + (\nabla_w^T \mathbf{H}) \Delta w + (\nabla_q^T \mathbf{H}) \Delta q + (\nabla_z^T \mathbf{H}) \Delta z \leq \bar{h} \\ \underline{u} \leq & z + \Delta z \leq \bar{u} \end{aligned} \quad (4-19)$$

The objective function for the least-squares minimization problem $\mathbf{F}(w, q, z)$ and the constraint functions $\mathbf{G}(w, q, z)$ and $\mathbf{H}(w, q, z)$ are linearized at the latest solution iterate (w_k, q_k, z_k) in order to obtain a quadratic program to find the new solution $w_{k+1} = w_k + \Delta w$, $q_{k+1} = q_k + \Delta q$ and $z_{k+1} = z_k + \Delta z$. The expressions for the linearized matrices are off-line analytically derived. In the iteration scheme, the linearization step merely consists of the evaluation of the vector functions \mathbf{F} , \mathbf{G} and \mathbf{H} and the matrices $\nabla_* \mathbf{F}$, $\nabla_* \mathbf{G}$, $\nabla_* \mathbf{H}$ in the latest solution, with $* \in \{w, q, z\}$.

4-3-4 Problem condensing

The *full-size* QP can be simplified greatly by decreasing the number of decision variables. Let us describe so-called condensing steps that can be executed in the preparation steps, thus without the knowledge of the most recent vehicle state. Using linearized continuity conditions of the states in the shooting nodes, the state variables can be eliminated from objective function and the inequality constraints. The gradient, Hessian and linearized constraints are projected onto the space of remaining decision variables (control input of the system and the initial condition).

$$\Delta w = -(\nabla_w^T \mathbf{G})^\dagger \mathbf{G} - (\nabla_w^T \mathbf{G})^\dagger (\nabla_q^T \mathbf{G}) \Delta q - (\nabla_w^T \mathbf{G})^\dagger (\nabla_z^T \mathbf{G}) \Delta z \quad (4-20)$$

The number of decision variables for Δw in the quadratic program can be eliminated. We obtain the partially condensed QP outlined in Eq. (4-21).

$$\begin{aligned} \min_{\Delta q, \Delta z} & \| \left(\mathbf{F} - (\nabla_w^T \mathbf{F}) (\nabla_w^T \mathbf{G})^\dagger \mathbf{G} \right) + \left((\nabla_q^T \mathbf{F}) - (\nabla_w^T \mathbf{F}) (\nabla_w^T \mathbf{G})^\dagger (\nabla_q^T \mathbf{G}) \right) \Delta q \\ & + \left((\nabla_z^T \mathbf{F}) - (\nabla_w^T \mathbf{F}) (\nabla_w^T \mathbf{G})^\dagger (\nabla_z^T \mathbf{G}) \right) \Delta z \|_W^2 \\ \text{s.t. } \xi(0) = & \frac{q + \Delta q}{\left(\mathbf{H} - (\nabla_w^T \mathbf{H}) (\nabla_w^T \mathbf{G})^\dagger \mathbf{G} \right)} \quad (4-21) \\ h \leq & + \left((\nabla_q^T \mathbf{H}) - (\nabla_w^T \mathbf{H}) (\nabla_w^T \mathbf{G})^\dagger (\nabla_q^T \mathbf{G}) \right) \Delta q \leq \bar{h} \\ & + \left((\nabla_z^T \mathbf{H}) - (\nabla_w^T \mathbf{H}) (\nabla_w^T \mathbf{G})^\dagger (\nabla_z^T \mathbf{G}) \right) \Delta z \\ \underline{u} \leq & z + \Delta z \leq \bar{u} \end{aligned}$$

This condensing step is computationally quite intensive, but the step significantly reduces the required steps in the so-called feedback step introduced in the subsequent chapter.

4-4 Feedback step in the Real-Time Iteration scheme

The condensed QP Eq. (4-21) is prepared and the most recent state measurement is taken. The remaining tasks to obtain a new solution iterate is to embed the initial condition and to solve the resulting fully condensed QP.

4-4-1 Initial value embedding of state measurements

Consider the quadratic program from Eq. (4-21), in similar fashion to the elimination of the decision variables Δw we can eliminate Δq using the most recent state measurement.

$$\Delta q = \xi(0) - q \quad (4-22)$$

We finally obtain a fully condensed QP where the sole decision variables are the free variables Δz in the piece-wise constant control input parameterization.

$$\begin{aligned} \min_{\Delta z} & \left\| \mathcal{F} + \nabla_z^T \mathcal{F} \Delta z \right\|_W^2 \\ \text{s.t.} & \quad \underline{h} \leq \mathcal{H} + \nabla_z^T \mathcal{H} \Delta z \leq \bar{h} \\ & \quad \underline{u} \leq z + \Delta z \leq \bar{u} \end{aligned} \quad (4-23)$$

with,

$$\mathcal{F} = \left(\mathbf{F} - (\nabla_w^T \mathbf{F})(\nabla_w^T \mathbf{G})^\dagger \mathbf{G} \right) + \left((\nabla_q^T \mathbf{F}) - (\nabla_w^T \mathbf{F})(\nabla_w^T \mathbf{G})^\dagger (\nabla_q^T \mathbf{G}) \right) (\xi(0) - q) \quad (4-24)$$

$$\nabla_z^T \mathcal{F} = \left((\nabla_z^T \mathbf{F}) - (\nabla_w^T \mathbf{F})(\nabla_w^T \mathbf{G})^\dagger (\nabla_z^T \mathbf{G}) \right) \quad (4-25)$$

$$\mathcal{H} = \left(\mathbf{H} - (\nabla_w^T \mathbf{H})(\nabla_w^T \mathbf{G})^\dagger \mathbf{G} \right) + \left((\nabla_q^T \mathbf{H}) - (\nabla_w^T \mathbf{H})(\nabla_w^T \mathbf{G})^\dagger (\nabla_q^T \mathbf{G}) \right) (\xi(0) - q) \quad (4-26)$$

$$\nabla_z^T \mathcal{H} = \left((\nabla_z^T \mathbf{H}) - (\nabla_w^T \mathbf{H})(\nabla_w^T \mathbf{G})^\dagger (\nabla_z^T \mathbf{G}) \right) \quad (4-27)$$

4-4-2 Quadratic programming and problem expansion

The resulting QP in Eq. (4-23) can be solved using an arbitrary quadratic programming solver. The solver that is used in this thesis work is `qpOASES`, it uses an active set strategy to solve the QP [23]. An active set strategy in quadratic programming exploits the fact that, assuming a set of active inequality constraints, the Karush-Kuhn-Tucker (KKT) conditions define a linear program (LP). By iterating the active set of inequality constraints, an optimal solution can be found. It is out of scope of this thesis to discuss the details of the QP solution strategy. However it is noted that the QP solver uses warm starting. Because of the repetitive nature of the RTI scheme, the next iteration will likely involve a similar problem for the QP solver. The primal and dual solution and the active set of constraints from the previous QP step can be used to initialize the new iteration. In general this leads to faster solution times of the feedback step of the RTI algorithm.

We have obtained the solution to the quadratic subproblem. The solution is expanded using Eq. (4-20) to retrieve Δw^k and Δq^k . Using the relations $w^{k+1} = w^k + \Delta w^k$ and $z^{k+1} = z^k + \Delta z^k$ we obtain predictions of the optimal trajectory and control input.

4-4-3 Assessment of solution quality

A general technique used to assess the solution quality of the latest SQP iterate is the so-called KKT-tolerance or commonly referred to as the KKT-value. It is a positive quantity that can be computed to find whether the KKT-conditions are satisfied. In the PhD-thesis we find the definition [24]:

$$\text{KKT-value} = \left| (\nabla_z^T \mathcal{F}) \Delta z \right| + \sum_{i=1}^l |\mathcal{H} \Delta \nu| \quad (4-28)$$

Where Δz is the primal solution of the QP and $\Delta \nu$ the vector of Lagrange multipliers from the dual solution. The smaller this measure is, the better the solution of the NLP. A zero-value for the KKT tolerance means we have likely converged to a KKT-point of the original

NLP. As mentioned in [24], this value is sometimes used to terminate an sequential quadratic programming algorithm (as soon as it is smaller than a certain absolute tolerance). In the case of the Real-Time Iteration algorithm, this is not an option. Only one QP solution is computed per sampling interval. The value is only tracked to assess the solution quality of this Real-Time Iteration-scheme.

A second assessment method for the solution quality is the objective value itself. Of course an optimum is not necessarily zero in the objective value, but the Gauss-Newton algorithm is affected by the objective value in the optimum. If it is non-zero, the Hessian approximation is biased. This naturally affects the convergence properties of the overall RTI-algorithm.

4-5 Summary

In this chapter the solution algorithm to the OCP is introduced. A direct multiple shooting optimization technique is employed and used in the RTI-algorithm. Instead of solving an NLP each time step, SQP steps are solved synchronous with the sampling intervals of the controlled process. The controller uses a piecewise constant control parameterization for the discretization. A Gauss-Newton algorithm is used to locally approximate the NLP each control interval, to obtain a QP for the latest iterate of the SQP-algorithm. The constrained Gauss-Newton algorithm assures a computationally light Hessian approximation. But its downside is that the Hessian approximation is only effective if the optimum has a zero objective. Problem condensing is applied to reduce the search-space of each intermediate QP, to be solved by an active set QP solver.

In the next chapter, the implementation of the control algorithms is described.

Chapter 5

Real-time implementation of the trajectory generator

This chapter discusses the real-time implementation of the trajectory generator. An optimal control problem (OCP) is formulated and a solution technique for this OCP is introduced in the previous chapters. The goal of this chapter is to give the reader insight in the implementation details and understand how challenges in this area are tackled.

First, in Section 5-1 the two simulation environments in which the trajectory generator is implemented are described. The controller is implemented in both SIMULINK environment and a motion simulation environment at the Swedish National Road and Transport Research Institute (VTI).

In Section 5-2 the general control architecture is described. The trajectory generator is placed in the overall truck control architecture. The separate subtasks of the trajectory generator are described to present the reader a clear view of the implemented routines related to the real-time implementation of the optimization-based trajectory generator.

Section 5-3 addresses two scheduling related issues in the real-time implementation of the trajectory generator. It describes how and when the control algorithms are triggered to compute a new optimal trajectory. The solutions schedule, that follows the philosophy of the Real-Time Iteration (RTI) algorithm, is briefly outlined and motivated.

In Section 5-4 reference calculation is discussed. The OCP is designed to be able to cope with the complete trajectory generation problem. However, there is no reason to be naive in providing the reference trajectories for the longitudinal velocity and the lateral position with respect to the lane center. For example, if the curvature of the road is such that the lateral acceleration at the desired speed are exceeding the constraints, we can ease the problem for the optimizer by lowering the velocity request.

Section 5-5 discusses the implementation of the solution strategy to the OCP, described in Chapter 4, using the code generation functionality of the ACADO Toolkit. Advantages of generating tailored code for solving an nonlinear program (NLP) contribute significantly to

the feasibility of implementing the trajectory generator with real-time performance. Using the modeling interface of the ACADO Toolkit, the control engineer is saved from a great deal of the programming work involved with the implementation. The OCP can be defined in a natural mathematical syntax and the solution strategy described in Chapter 4 is automatically translated to tailored C-code.

Finally, in Section 5-6 several details to the implementation of the controller are described for the interested reader. Details related to the implementation, that do not have direct contribution to the results, are discussed.

5-1 Simulation environments

The trajectory generator is implemented in two simulation environments. The so-called desktop simulation environment and the VTI motion simulator environment.

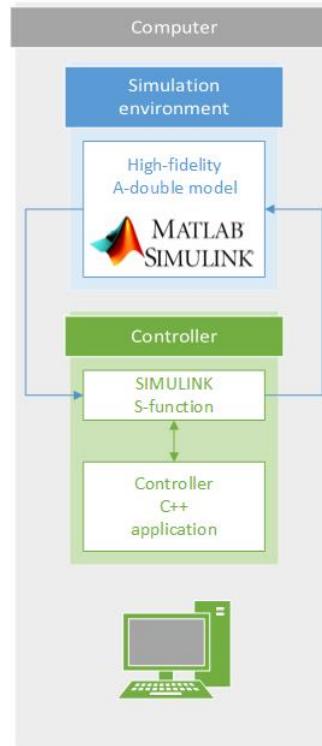


Figure 5-1: Graphical overview of SIMULINK implementation of the trajectory generator. The NMPC, written in C/C++, is interfaced with the SIMULINK models of the A-double combination using an S-function wrapper.

The main control routines are written in C/C++, completely independent from external libraries. Nevertheless, external libraries are used for the simulation environment specific interaction with the vehicle model.

In Figure 5-1 the desktop simulation structure is schematically depicted. Both the controller and the vehicle model run on the same machine. The interface between SIMULINK model of

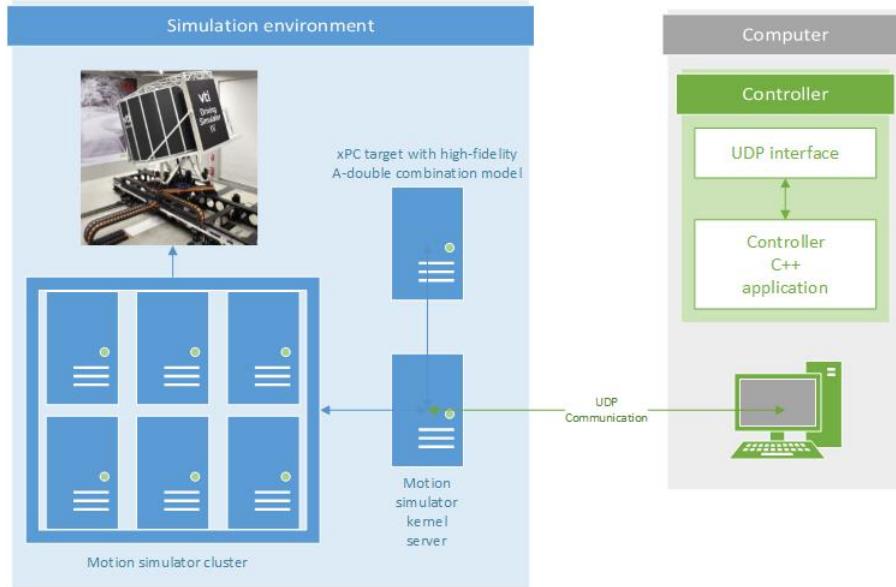


Figure 5-2: Graphical overview of the implementation at the VTI motion simulator. The NMPC is interfaced with the motion support and A-double combination simulation platform via a UDP connection. The controller implementation runs as a stand-alone application on a dedicated PC.

the A-double combination and the controller is arranged using an S-function¹. Timing of the control is dealt with by SIMULINK and no delay is involved (if not modeled explicitly). The road position and the surrounding traffic are also modeled in SIMULINK. The SIMULINK interface allows us to test the controller on both the single-track model introduced in Section 2-2 and the high-fidelity plant of the A-double combination.

Figure 5-2 illustrates the implementation of the trajectory generator with the VTI motion simulator. The vehicle model of the A-double combination is identical to the high-fidelity model that is used in the desktop simulations. The vehicle model runs on an xPC target² to achieve real-time execution. The controller runs as a stand-alone process on a dedicated computer³. A third computer runs a supervisory process that manages the motion control, vehicle model simulation and communicates with the control algorithm over a UDP connection in an internal network. All information that is required by the control routines is sent at a rate of 200[Hz].

The main difference between the SIMULINK implementation and the VTI implementation is undoubtedly the hard requirement for real-time performance of the control algorithm. If in the SIMULINK implementation the control algorithm does not terminate in due time, the simulation is simply delayed. In fact, all routines are assumed to be finished within the sampling time of 0.001[s] of the VTM plant. Hence, no source of delay is present in the simulation. One needs to explicitly model the delay if desired. The VTI implementation is arranged differently, the vehicle model does progress during the execution time of the controller. Hence, there is a delay that one cannot directly influence and if the controller

¹An S-function is a SIMULINK block written in C/C++. The C/C++ S-functions are compiled as MEX files and executed by SIMULINK during the simulation.

²An xPC target is a computer running a SIMULINK model in a real-time operating system.

³Arch Linux, Intel Xeon X5660 processor, 6GB DDR3-1333MHz memory

terminates too late, it directly means that the vehicle model has missed a control signal. If the controller is persistently too late, or if delay is not dealt with appropriately, artifacts in the closed-loop behavior of the plant model is to be expected. This handling of delay is subject to discussion in Section 5-3.

5-2 Control hierarchy

In Figure 5-3 a schematic representation of the overall control hierarchy is depicted. The trajectory generator is externally provided with information about the road $\{\kappa_R\}$ and l_R . Where $\{\kappa_R\}$ which represent the curvature of the road geometry, for both the tractor and the rear axle over the prediction horizon. And l_R is a collection of information about the drivable lanes, this includes the lane width, the number of lanes, etcetera. Also, for a set of obstacles the instantaneous distance $\Delta s_{o,*}$ from the front axle of the tractor, lane $l_{o,*}$, velocity $v_{o,*}$ and acceleration $a_{o,*}$ along the road geometry is provided. The state vector of the truck ξ_{lhvc} is available.

$$\xi_{lhvc} = \begin{pmatrix} v_{y,1} & \theta_0 & \dot{\theta}_0 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & v_{x,1} & a_{x,1} & d_1 & d_4 & \delta \end{pmatrix}^T \quad (5-1)$$

The intermediate desired acceleration for the cruise-control velocity set-point is provided as an initial condition for the prediction model in the OCP.

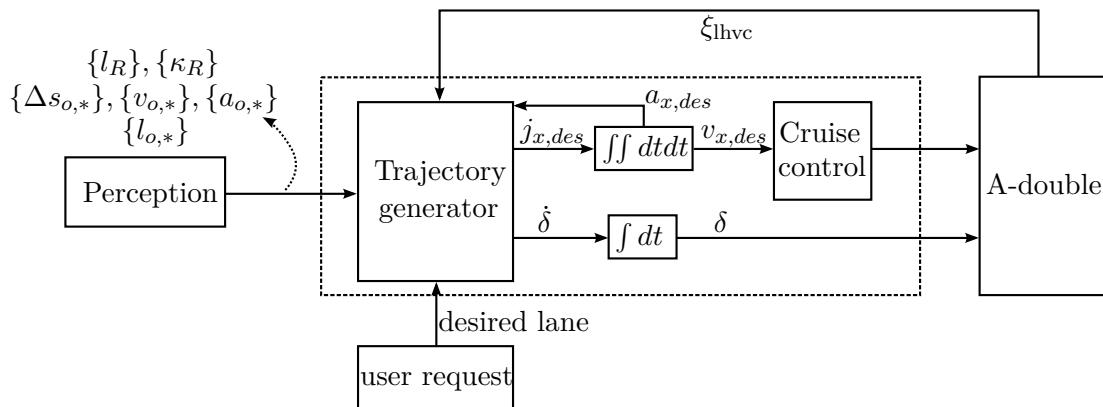


Figure 5-3: Control hierarchy of the trajectory generator in closed-loop with the A-double combination.

The trajectory generator generates a desired longitudinal jerk $j_{x,des}$ for the cruise controller and a desired steering angle rate $\dot{\delta}$. To interface the trajectory generator with the cruise controller, the desired longitudinal jerk is integrated twice over time. The steering angle rate is integrated over time once, to provide a reference for the steering actuator.

The driver can request the trajectory generator externally to change lane. The trajectory generator is provided with this request to make the necessary preparations and predict a suitable trajectory.

5-3 Control scheduling and handling of delay

The OCP for the trajectory generation controller solves a discrete problem each control interval. The discretization of the OCP is achieved by selecting points of the trajectory prediction on an equidistant grid over the spatial prediction horizon. As introduced in Chapter 4, the solution strategy to the NLP relies on a timing scheme that is synchronized with the problem discretization. Let us motivate how the discretization step size Δs is selected and how computational delay is handled.

5-3-1 Selection of sampling rate

Digital control research teaches us how to select a suitable sampling rate. In essence, one must choose a sampling rate that is low enough to avoid finite precision problems in computers. On the other hand, it must sufficiently high to avoid aliasing. A general *rule-of-thumb* suggests to pick a sampling rate that is at least ten times faster than the fastest dynamics in the system.

The fastest eigenvalue of lateral dynamics of the A-double combination prediction model is dependent on the velocity of the truck. Based on the minimum velocity constraint of $8[m/s]$ we see in Table 2-1 that the fastest eigenvalues are the pair located at $-13.68 \pm j2.50[rad/s]$. Hence, the fastest lateral dynamics of the A-double combination are in the range of $2[Hz]$. We notice though, that the cruise control velocity tracker is modeled as a first-order system with a time constant of $\tau_{cc} = 0.25[s]$ agreeing to $4[Hz]$ dynamics. Hence, we conclude that cruise-controller is the limiting factor. According to the rule of thumb, a sampling rate higher than $40[Hz]$ is desired.

On the other side of the trade-off is the computational performance of the control algorithm. The time that is required by the optimization algorithm to prepare the control input for the new interval. Two important considerations are made:

- The solution time of the trajectory generator must be lower than the sampling time.
- The solution time of the feedback step of the RTI-scheme must be an order of magnitude faster than the sampling time.

From the general results in Chapter 6 we see that this computation time is in the range of $0.02 - 0.03[s]$. The average time consumed by the feedback-step is in the range of $3 - 5[ms]$. Based on this observation, a temporal sampling in the range of $50[ms]$ is appropriate.

Recall that the discretization took place in a spatial domain, the temporal sampling must be converted. The spatial sampling rate can be approximated by $\Delta s = \frac{ds}{dt}\Delta t$. The A-double combination is allowed to drive between $8 - 23[m/s]$, which leads us to a sampling distance in the range of $0.4 - 1.2[m]$. A step size of $1[m]$ is used in the remainder of the work. It would be appropriate to switch control algorithms for different velocity levels, since the choice of $1[m]$ results in a temporal sampling of $8[Hz]$ at $8[m/s]$. This is on the edge of the Nyquist frequency for the cruise control dynamics.

5-3-2 Scheduling

The implementation for the triggering of the controller is based on the measurement from the simulation environment. The traveled distance along the road geometry is checked continuously. A new control step is triggered when the discrete step size Δs has passed. It depends on the implementation of the controller how often the triggering process can check the conditions. In the desktop simulations in SIMULINK, the controller is called at a rate of 1000[Hz]. At the VTI facilities, a UDP connection sends data at rate of 200[Hz].

If the distance Δs is traveled since the last iteration, the feedback step of the RTI-scheme is started. In an ideal scenario, the feedback step would terminate immediately. We would be able to apply the control input at the same time a measurement is taken. This is not the case however, time has passed and the initial condition for which the open-loop optimal trajectory is computed is not entirely valid anymore.

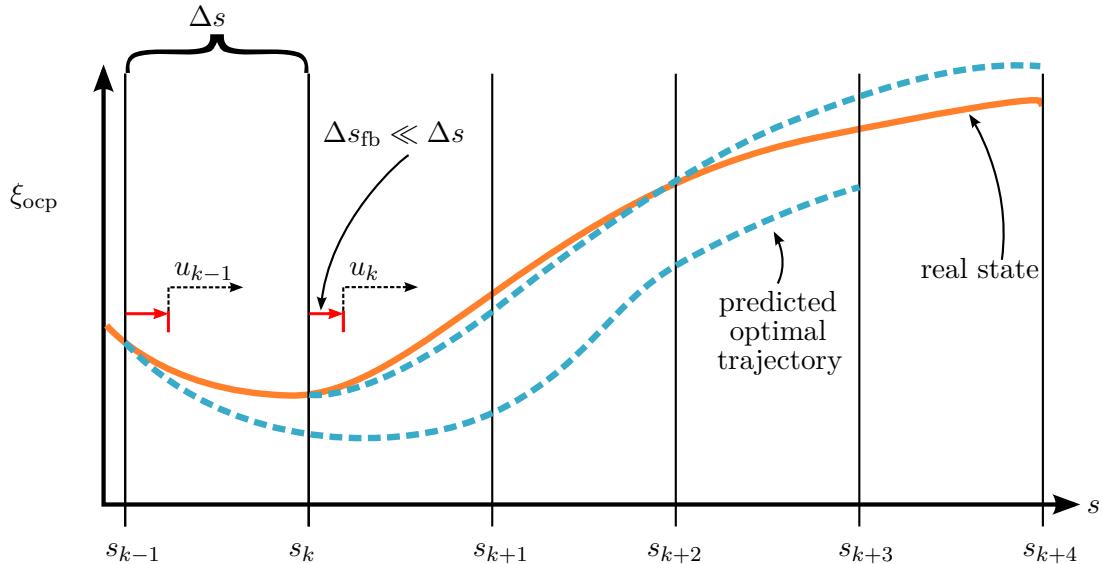


Figure 5-4: The control step is applied asynchronously from the measurement steps. For measurement step s_k , an optimal trajectory is computed. Δs_{fb} later, the time/space consumed by the feedback step of the solution algorithm, the control action is applied.

This problem of delay is commonly solved by making the delay deterministic and to account for this delay by design. The RTI-algorithm has a different philosophy regarding this matter. As extensively described in Section 4-1, the algorithm intends to minimize the number of numerical operations required after a measurement is obtained. By ensuring the time taken by this so-called feedback step is short, in fact an order of magnitude shorter than the sampling, the mismatch in initial condition is hopefully negligible. See Figure 5-4 for an illustration.

The control algorithm in this thesis follows this philosophy of the RTI-scheme. As we will see in Chapter 6, the feedback step indeed is an order of magnitude lower than the sampling rate. It can be concluded that no explicit technique for delay handling is in place, we rely on a short feedback step time.

Finally, the description of the general scheduling algorithm in the trajectory generator:

1. From time-step $k - 1$ we have an iterate of the optimal trajectory.

2. Shift the previous solution to the new receded horizon and obtain the unknown data point by integration.
3. Prepare the reference trajectories and the road parameterization for the upcoming prediction horizon $[s_k, s_k + s_f]$. Where s_f is the spatial prediction horizon.
4. Do all possible preparations (preparation step of RTI-scheme) independent from the knowledge on the most recent vehicle state at sampling instance s_k .
5. Wait for sampling instance s_k and take a measurement
6. Solve quadratic program (QP) (feedback step of RTI-scheme) and apply feedback u_k immediately.
7. Repeat.

5-4 Reference calculation

The OCP is designed to tackle the complete trajectory generation problem. However, with logic we are able to assist the optimization very easily. Consider the OCP as described in Section 3-4. The formulation bounds the lateral acceleration in the tractor and in the second semi-trailer. In the cost function however, there is no incentive to minimize the lateral acceleration whatsoever. This is perfectly in line with the desired driving behavior. But consider now that a tight corner appears in the prediction, cornering imposes a steady-state lateral acceleration. It cannot be our goal to find a trajectory that exactly satisfies the lateral acceleration bounds. The OCP reflects otherwise. However, since velocity is the main contributing factor to the lateral acceleration, we can simply adjust our desired velocity according to the curvature. In this section, a set of heuristics are described and motivated that lead to the reference calculation. The reference calculation is a pre-processing step to the optimization.

It is not uncommon in the field of model predictive control (MPC) to adjust the reference profile to the dynamic optimization problem. Target calculation [25] can be applied to find a reference that is strictly feasible for the system to achieve in steady-state. Situations can be found, where systems with more outputs than the number of inputs cannot converge to all reference points. Although the name *reference calculation* is derived from *target calculation*, the philosophies of the techniques do not entirely match. The reference calculation is not aimed to guarantee feasibility of the trajectory, but rather a method to involve human knowledge into the solution strategy.

Reference tracking is imposed on two states. The longitudinal velocity and the lateral position of the tractor in the center of mass (COM). Both reference calculation problems are considered independent from one another. Let us first introduce the velocity reference calculation and thereafter discuss the lateral reference calculation technique.

5-4-1 Reference calculation for the longitudinal velocity

Generating a velocity reference profile for the optimal control problem is considered as a trade-off between the following three matters.

1. We desire a velocity that agrees with the speed limit of the lane, to minimize travel time.
2. Find a velocity profile that has comfortable lateral acceleration properties caused by the lane curvature.
3. Generate a velocity profile that achieves intuitive and comfortable car following behavior.

The reference calculation is implemented by the heuristic to maximize the velocity reference on each shooting node. Considering each element in the trade-off poses a maximal velocity, one simply takes the minimum of the results.

Desired longitudinal velocity from truck driver or legal speed limit

The first requirement for the legal speed limit is the simplest of the list. We consider that the speed limit is constant over the prediction horizon and we obtain a velocity reference as follows:

$$v_{x,1,\text{ref}} = v_{x,1,\text{legal}} \quad (5-2)$$

Comfortable lateral acceleration due to lane curvature

The steady-state lateral acceleration due to the curvature of the lane solely depends on the velocity of the truck tangent to the road ($\frac{ds_1}{dt}$) and on the road curvature (κ_R). Consider we want to bound this steady-state lateral acceleration:

$$-a_{y,\text{comf}} \leq \kappa_R(s_1) \left(\frac{ds_1}{dt} \right)^2 \leq a_{y,\text{comf}} \quad (5-3)$$

We can rewrite this inequality to the following expression, that provides a bound on the velocity tangent to the road:

$$\frac{ds_1}{dt} \leq \sqrt{\left| \frac{a_{y,\text{comf}}}{\kappa_R(s_1)} \right|} \quad (5-4)$$

Because the controller is meant for high-way scenarios, we have: $\frac{ds_1}{dt} \approx v_{x,1}$. Hence the requirement is now compatible with the reference velocity.

$$v_{x,1,\text{ref}} \leq \sqrt{\left| \frac{a_{y,\text{comf}}}{\kappa_R(s_1)} \right|} \quad (5-5)$$

The road geometry properties are conveniently defined at each shooting node in the spatial prediction horizon. Hence, the velocity reference can be computed beforehand the optimization step.

Car following

The reference calculation will attempt to mimic human behavior in speed adaptation to preceding vehicles as much as possible. However, to avoid the use of many tuning variables it is also desired to have a conceptually simple approach to decide on the reference velocity.

In Figure 5-5 one finds an illustration of a tentative decision structure to the car following problem. Based on the intermediate velocity reference and the distance with the leading vehicle, we potentially decrease the velocity reference. The tuning variables are two parameters resembling time gaps. One time gap concerns the deceleration behavior when a leading vehicle is approached (τ_{lo}) and one time gap defines the desired acceleration behavior in case the distance with the preceding vehicle grows larger (τ_{hi}).

Using distinct time gaps, the sensitivity of the ego acceleration behavior to the trajectory of the preceding vehicle hysteresis. If the truck decelerates for a vehicle, it will not immediately accelerate when this same vehicle momentarily speeds up or varies its speed continuously.

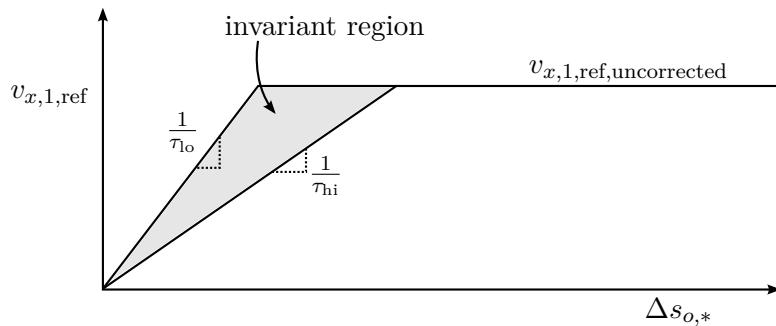


Figure 5-5: Illustration of the sensitivity areas to actions of the car that is being followed. At each time instant, the depicted figure represents a decision diagram. The velocity reference is defined by the projection parallel to velocity reference axis of the coordinate of by the current velocity and distance with the leading vehicle onto the graph. In the invariant region, the velocity reference is kept the same as the current velocity.

Naturally, this reference calculation technique requires integration and a temporal prediction of the surrounding vehicles. A velocity reference for each shooting node is obtained by integration of the velocity difference with the preceding vehicle, assuming that the velocity reference is perfectly tracked. In case of a lane change, the so-called preceding vehicle is actually a set of two. The minimum distance to the vehicle in the departure lane and the goal lane is decisive.

5-4-2 Reference calculation for the Lateral position

The lateral position reference calculation refers to the reference for state d_1 , the lateral position of the COM of the tractor with respect to the lane center. There is one main trade-off in the OCP for optimal trajectory of the the lateral dynamics. It is the trade-off between the lateral reference tracking performance and the lateral jerk.

If a lane change is initiated, the lateral reference transitions to desired lane center. The goal of the lateral reference calculation is to derive a pre-optimized reference trajectory. In this

section, the result of a simplified optimization problem is presented that defines the optimal reference trajectory for the lateral position d_1 . In Section A-1 one finds the full details of the derivation of the solution to the simplified optimization problem.

Consider the cost function for the lateral motion of a point mass system.

$$\min_u \int_0^{\Delta s_{lc}} u(\sigma) d\sigma \quad (5-6)$$

with Δs_{lc} the spatial length of the lane change maneuver along the road geometry. Where $u(s)$ is the lateral jerk, also the input to the point-mass model. Using an indirect solution strategy we find an analytical solution to the two-point boundary value problem. The optimal trajectory for the point mass system is defined by the following quintic polynomial.

$$d_{1,\text{ref}}(s) = -\frac{6(d_{1,0} - d_{1,1})(s_1 - s_{lc})^5}{\Delta t_{lc}^5 v_{x,\text{base}}^5} + \frac{5(d_{1,0} - d_{1,1})(s_1 - s_{lc})^3}{\Delta t_{lc}^3 v_{x,\text{base}}^3} - \frac{15(d_{1,0} - d_{1,1})(s_1 - s_{lc})}{8\Delta t_{lc} v_{x,\text{base}}} + \frac{1}{2}(d_{1,0} + d_{1,1}) \quad (5-7)$$

In Table 5-1, the explanation of the parameters in the quintic polynomial is given.

Table 5-1: Free parameters in the reference calculation expression for the lane change maneuver.

Parameter	Tuning/Fixed	Explanation
s_1	Fixed	Distance for lateral reference coordinate.
$d_{1,0}$	Fixed	The initial lateral distance of the reference profile.
$d_{1,1}$	Fixed	The final lateral distance of the reference profile.
s_{lc}	Tuning	The position half way the lane change maneuver.
Δt_{lc}	Tuning	The time it takes to complete the lane change maneuver.
$v_{x,\text{base}}$	Fixed	the base velocity of the lane change maneuver.

5-5 Code generation using the ACADO Toolkit

In Chapter 4, the solution strategy for the OCP is introduced. The implementation of this solution strategy is achieved using the automatic C-code generation functionality of the ACADO Toolkit [26]. The ACADO Toolkit implements a modeling environment in which the trajectory generation problem can be defined in a syntax close to the mathematics. A wealth of options can be defined to configure the resulting solution strategy in great detail. A brief description of the code generation steps are:

- First we formulate the OCP by defining the cost function, the inequality constraints and the prediction model.
- Externally defined parameters (for road parameterization and collision avoidance) can be defined on each shooting node, which are interfaced with the solution algorithm.

- The fixed discretization step size is chosen and we choose the length of the horizon.
- We do configurations regarding hard-coding the cost function, bounds for the constraints and the integration technique.
- Tailored C-code is generated for the problem formulation.

The **ACADO Toolkit** symbolically differentiates the expressions for the relevant Jacobian matrices. Online differentiation and inversion is achieved by evaluating analytic expressions that are derived offline.

Automatic source code generation for optimization solution algorithms experiences a recent increase in popularity [27]. The main advantage of automatic code generation is that only the essential algorithmic components are generated. The size of the problem is known in advance, thus fixed dimensions and sparsity patterns can be exploited during the code generation. Redundant computations are avoided as much as possible during run-time of the controller. Because the complete problem is known, static memory allocation in the generated code is possible. This avoids yet another source of overhead, increasing the likelihood of securing real-time performance.

The automatically generated C-code is interfaced with the open-source active set QP solver **qpOASES** [23], used in the feedback step of the RTI-scheme. At the time of writing, auto-generated C-code from **ACADO Toolkit** is interfaced with an embedded version of **qpOASES** (based on version 1.3). The OCP in this thesis work can lead to a positive semi-definite (PSD) Hessian for intermediate QP-steps because of the sigmoid functions for the distance keeping. The embedded version of **qpOASES** only works with positive definite (PD) Hessian matrices and is therefore not compatible with our distance keeping formulation in the OCP. A custom interface is implemented to the newest version of **qpOASES** (version 3.0beta), which employs regularization steps for PSD quadratic programs. It is not obvious whether the use of the newer version is beneficial to the computation time or not. The new version incorporate algorithmic improvements, but does not employ static memory allocation like the embedded version does.

5-6 Implementation details

5-6-1 Decision making

Decision making is a vital component in the trajectory generator. Changing lane, initiating an abort maneuver are inherently discrete decisions. In this thesis work, no attempt is made to introduce an *intelligent* algorithms regarding this matter. Since it is crucial that the trajectory generator is able to make decisions, a simple state-machine is implemented. External signals can influence the state of the controller, such as a request for a lane change or to initiate an evasive maneuver. This ensures insightful behavior, but is obviously not suitable for production implementation.

Consider that we have three vertices in our automaton, each vertex agrees to one of the following conditions:

- The truck is keeping lane.
- The truck is changing lane.
- The truck is executing an abort maneuver.

The rule to change from the lane keeping state, to the lane changing state is defined as:

A lane change is desired and the distance with both the trailing and leading vehicle in the desired adjacent lane are greater than a predefined value.

The condition to change from the lane changing state to the lane keeping state is defined as:

The complete truck is within the lane boundaries of the desired lane.

The abort maneuver is triggered by an external signal. An input to the controller propagates to the decision maker and causes the state to transition to the abort maneuver state. In the current implementation, the state cannot leave this vertex unless the controller is restarted.

5-6-2 Handling infeasibility

It can occur that the NLP and/or the QP are infeasible. Infeasibility can occur because of several reasons.

- Because the initial state of the vehicle violates constraints in the OCP.
- Because a corner is in the prediction horizon that is too tight and the velocity cannot be accounted for within the deceleration bounds.
- Because of convergence issues in the solution strategy of the NLP.

Irrespective of the cause for the infeasibility, it is not desired to use the solution from the optimization. An alternative plan is applied.

The solution algorithm generates a trajectory of length s_f . Hence, the last feasible solution to the NLP has a predicted the optimal control input for several control intervals in the future. The controller stores a set of the predicted future controls in case the optimization was successful. Based on the exit-flag of the QP solver, we either take the most recent solution or reuse the previous optimal trajectory. Note that no guarantee exists that the system will recover from the infeasibility using this strategy. It is merely a heuristic that is considered to be the better alternative compared to choosing to use the latest infeasible solution.

5-6-3 UDP communication

The communication at the VTI motion simulator happens via the User Datagram Protocol (UDP). Packets of data are sent over the local network that connect the different machines, among which the computer running the trajectory generator implementation.

The use of UDP is motivated by the relatively low overhead involved in the communication. The implementation uses the Boost C++ libraries. Normally, the UDP standard states that

the first message that is received has highest priority. In case the receive buffer is full, any new messages are disregarded. In case of our trajectory generator, the newest messages are most important. In fact, older ones lose all their significance as soon as a new one arrives. An algorithm was written to assure that the controller always works with the most recent information available. Separate threads asynchronously update local buffers for the vehicle state, road information and measurements of the surrounding traffic.

5-6-4 Considerations for embedded hardware application

All state variables in the control algorithm are of floating point type with double precision. Embedded application would likely require single precision floating point numbers. A single type definition change is necessary to change operation to single-precision floating point numbers.

The solution algorithm is relatively heavy and the embedded hardware will require similar performance to high-end consumer PCs. Results show that the trajectory generator algorithm consumes about 6MB of memory (double-precision setting), which is a lot for most embedded hardware. It would be challenging to find affordable hardware for the implementation in production vehicles. One would quickly end up with the requirement for Digital Signal Processor-like embedded hardware.

All trajectory generator routines are independent from external libraries. This would significantly decrease the effort required to port the code for embedded purposes. The version of qpOASES used in this thesis work (3.0beta) does not employ static memory allocation. Not all embedded operating systems have an efficient implementation for dynamic memory allocation.

5-7 Summary

In this chapter the main details of the real-time implementation of the trajectory generator are described. All control algorithms are written in C/C++ and implemented in two simulation environments. The controller is implemented on a SIMULINK model for the A-double combination and is interfaced using an S-function. The second implementation is at the motion simulator of the Swedish National Road and Transport Research Institute (VTI), where it is interfaced using UDP-communication.

Reference calculation techniques are used for both the longitudinal velocity and the lateral position-offset reference profiles. The longitudinal velocity reference accounts for curvature of the road and for car following behavior. The lateral position reference calculation assures a pre-optimized lane transition.

The solution algorithm from Chapter 4 is automatically generated using the automatic C-code generation functionality of the **ACADO Toolkit**.

In the next chapter the trajectory generator is tested in closed-loop simulations in SIMULINK. We will verify the performance of the controller applied to both the vehicle prediction model and a high-fidelity vehicle plant model.

Chapter 6

Simulation results and discussion

The previous chapters have described all ingredients to the solution algorithm to the general trajectory generation problem and its implementation. This chapter presents the results from closed-loop experiments of the controller applied to both the single-track model and the high-fidelity VTM plant for the A-double combination. In all experiments the solution time of the control algorithms is tracked in order to assess the real-time performance. The main goal of this chapter is to show the results for common highway maneuvers in insightful conditions and to compare the closed-loop performance for both plant models. We try to find the limitations for the use of the trajectory generator and how it can be improved in the future.

The tuning of the optimal control problem (OCP) is kept constant for the majority of the results presented in this chapter. In Table 6-1 the normalized weighting terms are depicted. For the evasive maneuver the tuning is slightly modified, the changes are introduced in the respective section.

In Section 6-1, a lane change is considered without the presence of surrounding traffic. Lane changes are executed on a straight road and on a road of constant curvature. Closed-loop simulations with both the single-track prediction model and the high-fidelity model show the general performance of the control algorithm to vehicle models of different complexity.

The second scenario presents the merging capabilities of the trajectory generator. The truck merges in a lane for which it needs to slow down. In Section 6-2 simulation results with the high-fidelity plant model are discussed.

The third scenario is focused to discuss the functionality of the controller to execute an evasive maneuver. In Section 6-3 results are presented for simulations of a return to lane maneuver of the single-track model on a curved road and an equivalent abort maneuver of the high-fidelity vehicle model on a straight road.

Finally, Section 6-4 is focused to show the lane center tracking performance of the trajectory generator in a road of constant increasing curvature.

To fully understand the figures from simulation results, one must be familiar with the naming convention. We will find abbreviations such as *STM*, *VMM*, etc. These abbreviations are to

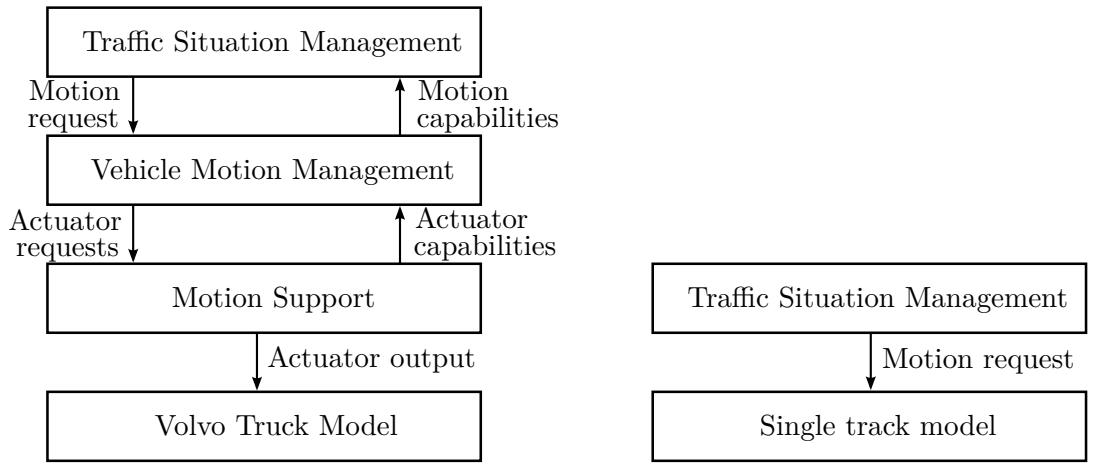
Table 6-1: Standard tuning configuration of controller for the majority of the simulation results.

Parameter	Symbol	Value
Prediction horizon	s_f	100[m]
Discrete step size	Δs	1[m]
Cost weight: d_1	K_{d_1}/n_{d_1}	800/2
Cost weight: $j_{y,1}$	$K_{j_{y,1}}/n_{j_{y,1}}$	100/3
Cost weight: $v_{x,1}$	$K_{v_{x,1}}/n_{v_{x,1}}$	50/17
Cost weight: $a_{x,1,des}$	$K_{a_{x,1,des}}/n_{a_{x,1,des}}$	20/1.75
Cost weight: $\dot{\delta}$	$K_{\dot{\delta}}/n_{\dot{\delta}}$	5/0.05
Cost weight: $j_{x,1,des}$	$K_{j_{x,1,des}}/n_{j_{x,1,des}}$	5/0.05
Cost weight: $\Delta s_{o,2}$	$K_{\Delta s_{o,2}}/n_{\Delta s_{o,2}}$	4200/1
Cost weight: $\Delta s_{o,3}$	$K_{\Delta s_{o,3}}/n_{\Delta s_{o,3}}$	4200/1
Cost weight: $\Delta s_{o,4}$	$K_{\Delta s_{o,4}}/n_{\Delta s_{o,4}}$	4200/1

clarify where signals have been probed in the model. Especially the high-fidelity Volvo Truck Model (VTM) plant consists of multiple modeling layers for the actuators and the vehicle body dynamics. For example, the steering angle that is requested by the trajectory generator will first be fed to an actuator allocation algorithm that evaluates the request. The allocation step decides how the user demand can be best achieved considering the different actuators, such as the steering, power train and brakes. These signals are then fed to models for the propulsion, brakes and steering actuation. Finally, the steering angle is applied to the wheels and its effect propagates through the multi-body dynamic system. Hence, the steering angle is rather an ambiguous term, it can either be the request to the control allocation algorithm, the reference to the steering actuator and the actual road wheel angles of the wheels themselves.

In Figure 6-1 a schematic representation is given of the signal propagation in the high-fidelity plant and single-track A-double combination model.

- Traffic Situation Management (TSM): in this work, the full functionality of this layer is attributed to the trajectory generator. Hence, the signals that are accompanied with the abbreviation TSM are from the control algorithms discussed in this thesis. It provides a motion request to the VMM layer.
- Vehicle Motion Management (VMM): this is the control allocation algorithm that takes input from the TSM and converts the request to low-level control actuator signals. A convex constrained optimization is the basis for this algorithm. In [28] one finds more details on this algorithm.
- Volvo Truck Model (VTM): a library of truck models. In case of this thesis work, this is the high-fidelity plant for the multi-body system that models the A-double combination. One can consider the measures from the VTM plant are closest to what one would measure in real-life experiments.



(a) Architecture for high-fidelity plant model.

(b) Architecture for the single track model.

Figure 6-1: Architectural diagrams of signals for the two vehicle models in SIMULINK.

- Single Track Model STM: the nonlinear single-track model of the A-double combination that is introduced in Section 2-2. In case the single-track model is used, no actuator models for steering, propulsion and brakes are modeled. This perfectly reflects the prediction model in the temporal domain.
- Motion Support: the models for the propulsion, transmission, brakes and the steering actuation. No abbreviation is assigned because no states are tracked in this subsystem.
- Cruise controller (CC): the cruise controller is technically part of the TSM. It consists of a PID controller that is designed to track a velocity reference. It interfaces with the VMM by contributing to the computation for the requested longitudinal acceleration reference.

Many figures show a set of predictions for the trajectory of the state. These predictions are indicated with dashed cyan lines. Not every prediction is plotted, the prediction signals are decimated (to approximately one every two seconds) to keep plots tidy. Recall that predictions from the nonlinear program (NLP) are made along the road geometry. The time equivalent prediction is obtained by relating each sample to a point in time:

$$t_0 + t(s_1) = t_0 + \int_0^{s_1} \frac{dt}{ds_1} ds_1 \quad (6-1)$$

where t_0 is the time at the start of the prediction step.

The so-called desktop simulations are run on a notebook pc¹, results are obtained in closed simulations in SIMULINK and MATLAB 2014a. The controller is compiled to a **mex**-file using GCC-4.7 with the most drastic optimization option (**-O3**). Code generated by the ACADO Toolkit supports parallel execution of the model integration. This can significantly reduce the time consumed by the so-called *preparation phase*. In SIMULINK this option is not enabled

¹Desktop simulations: Arch Linux, Intel i7-2630QM processor, 12GB DDR3-1333MHz memory

because it caused unexpected behavior, that was most likely due to the combination of the compiler and the MATLAB-libraries. It must be considered at all times that the simulations on the desktop computer are accompanied by lots of tasks on the operating system. This implies that any spike or other strange behavior in the solution time of the control algorithm can be caused by unrelated background processes. In an embedded implementation.

6-1 Lane change maneuver

The lane change maneuver is executed by providing the controller with an external lane-change request. Since no other vehicles are in the proximity of the truck, the decision maker immediately approves to initiate the maneuver. Approximately 5 seconds after the lane change request, the lateral distance reference moves from the initial lane to the target lane. The lateral distance off-set reference is pre-optimized for jerk using the lateral reference calculation step. During the lane change itself, the lateral distance constraints are temporarily relaxed to allow for a lane change to take place without violating lateral position constraints. In this section we will evaluate results from such lane change on both a straight road and on a curved road.

The value for the so-called lane change duration Δt_{lc} is identified from test scenarios with human drivers in the Swedish National Road and Transport Research Institute (VTI) motion simulator with the A-double combination. It is identified that a lane change consumes 9.7[s]. This tuning parameter is of relevance in the reference calculation for the lateral position.

6-1-1 Simulations with single-track model

Lane change on straight road

We start with the lane change maneuver on a straight road with simulations on the single-track model. In Figure 6-2 and Figure 6-3 eight sets of signals are plotted against time.

Table 6-2: Details for the lane change scenario on a straight road for the controller in closed-loop with the prediction model.

Parameter	Symbol	Value
Velocity	$v_{x,1}$	20[m/s]
Curvature	κ_R	0[rad/m]
Lane width		3.7[m]
Lane change request time		25[s]

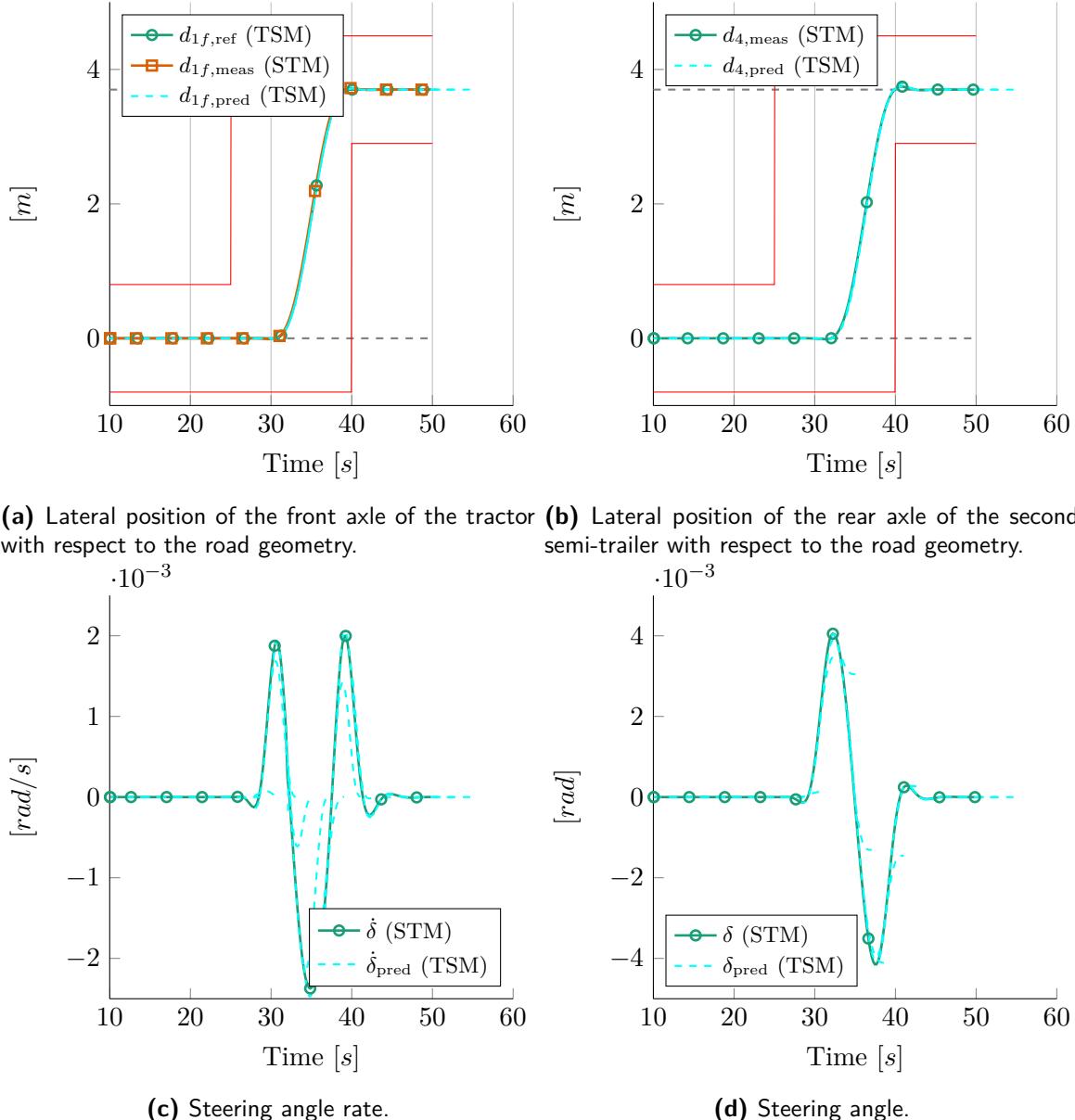
The lane change is requested at 25[s], not much later the vehicle actually moves to the adjacent lane. We see that the trajectory generator plans the trajectory ahead of time almost perfectly. The dashed lines, that represent the prediction, overlap the actual trajectory. This is also reflected by the KKT-value, it is nearly zero at all time. Which means that we are close to an optimum of the original NLP and all constraints are satisfied. One spike is visible in the KKT-value when the truck crosses the lane boundary. The reference frame for the lateral position in the solution algorithm is shifted to the new lane, causes a small disturbance.

The jerk approximation in the OCP seems to accurately represent the finite differences approximation for the time derivative of the lateral acceleration. From the results we see that the lateral jerk perceived at the front axle of the tractor is very low. At the same time, the lateral reference trajectory is tracked well. We cannot underestimate the contributions of a well-initialized reference for a lane change maneuver. If a step change was employed on the reference, it would be much harder to make a trade-off between tracking the lane center and obtaining an acceptable lane change trajectory. The model predictive control (MPC)-algorithm perfectly complements the preparation step by planning the complete trajectory for all states and input to track this jerk-optimal lateral distance profile.

The lateral acceleration in each unit stays very low, far from the imposed limit of $2.5[m/s^2]$. One can observe a slight effect of the rearward amplification (RWA), the lateral accelerations increase a little towards the rear unit and delayed time.

The solution time of the complete algorithm is far within the maximum solution time of $50[m/s]$, the time required to travel $1[m]$ with a speed of $20[m/s]$. The solution-time of the feedback-step is an order of magnitude shorter than the sampling time. This is what we expect from the Real-Time Iteration (RTI)-algorithm. The good performance is no surprise, the trajectory is far within the feasible solution space spanned by the constraints. The intermediate quadratic program (QP) requires very little, to no iterations to converge since active set changes are of little concern. For the lane change maneuver, without modeling-mismatch and an optimum away from the constraints, we conclude that the controller is applicable in real-time.

Let us now continue to a slightly more challenging scenario with the single-track vehicle model, a lane change on a curved road.



(a) Lateral position of the front axle of the tractor **(b)** Lateral position of the rear axle of the second semi-trailer with respect to the road geometry.

(c) Steering angle rate. **(d)** Steering angle.

Figure 6-2: Results from a lane change on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.

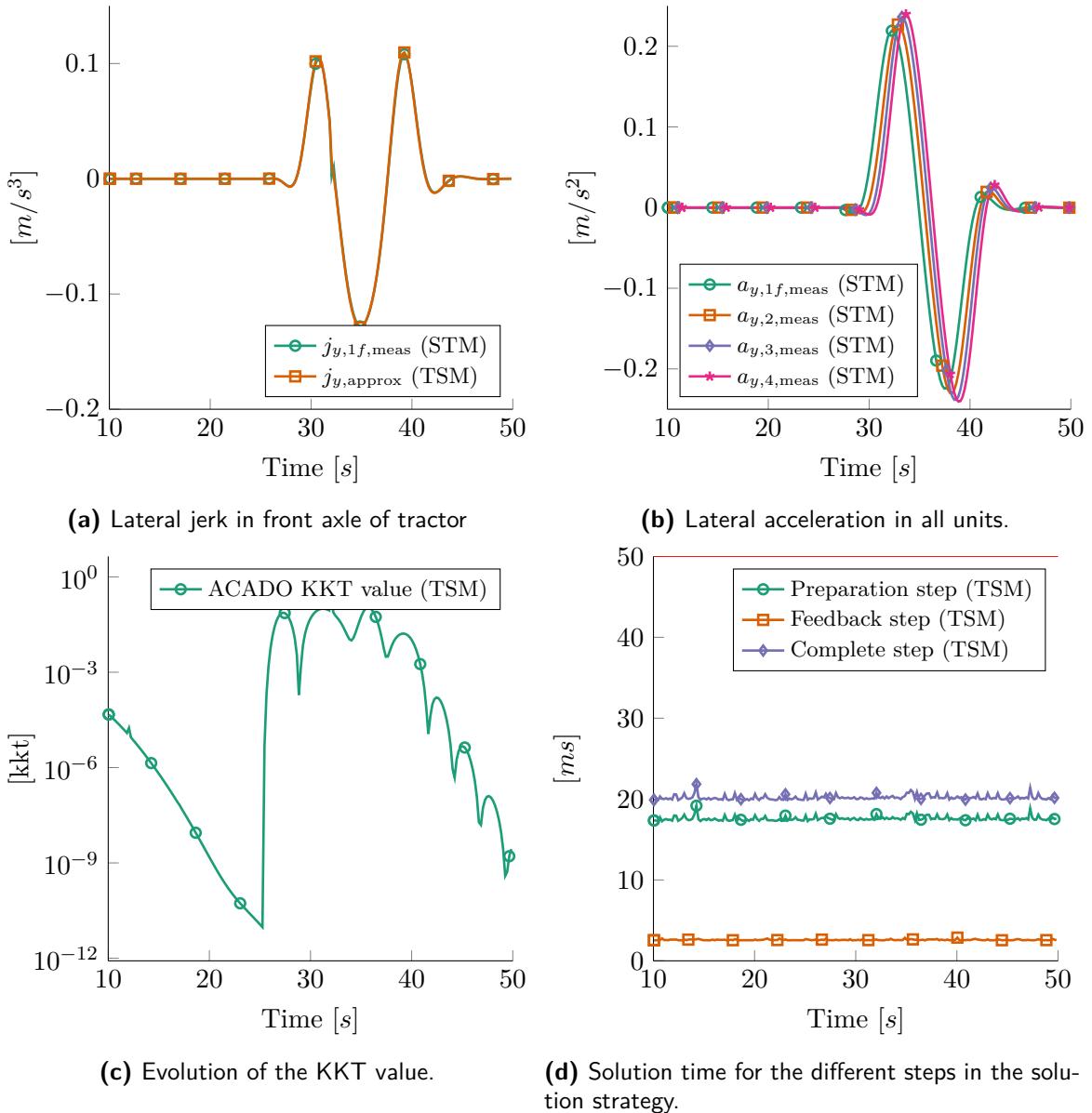


Figure 6-3: Results from a lane change on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.

Lane change on road of constant curvature

In the second scenario, the single-track model changes lanes on a curved highway. The curvature is set to $0.003[\text{rad}/\text{m}]$, which is equivalent to driving a circle with a radius of $333[\text{m}]$. A steady-state lateral acceleration of $1.2[\text{m}/\text{s}^2]$ is to be expected due to the road curvature alone. This can be considered as a high acceleration for the driver comfort. A natural action is to slow down if curves become much tighter. In Table 6-3 the details of the scenario are summarized.

Table 6-3: Details for the lane change scenario on a curved road for the controller in closed-loop with the prediction model. Cost weights for the distance keeping are not listed because of irrelevance in the scenario.

Parameter	Symbol	Value
Velocity	$v_{x,1}$	$20[\text{m}/\text{s}]$
Curvature	κ_R	$1/333[\text{rad}/\text{m}]$
Lane width		$3.7[\text{m}]$
Lane change request time		$25[\text{s}]$

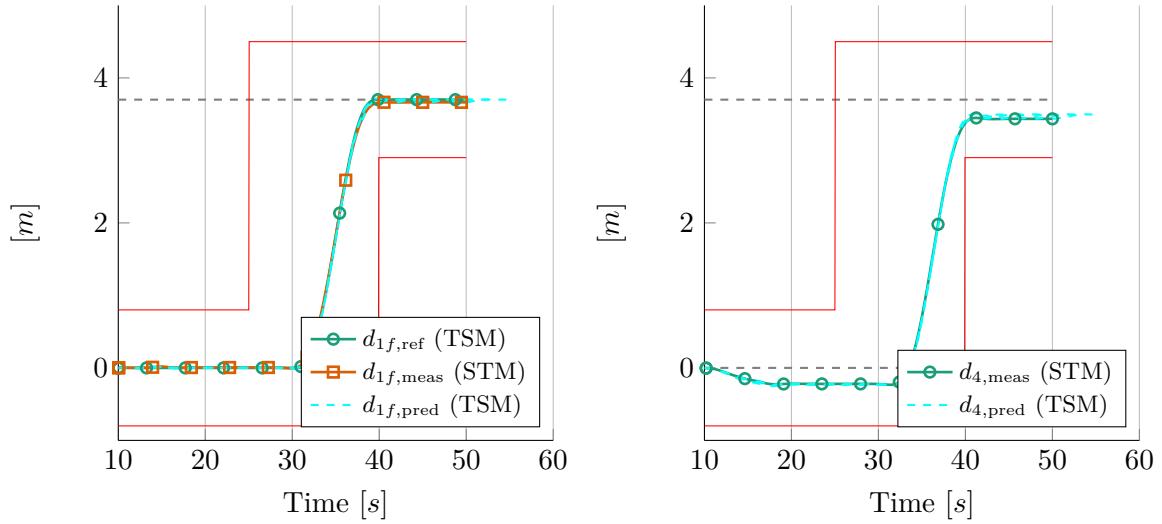
Figure 6-4 and Figure 6-5 depict a selection of signals during the lane change maneuver. We observe that the performance is again very acceptable. Between $10[\text{s}] - 15[\text{s}]$ the truck is entering the curve, any dynamics in that range of time are thus irrelevant to the lane change.

The system does not converge perfectly to the lane center. The few centimeter tracking error can be perfectly explained by the assumptions that are introduced in the spatial model formulation. Most notably, these are the assumption that $\kappa_{R,1}d_1 = 0$ and that the distance $s_1 - s_4 = \text{constant}$. We recognize in the evolution of the KKT-value over time that the solution converges close to an optimum for the NLP. Like in the lane change case on a straight road, a small glitch in the solution algorithm occurs when the truck enters the new lane. This happens at approximately 35 seconds and seems to affect the control action for the steering angle rate. In the measurement of the jerk a similar peak is observed, the jump may be noticeable by the truck driver. The author expects that properly manipulating the intermediate solution will avoid these jumps.

The lateral acceleration transient is close to identical in the case of the lane change on a straight road, shifted with the steady-state lateral acceleration of $\sim 1.2[\text{m}/\text{s}^2]$ due to the road curvature. Jerk is hardly influenced by driving a curved road, rather than a straight road. The optimal trajectory for a lane change thus neither.

Computational performance of the solution algorithm is very similar to the straight lane change simulations. Real-time applicability is no concern. We observe little spikes in the preparation-step of the algorithm. It is very likely these are not related to the algorithm execution time and can be attributed to the other processes on the PC.

We see that the control algorithm perfectly controls the prediction model. Next the performance for the same two maneuvers are evaluated on the high-fidelity vehicle model of the A-double combination.



(a) Lateral position of the front axle of the tractor with respect to the road geometry. **(b)** Lateral position of the rear axle of the second semi-trailer with respect to the road geometry.

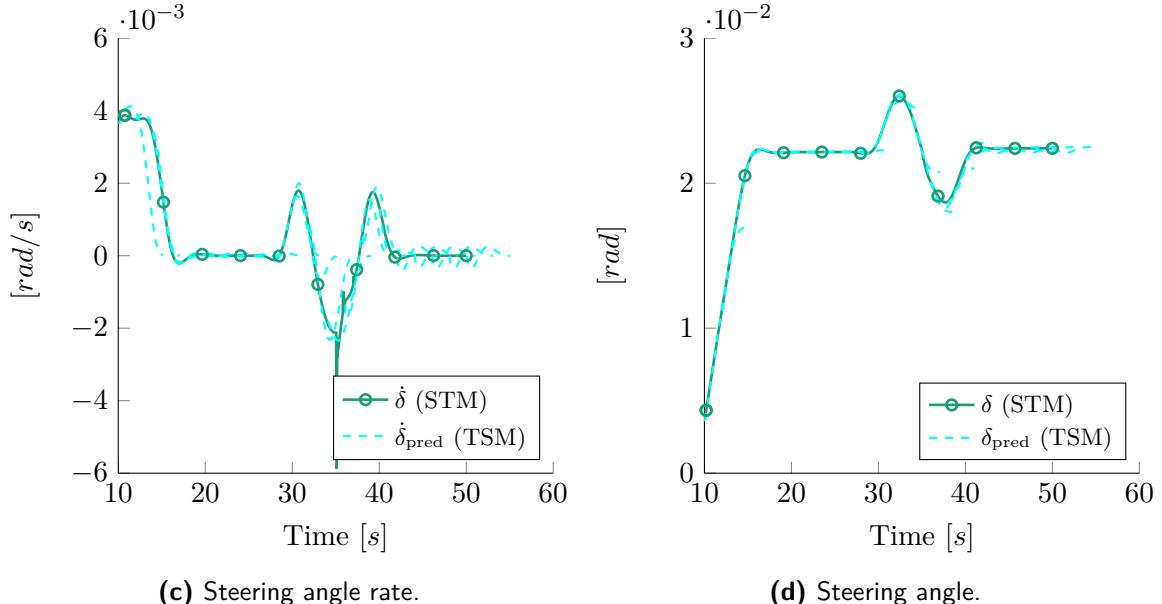


Figure 6-4: Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.

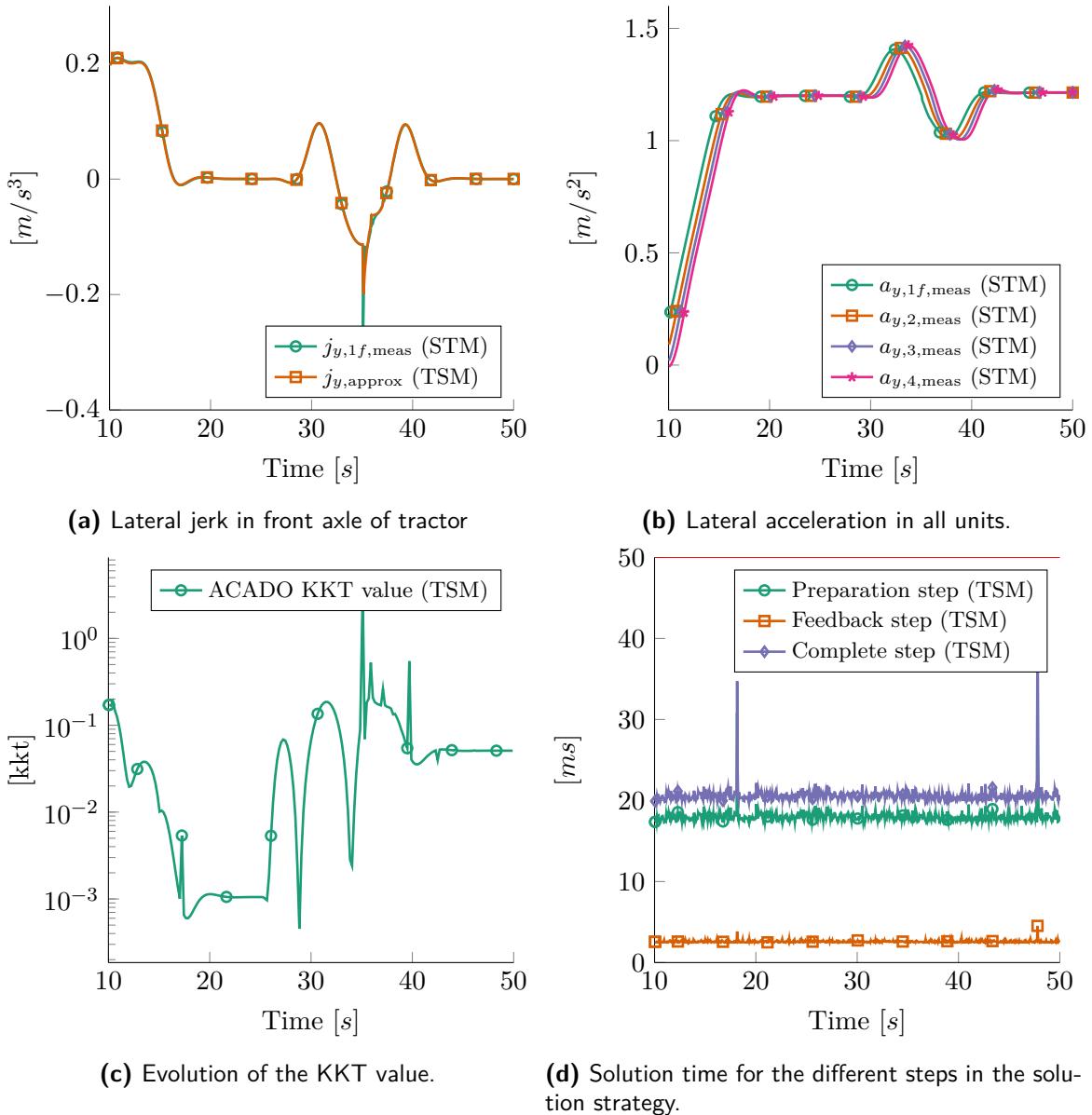


Figure 6-5: Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.

6-1-2 Simulations with high-fidelity plant

Lane change on straight road

We now evaluate the performance of the trajectory generator executing a lane change maneuver on the high-fidelity plant. We start with the straight road, the same velocity of $20[m/s]$ is considered. In Figure 6-6 and Figure 6-7 we observe that the prediction is very close to the actual trajectory for all signals. The overall performance of the lane change with the VTM plant is highly acceptable. The KKT-value evolution has the same trend but seems magnified with approximately a factor five compared to results with the single-track model. Nevertheless, it converges nicely an order of magnitude below unity. The solution time of the algorithm is about the same in the simulations with the single-track model, real-time performance is absolutely no issue.

The measured lateral jerk $j_{y,1,\text{meas}}$ is in the same range as the results with the single-track model. However, the approximation of the jerk is much higher. It is because of the lateral distance reference calculation step that the overall behavior is very similar. If a step change in the reference would have been applied, we could have expected different results. More surprisingly is that the lateral jerk approximation does not converge back to zero after the lane change is completed. This is because the jerk approximation relies on the single-track vehicle model to define the quantity $\ddot{\theta}_0$. Apparently the current measured state of the A-double combination would cause a rotational acceleration in the tractor. This is not observed in the high-fidelity plant model.

The prediction of the steering angle rate is of slightly less quality compared to results from the simulations with the single-track model. Two possible reasons are: model inaccuracies in the prediction model and secondly, because an optimum is not perfectly achieved as visible in the KKT-value. This can indicate that suboptimal trajectory is planned by the solution algorithm for the NLP. On the other hand, the prediction of the steering wheel angle is close to perfect.

Like with the single-track model, the difficulty of the maneuver will now be increased. The high-fidelity model is controlled on a constant curvature road. We will see that controller performance is prone to more discussion in this scenario.

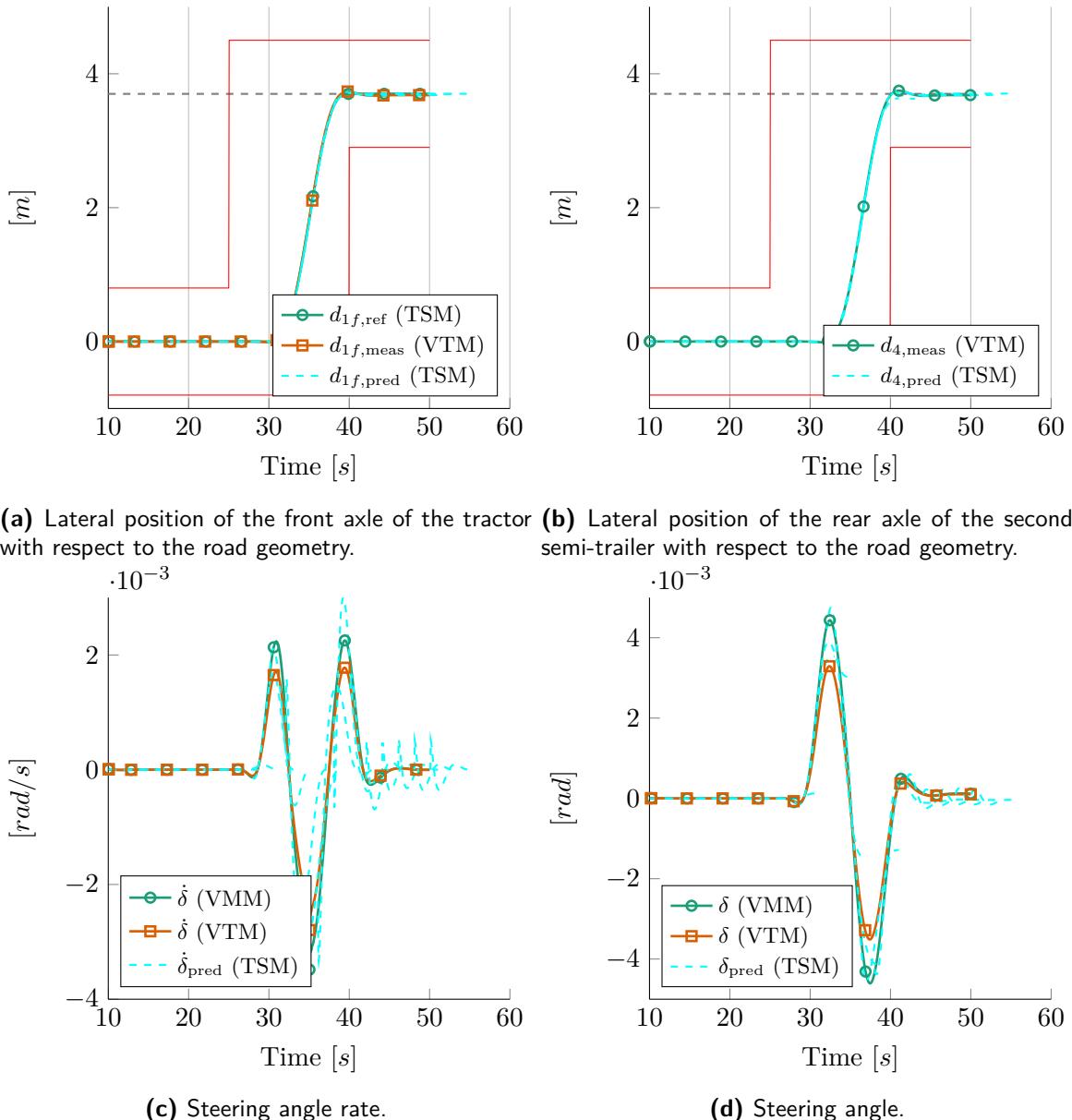


Figure 6-6: Results from a lane change on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

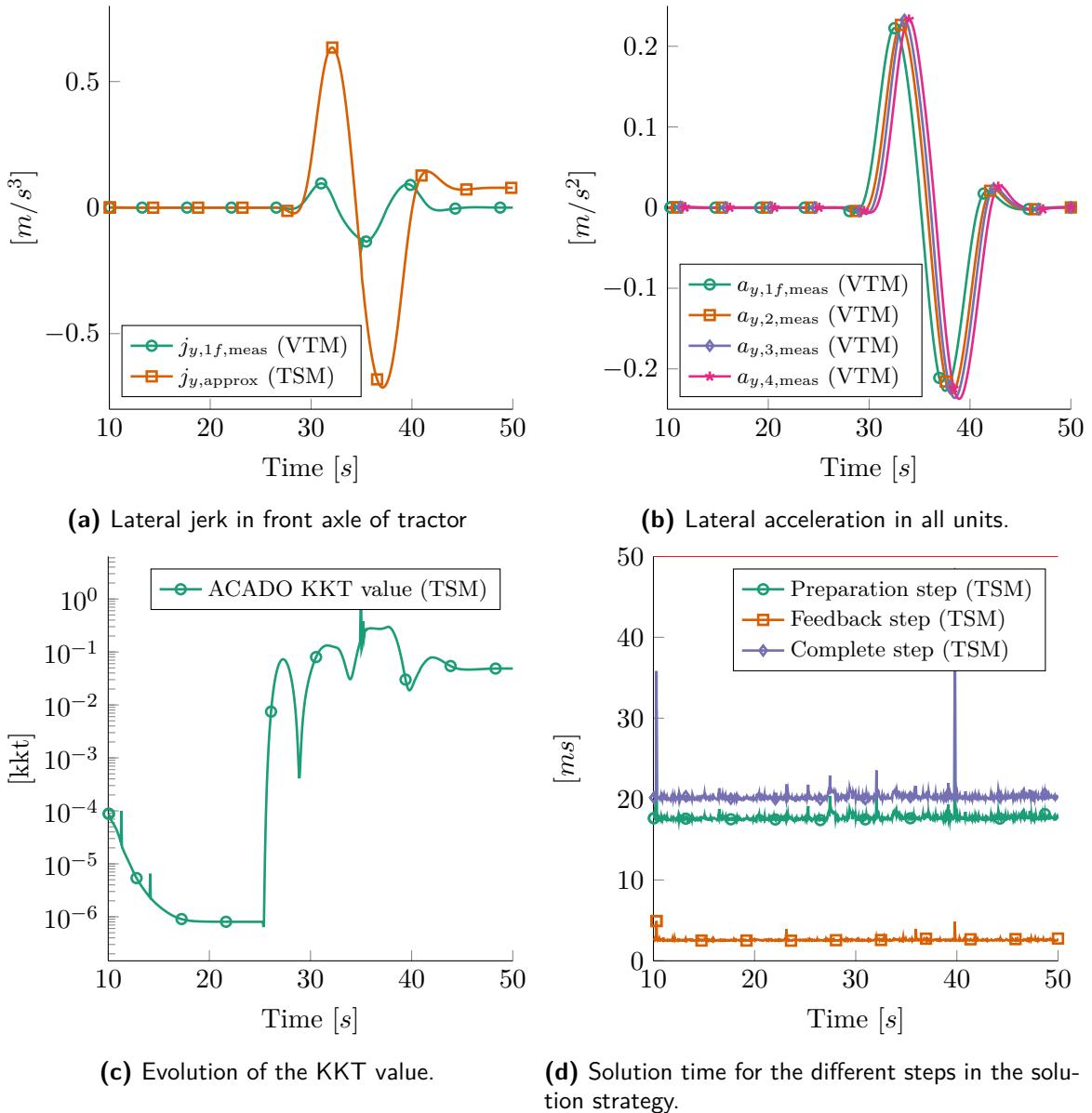


Figure 6-7: Results from a lane change on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

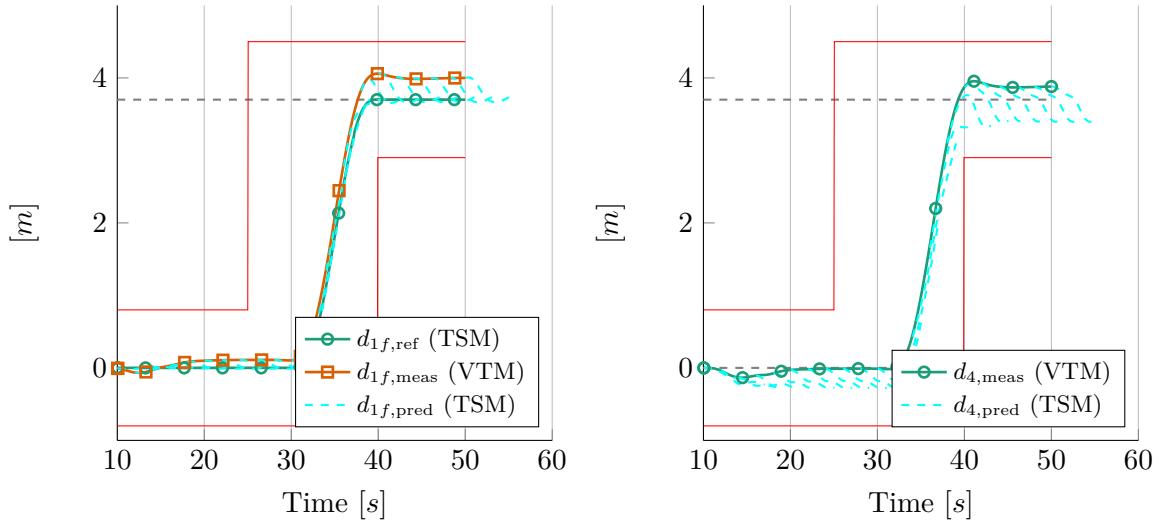
Lane change on road of constant curvature

The second lane change scenario for the high-fidelity plant is on a road of constant curvature of $1/333[\text{rad}/\text{m}]$. In Figure 6-8 and Figure 6-9 we see the tracking performance of the trajectory generator has decreased. The off-set in the lateral position tracking becomes notable. Also, the predictions are now seriously deviating from the measured trajectories. Whereas the KKT-value of the optimization result previously was in the order of unity, it has now increased an order of magnitude. The KKT-value fails to converge to zero, which indicates that optimality for the predicted trajectory is not achieved. Let us try to understand this by discussing the measured signals.

The lateral jerk profile in Figure 6-9a measures similar magnitudes as in the straight road lane change. The behavior in the range of $10 - 15[\text{s}]$ can be ignored, it reflects the jerk behavior when the curve is entered. The approximation of the jerk is now much worse. As proposed in the straight lane case for the lane change, the approximation for the rotational acceleration of the tractor as a function of the vehicle model states can explain this. The truck its state, the articulation angles, their time derivatives and the side slip would cause a change of yaw rate in the single-track model. Whereas the yaw acceleration of the high-fidelity plant is nearly zero. The jerk approximation clearly shows its limitations here.

The VMM-layer sets a reference to the low level steering control. The road wheel angle is not one-to-one to the actual measurement. It must be noted though, that a single steering angle signal is rather ambiguous in case of the high fidelity model. In reality, two separate wheels are actuated. The steering compliance and suspension dynamics make the effective road wheel angle change. We also see that the prediction of the control does not agree with the actual state evolution, see for example the steering signals and the lateral distance trajectories gathered in Figure 6-8. We have seen that the KKT-value did not converge to zero either. This in fact means that the overall sequential quadratic programming (SQP) of the RTI-scheme does not converge to the optimal solution. The suspicion rises that the Hessian approximation in the preparation step is inaccurate. Recall from the discussion on the constrained Gauss-Newton algorithm in Section 4-3-3 that we have recognized potential occasions that the Hessian approximation is wrong. When the objective value of the NLP is non-zero, the Gauss-Newton algorithm is not entirely appropriate to use. From this, we can conclude that the prediction inaccuracy is likely due to the failure to track the lane center and the faulty lateral jerk approximation. Both induce a rise in the objective value of the NLP.

Overall we can say that the controller does achieve acceptable behavior. A low-jerk transition is made to a new lane and lane-keeping is achieved reasonably well. The tracking error is very limited and no constraints are violated. We have only considered the lane change at a velocity of $20[\text{m}/\text{s}]$, in Section 6-4 results of a lane keeping scenario is considered on a road of constant increasing curvature. This allows us to informally evaluate the effect of the model mismatch on lane center tracking. But first, the effect of tuning the cost function is discussed with simulations on the single-track model.



(a) Lateral position of the front axle of the tractor with respect to the road geometry. **(b)** Lateral position of the rear axle of the second semi-trailer with respect to the road geometry.

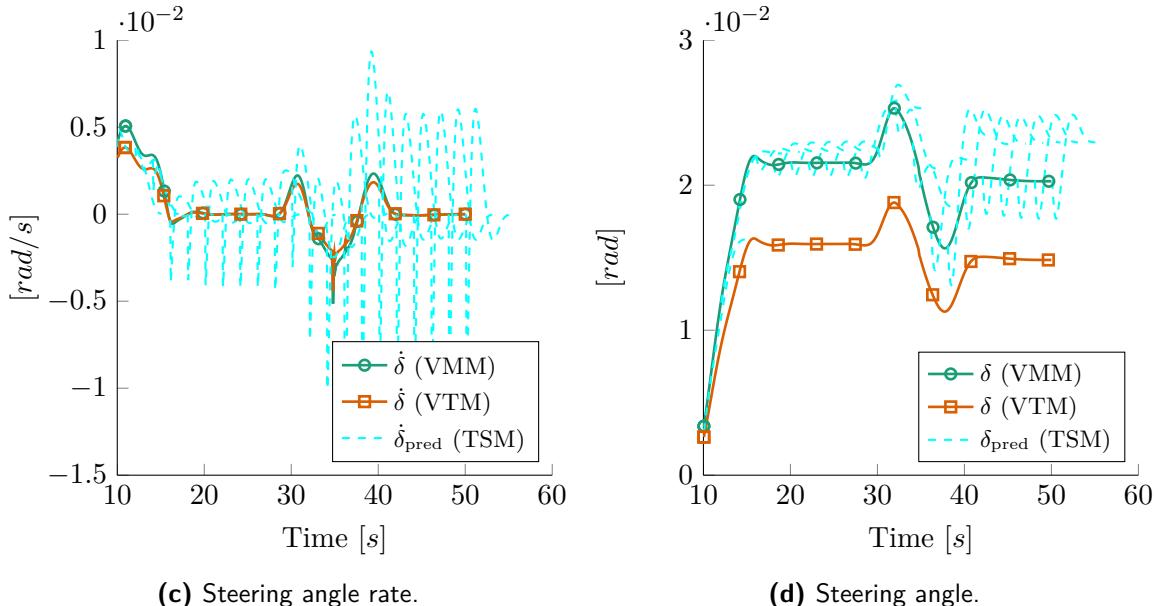


Figure 6-8: Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

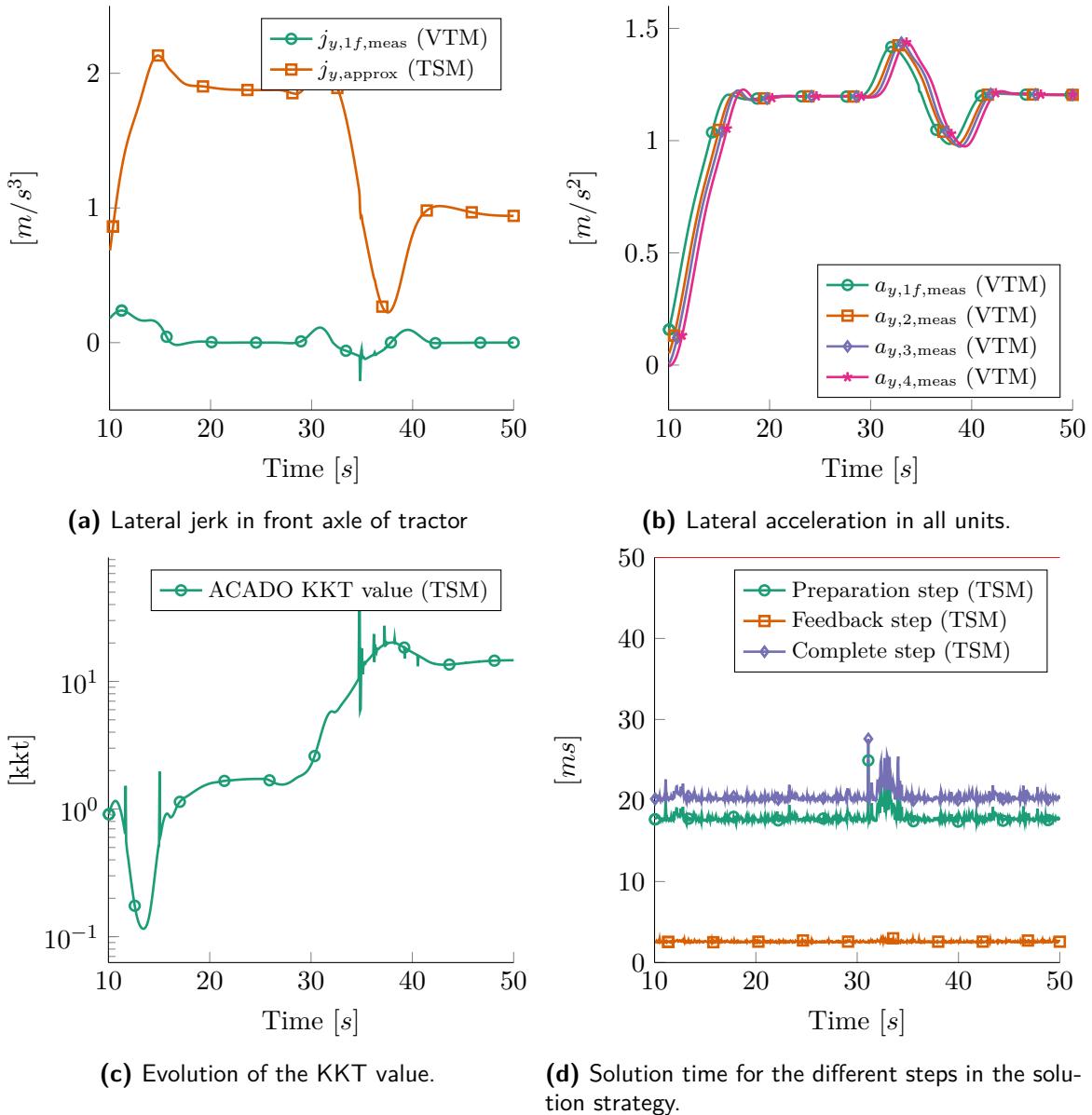


Figure 6-9: Results from a lane change on a curved road (to the left), the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

6-1-3 Effect of cost function tuning on lane change maneuver

One of the main appealing advantages of optimization-based control is the intuitive tuning phase. The state amplitudes that one would like to reduce or increase are directly available in the cost function of the optimization. In this section, simulations are executed on the single-track model for the A-double combination. Four sets of tuning parameters, weighting terms for the least-squares cost function, are considered and applied in the NLP formulation. Only the terms that affect the lateral dynamics directly are varied.

Note that the setup for the controller that is used in other parts of this thesis are not solely tuned on a lane change maneuver. A trade-off is made between good tracking performance of the center line in curved roads and the lane change trajectory. A very mild control setting may result in a very smooth lane change, but at the same time result in fishtailing on curved roads because of poor lane keeping. That said, in Table 6-4 the different cost settings are listed.

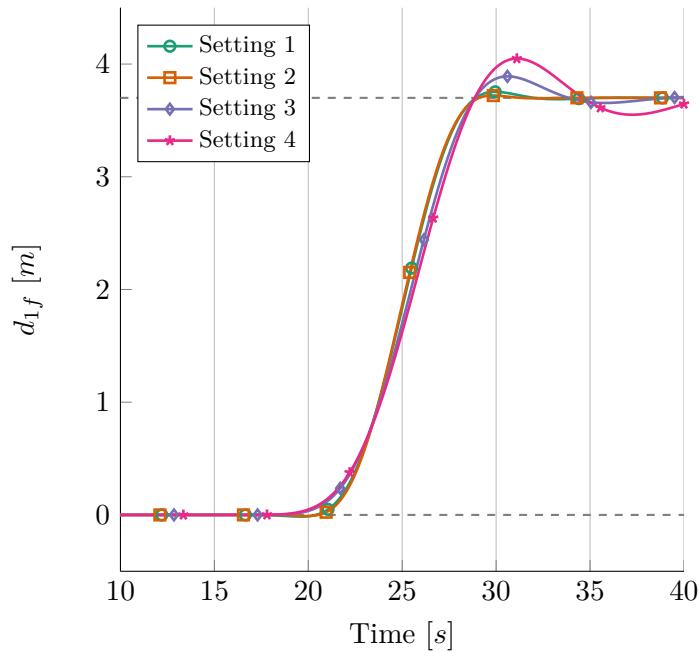
Table 6-4: Cost settings for different lane change simulations.

#	K_{d_1}/n_{d_1}	$K_{j_{y,1}}/n_{j_{y,1}}$	$K_{\dot{\delta}}/n_{\dot{\delta}}$
1	100/2	100/3	5/0.05
2	1000/2	100/3	5/0.05
3	100/2	1000/3	5/0.05
4	10/2	1000/3	5/0.05

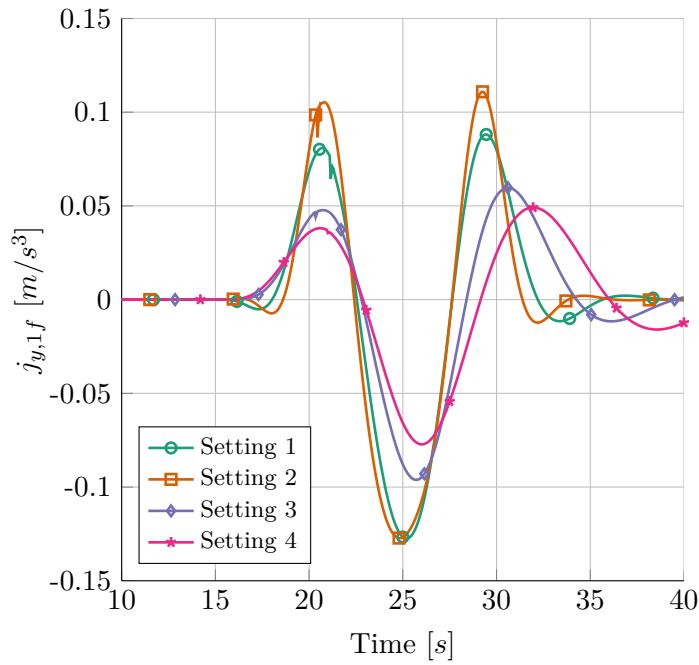
Simulations for the different cost settings in Table 6-4 are executed. The remainder of the weights in the cost function are identical to the other simulations in this chapter. Also, lateral reference calculation is in place to pre-optimize the lateral trajectory for minimum jerk.

See the results of the four cost settings in Figure 6-10. They show a trend of decreasing jerk when the trade-off shifts from a preference to decrease tracking error, to penalizing jerk. The lateral distance trajectories show increasing oscillatory behavior. Hence, the cost function tuning seems to have the desired effect.

The next topic of discussion is merging. In that scenario the optimization by the trajectory generator for longitudinal dynamics is visible. The effect of surrounding traffic is shown in closed-loop simulations with the high-fidelity vehicle model.



(a) Lateral position of the front axle of the tractor with respect to the road geometry.



(b) Lateral jerk measured in the front axle of the tractor.

Figure 6-10: A comparison of a lane change maneuver with four different cost function settings. The results are from closed-loop simulations with the single-track model.

6-2 Merging maneuver

The merging maneuver is a problem of increased complexity compared to the regular lane change. To focus on the longitudinal trajectory planning capabilities of the planner, only straight road conditions are executed. Only simulations with the high-fidelity plant model are discussed, simulation results with the single-track model are almost identical. The outline of the scenario is as follows:

The A-double combination drives on the left lane of a 70[km/h] highway. The truck initially drives 20[m/s] and approaches a vehicle in the same lane driving 19[m/s]. Meanwhile, in the adjacent lane to the right, two vehicles drive in front of each other. One is 40[m] ahead and one is 150[m], following the exact same velocity profile of the truck in the first 17[s]. After these 17[s], both vehicles in the adjacent lane slow down to 15[m/s]. A lane change is requested to merge between the two vehicles.

In this maneuver, multiple steps in the control algorithm actively contribute to the results. Firstly, the reference calculation for the longitudinal velocity. Secondly, the tuning of the NLP to keep distance from the three vehicles of interest. For a recapitulation of the so-called vehicles of interest, the reader is referred to Section 3-3. The tuning of the OCP is identical to the lane change simulations. In Table 6-5 one finds the selection of tuning parameters relevant to the merging maneuver.

Table 6-5: Details relevant for the merging maneuver of the A-double combination modeled using the VTM-plant.

Parameter	Symbol	Value
Reference calculation deceleration gap	τ_{lo}	3.0[s]
Reference calculation acceleration gap	τ_{hi}	5.0[s]
Sigmoid activation distance: $\Delta s_{o,2}$	$\tau_{o,2}$	2.0[s]
Sigmoid activation distance: $\Delta s_{o,3}$	$\tau_{o,3}$	2.5[s]
Sigmoid activation distance: $\Delta s_{o,4}$	$\tau_{o,4}$	2.0[s]

6-2-1 Simulation with high-fidelity plant

A simulation is executed for the merging maneuver with the high-fidelity plant. Let us first focus on the results in Figure 6-11, it visualizes the distances of the surrounding vehicles measured from the front axle of the tractor. Also, we see a plot of the longitudinal velocity of the A-double combination, and the velocity reference profile for the first shooting node of the NLP. The scenario is relatively complex, let us go through the results in chronological order:

1. The truck drives on the left lane of the highway with 20[m/s].
2. Two vehicles in the adjacent lane to the right, obstacles 3 and 4, drive with the same velocity as the truck. At 17[s], their speed rapidly adapts to 15[m/s].

3. At 25[s] a signal is sent to the controller that a lane change is desired. The decision making algorithm starts considering this option.
4. At approximately 26s the vehicle in front of the A-double combination, obstacle 2, is closing in and the reference calculation step advises the truck to slow down to the velocity of the leading vehicle, 19[m/s].
5. It is not before 40[s] that a lane change option is detected. The reference calculation adapts the desired velocity to the leading vehicle in the adjacent lane, since this vehicle drives slower.
6. At the same time, the OCP-terms for distance-keeping become active. The truck finds a position between the two vehicles, suitable for a lane change.
7. As soon as the lane change is fully completed, at about 60[s], the trailing vehicle is disregarded from the OCP and it is optimal to obey the velocity from the reference calculation.
8. The gap between the A-double combination and obstacle 4 is adapted to a more comfortable distance. In Table 6-5 one can see this preferred distance is specified to 3[s].

Note that the trailing vehicle is expected to slow down for the A-double combination as soon as the lane change is completed. This does not occur, because no human driver model is present in the simulation. The trailing vehicle and the truck will almost collide at the end of the simulation. In real-life this very unlikely to happen, the distance profile of vehicle $\Delta s_{o,3}$ can be ignored after 60[s].

In Figure 6-12 we observe that the lane change maneuver itself is executed at approximately 45[s], five seconds after the vehicles in the adjacent lane are considered in the problem. The lateral trajectory is smooth and follows the results from the lane change maneuver without surrounding obstacles on a straight road discussed earlier. Additionally, in Figure 6-13 some general diagnostics of the optimization are shown. The solution time is still within safe bounds from the maximum solution time.

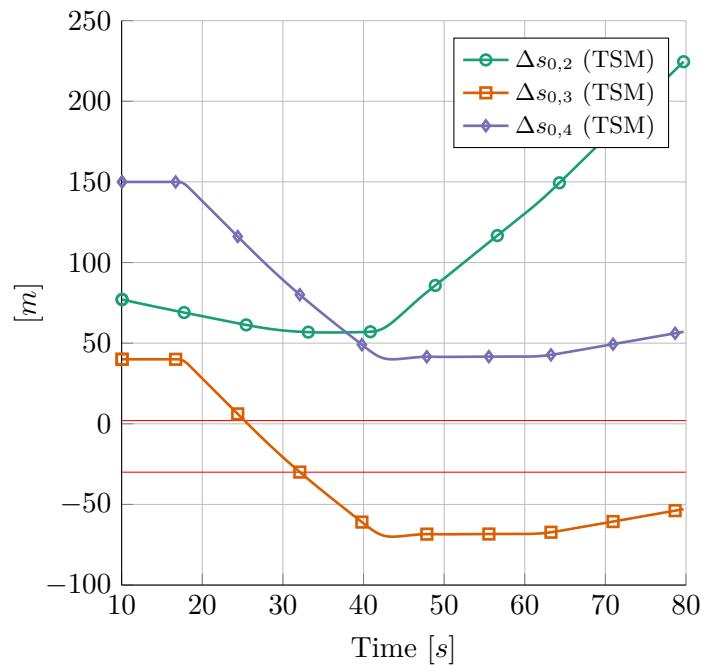
The KKT-value does not converge to zero during the merging maneuver itself. This can be explained by the inaccurate Hessian approximation of the Gauss-Newton algorithm in case the objective of the NLP does not converge to zero. This is especially visible between 40 – 60[s] in the simulation, the optimum of the NLP is in a non-zero objective value. There is an optimal position between the three vehicles under consideration. The Hessian approximation for the QP is in this case biased, the effect is visible in the prediction of the longitudinal dynamics. The prediction of the longitudinal velocity and the longitudinal acceleration, is mildly put, not nearly matching the actual behavior. As soon as the lane change is completed the objective function decreases significantly, this can be observed in Figure 6-13. Immediately, the prediction of the longitudinal trajectory increases in accuracy.

Another side-effect of the RTI scheme is visible in the results, at around 60[s] the problem for the OCP suddenly changes. The trailing obstacle is suddenly removed from the cost formulation. This causes the previous solution to be highly non-optimal in the next iteration. Although the optimization recovers very quickly, the solution time is temporarily longer. If many *unexpected* phenomena occur, this may cause potential problems for the applicability of the RTI scheme.

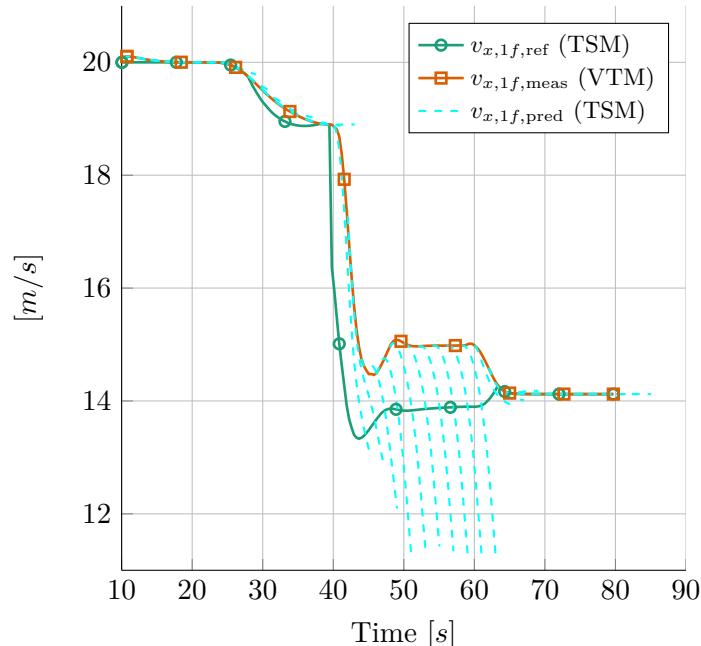
We conclude that the formulation of the predicted collision free corridor achieves our intentions. Be aware that the scenario is tailored to be compatible with the rectangular collision free corridor described in Section 3-3-4. The vehicles in the goal lane of the lane change drive slower than the traffic in the departure lane. In case this would not be true, the controller could fail. Additionally, the obstacles drive with a constant velocity throughout the merging maneuver. This is exactly as modeled in the OCP, but it does not necessarily reflect realistic behavior.

The results of the merging maneuver are at best a *proof of concept*. Future research is necessary for alternative formulations, such as the sigmoid shaped predicted collision free corridor (Section 3-3-4). Alternative formulations for the corridor and more accurate models for the surrounding traffic can enable less conservative decision making.

We have discussed the standard functionalities of the trajectory generator. Let us now shift our focus to challenge the algorithm. First, a so-called abort maneuver is simulated.

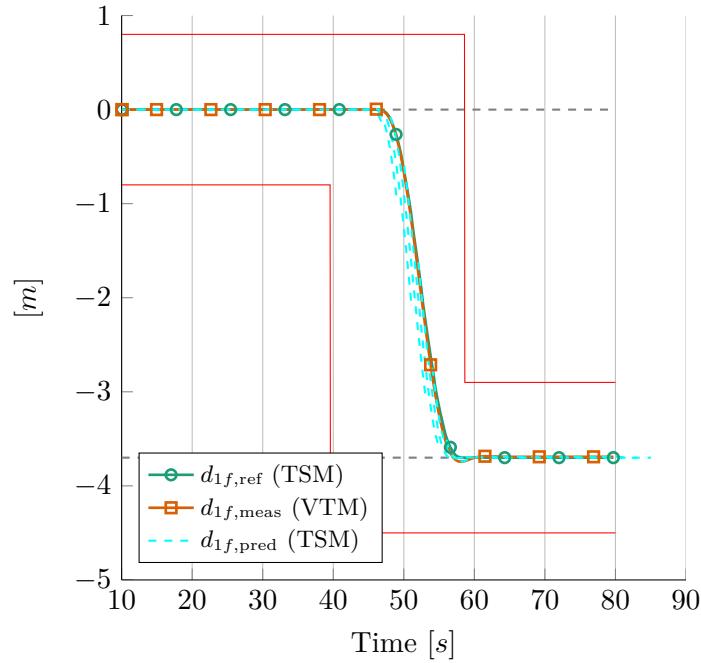


(a) Distances with each obstacle of interest. The red lines indicate the front and the back of the A-double combination.

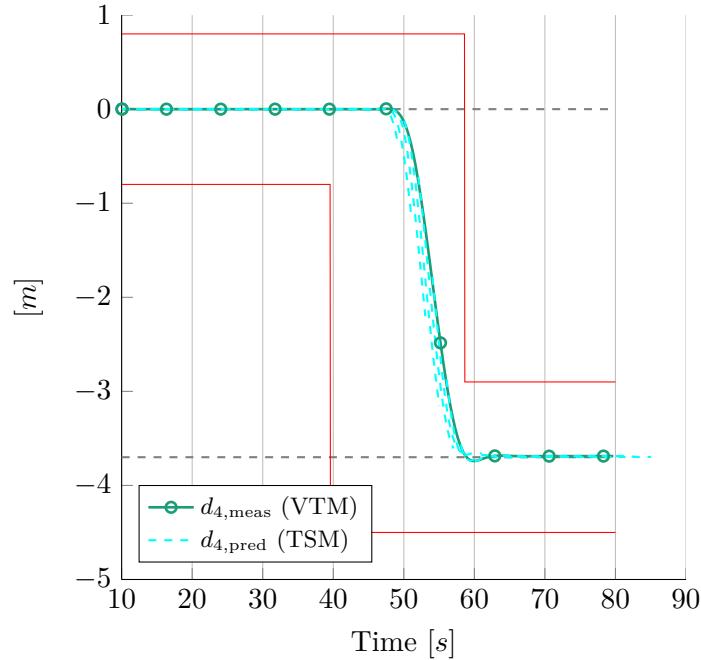


(b) Longitudinal velocity of the truck in the tractor, $v_{x,1f,ref}$ denotes the advised velocity from the reference calculation step.

Figure 6-11: Results from a merging scenario to the right, the trajectory generator is in closed-loop with the high-fidelity model.



(a) Lateral position of the front axle of the tractor with respect to the road geometry.



(b) Lateral position of the rear axle of the second semi-trailer with respect to the road geometry.

Figure 6-12: Results from a merging scenario to the right, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

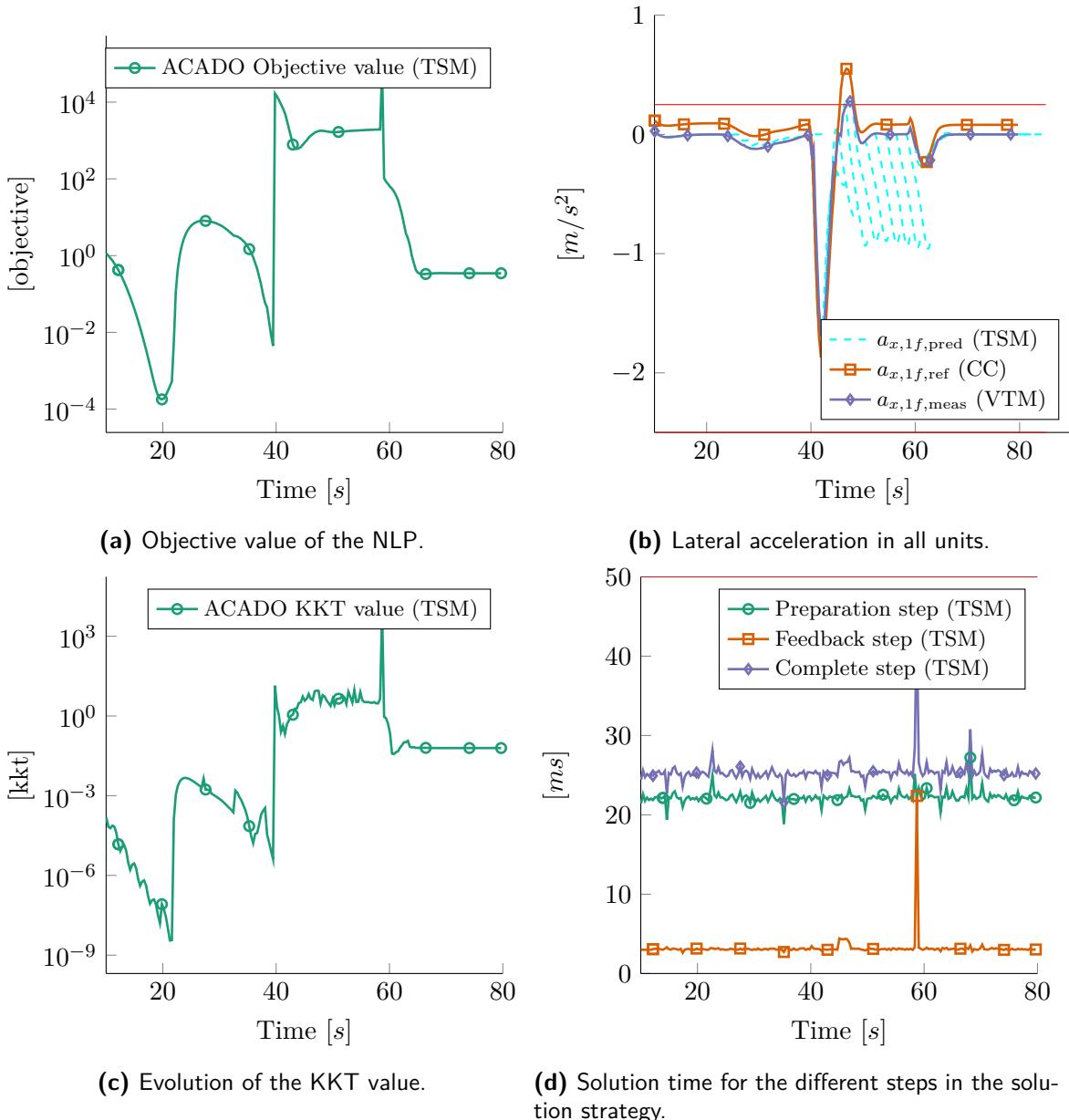


Figure 6-13: Results from a merging scenario to the right, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

6-3 Evasive return to lane maneuver

The abort maneuver, also referred to as the evasive return to lane maneuver, is the emergency capability of the trajectory generator. Assume we are in the almost the same scenario as the merging procedure that is described in the previous section. The difference is, that the vehicles in the adjacent lane now drive $18[m/s]$. Half-way the lane change, the leading vehicle in the departure lane suddenly slows down to $14[m/s]$. The vehicles in the adjacent lane do not slow down. Our distance with obstacle 2, $\Delta s_{o,2}$, decreases quickly. If we do not respond to this changed scenario promptly, the truck will have a problem. The union of the half-spaces $\{s_1 | \underline{\Delta}s_{o,2} \leq \Delta s_{o,2}\}$ and $\{s_1 | \underline{\Delta}s_{o,3} \leq \Delta s_{o,3}\}$ shrinks when obstacle 3 starts to overtake obstacle 2.

As a response to this changed conditions, the controller executes a so-called abort maneuver. The vehicle tries to return to the departure lane as quickly as possible. No lateral position reference calculation is performed and the jerk in the cabin is expected to be quite severe. The lateral accelerations observed in the axles of the different units are also expected to be much higher.

The cost function is re-tuned in order to avoid too aggressive maneuvers. As soon as evasive maneuver is initiated, the weighting on the lateral distance d_1 and the lateral jerk $j_{y,1}$ is adjusted. The controller cannot rely on the reference calculation path to generate a smooth reference trajectory. The weighting of the lateral position off-set and the lateral jerk are changed, see Table 6-6.

Table 6-6: The changed cost function weights for the abort maneuver.

Parameter	Symbol	Value
Cost weight d_1	K_{d_1}/n_{d_1}	$100/2$
cost weight $j_{y,1}$	$K_{j_{y,1}}/n_{j_{y,1}}$	$50/3$

In a production implementation, one could change the cost function online. In the presented results, the cost function of OCP is hard-coded for increased computational efficiency. Hence, in the trajectory generator implementation of this thesis, the cost function weights are static.

Simulations are executed on both the single-track model and the high-fidelity plant. The scenarios are not identical. The simulation with the single-track model is executed on a curved road. Whereas the high-fidelity plant performs the abort maneuver on a straight road.

6-3-1 Simulation with the single-track model on a curved road

Results of the simulation for the evasive maneuver with the single-track model are depicted in Figure 6-14, Figure 6-15 and Figure 6-16. The results are from a situation driving a curved road of $0.003[rad/m]$, which is identical to the curvature in the lane change discussed previously.

The scenario starts with a merging maneuver. At $25[s]$ the lane change is requested, it is approved immediately since the gaps with obstacles 3 and 4 in the adjacent lane are sufficient. At $30[s]$ the lateral reference shifts from the departure lane to the goal lane to the right. The

velocity is decreased to match the speed of the vehicles in the adjacent lane. At 36[s] obstacle 2 starts to rapidly slow down to 14[m/s], for an unknown reason. It is considered that an abort maneuver is appropriate, at 38[s] the maneuver is actually triggered. Immediately the lateral distance reference returns to the departure lane. Obstacles 3 and 4 are removed from the NLP formulation, only the leading vehicle in the departure lane (obstacle 2) is left under consideration.

The A-double combination returns to the departure lane perfectly, almost no overshoot is observed. The tracking error with the lane center is negligible after the abort maneuver is finished. The curved road causes a steady-state lateral acceleration of approximately 1 – 1.3[m/s²] for the velocity range of the scenario. This off-set in the lateral acceleration causes the activation of the lateral acceleration constraints for the first unit and the rear axle of the second semi-trailer. In Figure 6-14b we observe that the generated trajectory seemingly obeys the constraints in the optimization algorithm. In fact, the constraints are slightly violated. As visible in Figure 6-16c, the QP solver returns an exit flag that the optimization is infeasible for a considerable amount of time. However, the trajectory generator remains working properly and the abort maneuver is successfully completed.

The lateral jerk is depicted in Figure 6-14a. At the time the abort maneuver is initiated a sharp peak is visible. This peak can be explained by two causes. First, the rapid change in the reference actually requires an aggressive steering maneuver to return to the lane. Secondly, the optimum to the optimization suddenly shifts a lot. The RTI needs to re-converge to an optimal trajectory. The steering may be highly non-optimal to the NLP.

At the time the abort maneuver is request, the solution time spikes and violates the 50[ms] limit. The QP seems to be the main cause for the increased computation time. The increased solution time is explained by the sudden change of the overall NLP problem. It would be possible to avoid this, two instances of the trajectory generator can run side-by-side. One for the abort maneuver and one for normal operation. We could continuously optimize for a return-to-lane trajectory, even if it is not requested. The SQP for this separate problem would already have a close to optimal trajectory available.

We can conclude that the trajectory generator can successfully execute an evasive maneuver on a curved road in closed-loop with the single track model. Next, results for the high-fidelity plant in an abort maneuver on a straight road are discussed.

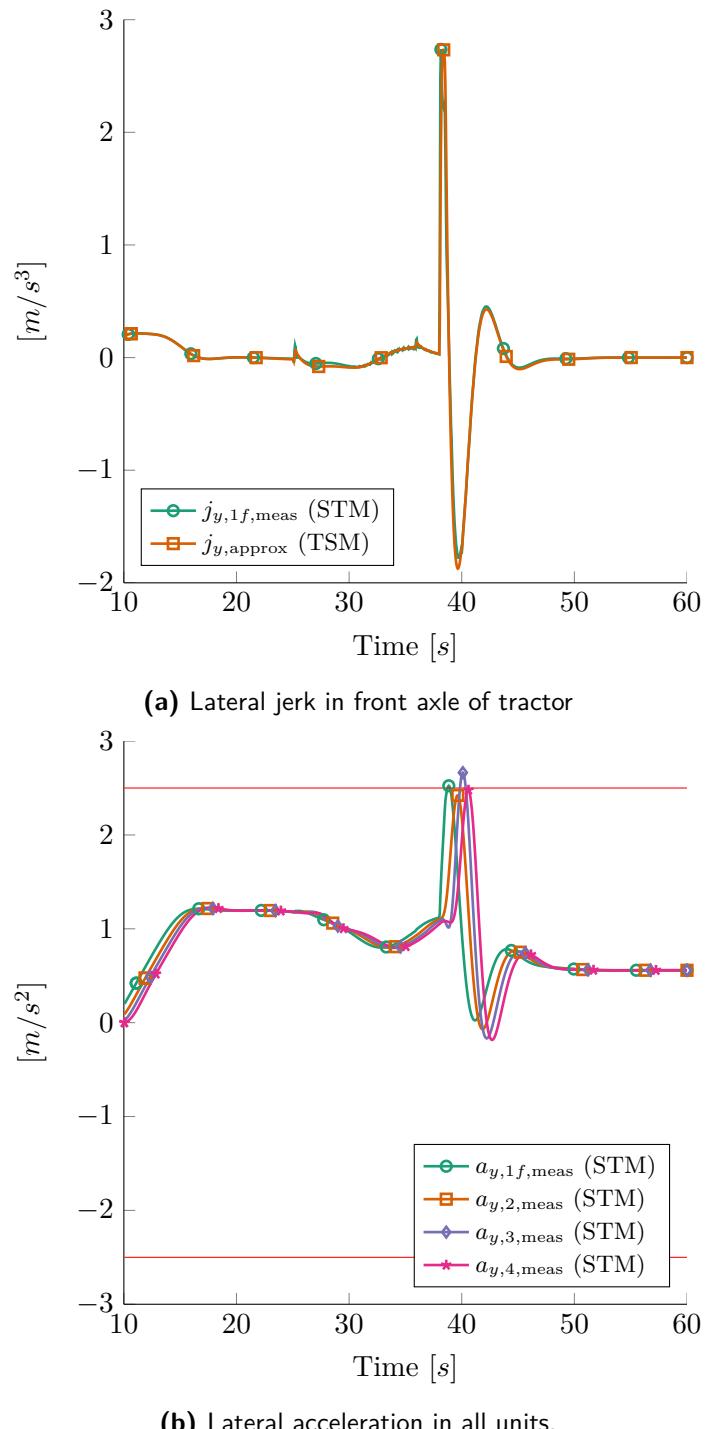
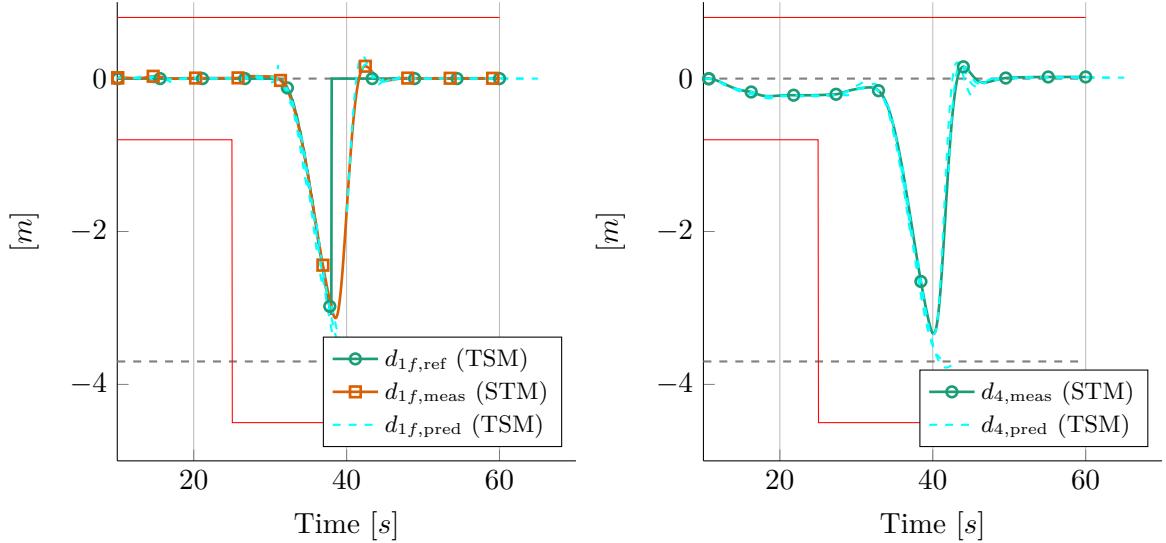
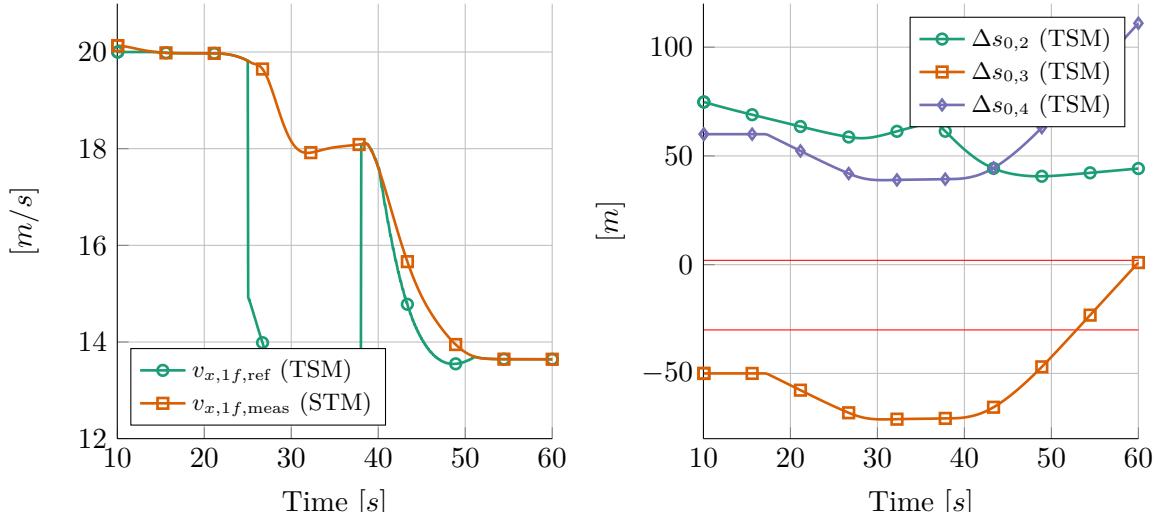


Figure 6-14: Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.

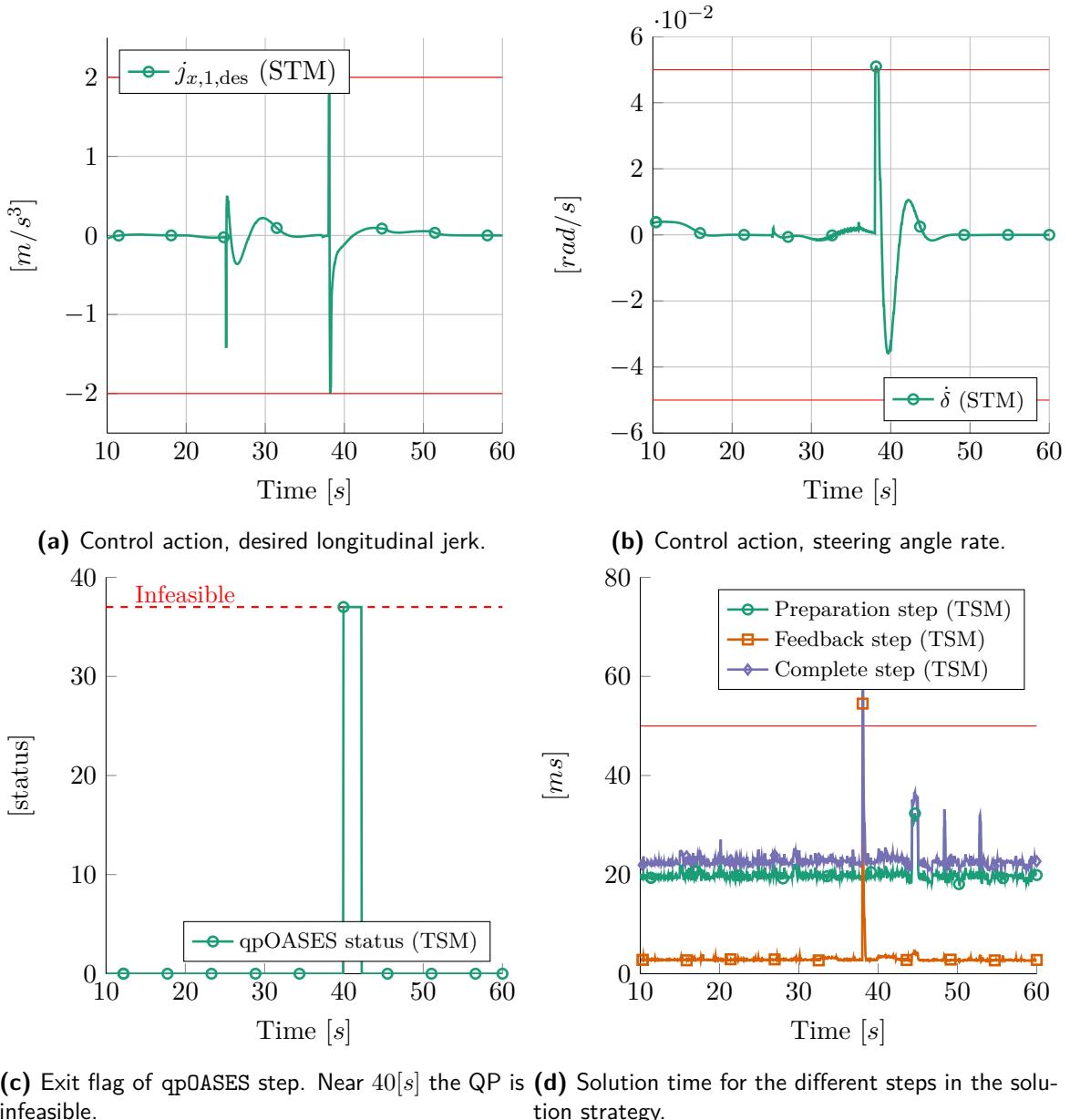


(a) Lateral position of the front axle of the tractor **(b)** Lateral position of the rear axle of the second semi-trailer with respect to the road geometry.



(c) Longitudinal velocity of the tractor in the front axle. **(d)** Distances with each obstacle of interest. The red lines indicate the front and the back of the A-double combination.

Figure 6-15: Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the single-track model. Bounds on signals are indicated with red lines, if appropriate in the situation.



6-3-2 Simulation with the high-fidelity plant on a straight road

The second simulation for the abort maneuver is with the high-fidelity vehicle model of the A-double combination. It drives a straight road, the scenario is otherwise identical to the single-track model simulation for the abort maneuver. Curved road simulations have been executed, but the trajectory generator fails to complete the maneuver without violating constraints and needs more work.

Like in the scenario with the single-track model, at 25[s] a merging maneuver is requested. Since obstacles 3 and 4 are both sufficiently far away, the request is immediately approved. The lateral distance reference starts to move to the right lane at 30[s]. The A-double combination decelerates to 18[m/s] in order to adapt for the velocity of the vehicles in the adjacent lane. At 36[s] obstacle 2 starts to decelerate to 14[m/s]. At 38[s] the evasive maneuver is initiated. The lateral distance reference immediately shifts to the departure lane, without performing reference calculation.

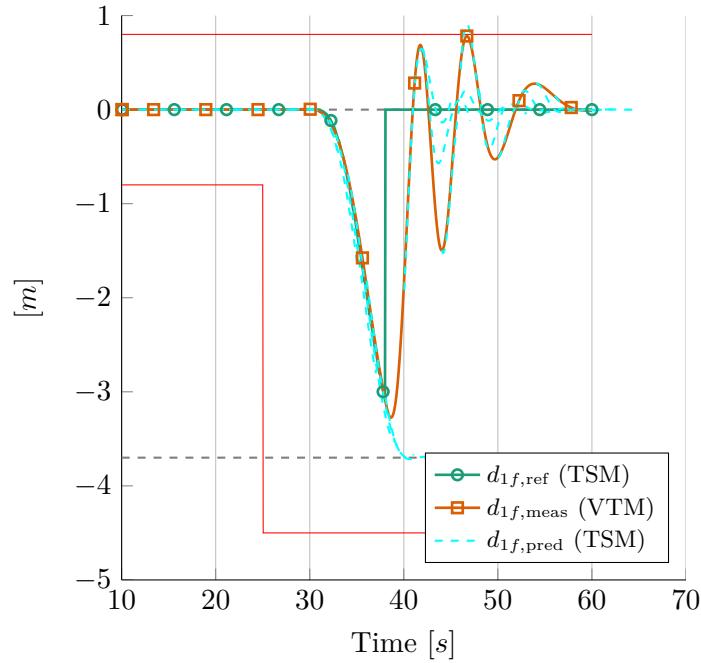
In Figure 6-17a and Figure 6-17b we observe that the A-double combination is successfully brought back to the departure lane. The vehicle respects the lane boundaries and oscillations disappear over a time span of approximately 15 – 20[s]. Notice that the oscillations are fare more abundant than with the single-track model. It is expected that acceleration transients with an amplitude of 2[m/s²] cause dynamics that are not modeled in the prediction model to influence the behavior of the truck.

The approximation of the jerk seems to successfully capture the trend of the measured lateral jerk in the high-fidelity model. As observed in previous simulations, the jerk is over-approximated. The high jerk levels in the range of 2 – 3[m/s³] is likely perceived as unpleasant by the truck driver.

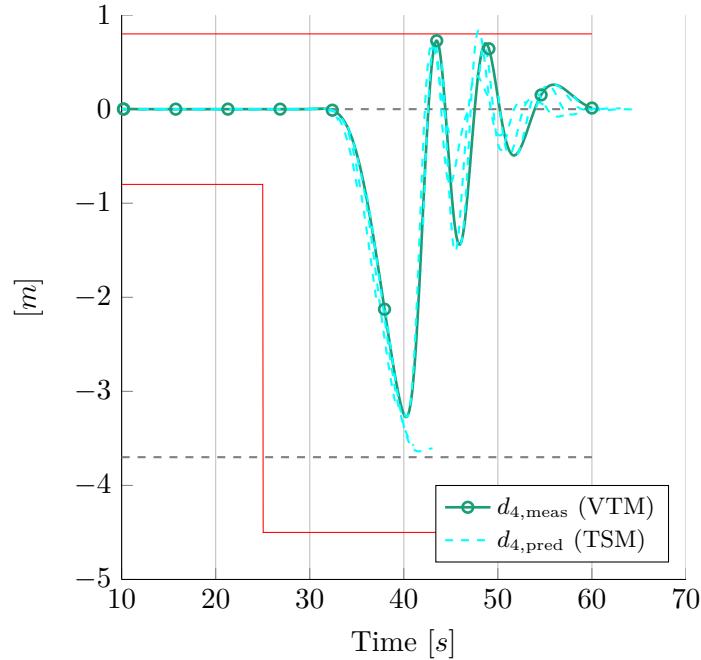
In general the abort maneuver seems to be executed successfully. The computation time stays within bounds and constraints are complied with. However, not all intermediate QP steps have been feasible. In fact, in Figure 6-19c it is depicted that the exit flag indicates infeasibility in many occasions between 40 – 46[s]. Even though the actual path stays within bounds, the QP fails to find a feasible trajectory prediction.

The abort maneuver functionality clearly needs extra work. However, the trajectory generator is shown to be able to execute a return to lane maneuver.

In the next section, the last simulation results are discussed. The trajectory generator is applied to the high-fidelity plant to a road of constantly increasing curvature.



(a) Lateral position of the front axle of the tractor with respect to the road geometry.



(b) Lateral position of the rear axle of the second semi-trailer with respect to the road geometry.

Figure 6-17: Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

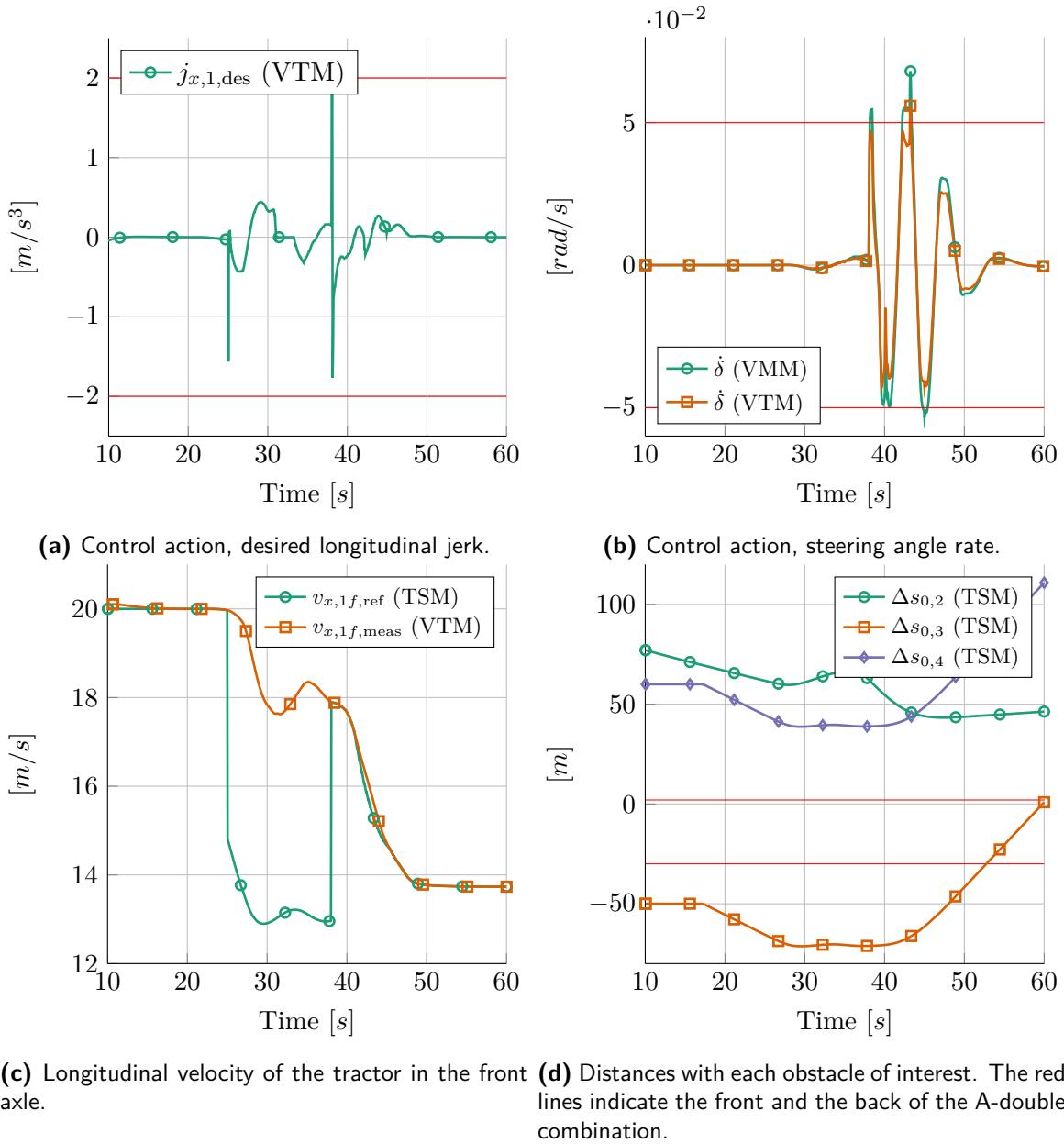


Figure 6-18: Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

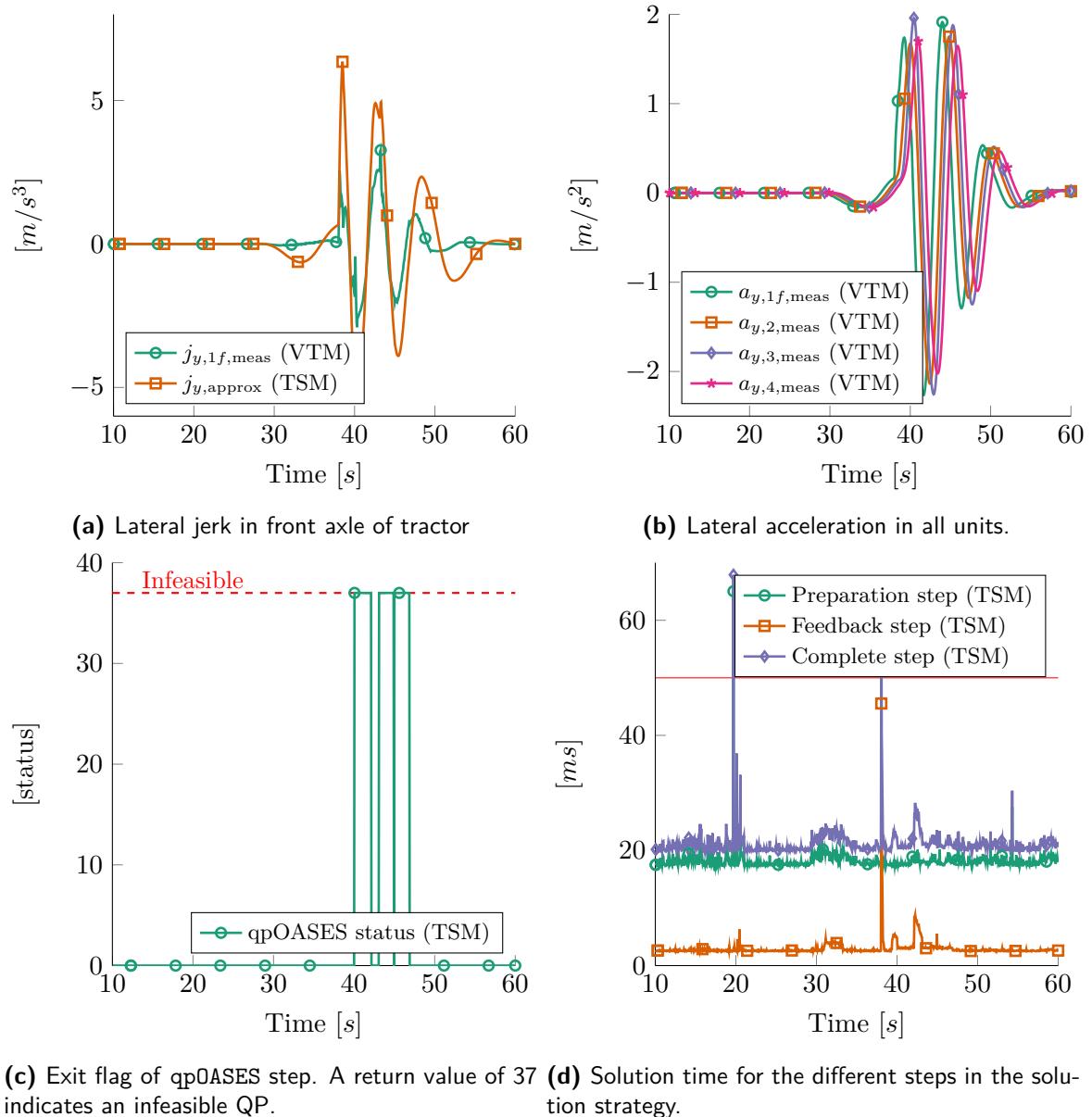


Figure 6-19: Results from an abort maneuver on a straight road, the trajectory generator is in closed-loop with the high-fidelity model. Bounds on signals are indicated with red lines, if appropriate in the situation.

6-4 Lane keeping on increasing curvature road

The high-fidelity plant model mismatch has shown to have an effect on the lane center tracking performance. In an attempt to test this tracking performance we have the high-fidelity vehicle model to drive on a road of constantly increasing curvature. The A-double combination starts on a straight road with a velocity of $20[m/s]$. The maximum curvature reached is approximately $1/64[rad/m]$. See Figure 6-20 for an illustration of the road profile.

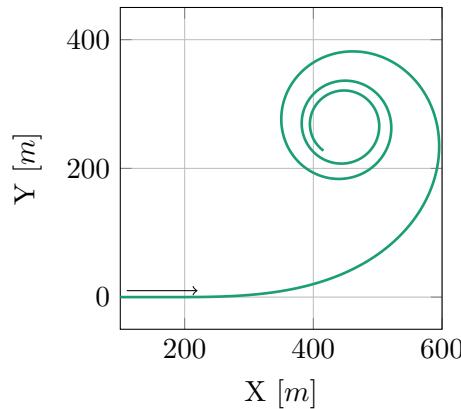
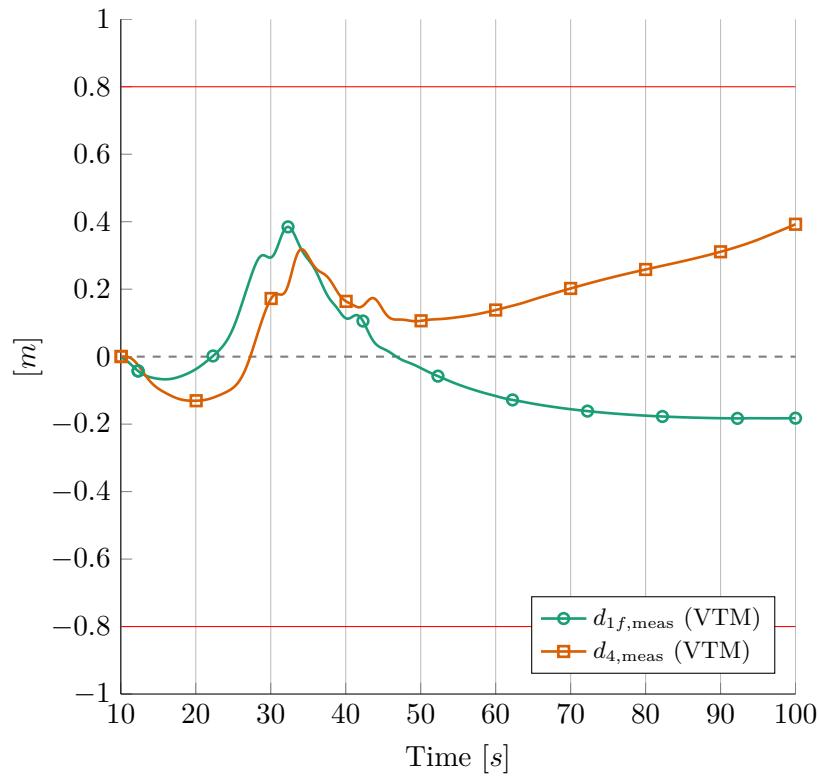


Figure 6-20: Illustration of the road profile for the simulation with the high-fidelity model on an increasing curvature road. The arrow indicates the driving direction.

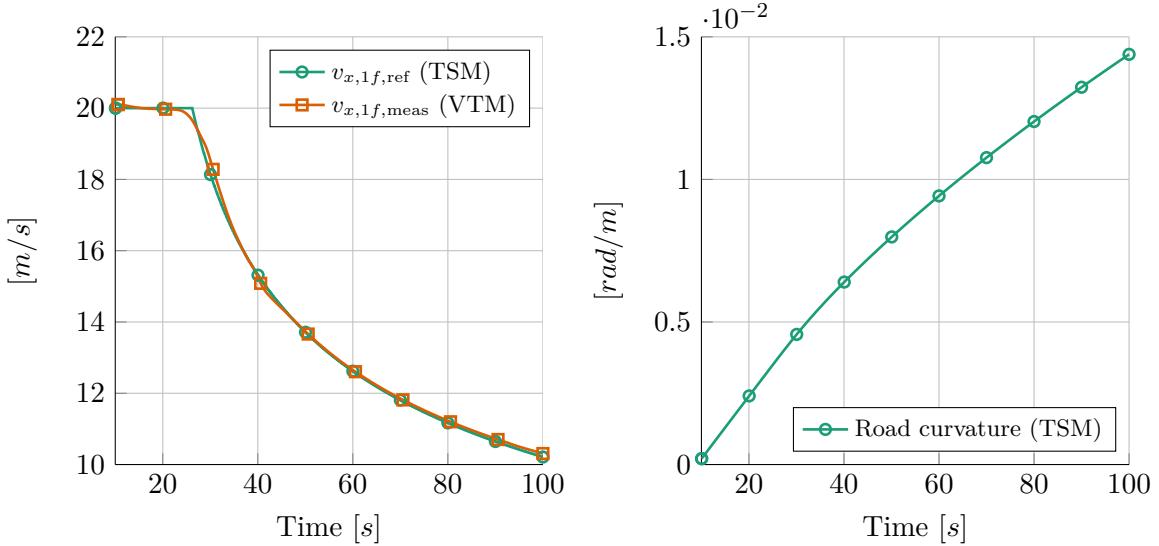
Until $25[s]$ the velocity of the truck is kept at $20[m/s]$. Due to the increasing curvature, the lateral acceleration increases to $1.5[m/s^2]$. The lateral distance offset of the tractor increases as the curvature grows. To maintain the steady-state lateral acceleration of $1.5[m/s^2]$, the velocity is decreased by the trajectory generator. In Figure 6-21a we see clearly how the lateral position behaves and how the longitudinal velocity is decreased while the curvature grows. The profile gives an interesting insight to the tracking error. Maintaining a constant lateral acceleration, the off-set gradually becomes smaller. For low velocities, the controller is better able to keep center lane. But inward off-tracking becomes significant at low velocities as well. One may consider to penalize the distance off-set of the rear axle of the second semi-trailer from the lane center.

The lateral jerk and its approximation are presented in Figure 6-22d, we observe that the lateral jerk measurement is close to zero at all times. The approximation however becomes increasingly inaccurate for higher road curvature. The state approximation $\ddot{\theta}_0$ seems to get worse for increasing curvature, it is unlikely due to the decreasing velocity. The KKT-value of the NLP solution increases, which indicates that the quality of the solution decreases. This can be explained by the increase of lateral jerk in the approximation and the convergence issues of the Gauss-Newton algorithm with non-zero objective values.

Finally, the distance $s_{1f} - s_4$ is shown in Figure 6-22b. The distance between the front axle of the tractor and the rear axle of the second semi-trailer in a straight orientation is $27.35[m]$. Recall the assumption in the OCP-implementation that the distance $s_{1f} - s_4$ is constant. We observe that the variation in this scenario is very small. For long horizons, such a small error can lead to significant drift in the lateral distance trajectory. But for a short horizon this is not an issue, the assumption is considered valid.



(a) Lateral position of front axle of tractor and rear axle of second semi-trailer.



(b) Longitudinal velocity of tractor.

(c) Road curvature.

Figure 6-21: Results from lane keeping on a constantly increasing road curvature. Trajectory generator is in closed-loop with high-fidelity plant model.

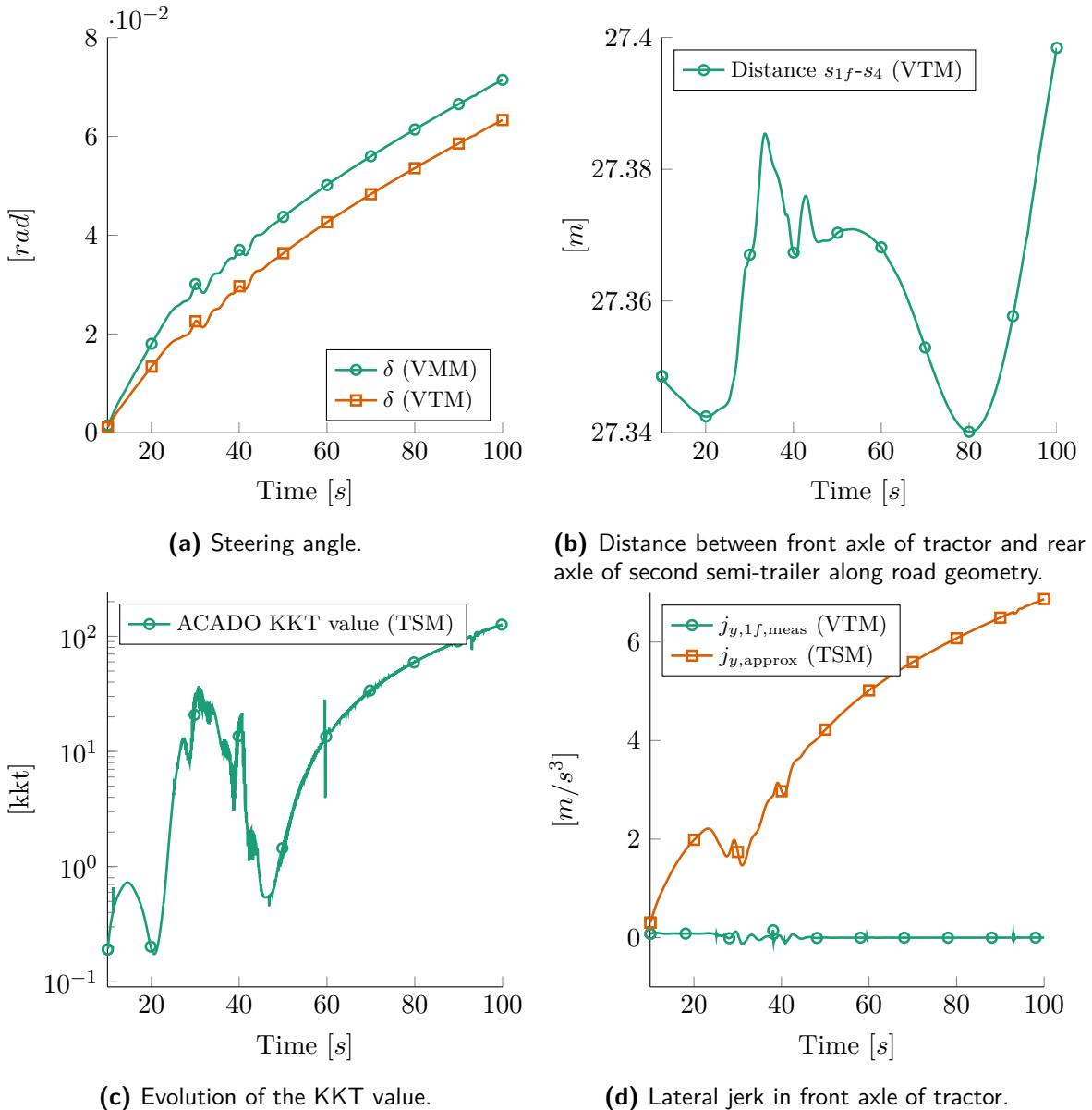


Figure 6-22: Results from lane keeping on a constantly increasing road curvature. Trajectory generator is in closed-loop with high-fidelity plant model.

6-5 Summary

A considerable number of simulations are discussed in this chapter. It is attempted to identify the strengths and weaknesses of the RTI-algorithm applied to the A-double combination in the most common highway maneuvers. From the results we mainly learned the following:

- The RTI scheme can control the single-track model barely losing optimality of the NLP over time.
- Despite a slight model-mismatch, a high-fidelity plant model of the A-double combination is successfully controlled in closed-loop with the trajectory generator. The trajectories were directly applied to the low-level steering control and the velocity tracker.
- Real-time performance for the receding horizon optimization-based trajectory generator is achieved. Solution times are in the range of $28 - 25[ms]$ for the complete step and a feedback delay of approximately $3[ms]$ is maintained.
- The cost function of the OCP successfully minimizes lateral jerk and achieves reference tracking of lateral distance and speed. The jerk approximation is inaccurate on highly curved roads when applied to the high-fidelity plant.
- With the predicted collision free corridor formulation, a (sub)optimal trajectory is obtained in a merging maneuver to lane of lower velocity. Future research in a nonlinear formulation of the predicted collision free corridor may enable compatibility of the OCP with merging maneuvers to lane with a positive velocity difference.

The trajectory generator can handle the main foreseeable scenarios: lane keeping, lane changing and merging on highways of arbitrary curvature. The abort maneuver was identified as the most challenging action. Nonlinear constraints on the lateral acceleration are satisfied in closed-loop simulations with the single-track model on a curved road. Lane boundary constraints are obeyed in closed-loop simulations with the high-fidelity plant in an abort maneuver on a straight road.

In the next and final chapter, the main conclusions from this thesis work are formulated, the contributions are summarized and opportunities for future work are outlined.

Chapter 7

Conclusions and recommendations for future work

This thesis presents the real-time implementation of a computationally efficient receding horizon scheme for trajectory generation of an A-double combination truck. The Real-Time Iteration (RTI) solution scheme [6] is applied to an optimal control problem (OCP) defining open-loop optimal highway driving. Simulations with a high-fidelity vehicle model show acceptable results in terms of dynamic response of the vehicle and computational time. Real-time performance is not a tale for the future.

The main contributions of this thesis are:

- The formulation of an OCP for highway driving of an A-double combination, with a spatial prediction horizon defined along the road geometry.
- The formulation in an OCP for collision avoidance and distance-keeping from other traffic in highway lane keeping and merging.
- Real-time implementation of the optimization-based trajectory generator applied a long heavy vehicle combination (LHVC) using the RTI-algorithm.

Additionally, the implementation of the presented algorithms in stand-alone executable C/C++ code at the VTI motion simulator will allow for future research in driver acceptance of the active safety functionalities in long heavy vehicle combinations (LHVCs).

7-1 Summary of the conclusions

The main conclusions that are made in this thesis work, with regard to the real-time implementation of an optimization-based trajectory generator for LHVCs on highways, are:

1. Using the RTI-scheme [6], trajectory generation with an eighteenth-order non-linear prediction model can be implemented for highway maneuvers maintaining real-time performance. Average solution times of 20[ms] and a feedback delay of 3[ms] for a spatial OCP with a prediction horizon of 100[m] and a discrete step size of 1[m] is achieved.
2. The optimization objective, to minimize jerk, track a lane reference and velocity, can be effectively formulated in an OCP in the model states of a single-track vehicle model of a LHVC. The approximation of the lateral jerk in the states of a single-track vehicle model is shown to be effective in low curvature conditions, but is highly inaccurate for high-curvature conditions combined with the high-fidelity vehicle plant.
3. The optimization-based trajectory generator can successfully execute common highway maneuvers, among which, lane changing and merging. The abort maneuver functionality can safely plan and execute an evasive trajectory for the A-double combination, which satisfies the constraints imposed in the OCP.
4. Closed-loop simulation results of the trajectory generator with a high-fidelity vehicle model showed that a nonlinear single-track prediction model of the A-double combination is suitable for optimal control highway driving. Model-mismatch of the single-track model with the high-fidelity plant has limited .
5. Collision avoidance can be effectively and efficiently incorporated to the OCP using sigmoid functions for the logic nature for the distance-keeping incentive.

7-2 Recommendations for future work

1. **Alternative Hessian approximation algorithms** The Hessian approximation algorithm in the solution strategy of the RTI-algorithm makes drastic assumptions on the least-squares objective function of the OCP near the optimum. The assumption that the objective value of the function, of which a least-squares minimum is attempted to be found, is either close to zero or linear near the optimum has a notable effect on the result of the optimization. Although known globalization techniques, would not trivially fit the RTI-scheme, it would be interesting to see what progress can be made in this area.
2. **Single-track model mismatch** A nonlinear single-track model of an LHVC simplifies the truck dynamics to occur as if the truck has zero width and as if tires behave linear. This introduces a discrepancy with the real behavior, especially when weight shift and high slip occurs. Future work can be done to finding methods dealing with this imposed model-mismatch in high curvature scenarios.
3. **Modeling of longitudinal dynamics** In this work the longitudinal dynamics were modeled as a first-order differential equation in the acceleration. It is assumed that the cruise control always behaves as such. This is achieved by using a relatively slow cruise control. To be less conservative, a longitudinal dynamics model can be incorporated in the OCP. This would involve modeling aerodynamics, approximating the propulsion system, brakes and road inclination.

4. **Non-convex predicted collision-free corridors** Effectively incorporating the surrounding traffic in the OCP involves finding a parameterization that fits most common scenarios. In this thesis work, only one type of parameterization is implemented that assumes a collision-free corridor that is convex at each prediction step. This type of allowed space does not support overtaking another vehicle in the planned optimal trajectory. An alternative formulation that is non-convex, yet compatible with a nonlinear program (NLP) is proposed in this thesis, but not implemented. Future work can be directed to find and analyze improved collision avoidance formulations.
5. **Vehicle model in 3D Euclidean space** In real-life traffic situations, road banking and road inclination [4] may affect performance of the closed-loop system. Unmodeled phenomena in the planar model are e.g. off-tracking due to a laterally inclined road and the effects of driving a hill. Road banking imposes off-tracking for which an appropriate response would be to drive off-center. Instability can occur in case the LHVC drives down hill, because the multiple towed units may act as inverted pendulums and only limited natural yaw damping is present.
6. **Observations of vehicle state, road and surrounding traffic** This thesis work assumed perfect knowledge on the vehicle state, road properties and the behavior of surrounding traffic. It was assumed that the road curvature was available over a horizon of 50[m] behind the truck to 150[m] ahead. Affordable sensors and other sources of information, such as map data, should be in place to provide the trajectory generator with the data it needs. Work needs to be done on analysis in the sensitivity to inaccurate data and noisy measurements. Sensing of the motion of surrounding traffic of all vehicles in the near proximity is required.
7. **External disturbances** Perfect prediction would not only require a high-quality vehicle model, but also an accurate world model for the road and weather. The effect of road friction and other influence on the vehicle dynamics and e.g. the influence cross-wind need to be dealt with explicitly.
8. **Modeling fellow road users** It was assumed that the road users surrounding the truck drove a fully predefined path. In the field of microscopic traffic modeling, models for human driving are researched. It is interesting to see in what degree intelligent human driver models can be incorporated in the trajectory generator. A first step would be to validate the control algorithms on roads occupied by human driver models.
9. **Differential flatness and collocation methods** Previous work in real-time path planning for LHVC used differential flatness of a kinematic model of an LHVC to obtain a computationally efficient formulation of an NLP. It is known that a dynamic single-track model of a car is also differentially flat [29]. The property of differential flatness has already shown to be a great advantage for efficient trajectory planning. Research can be spent to find a differentially flat formulation of a dynamic LHVC model.
10. **Driver acceptance** Finally, it must be emphasized that a lot of work has to be done in the driver acceptance of active safety systems in which the proposed trajectory generator has great influence. It would be interesting to compare the trajectories of professional truck drivers with the closed-loop behavior of the optimization-based control algorithm.

Research in human machine interaction can help to study the effect of a discrepancy between the short-term driver intentions and the control behavior.

Appendix A

Mathematical derivations

A-1 Lateral reference calculation

In lateral reference calculation we solve a simplified optimization problem for the lateral path of the truck. We want to find a balance between a high penalty on desired lateral position with respect to the lane center and smooth lane transition behavior. One method to achieve this, would be to change the cost function of the optimization in different scenarios. However, it is much more elegant to provide a pre-optimized reference to tell the optimization problem to change lane. In this section it is explained how we can find a pre-optimized path for a lane-change maneuver.

Approach to lateral reference trajectory generation

The basis of the reference calculation approach is Pontryagin's maximum principle [30]. Pontryagin's maximum principle is an example of an indirect method to optimal control. In contrast to direct methods, indirect methods tend to solve for the optimality conditions of an optimal control problem instead of trying to minimize the cost functional directly.

Consider we have a dynamical system with the state vector $\zeta(t) \in \mathbb{R}^n$ and control input vector $u(t) \in \mathbb{R}^m$:

$$\frac{d\zeta}{dt} = f(\zeta(t), u(t)) \quad (\text{A-1})$$

We want to control this system over time interval $t \in [0, T]$ using an admissible control signal $u(t) \in \mathcal{U}$.

We define a cost functional $\mathcal{J}(\zeta(t), u(t)) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ that describes an optimality measure for the trajectory:

$$\mathcal{J}(\zeta(t), u(t)) = \int_{t_0}^{t_1} \mathcal{L}(\zeta(t), u(t)) dt \quad (\text{A-2})$$

With the Lagrangian \mathcal{L} to define the cost of the trajectory. The system dynamics constraints Eq. (A-1) are adjoined to the Lagrangian Eq. (A-2) using the vector of Lagrange multipliers $\lambda(t) \in \mathbb{R}^n$ to construct the Hamiltonian $\mathcal{H}(\zeta(t), u(t), \lambda(t)) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$:

$$\mathcal{H}(\zeta(t), u(t), \lambda(t)) = \mathcal{L}(\zeta(t), u(t)) + \lambda^T f(\zeta(t), u(t)) \quad (\text{A-3})$$

Subsequently we turn this optimal control problem into a two-point boundary value problem by introducing start- and end-state constraints:

$$\zeta(t_0) = \zeta_0 \quad \text{and} \quad \zeta(t_1) = \zeta_1 \quad (\text{A-4})$$

Minimization of the Hamiltonian

According to Pontryagin's maximum principle, minimizing the optimal control problem is equivalent to minimizing the Hamiltonian. Consider the optimal control input $u^*(t)$, a necessary condition is:

$$\mathcal{H}(\zeta(t), u^*(t), \lambda(t)) \leq \mathcal{H}(\zeta(t), u^*(t), \lambda(t)) \quad (\text{A-5})$$

over time interval $t \in [t_0, t_1]$.

For a quadratic cost function combined with a linear dynamic system, this $u^*(t)$ can be found using the first order optimality condition:

$$\frac{\partial \mathcal{H}}{\partial u} = 0 \quad (\text{A-6})$$

As we will see later, this first-order optimality condition will be a sufficient condition for optimality in the case that is considered here.

The adjoint equations

The constraints of the minimization problem are represented by the adjoint equations and the adjoint variables $\lambda(t)$. The adjoint equations state that:

$$\frac{d\lambda(t)}{dt} = -\frac{\partial \mathcal{H}}{\partial \zeta} \quad (\text{A-7})$$

The reference calculation problem

As introduced, we would like to find a suboptimal lateral position trajectory that will serve as a reference to the model predictive control algorithm. Let us construct the ingredients for the optimal control problem:

- The lateral spatial system dynamics;
- The cost functional over the spatial horizon;
- The boundary conditions.

Lateral system dynamics

The system dynamics parameterized in the spatial coordinate system defined along the road geometry.

$$\frac{d\zeta(s)}{ds} = \frac{A\zeta(s) + Bu(s)}{ds/dt(s)}$$

with $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

(A-8)

with the state-vector and control input:

$$\zeta(s) = [d_1(s) \quad \dot{d}_1(s) \quad \ddot{d}_1(s)]^T, \quad u(s) = \ddot{d}_1(s)$$
(A-9)

We can simplify this problem by keeping $\frac{ds}{dt}(s)$ constant using the measured longitudinal velocity of the truck during the initialization of the lane change. Hence, we replace $\frac{ds}{dt}(s) \rightarrow v_{x,\text{base}}$ which will be referred to as the base velocity of the lane change maneuver. We can write the system equations as follows:

$$\begin{aligned} \frac{d\zeta_1}{ds} &= \frac{\zeta_2}{v_x(s_0)} \\ \frac{d\zeta_2}{ds} &= \frac{\zeta_3}{v_{x,\text{base}}} \\ \frac{d\zeta_3}{ds} &= \frac{u}{v_{x,\text{base}}} \end{aligned}$$
(A-10)

Cost function for the optimality measure

The goal of the reference calculation algorithm is the same as the model predictive control algorithm: minimize lateral jerk. Hence we define the cost function as:

$$\mathcal{J}(\zeta(t), u(t)) = \int_{s=s_0}^{s_0+\Delta s_{lc}} \frac{1}{2} u(\sigma)^2 d\sigma$$
(A-11)

for the minimization problem:

$$\min_{u(s)} \mathcal{J}(\zeta(t), u(t))$$
(A-12)

We can interpret this cost functional as minimizing the energy in the jerk over the horizon $s \in [s_0, s_0 + \Delta s_{lc}]$.

Boundary conditions of the minimization problem

As boundary conditions for the minimization problem, we choose to constraint the states in both s_0 and $s_0 + \Delta s_{lc}$. Consider that the initial and end states are equal to:

$$\begin{aligned} \zeta_1(s_0) &= d_{1,0} & \zeta_1(s_0 + \Delta s_{lc}) &= d_{1,1} \\ \zeta_2(s_0) &= 0 & \zeta_2(s_1) &= 0 \\ \zeta_3(s_0) &= 0 & \zeta_3(s_1) &= 0 \end{aligned}$$
(A-13)

As one can see, we desire that the reference profile moves from y_0 to y_1 where lateral velocity and acceleration are zero outside the transition interval $s \in (s_0, s_0 + \Delta s_{lc})$.

Solution to the reference calculation problem

Now we have all the ingredients, we can actually solve the problem. First we construct the Hamiltonian:

$$\mathcal{H}(\zeta(s), u(s), \lambda(s)) = \frac{1}{2}u(s)^2 + \lambda_1(s)\frac{\zeta_2(s)}{v_{x,\text{base}}} + \lambda_2(s)\frac{\zeta_3(s)}{v_{x,\text{base}}} + \lambda_3(s)\frac{u(s)}{v_{x,\text{base}}} \quad (\text{A-14})$$

In order to find the optimal $u^*(s)$ that minimizes the Hamiltonian we use the first order optimality condition:

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial u} &= 0 \\ u^*(s) &= -\frac{\lambda_3(s)}{v_{x,\text{base}}} \end{aligned} \quad (\text{A-15})$$

Secondly we use the adjoint equations to find the trajectories for the Lagrange multipliers:

$$\begin{aligned} -\frac{\partial \mathcal{H}}{\partial \zeta_1} &= \frac{d\lambda_1}{ds} : \quad \frac{d\lambda_1}{ds} = 0 \\ -\frac{\partial \mathcal{H}}{\partial \zeta_2} &= \frac{d\lambda_2}{ds} : \quad \frac{d\lambda_2}{ds} = -\lambda_1 \\ -\frac{\partial \mathcal{H}}{\partial \zeta_3} &= \frac{d\lambda_3}{ds} : \quad \frac{d\lambda_3}{ds} = -\lambda_2 \end{aligned} \quad (\text{A-16})$$

The trajectories for the Lagrange multipliers can now be found easily by integrating over s , let us start with λ_1 :

$$\begin{aligned} \lambda_1(s) &= C_1 \\ \lambda_2(s) &= -C_1s + C_2 \\ \lambda_3(s) &= \frac{1}{2}C_1s^2 - C_2s + C_3 \end{aligned} \quad (\text{A-17})$$

We can connect these trajectories using the first order optimality. We obtain the following expression for the control input and at the same time the lateral jerk profile:

$$u^*(s) = \frac{-\frac{1}{2}C_1s^2 + C_2s - C_3}{v_{x,\text{base}}} \quad (\text{A-18})$$

With the point-mass system model we obtain the solution to the first order differential equations by simply integrating over s .

$$\begin{aligned} \zeta_3(s) &= \frac{1}{v_{x,\text{base}}^2} \left(-\frac{1}{6}C_1s^3 + \frac{1}{2}C_2s^2 - C_3s + C_4 \right) \\ \zeta_2(s) &= \frac{1}{v_{x,\text{base}}^3} \left(-\frac{1}{24}C_1s^4 + \frac{1}{6}C_2s^3 - \frac{1}{2}C_3s^2 + C_4s + C_5 \right) \\ \zeta_1(s) &= \frac{1}{v_{x,\text{base}}^4} \left(-\frac{1}{120}C_1s^5 + \frac{1}{24}C_2s^4 - \frac{1}{6}C_3s^3 + \frac{1}{2}C_4s^2 + C_5s + C_6 \right) \end{aligned} \quad (\text{A-19})$$

We have obtained the quintic polynomial that describes a jerk-optimal lateral reference profile for the lane change maneuver. All that is left is to find the coefficients C_i , $\forall i \in \{1, \dots, 6\}$ using the boundary conditions we defined earlier. Additionally, the term Δs_{lc} is replaced by $\Delta t_{lc} v_{x,base}$ to express the lane change duration in time rather than distance.

$$\begin{aligned}
 C_1 &= \frac{720(d_{1,0} - d_{1,1})}{\Delta t_{lc}^5 v_{x,base}} \\
 C_2 &= \frac{720(d_{1,0} - d_{1,1}) s_{lc}}{\Delta t_{lc}^5 v_{x,base}} \\
 C_3 &= -\frac{30(d_{1,0} - d_{1,1})(\Delta t_{lc}^2 v_{x,base}^2 - 12s_{lc}^2)}{\Delta t_{lc}^5 v_{x,base}} \\
 C_4 &= -\frac{30(d_{1,0} - d_{1,1})(\Delta t_{lc}^2 s_{lc} v_{x,base}^2 - 4s_{lc}^3)}{\Delta t_{lc}^5 v_{x,base}} \\
 C_5 &= -\frac{15(d_{1,0} - d_{1,1})(\Delta t_{lc}^2 v_{x,base}^2 - 4s_{lc}^2)^2}{8\Delta t_{lc}^5 v_{x,base}} \\
 C_6 &= \frac{6(d_{1,0} - d_{1,1})s_{lc}^5}{\Delta t_{lc}^5 v_{x,base}} + \frac{5(d_{1,1} - d_{1,0})s_{lc}^3 v_{x,base}}{\Delta t_{lc}^3} \\
 &\quad + \frac{15(d_{1,0} - d_{1,1})s_{lc}v_{x,base}^3}{8\Delta t_{lc}v_{x,base}} + \frac{1}{2}(d_{1,0} + d_{1,1})v_{x,base}^4
 \end{aligned} \tag{A-20}$$

Results

We now have a quintic polynomial that describes a jerk optimal reference trajectory:

$$\begin{aligned}
 d_{1,ref}(s) &= -\frac{6(d_{1,0} - d_{1,1})(s - s_{lc})^5}{\Delta t_{lc}^5 v_{x,base}^5} + \frac{5(d_{1,0} - d_{1,1})(s - s_{lc})^3}{\Delta t_{lc}^3 v_{x,base}^3} \\
 &\quad - \frac{15(d_{1,0} - d_{1,1})(s - s_{lc})}{8\Delta t_{lc}v_{x,base}} + \frac{1}{2}(d_{1,0} + d_{1,1})
 \end{aligned} \tag{A-21}$$

Explanations of the parameters can be found in Table A-1.

Table A-1: Free parameters in the reference calculation expression for the lane change maneuver.

Parameter	Tuning/Fixed	Explanation
s	Fixed	Distance for lateral reference.
$d_{1,0}$	Fixed	The initial lateral distance of the reference profile.
$d_{1,1}$	Fixed	The final lateral distance of the reference profile.
s_{lc}	Tuning	The position half way the lane change maneuver.
Δt_{lc}	Tuning	The time it takes to complete the lane change maneuver.
$v_{x,base}$	Fixed	the base velocity of the lane change maneuver.

This relatively simple expression is of tremendous help for the trajectory generation controller.

Implementation

The reference calculation algorithm will recompute the reference trajectory every time step. This is due to the fact that the parameters change throughout the maneuver. The main reason is that the coordinate frame of the truck changes during a lane change. This is caused by a change in the lane in which the truck is residing. The controller deals with this changing reference frame by abstracting the lanes as objects independent from the coordinate frame. This means that especially the values $d_{1,0}$ and $d_{1,1}$ need to be recomputed each time step.

- The values for $d_{1,0}$ and $d_{1,1}$ change each time the coordinate frame of the truck changes.
- The values for s_{lc} and Δt_{lc} are fixed for the controller and will not change after tuning.
- The value for $v_{x,\text{base}}$ is chosen once per lane change maneuver. In general, the longitudinal velocity of the truck at time of a lane change initiation is used.

Bibliography

- [1] J. Aurell and T. Wadman, “Vehicle combinations based on the modular concept,” Tech. Rep. 1, NVF-reports, 2007.
- [2] M. Kindt, A. Burgess, M. Quispel, S. Van Der Meulen, and M. Bus, “Monitoring Modal Shift: Longer and heavier vehicles The follow-up measurement,” tech. rep., Dutch Ministry of Infrastructure and the Environment, 2011.
- [3] J. Bing, M. Roelofsen, L. Jin, and B. van Arem, “Lane Change and Overtaking Collisions: Causes and Avoidance Techniques,” in *Wireless Vehicular Networks for Car Collision Avoidance* (R. Naja, ed.), pp. 143–187, Springer New York, 2013.
- [4] M. S. Kati, “Definitions of Performance Based Characteristics for Long Heavy Vehicle Combinations,” tech. rep., Chalmers University of Technology, 2013.
- [5] M. W. Disse, *Real-time Path Planning for Long Heavy Vehicle Combinations*. Msc thesis, Delft University of Technology, 2012.
- [6] M. M. Diehl, *Real-Time Optimization for Large Scale Nonlinear Processes*. Phd thesis, University of Heidelberg, 2001.
- [7] B. Houska, H. J. Ferreau, and M. M. Diehl, “An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range,” *Automatica*, vol. 47, pp. 2279–2285, Oct. 2011.
- [8] J. V. Frasch, A. J. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. M. Diehl, “An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles,” in *Proc. of the European Control Conference*, (Zürich, Switzerland), pp. 4136–4141, 2013.
- [9] P. Nilsson and K. Tagesson, “Single-track models of an A-double heavy vehicle combination,” tech. rep., Chalmers University of Technology, Göteborg, Sweden, 2013.

- [10] B. Houska, H. J. Ferreau, and M. M. Diehl, “ACADO toolkit - An open-source framework for automatic control and dynamic optimization,” *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [11] R. Rajamani, *Vehice Dynamics and Control*. Springer Science, 2nd ed., 2012.
- [12] H. Pacejka, *Tyre and vehicle dynamics*. Butterworth-Heinemann, 2006.
- [13] P. d. C. Manfredo, *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.
- [14] Y. Gao, A. J. Gray, J. V. Frasch, T. Lin, H. E. Tseng, J. K. Hedrick, and F. Borrelli, “Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles,” in *Proc. of the 11th International Symposium on Advanced Vehicle Control*, (Seoul, South Korea), 2012.
- [15] N. J. Van Duijkeren, *Real-time path planning and obstacle avoidance for a long heavy vehicle combination in a lane-change maneuver*. Literature survey, Delft University of Technology, 2013.
- [16] T. Toledo, “Driving Behaviour: Models and Challenges,” *Transport Reviews*, vol. 27, pp. 65–84, Jan. 2007.
- [17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [18] F. Borrelli, T. Keviczky, and G. J. Balas, “Collision-free UAV formation flight using decentralized optimization and invariant sets,” in *2004 43rd IEEE Conference on Decision and Control (CDC)*, (Atlantis, Bahamas), pp. 1099–1104, Ieee, 2004.
- [19] J. T. Betts, “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [20] M. M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations,” *Journal of Process Control*, vol. 12, pp. 577–585, 2002.
- [21] H. G. Bock and K. J. Plitt, “A multiple shooting algorithm for direct solution of optimal control problems,” in *Proc. of the 9th IFAC Wolrd Congress*, pp. 242–247, 1984.
- [22] P. T. Boggs and J. W. Tolle, “Sequential Quadratic Programming,” *Acta Numerica*, vol. 4, pp. 1–51, 1995.
- [23] H. J. Ferreau, H. G. Bock, and M. M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
- [24] D. B. Leineweber, *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*. Phd thesis, Ruprecht-Karls-Universität Heidelberg, 1998.

-
- [25] J. B. Rawlings, "Tutorial Overview of Model Predictive Control," *Control Systems, IEEE*, vol. 20, no. 3, pp. 38–52, 2000.
 - [26] B. Houska and H. J. Ferreau, *ACADO Toolkit User's Manual*. <http://www.acadotoolkit.org/>, jan 2014 ed., 2014.
 - [27] J. Mattingley and S. Boyd, "Real-Time Convex Optimization in Signal Processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 50–61, 2010.
 - [28] K. Tagesson, P. Sundstrom, L. Laine, and N. Dela, "Real-time Performance of Control Allocation for Actuator Coordination in Heavy Vehicles," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 685–690, 2009.
 - [29] J. hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, T. Panagiotis, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proc. of the American Control Conference*, pp. 188–193, Ieee, 2013.
 - [30] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. Wiley-Interscience, english ed., 1962.

Glossary

List of Acronyms

Volvo GTT	Volvo Group Truck Technologies
VTI	Swedish National Road and Transport Research Institute
VTM	Volvo Truck Model
UDP	User Datagram Protocol
STM	single track model
RWA	rearward amplification
RTI	Real-Time Iteration
MPC	model predictive control
NMPC	nonlinear model predictive control
RT	real-time
OCP	optimal control problem
NLP	nonlinear program
LP	linear program
PSD	positive semi-definite
PD	positive definite
QP	quadratic program
SQP	sequential quadratic programming
COM	center of mass
KKT	Karush-Kuhn-Tucker

LHVC	long heavy vehicle combination
LHVCs	long heavy vehicle combinations
TSM	Traffic Situation Management
VMM	Vehicle Motion Management
CC	cruise controller

List of Symbols

δ	Road wheel angle of front axle of tractor.
\mathcal{H}	The Hamiltonian of a system.
\mathcal{L}	The Lagrangian of a system.
θ_0	Yaw angle of tractor (unit 1) with some global reference frame. Equal to ϕ_1 .
θ_1	Articulation angle between the tractor and the first semi-trailer.
θ_2	Articulation angle between the first semi-trailer and the converter dolly.
θ_3	Articulation angle between the converter dolly and the second semi-trailer.
ϕ_1	Yaw angle of tractor (unit 1) with some global reference frame. Equal to θ_0 .
ϕ_2	Yaw angle of first semi-trailer (unit 2) with some global reference frame.
ϕ_3	Yaw angle of dolly (unit 3) with some global reference frame.
ϕ_4	Yaw angle of second semi-trailer (unit 4) with some global reference frame.
$\phi_{R,*}$	Road heading in perpendicular projection of point * on the truck.
ξ_{lat}	State vector for the A-double combination for the lateral dynamics.
ξ_{lhvc}	State vector for the A-double combination for the lateral and point mass longitudinal dynamics.
ξ_{ocp}	State vector of the prediction model in the OCP.
$v_{x,*}$	Longitudinal velocity of the truck in position *.
$v_{y,*}$	Lateral velocity of the truck in position *.
Δs	Spatial discretization step size of OCP and prediction model.
$\Delta s_{o,*}$	Distance from obstacle * measured from front axle. In optimal control problem (OCP) the state is measured from the rear or front end of the truck.
$\hat{*}$	Normalized vector of *.
$\bar{*}$	Lower bound on state *.
$\bar{\bar{*}}$	Upper bound on state *.
d_*	Perpendicular distance of position * in truck with road geometry.
f_*	Generic notation for functions, identified by *.
g	Set of first order differential equations for spatial OCP prediction model.
K_*	Weight of state or expression * in least squares cost function of the OCP.
l	Number of external parameters in OCP formulation.

m	Number of control input in the OCP prediction model.
N	Number of shooting nodes in the OCP discretization.
n	Number of states in the OCP prediction model.
n_*	Normalization term for weight on * in cost function.
N_c	Number of inequality constraints in the OCP formulation.
N_f	Number of least-squares objective terms in the OCP formulation.
s_*	Coordinate of orthogonal projection of position * in truck on road geometry.
s_f	Length of the spatial prediction horizon in OCP.
u	Control input vector.
${}^G *$	The state(-vector) * with respect to a global Euclidean coordinate frame.
${}^L_{1f} *$	The state(-vector) * as observed in the local body-fixed coordinate frame in the front axle of the tractor.
${}^L_1 *$	The state(-vector) * as observed in the local body-fixed coordinate frame in the center of mass of the tractor.
${}^L_2 *$	The state(-vector) * as observed in the local body-fixed coordinate frame in the rear axle of the first semi-trailer.
${}^L_3 *$	The state(-vector) * as observed in the local body-fixed coordinate frame in the rear axle of the dolly.
${}^L_4 *$	The state(-vector) * as observed in the local body-fixed coordinate frame in the rear axle of the second semi-trailer.

