Eindhoven University of Technology
Department of Mechanical Engineering
Dynamics & Control

Online Motion Planning for All-Wheel Drive Autonomous Race Cars.

MSc. Thesis
DC 2022.088

M.R. (Mischa) Huisman
0971152

Supervisors:
dr.ir. A.A.J. Lefeber

Third Version

Eindhoven, January 2022

Abstract

This thesis considers an online optimization-based motion planning system for an all-wheel drive autonomous racing car, driving in a formula student competition. These competitions consist of unknown track layouts where no head-to-head racing is allowed. The motion planning system is responsible for the localization of the vehicle with respect to a reference path and determining the optimal control input to drive this reference path as fast as possible. First, a localization method is defined in the Frenet coordinate system to determine the error coordinates. Then, to mitigate external disturbances on the controller input, an Extended Kalman Filter is developed to observe the vehicle velocities and error states. The optimal control input is determined with nonlinear model predictive control, which is often limited by the computational burden, simplified vehicle models, or relying on a hierarchical control strategy. In this thesis, the principle of cascading vehicle models within a single horizon is applied to obtain a long horizon while using a two-track model, including a nonlinear tire model, to determine the optimal control input. The considered inputs are the desired longitudinal force at each of the wheels and the desired steering rate. The primary objective is to minimize the time to reach the end of the prediction horizon, where a transformation to the spatial domain is used to formulate time as an optimization variable. Via simulations, the performance of different horizon configurations is compared using lap time, computation time, and vehicle behavior. These results indicate that similar performance can be obtained while reducing the average solve time compared with a single two-track model.

Acknowledgements

In this master's thesis report you will find the result of my final project to graduate from my Master of Science in Automotive Technology at the Technical University of Eindhoven. This thesis work took place from February 2022 till November 2022. During this time the world had slowly recovered from the COVID-19 pandemic, resulting in a more open society which essentially led to more brainstorming, discussions, and suggestions. It is safe to say I could not have achieved the results I have now without the people I had all these conversations with, and therefore I would like to thank the people who have supported me during my final year as master student.

First of all, I would like to thank Dr.Ir. Erjen Lefeber, who functioned as my supervisor during the entire project and tirelessly supported me throughout my final year as master student at TU/e. Not only did he provide me with useful feedback during the weekly meetings, but he also provided me with new literature and gave clever comments or suggestions which contributed to the quality of my work. Our meetings helped me structure my work and ensuring my progression throughout the project. I am happy and looking forward to continue working together during my time as Ph.D. student.

Secondly, I would like to express my gratitude towards Dr.Ir. Mircea Lazar for sharing his knowledge about model predictive control. During my internship you already showed great interest in my work, and now again during my master thesis. When I reached out to you, you got me in touch with other students working with similar tools, allowing me to have more detailed discussions about the implementation of the CasADi tool.

I would also like to express my gratitude towards Dr. Ir. Mauro Salazar, who took the time during my literature studies to provide me with useful literature regarding autonomous racing. Not only this literature, but also the discussions we had inspired me to formulate the research goal for this project.

Finally, I would like to thank all the people at University Racing Eindhoven. The team showed great interest in my project, and we will continue to work towards the final goal and which is the online implementation of model predictive control.

Over the past few months, I have learned a lot about optimization tools and online motion planning, but most of all I learned a lot about myself. This project showed me the joy I get while doing research and having all different sorts of discussions to achieve different goals. Therefore, I was really happy when the opportunity arose to do a Ph.D. study at TU/e, which I gladly accepted. Thanks to everyone who showed interest in this project and read my project report, and to whoever will.

Mischa Huisman
November 2022

Table of Contents

Abstract i

Acknowledgements ii

Nomenclature v

1 Introduction 1

1.1 Control Challenges . . . 2
1.2 Current Solutions . . . 2
1.3 Research Goal . . . 6
1.4 Thesis Outline . . . 7

2 Preliminaries 8

2.1 Theories . . . 8
2.2 Dynamics . . . 9
2.3 Discretization Methods . . . 13
2.4 Observations and Conclusion . . . 16

3 Localization 17

3.1 Footpoint . . . 17
3.2 Extended Kalman Filter . . . 19
3.3 Results . . . 24
3.4 Observation and Conclusion . . . 35

4 Optimization Problem 36

4.1 Cascaded Vehicle Model . . . 37
4.2 Problem Formulation . . . 38
4.3 Constraints . . . 43

4.4 Variable Step Size 49
4.5 Observations and Conclusions 53

5 Results **55**
5.1 Feasibility 55
5.2 Model Comparison 64
5.3 Linear Tyre Constraints 72
5.4 Convergence 75
5.5 Conclusions and Observations 77

6 Conclusion and Recommendations **78**
6.1 Conclusion 78
6.2 Recommendations 79

A Variable Step Size **I**

B Extended Kalman Filter Implementation **IV**

Nomenclature

Accents

$\dot{}$	Time-derivative
$'$	Derivative
$\bar{}$	Associated with the point mass model
$\hat{}$	Extended Kalman Filter state
$\tilde{}$	Associated with the single track model

Abbreviations

CG	Center of Gravity
EM	Electric motor
EKF	Extended Kalman Filter
FSD	Full Self-Driving
HMPC	Hierical Model Predictive Control
LLC	Low Level vehicle Control
LTO	Lap Time Optimisation
LPV	Linear Parameter Varying System
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
MPCC	Model Predictive Contouring Control
NMPC	Nonlinear Model Predictive Control
NLP	Nonlinear Programming
RC	Radio-Controlled
RK2	Second Order Runge-Kutta Discretization
RK4	Fourth Order Runge-Kutta Discretization
RMSE	Root Mean Square Error
SQP	Sequential Quadratic Programming
SLAM	Simultaneous Localization And Mapping
TRO	Trajectory Optimization
URE	University Racing Eindhoven

Sub- and superscripts

0	Initial condition
1	Left front tire
2	Left rear tire
3	Right front tire
4	Right rear tire
ψ	Heading
b	Bound
D	Drag
f	Front
i	Prediction step two track model
j	Prediction step single track model
k	Time step
l	Left
min	Minimum
max	Maximum
N	Horizon length
R	Rolling
r	Rear
r	Reference
r	Right
ss	Steady-State
T	Transpose
x	Longitudinal direction
y	Lateral direction
z	Vertical direction
z	Prediction step point mass model

Matrics/Columns

Θ	Observation space
Φ	Observability contribution
e	Error vector
F	State transition matrix
H	Observation matrix
I	Identity matrix
K	Kalman gain
P	Covariance matrix
Q	Process noise covariance
Q	State cost matrix
R	Input cost matrix
R	Observation noise covariance

Sets/Maps

$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
Φ	Observability contribution
Θ	Observation space
$\hat{\Theta}$	Observation map
$\mathcal{U}$	Set of feasible inputs
U	Predicted MPC inputs
$\mathcal{X}$	Set of feasible states
X	Predicted MPC states

Operators

$\dot{(\cdot)}$	Time derivative
∂	Partial derivative
$(\cdot)'$	Derivative
Δ	Difference
$\times$	Times
$\approx$	Approximation
$\forall$	For all
$\neq$	Not equal to
$>$	Greater than
$\geq$	Greater or equal
$\gg$	Much greater than
$<$	Less than
$\leq$	Less or equal
$\ll$	Much less than
$\in$	Is element of
$\{\}$	Set
$\subset$	Proper subset
$\implies$	Implies
$\rightarrow$	Maps to
$\Leftrightarrow$	Equivalent
$\mid$	Such That
$\lvert\cdot\rvert$	Absolute
$\lvert\lvert(\cdot)\rvert\rvert$	Norm
$\int$	Integral
$\sum$	Sum
$\intercal$	Transpose
det	Determinant
d	Derivative
diag	Diagonal Matrix
max	Maximum
min	Minimum
L_f	Lie derivative

Symbols

α	Tire side slip angle	$[rad]$
β	Vehicle side-slip angle at CG	$[rad]$
δ	Steering angle	$[rad]$
∞	Infinity	
κ	Curvature	$[m^{-1}]$
μ	Friction coefficient	
ω	Rotational velocity	$[rad\,s^{-1}]$
ψ	Heading Vehicle	$[rad]$
ρ	Density	$[kg\,m^{-3}]$
θ	Path heading	$[rad]$
A	Surface	$[m^2]$
a	Acceleration	$[m\,s^{-2}]$
a	Slope linear function	
B	Stiffness factor Pacjeka tire model	
b	Intercept linear function	

C	Constant	
C	Shape factor Pacjeka tire model	
C	Cornering stiffness	$[N\,rad^{-1}$
D	Peak factor Pacjeka tire model	
e	Error	
F	Force	$[N]$
f	Friction coefficient	
g	Inequality constraints	
H	Horizon length single track model	
h	Equality constraints	
h	Output mapping	
h	Step size	$[s]$
I	Moment of Inertia	$[kgm^2]$
i	Prediction step two track model	
J	Cost function	
j	Prediction step point mass model	
k	Time step in discrete time	
k	Slope	
L	Distance	$[m]$
L	Wheelbase	$[m]$
l	Length from CG till respective axle	$[m]$
M	Horizon length point mass model	
M	Moment	$[Nm]$
m	Mass	$[kg]$
m	Output dimension	
N	Horizon length two-track model	
n	State dimension	
O	Origin	
P	Altitude triangle footpoint	$[m]$
P	Number of intersection points	
p	Distance reference point and CG	$[m]$
R	Radius	$[m]$
R	Cartesian coordinate frame	
S	Semiperimeter	$[m]$
s	Progress along reference path	$[m]$
T	Frenet coordinate frame	
t	Time	$[s]$
u	Model input	
V	Resulting vehicle velocity	$[m\,s^{-1}]$
v	Velocity	$[m\,s^{-1}]$
w	Vehicle's trackwidth	[m]
w	Weight distribution	
x	x-position	$[m]$
x	Model states	
y	y-position	$[m]$
y	Model output	
z	Prediction step single track model	

Chapter 1

Introduction

The automotive industry has evolved in the past few years, focusing on replacing the human driver step-by-step, where the ultimate goal is towards Full Self-Driving (FSD) vehicles. In 2020 Tesla released their FSD beta software, allowing its customers to exploit the advantages of entirely autonomously driving cars on public roads [1]. To allow FSD vehicles on the public road, the software has to handle all situations that can occur on the road, including emergency situations. These particular situations result in the software handling the vehicle at its performance limits to avoid accidents. Yet research showed that current software could not reach the level of performance of an expert human driver. To bridge the gap between autonomous driving and driver performance, the main improvements for future work are online adaption of the safety distances, online assessment of the vehicle performance limit considering velocity-dependent influences, and robust controlling of the vehicle at the handling limits [2].

Many future challenges lay within the scope of motion planning, which can be divided into three different tasks; 1. perception/localization, 2. motion strategy/path planning, and 3. motion control [3] [4]. This project will focus on the path planning and vehicle control aspect, where the goal is to seek the limits of a car while driving. Seeking the car's limits is one of the main goals of racing, where racing has proven itself over the years to ensure fast testing, development, and enhancement of new technologies [5], resulting in the typical vehicle we know nowadays [6]. Furthermore, the advent of autonomous racing events, such as Roborace [7] and Formula Student Driverless Cup [8], will contribute towards developing the typical vehicle of the future.

Available motion planners for autonomous racing applications can be distinguished into three categories. The first category includes motion planning systems that track different control inputs using static feedback control. The static feedback controller uses a trajectory, which is also determined offline. The second category consists of pure online optimization-based motion planning, where both path planning and reference tracking are optimized online, providing more flexibility to the system. At last, the third category combines the benefits of offline trajectory optimization to perform path planning and uses an online optimization-based method to follow this trajectory, providing computation times that allow for real-time implementations [9], but can only be used when the track layout is known beforehand.

This project focuses on developing an online motion planning system that fits under category two and is therefore responsible for online path planning and tracking control. Additionally, an autonomous racing car, in the presence of actuator and track limits, and no overtaking of any competitors are considered. The considered race car for this project is shown in Figure 1.1, which is the autonomous racing car of University Racing Eindhoven (URE). The racing car uses a steering actuator, capable of tracking the desired steering trajectory, and four electric in-wheel motors, which all can be controlled independently to achieve the desired traction. This means that

the control inputs of the vehicle are the steering wheel angle and four motor torques. Furthermore, it is assumed that a state estimator is available to provide all states for online implementation.

Figure 1.1: URE15D at Formula Student Spain 2021.

1.1 Control Challenges

First of all, driving at the limits of the car result in highly nonlinear dynamics. Controlling highly nonlinear dynamics often requires complex controllers and results in complex optimization problems. A dynamic vehicle model including slip is required to include all the nonlinear vehicle behavior, such as tire forces. However, using a vehicle model with slip introduces a singularity into the system when we consider driving at low velocities.

One of the significant difficulties in developing a safe and robust motion planner is the variety of driving situations that occur during racing. Finding a solution for all these situations often requires nonlinear optimization with non-convex cost functions. Unfortunately, solving a non-convex optimization problem heavily depends on the initialization, resulting in only local rather than global minima can be found [10]. Moreover, non-convex optimizations do not guarantee convergence to the same solution for different driving scenarios.

Using model predictive control (MPC) to implement online motion planning requires an efficient algorithm. Since the introduction of MPC, engineers have been challenged to find the balance between accurate results and the required computational power. The prediction horizon length is a crucial parameter to obtain stability and determines the complexity of the optimization problem. In general, the applications with long prediction horizons should provide smoother results than a short prediction horizon [11] [12]. Ideally, one would like to use an infinite horizon length, tending to find the optimal control for all given allowable initial states, but the problem occurs that the open-loop optimal control problem cannot be solved sufficiently fast. As a result, a sufficiently large finite prediction horizon is chosen to obtain computational feasibility and maintain stability [13]. However, providing theoretical proof of NMPC stability is often hard due to the complexity of the optimization problem. Therefore, most often, the stability is analyzed using simulation.

1.2 Current Solutions

Path Tracking Controllers

Over the years, different path-tracking controllers have been developed for autonomous vehicles. The position of a car can be controlled by defining its error coordinates from a given path. A

simple layout for controlling the position on the road can be found via a Pure Pursuit [14], or Stanley Controller [15]. Such controllers use a simple kinematic bicycle model to obtain the desired steering angle. The desired steering angle is based on the error coordinates with respect to a look-ahead point in front of the vehicle and the vehicle heading. The Pure Pursuit and Stanley controllers have been proven to work in low-velocity environments but have poor performance in cases of high velocities, and varying curvature [16].

Among the existing path-tracking controllers, the PID controller is popular due to its simplicity and engineering applications. Autonomous racing requires driving at the vehicle's limits, wherein [17] it is shown that this is possible via a PID controller. The PID controller is combined with feedforward control to minimize the error between the vehicle's center of percussion and a given path. Vehicle stability can be guaranteed via yaw stability control, and the closed-loop system's stability is proven by utilizing Lyapunov theory, even when the rear tires are saturated. However, the PID controller comes with a lack of versatility [18]. The controller parameters can be tuned towards an optimal and desired performance, but once the operating conditions change, the control parameters are no longer optimal [16].

Therefore, optimal control theory is used to find the optimal control input which minimizes an objective function for a given dynamical system. In the case of a linear control system and a quadratic objective function, a linear-quadratic regulator (LQR) can be used as a feedback controller to obtain the optimal control input. An LQR provides the optimal control feedback gains to enhance closed-loop stability while maintaining high performance. The disadvantage for LQR arises when disturbances and transient behavior occur in the system [18]. LQR provides robust stability and is computationally efficient when it comes to linear systems, but non-linear behavior results in a decrease in performance [16]. Therefore, the LQR system often requires additional controllers, such as a PID controller [18], or a non-linear optimal control problem has to be solved via either an augmented LQR [19] or MPC.

MPC is an optimal control method that has been applied since the 1980s but has become a more viable control method due to the improved computing technologies over recent years. The popularity of model predictive control is growing with the increasing system complexity engineers have to deal with these days [16]. An MPC can be implemented with a direct physical understanding of the parameters once the dynamical system has been identified. The growing popularity is also due to the constraint handling of MPC and the key characteristic of solving optimization problems online [20]. However, applying non-linear model predictive control (NMPC) for online reference tracking still results in computational disasters. Also, the NMPC comes with a great challenge to prove the system's stability, leading to the fact that most controllers are verified through simulations rather than mathematical proof [16].

This thesis focuses on a path-tracking controller for autonomous racing applications, which requires the controller to push the vehicle to its limits. To achieve this goal, a complex optimization problem has to be solved while considering different vehicle and path constraints. Based on the information provided here, NMPC is regarded as the most viable solution to push the car to its limits. However, as mentioned, NMPC requires solving computational and stability challenges. Therefore, the following part discusses NMPC methods developed for online reference tracking with autonomous vehicles.

Curvilinear Model Predictive Control

In [9], a hierarchical motion planning strategy is used for autonomous racing. First, a high-level NMPC is used to perform trajectory optimization (TRO) offline based on a curvilinear dynamic bicycle model, which includes nonlinear tire dynamics. Subsequently, the online reference tracking controller uses both the track and optimal trajectory. The online reference tracking controller is an NMPC based on the same curvilinear dynamic bicycle model. The curvilinear dynamic bicycle model includes a torque vectoring moment determined via a proportional controller. The

proportional controller uses some gain p_{tv} and the difference between the desired yaw rate and the actual yaw rate. The use of the proportional controller introduces difficulties in tuning the system since the optimal solution is now dependent on controller input. Secondly, the desired yaw rate is based on a kinematic steady-state cornering bicycle model, which does not provide accuracy at high velocity and does not account for the tire characteristics, potentially causing a performance cap.

Consequently, [21] expanded on the work in [9], with the focus on co-designing the low-level vehicle control (LLC) to work in harmony with the higher-level control. At first, the mid-level controller is adapted to consider the torque vectoring moment as a system input, allowing the high-level trajectory planner to fully utilize the torque vectoring capabilities. Secondly, the solution from the mid-level controller is used as a set point for the LLC, where the yaw rate and acceleration are used to determine the required motor torque and steering input. Computing the required torque per wheel provides a performance boost to the autonomous vehicle, with the potential to outperform an expert racing driver. Since the system can determine the optimal torque trajectory over the horizon, the controller distinguishes itself from the typical torque vectoring controller used in non-autonomous racing vehicles, where the driver's command is directly mapped to the total torque demand [22]. After equally distributing the desired torque between the left and right motors, the normal force is used to scale the torque, accounting for the gravitational and downforce working on the tires. Since a bicycle model without an elevated center of gravity is used, the considered normal force on the tire does not account for any load transfer due to vehicle roll or pitch. To increase the vehicle's performance, it is desirable to include all motions that affect the normal force since the available grip directly depends on the normal force acting on the tire [23].

Using nonlinear vehicle models for online optimization is computationally expensive and often does not allow for real-time implementation. Therefore, simplified models are required to reduce computation time. For example, the hierarchy control strategy presented in [9] and [21] both rely on offline trajectory planning since it is computationally not possible to rely on online trajectory planning. In [24] a linear parameter varying (LPV) model is introduced to implement online path planning, reducing the computation time by a factor of 50 compared to the nonlinear model. To implement the LPV, [24] assumed linear tire dynamics and excluded any torque vectoring, reducing the vehicle's overall performance.

LPV systems also can be used to reduce the computation time of the nonlinear reference tracking controller. For example, in [25], a similar control strategy as [9] is implemented, but using an LPV vehicle model. To discretize the system, [25] used an Euler approach, where [9] applied a fourth-order Runge-Kutta discretization to increase accuracy, resulting in longer computation times. Due to the LPV model, running the system with a sampling frequency of 33 Hz is possible while using Python 2.7 as the computer programming language. The nonlinear approach only allows for a sampling frequency of 20 Hz, while the algorithm is programmed in the more efficient computer language C++ [26].

All previous MPC formulations using the curvilinear approach use a hierarchical approach, where the trajectory is optimized separately from the tracking control due to the computational benefits this provides. However, due to the increasing computational power available nowadays, [27] introduced a single-layer control strategy to optimize path planning and tracking based on a curvilinear bicycle model. By using a quadratic cost and a centerline reference that is slightly out of reach, it is possible to maximize the progress on track resulting in a time-optimal racing line. Furthermore, the quadratic cost function makes it possible to solve the optimization problem with computationally efficient Gauss-Newton algorithms. Nevertheless, a bicycle model without slip is introduced, and no lateral motion is considered to make this possible. The controller in [27] is verified via an experiment with 1:43 scaled RC cars. This experiment is performed at low vehicle velocities, where the assumption of no occurring vehicle slip holds, but this is not the case when considering full-scale racing cars.

This single-layer control structure, where path planning and reference tracking is solved in one control problem, is also considered in [28]. The main challenge with online NMPC is the required horizon length to guarantee safe driving and stability of the system. This problem is solved by using a cascaded vehicle model, where the bicycle model's prediction horizon, including slip, is extended with a point mass that does not require a small step size to guarantee stability. The controller concept is proven to work on full-scale racing cars by performing tests with a Volkswagen golf racing car. However, offline velocity optimization is required to determine the safe set of velocities on track to guarantee safety. Secondly, the controller is developed for a fixed drive distribution between the front and rear axle, where the future challenges lay at all-wheel drive racing cars using torque vectoring to maximize its performance.

Model Predictive Contouring Control

Model Predictive Contouring Control (MPCC) is introduced for multi-axis contouring systems. The control object is to maximize the system's accuracy and minimize the traversal time while the system is subject to constraints. The trade-off between productivity and accuracy can be determined by adjusting the weights in the cost function [29].

In [30], the framework from [29] is adapted to obtain a high-performance MPCC for autonomous racing applications, which can maximize performance along the centerline within the prediction horizon. By considering the nonlinear projection of the vehicle's position onto the centerline and selecting low weights on the tracking error, the MPCC determines the optimal progress path. The controller can plan and follow the optimal trajectory, resulting in a similar path to a time-optimal path driven by an expert driver. The resulting controller is compared with a hierarchical NMPC by implementing both types of controllers on 1:43 scaled radio-controlled cars. The MPCC resulted in an additional computation time of around a factor of five compared with the hierarchical approach since both planning and following are computed in the same control layer. Since MPCC is a single-layer control strategy, it is more sensitive to external disturbances and infeasible trajectories, as hierarchical model predictive control provides more robustness due to a hierarchical multi-layered control strategy.

The lack of robustness in [30] was mainly caused by using an LPV system to linearise the system. In [31], the MPCC is adapted for it to be solved with nonlinear dynamics. Instead of maximizing the progress along the centerline, the controller's objective is to follow a reference path and velocity given to the MPCC. The adjustments of the cost functions allow for a quadratic programming problem, improving solving time and result in a real-time implementation with a sample rate of 20Hz, which is similar to [9] comparing computation time. The same algorithm is also used for a kinematic bicycle model in [32], which increases the complexity of the control model. The MPCC with a kinematic bicycle model still allows for a real-time implementation with a sample rate of 25Hz, but also has more computation power available to solve the optimization problem. Consequently, using kinematic models allows accuracy for low velocities and is, therefore, not suitable for racing applications. Furthermore, the transformation to a QP problem provides computational advantages but does not allow for online optimization of the reference trajectory.

Observations and Conclusions

In this chapter the state of the art motion planning systems are discussed, with the focus on using MPC, where the distinction is made between the contouring and the curvilinear formulation. Real-time application using a curvilinear system have been proven to work on an autonomous formula student racing car [9] [21], by using a hierarchical control strategy and relying on offline trajectory optimisation. Single-layer control strategies, such as the contouring formulation in [33] [32], but also the curvilinear approach in [27], have not yet been proven to work on a full scale autonomous racing car due to the computational challenges and the use of slip free models.

Most systems use a bicycle model to model the vehicle motion, due to a good balance between accuracy and computational efficiency. However, due to the electrification of the vehicles these days, especially when considering autonomous driving vehicles, new opportunities arise where a racing car is equipped with four in-wheel motors rather than a single motor providing power to one or two axles. Therefore, using a two track model, where every wheel is considered a separate input, can fully exploit the torque vectoring possibilities of all-wheel drive racing cars. Using a two track model in autonomous racing is already used in velocity control [34], but not yet used in order to follow a reference path on a racing track. As a result, state of the art motion planners do not fully utilize the potential of all wheel drive in autonomous applications.

Therefore, this project investigates online optimisation based motion planning systems for all-wheel drive autonomous racing vehicles. In the next section the research goal is introduced with the required sub-goals to achieve this.

1.3 Research Goal

Based on the state-of-the-art solutions presented in the previous chapter, the main goal of this research is defined as:

> **"Develop an online optimization-based motion planning system for all-wheel drive autonomous race cars."**

From this goal, several subgoals can be determined, outlying the requirements and challenges of an online motion planner for autonomous race cars. First of all, the considered vehicle is the autonomous racing car from formula student team URE. The race car has four electric in-wheel motors, capable of both braking and accelerating. Also, the car is front-wheel steered, excluding rear-wheel steering from the vehicle model. Therefore, the motion planning should provide the desired longitudinal force at each wheel and the desired steering rate.

Secondly, the motion planner is designed to minimize the lap time, using the centerline as a reference path while fully utilizing the available track width. A two-track model can be used to determine the optimal control input over the prediction horizon, providing a safe but fast trajectory over the track. Including tire dynamics is essential when driving at the limits of the car, which requires a vehicle model including slip. A single-layer controller is considered, solving the optimization problem to maximize the performance on track by performing both path planning and reference tracking along the centerline. Since the formula student competition does not use obstacles, nor does it allow for head-to-head racing, obstacle avoidance is not inside the scope of this thesis.

Finally, the controller is designed to operate online and therefore the localization of the vehicle with respect to the centerline should be considered. Computing the centerline and track width is done via the simultaneous localization and mapping (SLAM) system of URE and is assumed to be known during the development of the controller. The control input from the localization system should be smooth to enhance stability for various look-ahead distances and ensure continuous inputs for both control and drivetrain durability. Furthermore, to account for the variety in race cars, the controller should be adjustable, and preferably tunable, for different vehicle configurations.

This chapter introduced the research goal, which specified the requirements and preferences for developing an online motion planner for autonomous racing applications. In short, this thesis aims to extend and improve existing motion planning systems by exploring optimization-based motion planning for all-wheel drive autonomous racing vehicles. It should consider different track layouts to guarantee robustness, and satisfying dynamic and actuator constraints.

1.4 Thesis Outline

Based on the literature review, this thesis proposes an online optimization based motion planning algorithm for all-wheel drive autonomous racing cars. First, in Chapter 2 the preliminaries are introduced which are used as mathematical foundation for some theoretical results. The chapter also contains all the vehicle models used throughout the thesis and different discretization methods. In Chapter 3 the localization algorithm is introduced, where an Extended Kalman Filter is developed to mitigate any external disturbances. Consequently, in Chapter 4 the optimization problem to solve the minimum time problem is proposed, which is then used in Chapter 5 to analyse the performance for different prediction horizon lengths. At last, the conclusions of this thesis are drawn in Chapter 6, where additionally some recommendations for future work are given.

Chapter 2

Preliminaries

In this chapter, the theory and vehicle models are introduced which are used in this thesis.

2.1 Theories

Let us consider the smooth affine control system with an output map

$$\begin{aligned} \dot{x} &= f(x) + \sum_{j=1}^{m} g_j(x) u_j, \quad u = (u_1, ..., u_m) \in U \subset \mathbb{R}^m, \\ y_i &= h_i(x), \quad i \in \{\underline{p}, \underline{p} \in \mathbb{N}\}, \end{aligned} \tag{2.1}$$

where $x = (x_1, ..., x_n)$ represent the coordinates for a smooth state-space manifold M, and $h = (h_1, ..., h_p)^T : M \to Y = \mathbb{R}^p$ is the smooth output map of the system.

Definition 2.1.1 Lie Derivative. The first order Lie derivative of output $h(x)$ along vector field $f(x)$ is given by

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) \ \in \mathbb{R}^p, \tag{2.2}$$

where $h(x) \in \mathbb{R}^p$ and $x \in \mathbb{R}^n$. The nth order Lie derivative is subsequently

$$L_f^n h(x) = \frac{\partial L_f^{n-1} h(x)}{\partial x} f(x) \ \in \mathbb{R}^p, \tag{2.3}$$

with the zero order Lie derivative $L_f^0 h(x) = h(x)$.

Definition 2.1.2. Observation space [35]. *Consider the system* (2.1). *The observation space* Θ *of* (2.1) *is the linear space (over* $\mathbb{R}$*) of functions on* M *containing* $h_1, ..., h_p$ *and all repeated Lie derivatives.*

Theorem 2.1.3. Observability rank condition [35]. *Consider the system* (2.1) *with* $dim M = n$. *Assume that*

$$\dim \, \mathrm{d}\Theta(x_0) = n, \tag{2.4}$$

then the system is locally observable at x_0.

2.2 Dynamics

This section introduces all the vehicle models that are used in this thesis. To distinguish the different vehicle models, a different notation is applied for each model. When the point-mass is used, all the associated states and inputs are denoted with $\bar{(\cdot)}$, the states and inputs from the single track model are denoted with $\tilde{(\cdot)}$, and the two-track model uses the plain notation.

Point Mass Model

A planar point-mass model is schematically represented in Figure 2.1. The point-mass model is used as a simple representation of the Center of Gravity (CG), avoiding stiff dynamics by considering only one velocity state and the forces acting at the CG as inputs of the system. In this project, a larger discretization step can be applied due to the reduced model stiffness compared with a single or two-track model.

The model consists of one velocity state, the total horizontal velocity $\bar{V} \in \mathbb{R}$, given by [28]

$$\dot{\bar{V}} = \frac{\bar{F}_x - \bar{F}_{\text{loss}}}{m}, \tag{2.5}$$

with m the vehicle mass, $\bar{F}_x$ the longitudinal force input, and $\bar{F}_{loss}$ the combined drag and rolling resistance force.

Furthermore, the point-mass model considers three position states in the curvilinear reference frame $\bar{T}$, describing its position concerning the reference path. Reference frame $\bar{T}$ is obtained by rotating Cartesian frame R about the angle $\theta(\bar{s})$, as is depicted in Figure 2.1. This frame $\bar{T}$ describes the position of the vehicle about the centerline, where the progress made along the reference path is denoted by $\bar{s} \in \mathbb{R}$, the lateral distance to the path by $\bar{e}_y \in \mathbb{R}$, and the difference in the direction of the velocity vector and the reference frame $\bar{T}$ by $\bar{e}_\psi \in \mathbb{R}$. The motion of the three position states can be described by [28]

$$\dot{\bar{s}} = \frac{\bar{V}\cos(\bar{e}_\psi)}{1-\kappa(\bar{s})\bar{e}_y}, \tag{2.6a}$$

$$\dot{\bar{e}}_y = \bar{V}\sin(\bar{e}_\psi), \tag{2.6b}$$

$$\dot{\bar{e}}_\psi = \frac{\bar{F}_y}{m\bar{V}} - \kappa(\bar{s})\frac{\bar{V}\cos(\bar{e}_\psi)}{1-\kappa(\bar{s})\bar{e}_y}, \tag{2.6c}$$

where $\kappa(\bar{s})$ denotes the curvature of the reference path and $\bar{F}_y$ the lateral force input.

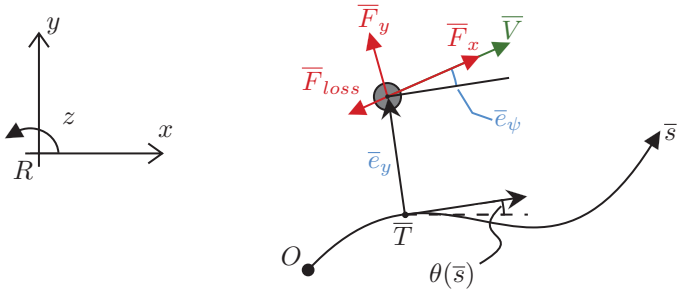

Figure 2.1: Planar point-mass model.

Single Track model

The single-track model represents a rigid two-axle vehicle body model, including longitudinal, lateral, and yaw motion. Such a vehicle model results in a physically plausible description of the

driving behavior.

The model has three velocity states associated with its CG: longitudinal and lateral velocity $\tilde{v}_x, \tilde{v}_y \in \mathbb{R}$, respectively, and yaw velocity $\tilde{\omega} \in \mathbb{R}$. Similarly to the point-mass model is the position of the CG described in the curvilinear reference frame. However, instead of defining heading error as the error between the velocity vector and frame $\tilde{T}$, the heading error represents the difference in the chassis heading with respect to the heading of $\tilde{T}$. The resulting vehicle model is displayed in Figure 2.2, where the motion of the vehicle is described as

$$\dot{\tilde{v}}_x = \frac{1}{m}(\tilde{F}_x - \tilde{F}_{y,f}\sin(\tilde{\delta}) - \tilde{F}_D - \tilde{F}_R + m\tilde{v}_y\tilde{\omega}), \tag{2.7a}$$

$$\dot{\tilde{v}}_y = \frac{1}{m}(\tilde{F}_{y,f}\cos(\tilde{\delta}) + \tilde{F}_{y,r} - m\tilde{v}_x\tilde{\omega}), \tag{2.7b}$$

$$\dot{\tilde{\omega}} = \frac{1}{I_{zz}}(l_f\tilde{F}_{y,f}\cos(\tilde{\delta}) - l_r\tilde{F}_{y,r}), \tag{2.7c}$$

$$\dot{\tilde{s}} = \frac{\tilde{v}_x\cos(\tilde{e}_\psi) - \tilde{v}_y\sin(\tilde{e}_\psi)}{1-\kappa(\tilde{s})\tilde{e}_y}, \tag{2.7d}$$

$$\dot{\tilde{e}}_y = \tilde{v}_x\sin(\tilde{e}_\psi) + \tilde{v}_y\cos(\tilde{e}_\psi), \tag{2.7e}$$

$$\dot{\tilde{e}}_\psi = \tilde{\omega} - \kappa(\tilde{s})\frac{\tilde{v}_x\cos(\tilde{e}_\psi) - \tilde{v}_y\sin(\tilde{e}_\psi)}{1-\kappa(\tilde{s})\tilde{e}_y}, \tag{2.7f}$$

$$\dot{\tilde{\delta}} = \dot{\tilde{\delta}}_{in}, \tag{2.7g}$$

where m and I_{zz} represent the mass and moment of inertia of the vehicle, respectively, $\tilde{F}_x$ is the longitudinal force input from the motors, $\dot{\tilde{\delta}}_{in}$ is the steering rate input from the steering actuator, $\tilde{F}_D$ and $\tilde{F}_R$ are the drag and rolling resistance force losses, respectively, and $\kappa(\tilde{s})$ represents the curvature of the path at $\tilde{s}$. The distance from the CG to the front and rear axle are denoted by l_f and l_r, respectively, summing up to the wheelbase L.

The lateral tire forces $\tilde{F}_{y,l}$ can be approximated via a simplified Pacejka tire model [23]:

$$\tilde{F}_{y,l} = \tilde{D}_{y,l}\sin(C_{y,l}\arctan(B_{y,l}\tilde{\alpha}_l)), \tag{2.8}$$

where $l \in \{f, r\}$ representing the front and rear axle, respectively. The stiffness factor $B_{y,l}$, the shape factor $C_{y,l}$ and the peak factor $\tilde{D}_{y,l}$ are all lateral tire parameters, which are determined via experiments.

The tire side slip angles are defined as

$$\tilde{\alpha}_f = \tilde{\delta} - \arctan\left(\frac{\tilde{v}_y + l_f\tilde{\omega}}{\tilde{v}_x}\right), \tag{2.9a}$$

$$\tilde{\alpha}_r = -\arctan\left(\frac{\tilde{v}_y - l_r\tilde{\omega}}{\tilde{v}_x}\right), \tag{2.9b}$$

and the force losses as

$$\tilde{F}_D = \frac{1}{2}\rho_{air}\tilde{v}_x^2 C_D A_s, \tag{2.10a}$$

$$\tilde{F}_R = f_r F_z, \tag{2.10b}$$

where ρ_{air} is the density of air, C_d is the drag coefficient, A_s the frontal surface of the vehicle, f_r the rolling friction and F_z the total vertical load of the car.

Two Track Model

The model has three velocity states associated with its CG: longitudinal and lateral velocity $\tilde{v}_x, \tilde{v}_y \in \mathbb{R}$, respectively, and yaw velocity $\tilde{\omega} \in \mathbb{R}$. In addition, the position of the vehicle is defined

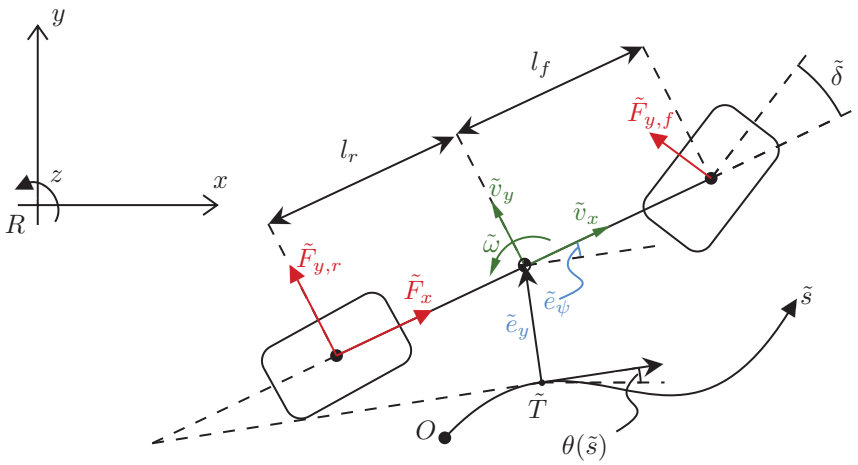

Figure 2.2: Single-track vehicle model.

likewise as the single track model, which is displayed in Figure 2.3, also yielding the same dynamics as (2.7d) - (2.7f).

The resulting vehicle model is displayed in Figure 2.4, where the motion of the vehicle is described as

$$\begin{aligned}
m\dot{v}_x &= F_{x,1}\cos(\delta_1) - F_{y,1}\sin(\delta_1) + F_{x,4}\cos(\delta_2) - F_{y,4}\sin(\delta_2) + F_{x,2} + F_{x,3} + ... \\
&\quad mv_y\omega - F_{loss}, && \text{(2.11a)}\\
m\dot{v}_y &= F_{x,1}\sin(\delta_1) + F_{y,1}\cos(\delta_1) + F_{x,4}\sin(\delta_2) + F_{y,4}\cos(\delta_2) + F_{y,2} + F_{y,3} - mv_x\omega, && \text{(2.11b)}\\
I_{zz}\dot{\omega} &= (F_{y,1}\cos(\delta_1) + F_{x,1}\sin(\delta_1) + F_{y,4}\cos(\delta_2) + F_{x,4}\sin(\delta_2))l_f ... \\
&\quad \frac{w_f}{2}(F_{y,1}\sin(\delta_1) - F_{x,1}\cos(\delta_1) - F_{y,4}\sin(\delta_2) + F_{x,4}\cos(\delta_2)) ... \\
&\quad -(F_{y,2} + F_{y,3})l_r + \frac{w_r}{2}(F_{x,3} - F_{x,2}), && \text{(2.11c)}\\
\dot{s} &= \frac{v_x\cos(e_\psi) - v_y\sin(e_\psi)}{1 - e_y\kappa(s)}, && \text{(2.11d)}\\
\dot{e}_y &= v_x\sin(e_\psi) + v_y\cos(e_\psi), && \text{(2.11e)}\\
\dot{e}_\psi &= r - \kappa(s)\frac{v_x\cos(e_\psi) - v_y\sin(e_\psi)}{1 - e_y\kappa(s)} && \text{(2.11f)}\\
\dot{\delta} &= \dot{\delta}_{in}. && \text{(2.11g)}
\end{aligned}$$

where the longitudinal tire forces $F_{x,l}, l \in \{1,2,3,4\}$ are considered inputs, as well as the front steering rate $\dot{\delta}_i n$. Notice that a different steering angle for the left δ_1 and right δ_2 wheel is considered, to account for toe, where $\delta_1 = \delta - \delta_{toe}$ and $\delta_2 = \delta + \delta_{toe}$. Furthermore, m and I_{zz} represent the vehicle's mass and moment of inertia, w_f and w_r denote the track width at the front and rear, respectively.

Similarly to the single track model, the lateral tire forces are modeled via the simplified Pacejka tire model as given in (2.8). However, each tire has a different side slip angle, depending on the longitudinal and lateral wheel velocity $v_{x,l}, v_{y,l}$, respectively. These wheel velocities are calculated

via

$$v_{x,1} = v_x - \frac{w_f}{2}\omega, \quad v_{y,1} = v_y + l_f\omega, \tag{2.12a}$$

$$v_{x,2} = v_x - \frac{w_r}{2}\omega, \quad v_{y,2} = v_y - l_r\omega, \tag{2.12b}$$

$$v_{x,3} = v_x + \frac{w_r}{2}\omega, \quad v_{y,3} = v_y - l_r\omega, \tag{2.12c}$$

$$v_{x,4} = v_x + \frac{w_f}{2}\omega, \quad v_{y,4} = v_y + l_f\omega. \tag{2.12d}$$

The wheel velocities can then be used to determine the resulting tire slip angles via

$$\alpha_1 = \delta_1 - \arctan\left(\frac{v_{y,1}}{v_{x,1}}\right), \tag{2.13a}$$

$$\alpha_2 = -\arctan\left(\frac{v_{y,2}}{v_{x,2}}\right), \tag{2.13b}$$

$$\alpha_3 = -\arctan\left(\frac{v_{y,3}}{v_{x,3}}\right), \tag{2.13c}$$

$$\alpha_4 = \delta_2 - \arctan\left(\frac{v_{y,4}}{v_{x,4}}\right). \tag{2.13d}$$

The force losses F_D and F_R represent the drag and rolling resistance force losses, respectively, similarly defined as (2.10).

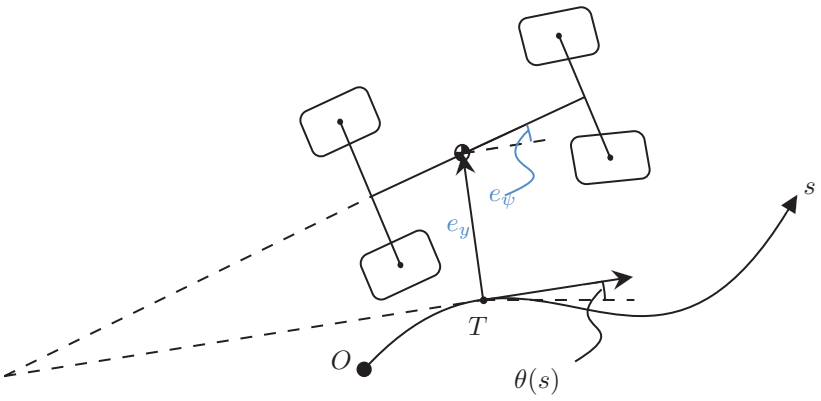

Figure 2.3: The position states of the two-track model.

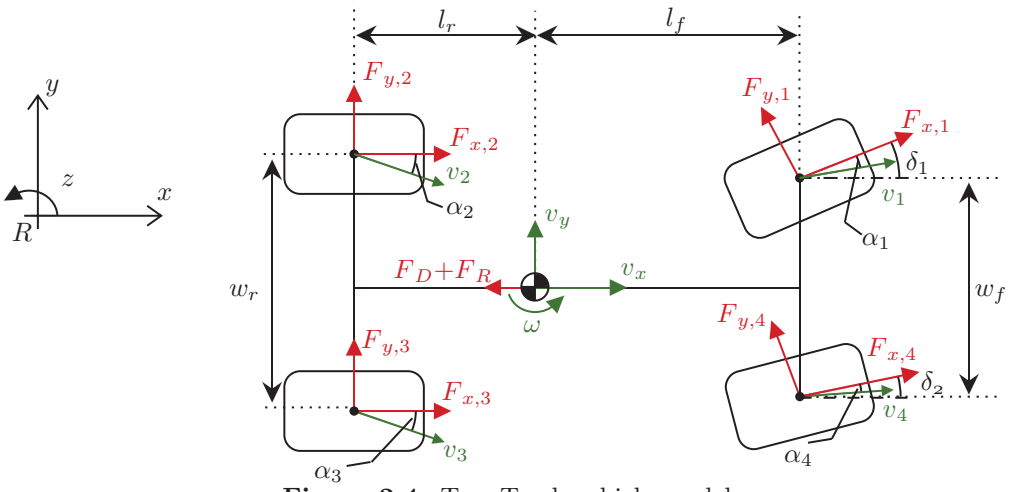

Figure 2.4: Two-Track vehicle model.

2.3 Discretization Methods

In this thesis, different discretization methods are applied. In this section, all used methods are briefly introduced.

Forward Euler

In many cases, the integration of a dynamical system must be performed numerically. A system simulator can be constructed by breaking time t into smaller intervals h and computing numerical solutions to differential equations. The Euler discretization is one method for computing such a numerical system simulator. Consider a general system formulation

$$\dot{x} = f(x(t_k), u(t_k)), \tag{2.14a}$$
$$x(t_k) = x_k. \tag{2.14b}$$

Suppose that $x(t_k)$ and $u(t_k)$ are known. By performing integration over time step h, the state $x(t_k + h)$ can be determined as [36]:

$$x(t_k + h) = x(t_k) + \int_{t_k}^{t_k+h} f(x(t_k), u(t_k))dt. \tag{2.15}$$

The issue arises that the integral cannot be evaluated directly due to $x(t_k)$ being in the integrand. Using the fact that the integrand can be approximated as

$$f(x(t_k), u(t_k)) \approx \frac{x(t_k + h) - x(t_k)}{h}, \tag{2.16}$$

solving (2.15) for $x(t_k + h)$ and the given approximation of $f(x(t_k), u(t_k))$ provides the Forward Euler discretization method [36]:

$$x(t_k + h) \approx x(t_k) + hk_1, \tag{2.17}$$

where

$$k_1 = f(x(t_k), u(t_k)). \tag{2.18}$$

In Figure 2.5, the forward Euler discretization is graphically displayed.

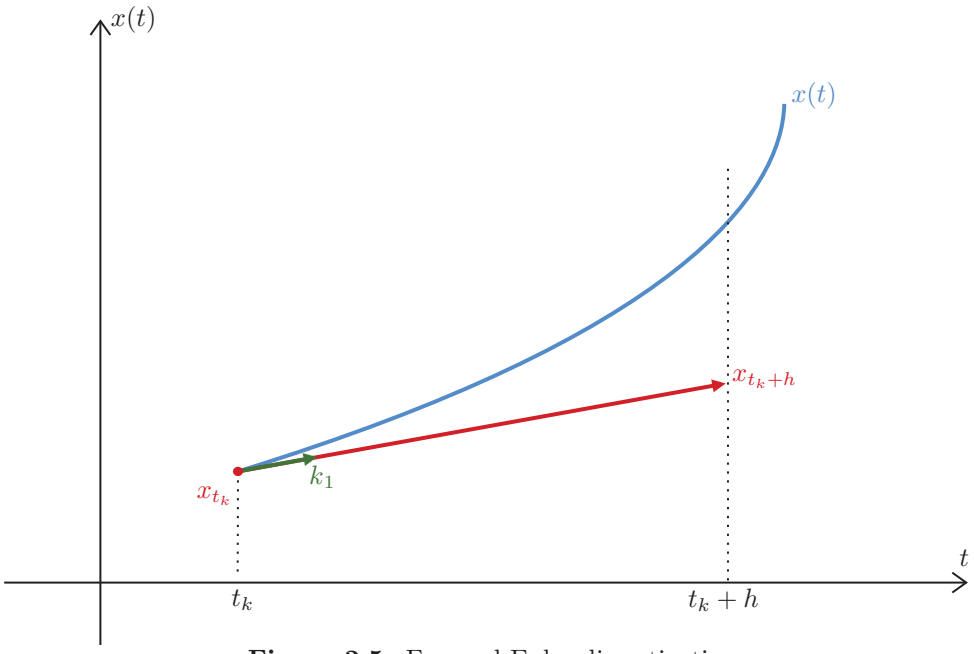

Figure 2.5: Forward Euler discretization.

Second Order Runge-Kutta

The general formulation of a Runge-Kutta discretization method of order s is written as

$$x(t_k + h) = x(t_k) + h \sum_{i=1}^{s} a_i k_i + O(h^{s+1}), \tag{2.19}$$

where

$$k_i = x(t_k) + h \sum_{j=1}^{s} \beta_{ij} f\left(x(t_k) + \alpha_i h k_j, u(t_k + \alpha_i h)\right), \tag{2.20}$$

are the slopes which are obtained evaluating the derivatives of $x(t_k)$ at the i-th order. For the second order Runge-Kutta (RK2) discretization $s = 2$, which provides that

$$x(t_k + h) = x(t_k) + h x'(t_k) + \frac{h^2}{2} x''(t) + O(h^3). \tag{2.21}$$

One can observe that the forward Euler discretization is simply the first-order term of the Runge-Kutta discretization. The numerical method can be obtained via Taylor expansion, resulting in

$$x(t_k + h) = x(t_k) + h \left(\frac{1}{2} k_1 + \frac{1}{2} k_2\right), \tag{2.22}$$

where

$$k_1 = f(x(t_k), u(t_k)), \tag{2.23}$$
$$k_2 = f(x(t_k) + h k_1, u(t_k)). \tag{2.24}$$

RK2 is also known as Heun's method or the improved Euler method [37]. The output at $t_{k+1} = t_k + h$ is approximated by determining the slope at two different times: t_k and t_{k+1}, which can be observed in Figure 2.6.

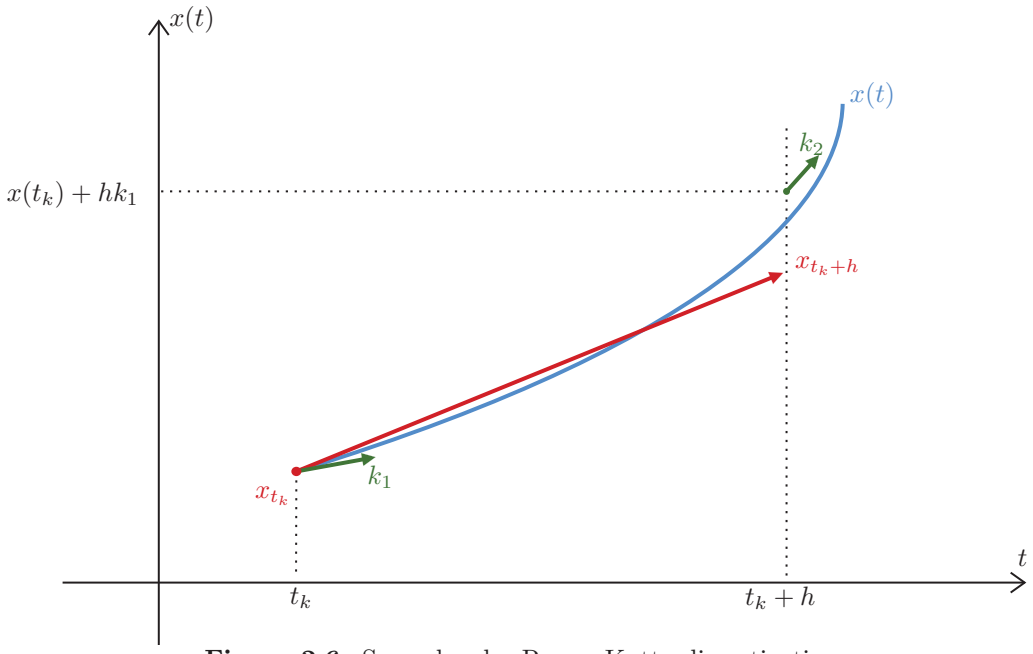

Figure 2.6: Second-order Runge-Kutta discretization.

Fourth Order Runge-Kutta

The fourth order Runge-Kutta (RK4) method can be obtained by evaluating (2.19) and (2.20) for $s = 4$, resulting in

$$x(t_k + h) = x(t_k) + hx'(t_k) + \frac{h^2}{2!}x''(t_k) + \frac{h^3}{3!}x^{(3)}(t_k) + \frac{h^4}{4!}x^{(4)}(t_k) + O(h^5). \tag{2.25}$$

Evaluating this with the Taylor series of $x(t_k + h)$ around t_k results in the numerical method [38]

$$x(t_k + h) \approx x(t_k) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{2.26}$$

in which

$$\begin{aligned} k_1 &= f(x(t_k), u(t_k)), \\ k_2 &= f(x(t_k) + \frac{h}{2}k_1, u(t_k)), \\ k_3 &= f(x(t_k) + \frac{h}{2}k_2, u(t_k)), \\ k_4 &= f(x(t_k) + hk_3, u(t_k)). \end{aligned} \tag{2.27}$$

The output at $t_{k+1} = t_k + h$ is approximated by determining the slope at three different times: t_k, $t_k + \frac{h}{2}$ and t_{k+1}, which can be observed inFigure 2.7.

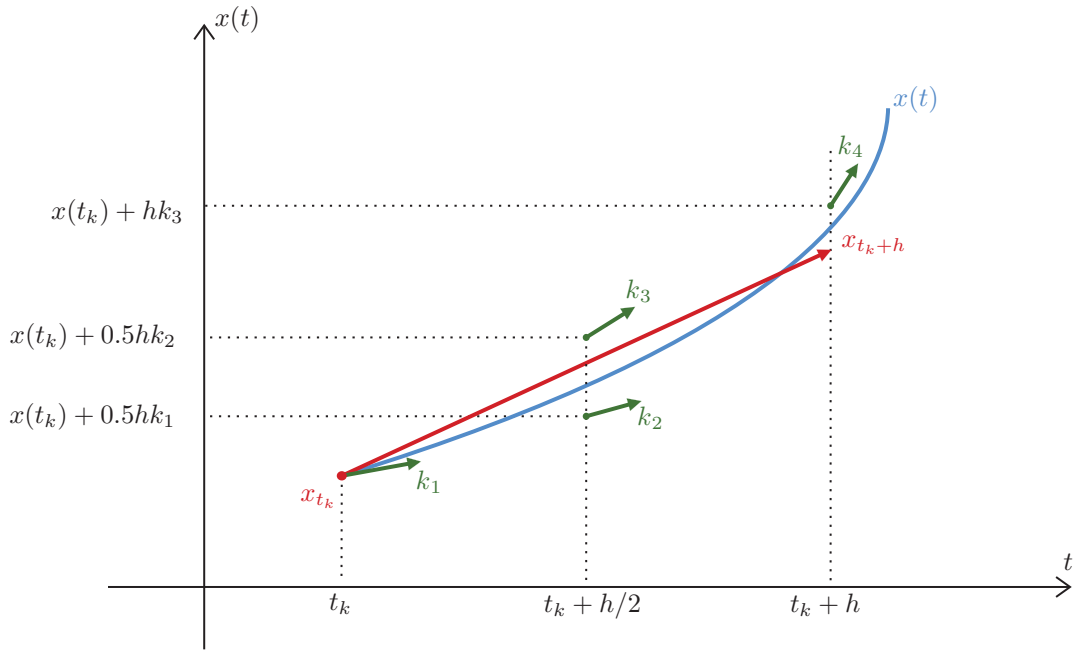

Figure 2.7: Fourth order Runge-Kutta discretization.

2.4 Observations and Conclusion

In this chapter, the required theorems used in this thesis have been introduced. Additionally, different representations of the dynamic behavior of an autonomous racing car have been presented, which are the basis for the optimization problem. The various vehicle models are transformed into the Frenet coordinate system. The single-track model is used in Chapter 3 to solve the localization problem. In Chapter 4, the different vehicle models are used to solve the path planning and reference tracking problem. Notice that all vehicle models are denoted using other notations to distinguish between the states, inputs, and parameters. At last, three different discretization methods are introduced, which are fundamental in the implementation of the Extended Kalman Filter in section 3.2 and the discretization of the prediction horizon in Chapter 4.

Chapter 3

Localization

One challenge in online implementation is determining the vehicle's position with respect to the reference path. Instead of relying on offline path planning, this thesis considers both path planning and reference tracking online. Moreover, A Simultaneous Localization and Mapping (SLAM) logarithm provides the reference points and the car's location, which combined provides the footpoint of the vehicle. This chapter discusses the calculation of the footpoint and the implementation of an Extended Kalman Filter (EKF), dealing with the non-smoothness, which is a result of discontinuous look-ahead distance and interpolation of the reference points, both side-effects of online path finding. Smoothing the coordinates in the Frenet coordinate frame directly smoothness the input of the NMPC, potentially avoiding infeasibility or undesired behavior in the NMPC output.

3.1 Footpoint

The path and vehicle pose are updated as the car drives on the track. Since motion planning is working with the reference frame T, and the path is defined in the Cartesian frame R, the location of frame T has to be determined, which is called the footpoint. This point is determined via linear interpolation between the reference point in front of the car $(x_{r,f}, y_{r,f})$ and the reference point behind the car $(x_{r,r}, y_{r,r})$, which is graphically displayed with the green dot in Figure 3.1, where p_f and p_r denote the distance from the CG to the front and rear reference point, respectively.

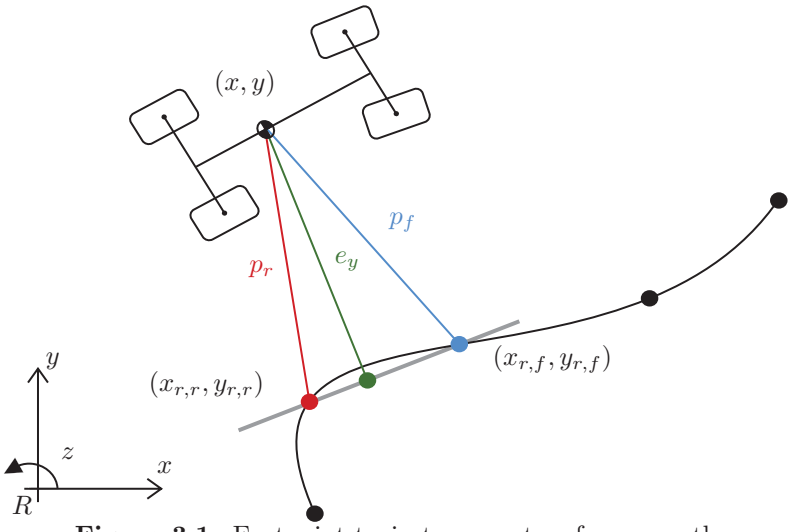

Figure 3.1: Footpoint trajectory w.r.t. reference path.

The footpoint assumes a straight line between the reference points and tries to find the footpoint, which is the projection of the altitude of the triangle on the track using simple geometry. Using the semi-perimeter of the triangle, calculated via

$$S = \frac{P_{r,f} + p_f + p_r}{2}, \tag{3.1}$$

the altitude of the triangle can be determined. The altitude of the triangle represents the error between the CG and the Frenet coordinate frame T, hence e_y, and can be calculated via

$$e_y = 2\frac{\sqrt{S(S - P_{r,f})(S - p_f)(S - p_r)}}{P_{r,f}}, \tag{3.2}$$

where $P_{r,f}$ represents the distance between the front and rear waypoint, calculated via

$$P_{r,f} = ||(x_{r,r}, y_{r,r}) - (x_{r,f}, y_{r,f})||_2. \tag{3.3}$$

At last, the heading of the path θ can be determined based on the two reference points via

$$\theta = \arctan\left(\frac{x_{r,f} - x_{r,r}}{y_{r,f} - y_{r,r}}\right), \tag{3.4}$$

from which the heading error e_ψ can be determined via

$$e_\psi = \psi - \theta. \tag{3.5}$$

The implementation of this localization strategy comes with two challenges. First, since the footpoint is determined via interpolating two reference points, the output can shift when switching to two new reference points, which can be observed when analyzing Figure 3.1. Also, as the car is driving, the reference points are updated, which can result in different path trajectories between time samples, as seen in Figure 3.2. For example, the figure shows three consecutive time samples, where SLAM shifts the reference points from the black path towards the blue path. As a result, the footpoint is shifting from a rather large positive error towards a small negative error.

Both these challenges cause the input for motion planning to be non-smooth, resulting in longer computation times since the new vehicle position moved away from the previous optimal solutions, which can even result in an infeasible control problem. Therefore, in the following section, the footpoint is observed with an Extended Kalman Filter, providing a more smooth estimation.

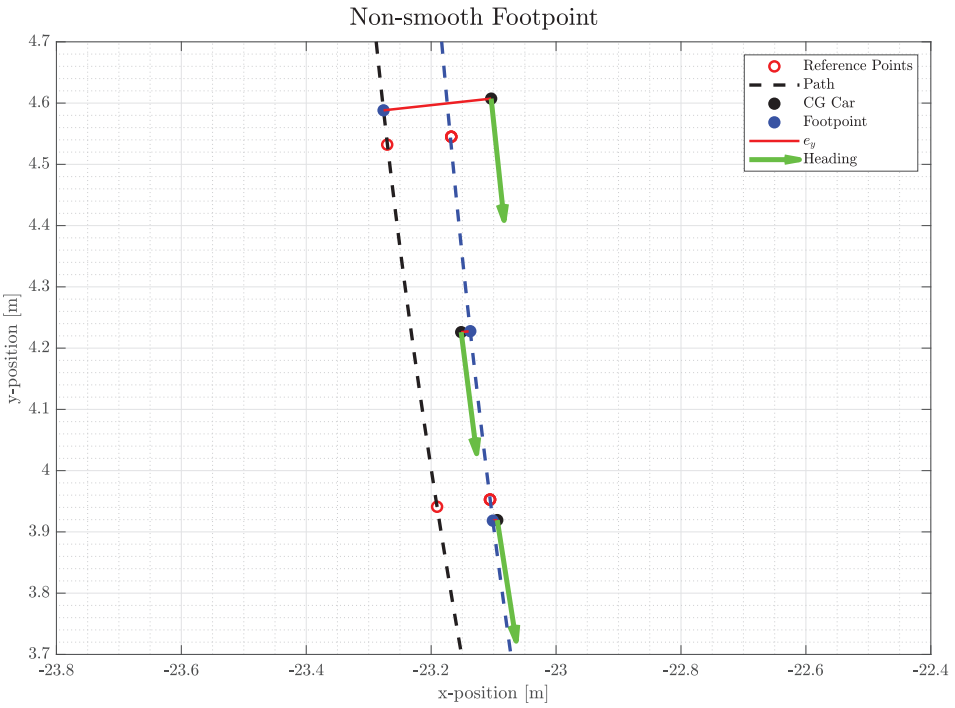

Figure 3.2: SLAM provides two different paths between consecutive sample timings. Black path for $k = 1$, blue path for $k \in 2, 3$. Footpoint is shifting accordingly.

3.2 Extended Kalman Filter

The goal of the EKF is to observe the vehicle states described by the nonlinear single track model in section 2.2, with the main focus on estimating the error states $\tilde{e}_y$ and $\tilde{e}_\psi$. Observing these states result in a smoother controller input, excluding any external disturbances that might occur in the calculation of the footpoint its location.

Prediction Step

At first, an approximation of the state $\hat{x}_{k|k-1}$ and covariance $P_{k|k-1}$ is made based on the previous estimation via

$$\hat{x}_{k|k-1} = f_{pred}(\hat{x}_{k-1|k-1}, u_k), \tag{3.6a}$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^\intercal + Q_k, \tag{3.6b}$$

where the notation $\hat{x}_{n|m}$ and $P_{n|m}$ represents the estimate of x and P at time n given the estimations up to and including at time $m \leq n$. The diagonal matrix Q_k represents the process noise covariance matrix, and F_k is the state transition matrix. The function $f_{pred}(\hat{x}_{k-1|k-1}, u_k)$ represents the model that is used for the prediction of the states, but in discrete time. Therefore, the single-track model is discretized with a fourth-order Runge-Kutta discretization, for which the reader is referred to section 2.3 regarding the details of the implementation.

The single track model in section 2.2 also includes the progress that is made along the reference path, indicated by s. However, the reference points provided by SLAM are a fixed number of points behind and in front of the car. As a result, s is increasing and decreasing, depending on the position of reference points combined with the vehicle pose. For that reason, observing s is not possible as there is no dynamic model which can predict such behavior. Therefore, due to s being redundant, a model reduction is applied to the six states $[\tilde{v}_x, \tilde{v}_y, \tilde{\omega}, \tilde{e}_y, \tilde{e}_\psi, \tilde{\delta}]$. The location of the footpoint directly provides the position of the vehicle w.r.t. the reference path using s.

The state transition matrix F_k is obtained via

$$F_k = \left. \frac{\partial f_{F_k}}{\partial x} \right|_{\hat{x}_{k-1|k-1}, u_k}, \tag{3.7}$$

where $f_{F_k}(\hat{x}_{k-1|k-1}, u_k)$ is the discretized single track model. However, due the non-linear nature of the model two assumptions are made to simplify $f_{F_k}(\hat{x}_{k-1|k-1}, u_k)$. The first assumption regards a small angle approximation. The second assumption is that the tire dynamics are linear, and therefore can be described as

$$F_{y,f} = C_f \alpha_f, \tag{3.8a}$$
$$F_{y,r} = C_r \alpha_r, \tag{3.8b}$$

where C_f and C_r are the cornering stiffness of the front and rear axle, respectively, and

$$\alpha_f = \tilde{\delta} - \frac{\tilde{v}_y + l_f \tilde{\omega}}{\tilde{v}_x}, \tag{3.9a}$$
$$\alpha_r = -\frac{\tilde{v}_y - l_r \tilde{\omega}}{\tilde{v}_x}. \tag{3.9b}$$

Based on these assumptions, the vehicle model described in (2.7) can be reformulated as

$$\dot{\tilde{v}}_x = \frac{1}{m}(\tilde{F}_x - C_f \left(\tilde{\delta} - \frac{\tilde{v}_y + l_f \tilde{\omega}}{\tilde{v}_x} \right) \tilde{\delta} + m \tilde{v}_y \tilde{\omega}), \tag{3.10a}$$
$$\dot{\tilde{v}}_y = \frac{1}{m}(C_f \left(\tilde{\delta} - \frac{\tilde{v}_y + l_f \tilde{\omega}}{\tilde{v}_x} \right) - C_r \left(\frac{\tilde{v}_y - l_r \tilde{\omega}}{\tilde{v}_x} \right) - m \tilde{v}_x \tilde{\omega}), \tag{3.10b}$$
$$\dot{\tilde{\omega}} = \frac{1}{J}(l_f C_f \left(\tilde{\delta} - \frac{\tilde{v}_y + l_f \tilde{\omega}}{\tilde{v}_x} \right) - l_r C_r \left(\frac{\tilde{v}_y - l_r \tilde{\omega}}{\tilde{v}_x} \right)), \tag{3.10c}$$
$$\dot{\tilde{s}} = \frac{\tilde{v}_x - \tilde{v}_y(\tilde{e}_\psi)}{1 - \kappa(\tilde{s}) \tilde{e}_y}, \tag{3.10d}$$
$$\dot{\tilde{e}}_y = \tilde{v}_x \tilde{e}_\psi + \tilde{v}_y, \tag{3.10e}$$
$$\dot{\tilde{e}}_\psi = \tilde{\omega} - \kappa(\tilde{s}) \frac{\tilde{v}_x - \tilde{v}_y \tilde{e}_\psi}{1 - \tilde{e}_y \kappa(\tilde{s})}, \tag{3.10f}$$
$$\dot{\tilde{\delta}} = \dot{\tilde{\delta}}_{in}. \tag{3.10g}$$

The function $f_{F_k}(\hat{x}_{k-1|k-1}, u_k)$ is obtained via a Forward Euler discretization of the simplified vehicle model in (3.10). For the implementation of the forward Euler discretization, the reader is referred to section 2.3. Based on the discretized model, the state transition matrix can be derived via (3.7), resulting in

$$F_k = \begin{bmatrix} F_{k,1.1} & h\,\tilde{\omega} + \frac{C_f\,\tilde{\delta}\,h}{m\,\tilde{v}_x} & h\,\tilde{v}_y - \frac{C_f\,l_f\,\tilde{\delta}\,h}{m\,\tilde{v}_x} & 0 & 0 & F_{k,1.6} \\ F_{k,2.1} & 1 - \frac{h\,(C_f + C_r)}{m\,\tilde{v}_x} & \frac{h\,(C_r\,l_r - C_f\,l_f)}{m\,\tilde{v}_x} - h\,\tilde{v}_x & 0 & 0 & \frac{C_f\,h}{m} \\ F_{k,3.1} & -\frac{h\,(C_f\,l_f - C_r\,l_r)}{J\,\tilde{v}_x} & 1 - h\frac{C_f\,l_f^2 - C_r l_r^2}{J\,\tilde{v}_x} & 0 & 0 & \frac{C_f\,l_f\,h}{J} \\ \tilde{e}_\psi\,h & h & 0 & 1 & h\,\tilde{v}_x & 0 \\ \frac{h\,\kappa}{\tilde{e}_y\,\kappa - 1} & -\frac{\tilde{e}_\psi\,h\,\kappa}{\tilde{e}_y\,\kappa - 1} & h & F_{k,5.4} & F_{k,5.5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.11}$$

where

$$F_{k,1.1} = 1 - h\frac{C_f\,\tilde{\delta}\,(\tilde{v}_y + l_f\,\tilde{\omega})}{m\,\tilde{v}_x^2}, \tag{3.12a}$$

$$F_{k,1.6} = -h\frac{C_f\,\left(2\,\tilde{\delta}\,\tilde{v}_x - l_f\,\tilde{\omega} - \tilde{v}_y\right)}{m\,\tilde{v}_x}, \tag{3.12b}$$

$$F_{k,2.1} = h\frac{C_f\,(\tilde{v}_y + l_f\,\tilde{\omega}) + C_r\,(\tilde{v}_y - l_r\,\tilde{\omega})}{m\tilde{v}_x^2} - h\tilde{\omega}, \tag{3.12c}$$

$$F_{k,3.1} = \frac{h}{J}\left(\frac{C_f l_f\,(\tilde{v}_y + l_f\,\tilde{\omega}) - C_r\,l_r\,(\tilde{v}_y - l_r\,\tilde{\omega})}{\tilde{v}_x^2}\right), \tag{3.12d}$$

$$F_{k,5.4} = -\frac{h\,\kappa^2\,(\tilde{v}_x - \tilde{e}_\psi\,\tilde{v}_y)}{(\tilde{e}_y\,\kappa - 1)^2}, \tag{3.12e}$$

$$F_{k,5.5} = 1 - \frac{h\,\kappa\,\tilde{v}_y}{\tilde{e}_y\,\kappa - 1}. \tag{3.12f}$$

Update Step

The state and covariance estimation from (3.6) are updated based on the available sensor data at time k, providing a more accurate state estimate via

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_k, \tag{3.13}$$

where e_k represents the error between the measured output y_k and the predicated state estimate, K_k is the Kalman gain, calculated via

$$e_k = y_k - H_k\hat{x}_{k|k-1}, \tag{3.14a}$$

$$K_k = P_{k|k-1}H_k^T(H_k P_{k|k-1}H_k^T + R_k)^{-1}, \tag{3.14b}$$

where R_k represents the observation noise covariance, and H_k is the observation matrix. To determine H_k, an output mapping $h(x)$ is defined based on the available sensor data at URE.

The footpoint calculation in section 3.1 provides the measurement model for both e_y and e_ψ. Additionally, the wheel-speed encoders provide the longitudinal velocity, the yaw velocity is estimated based on the inertial measurement unit, and the steering angle is measured via the steering rack sensor. This provides the output mapping to be

$$h(x) = [v_x, \omega, e_y, e_\psi, \delta]. \tag{3.15}$$

From this output mapping, the observation matrix can be determined

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}, u_k}, \tag{3.16}$$

resulting in

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.17}$$

Also the covariance matrix $P_{k|k-1}$ has to be updated to obtain $P_{k|k}$, which is calculated via

$$P_{k|k} = (I - K_k\,H_k)\,P_{k|k-1}. \tag{3.18}$$

Once the covariance matrix is updated, the model starts consecutively with the prediction step. Therefore, updating the covariance matrix is considered the last step of the EKF.

Observability

To determine if the vehicle behavior can be described with the selected output map, an analysis is required to determine the observability of the vehicle states. A system is said to be observable if the states of the system can be determined from the system's output. However, determining the observability of a non-linear system is often hard and can only be determined locally. A tool for checking local observability for a nonlinear system in the form of (2.1), is the rank condition (Theorem 2.1.2) of the so-called observation space Θ (Definition 2.1.1).

An observation space is obtained via its definition

$$\Theta = \begin{bmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \\ \vdots \end{bmatrix}, \tag{3.19}$$

where $h(x) \in \mathbb{R}^p$, $x \in \mathbb{R}^n$, $\Theta \in \mathbb{R}^{p\,n}$ and $L_f h(x)$ refers to the Lie derivative (Definition 2.1.3). The states are considered locally observable at x_0 if the observability contribution, denoted as $\Phi(x)$, is full rank. The observability contribution is calculated via

$$\Phi(x) = \frac{\partial \Theta}{\partial x}. \tag{3.20}$$

The rank condition states that when

$$\dim \Phi(x_0) = n, \tag{3.21}$$

there is local observability at x_0. This condition can only be achieved when $\Phi(x) \in \mathbb{R}^{q \times n}$, where $q \geq n$. Since the considered system is a multi-output system, an observation map $\hat{\Theta}$ can be determined, which proves local observability if

$$\dim \frac{\partial \hat{\Theta}(x_0)}{\partial x} = n \tag{3.22}$$

where the observation map can be determined via a smart selection of the output mapping $h(x)$ and its Lie derivatives.

When analyzing $h(x)$ it can be concluded that $\tilde{e}_y$ and e_ψ must always be provided as measurement. Missing these states indicate that SLAM has not managed to construct a reference path for motion planning, which is a situation when the car is not allowed to drive. Therefore, the states $\tilde{e}_y$ and $\tilde{e}_\psi$ are assumed to always be available when driving. Also, the dynamics for $\tilde{e}_y$ are singular when driving at the origin of the corner radius. This kinematic singularity occurs when $\tilde{e}_y = R$, which in practical applications does not happen due to track limits.

Furthermore, the EKF experiences a singularity at $\tilde{v}_x = 0$ due to the tire model. The accuracy of the EKF is affected while operating near this singularity, and the states are not observable when operating at this singularity. As a solution to this singularity, the EKF is not enabled when the velocity is below a certain threshold $\tilde{v}_{x,\text{switch}}$, for which the EKF will pass through raw measurement data to the controller, without updating the Kalman gain K_k and covariance matrix $P_{k|k}$. This threshold $\tilde{v}_{x,\text{switch}}$ is determined via simulation in Section 3.3, showing the performance for different $\tilde{v}_{x,\text{switch}}$.

To determine the observability of the system, the output mapping is adjusted to

$$h(x) = [\tilde{v}_x,\, \tilde{e}_y,\, \tilde{e}_\psi]. \tag{3.23}$$

If observability around x_0 can be found for (3.23), the same can be said for the original output mapping in (3.15). The observation map $\hat{\Theta}$ is set as

$$\hat{\Theta} = \begin{bmatrix} \tilde{v}_x \\ \tilde{e}_y \\ L_f \tilde{e}_y \\ L_f^2 \tilde{e}_y \\ \tilde{e}_\psi \\ L_f \tilde{e}_\psi \end{bmatrix}, \tag{3.24}$$

where the observability contribution yields

$$\Phi(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \tilde{e}_\psi & 1 & 0 & 0 & \tilde{v}_x & 0 \\ \Phi_{4.1} & \Phi_{4.2} & \Phi_{4.3} & \Phi_{4.4} & \Phi_{4.5} & \Phi_{4.6} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\kappa}{\tilde{e}_y\,\kappa - 1} & -\frac{\tilde{e}_\psi\,\kappa}{\tilde{e}_y\,\kappa - 1} & 1 & -\frac{\kappa^2\,(\tilde{v}_x - \tilde{e}_\psi\,\tilde{v}_y)}{(\tilde{e}_y\,\kappa - 1)^2} & -\frac{\kappa\,\tilde{v}_y}{\tilde{e}_y\,\kappa - 1} & 0 \end{bmatrix}, \tag{3.25}$$

where

$$\Phi_{4.1} = \frac{C_f\,(\tilde{v}_y + l_f\,\tilde{\omega})}{m\,\tilde{v}_x{}^2} + \frac{C_r\,(\tilde{v}_y - l_r\,\tilde{\omega})}{m\,\tilde{v}_x{}^2} + \frac{\kappa\,(2\,\tilde{v}_x - \tilde{e}_\psi\,\tilde{v}_y)}{\tilde{e}_y\,\kappa - 1} - \frac{C_f\,\tilde{\delta}\,\tilde{e}_\psi\,(\tilde{v}_y + l_f\,\tilde{\omega})}{m\,\tilde{v}_x^2}, \tag{3.26a}$$

$$\Phi_{4.2} = \frac{\tilde{e}_\psi\,\left(C_f\,\tilde{\delta} + m\,\tilde{\omega}\,\tilde{v}_x\right)}{m\,\tilde{v}_x} - \frac{C_f + C_r}{m\,\tilde{v}_x} - \frac{\tilde{e}_\psi\,\kappa\,\tilde{v}_x}{\tilde{e}_y\,\kappa - 1}, \tag{3.26b}$$

$$\Phi_{4.3} = \tilde{e}_\psi\,\tilde{v}_y + \frac{C_r\,l_r - C_f\,l_f + C_f\,l_f\,\tilde{\delta}\,\tilde{e}_\psi}{m\,\tilde{v}_x}, \tag{3.26c}$$

$$\Phi_{4.4} = -\frac{\kappa^2\,\tilde{v}_x\,(\tilde{v}_x - \tilde{e}_\psi\,\tilde{v}_y)}{(\tilde{e}_y\,\kappa - 1)^2}, \tag{3.26d}$$

$$\Phi_{4.5} = \frac{\tilde{F}_x - C_f\,\tilde{\delta}\,\left(\tilde{\delta} - \frac{\tilde{v}_y + l_f\,\tilde{\omega}}{\tilde{v}_x}\right) + m\,\tilde{\omega}\,\tilde{v}_y}{m} - \frac{\kappa\,\tilde{v}_x\,\tilde{v}_y}{\tilde{e}_y\,\kappa - 1}, \tag{3.26e}$$

$$\Phi_{4.6} = \frac{C_f\,\tilde{e}_\psi\,(\tilde{v}_y + l_f\,\tilde{\omega})}{m\,\tilde{v}_x} - \frac{C_f\,\left(2\,\tilde{\delta}\,\tilde{e}_\psi - 1\right)}{m}. \tag{3.26f}$$

Since (3.25) is a square matrix it is possible to determine its determinant. If the determinant is non-zero, it indicates that the matrix is full rank and satisfies the condition for local observability. The resulting determinant equals

$$\det \Phi(x) = \frac{C_f\,(2\,\tilde{\delta}\,\tilde{e}_\psi - 1)}{m} - \frac{C_f\,\tilde{e}_\psi\,(\tilde{v}_y + l_f\,\tilde{\omega})}{m\,\tilde{v}_x}, \tag{3.27}$$

which can be rewritten as

$$\begin{aligned} \det \Phi(x) &= \frac{C_f\,(2\,\tilde{\delta}\,\tilde{e}_\psi - 1)}{m} + \frac{C_f\,\tilde{e}_\psi}{m}(\alpha_f - \tilde{\delta}), \\ &= \frac{C_f}{m}\left(2\,\tilde{\delta}\,\tilde{e}_\psi - 1 + \tilde{e}_\psi\,(\alpha_f - \tilde{\delta})\right), \\ &= \frac{C_f}{m}\left(\tilde{e}_\psi\,(\alpha_f + \tilde{\delta}) - 1\right), \end{aligned} \tag{3.28}$$

by using the definition of the front tire side slip angle in (3.9). From the determinant, it can be concluded that the system is not locally observable when

$$(\tilde{e}_\psi\,(\alpha_f + \tilde{\delta}) - 1) = 0 \Longleftrightarrow \alpha_f + \tilde{\delta} = \frac{1}{\tilde{e}_\psi}. \tag{3.29}$$

The condition to obtain local observability is dependent on five states and therefore it is difficult to derive a general expression. However, by using testing data, empirical analyses can be done to determine if (3.29) occurs during online implementation. This analysis provides a visual indication of the system's observability.

In Figure 3.3, $\tilde{\delta} + \alpha_f$ is plotted versus $\tilde{e}_\psi^{-1}$, which are both derived from testing data which will be referred to as Data Set 1. The left figure show that the system is not locally observable as a few data points tend towards $\alpha_f + \tilde{\delta} \approx \tilde{e}_\psi^{-1}$. However, these particular points are a result of the singularity at $\tilde{v}_x = 0$, where these particular points are being observed near this singularity. The right figure shows the results when $\tilde{v}_x$ for the EKF is lower-bounded by $\tilde{v}_{x,\text{switch}}$, which in this case equals $1\,\text{m}\,\text{s}^{-1}$. This figure indicates that $\alpha_f + \tilde{\delta} \neq \tilde{e}_\psi^{-1}$, resulting in local observability of the EKF when $\tilde{v}_x \geq \tilde{v}_{x,\text{switch}}$. As this approach is purely pragmatic, it can not be guaranteed that the system is locally observable for all conditions, but it does indicate that the system is locally observable for a specific domain.

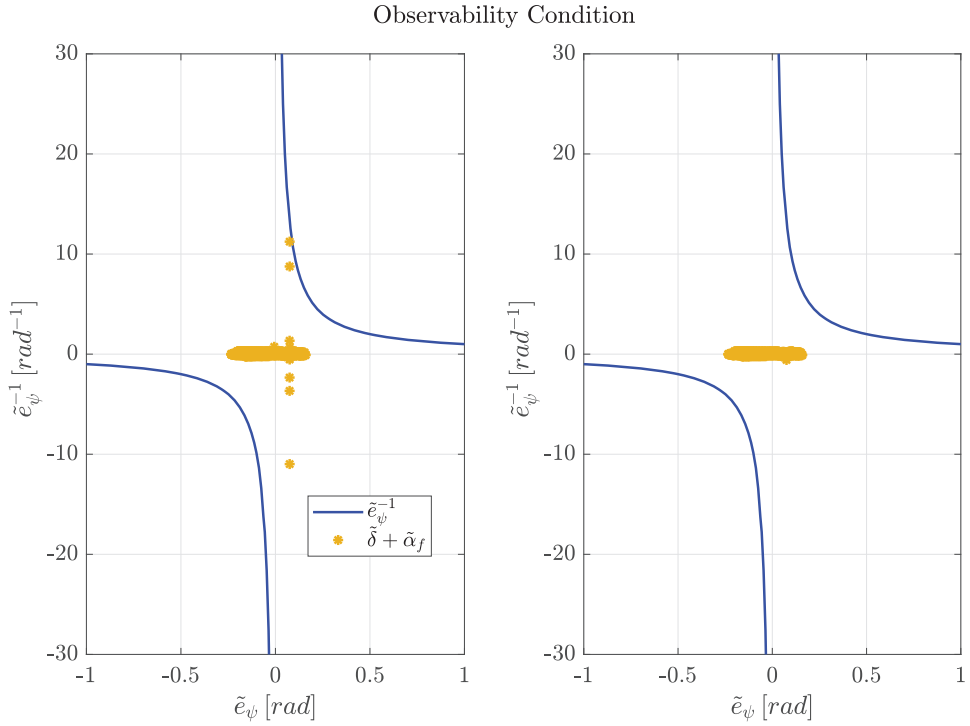

Figure 3.3: Left: result for $(\tilde{\delta} + \alpha_f)$ excluding lower-bound $\tilde{v}_{x,\text{switch}}$. Right: result for $(\tilde{\delta} + \alpha_f)$ including lower-bound $\tilde{v}_{x,\text{switch}} = 1\,\text{m}\,\text{s}^{-1}$.

3.3 Results

To validate the EKF, the observer is tested on the formula student racing car of University Racing Eindhoven. The observer is implemented in Simulink, which University Racing Eindhoven uses to run the autonomous system at 100 Hz. In Appendix B the implementation of the EKF is graphically shown, notice that additional steps were required based on the system's driving state, which is required by the Formula Student Rulebook, for which the reader is referred to [39]. The corresponding vehicle parameters are found in Table 3.1.

First, it is important to determine the accuracy of the prediction step in the EKF, which is analyzed by using different Data Sets and comparing the prediction with the resulting measurement. The results of the model validation give insight into the model's accuracy and the covariance of the

model prediction. Afterward, the covariance matrices Q_k and R_k are tuned to determine the desired behavior. Tuning these matrices is performed using the same data sets as the model validation, showing the difference in the accuracy of the model estimation. Last, since the EKF has a singularity at $\tilde{v}_x = 0$, simulations are performed for varying lower bounds on the velocity to determine the minimum required lower bound to guarantee enough accuracy for lower velocities.

Table 3.1: Vehicle parameters.

Parameter	Symbol	Value	Unit
Vehicle Mass	m	192	kg
Yaw moment of inertia	J	82	$kg\,m^2$
Distance front axle to CG	l_f	0.88	m
Distance rear axle to CG	l_r	0.64	m
Rolling friction coefficient	f_r	0.072	-
Aerodynamic drag coefficient	C_d	1.23	$N\,(m/s)^{-2}$
Cornering stiffness front axle	C_f	32000	$N\,rad^{-1}$
Cornering stiffness rear axle	C_r	45000	$N\,rad^{-1}$

Model validation

To validate the prediction model and its parameters, the EKF is used while it mainly relies on the prediction step and assumes that the measurement data has a high covariance, resulting in $Q_k << R_k$. The results from this model validation provide insight into the prediction step accuracy compared to the measured states. Inaccurate model parameters, such as cornering stiffness, result in a model mismatch and therefore an inaccurate prediction.

Let us define the matrices

$$Q_k = \text{diag}(1\text{e}{-4},\ 1\text{e}{-3},\ 1\text{e}{-4},\ 5\text{e}{-4},\ 3\text{e}{-3},\ 4\text{e}{-3}), \tag{3.30a}$$

$$R_k = \text{diag}(0.1\text{e}4,\ 0.1\text{e}4,\ 0.5\text{e}4,\ 0.15\text{e}4,\ 0.08\text{e}4), \tag{3.30b}$$

where the results of this validation, using Data Set 2, are displayed in Figure 3.4. It is observed that the prediction model is deviating from the measurement, indicating a model mismatch. In Figure 3.5, the state estimation for $\tilde{v}_y$ and $\tilde{\omega}$ is plotted with the respective measurement data, where $\tilde{v}_y$ is excluded for now since there is no measurement data available to validate the estimation.

It can be concluded from Figure 3.5 that the model estimation for both $\tilde{v}_x$ and $\tilde{\omega}$ provides a reasonable estimate of what the vehicle states are. However, the estimation for $\tilde{v}_x$ shows a slight offset compared with its measurement, and the estimation of $\tilde{\omega}$ mimics the measurement data except for the peaks around $t = 28[s]$ and $t = 33[s]$. Therefore, the cause for the model mismatch in the error estimates is likely caused by the estimation of $\tilde{v}_y$, and especially the tire model parameters C_f and C_r. Furthermore, the error in the tire dynamics accumulates for the lateral dynamics since the lateral tire forces act in the same direction, where the yaw dynamics is a ratio between the front and rear tire forces based on the weight distribution. Since this is a ratio, the error in dynamics cancels each other out, resulting in a better estimation for $\tilde{\omega}$ than $\tilde{v}_y$.

The influence of $\tilde{v}_y$ on the model mismatch in the error estimation is further examined by adjusting the prediction model dynamics for $\tilde{e}_y$ and $\tilde{e}_\psi$, where $\tilde{v}_y$ is excluded

$$\dot{\tilde{e}}_y = \tilde{v}_x \sin(\tilde{e}_\psi), \tag{3.31a}$$

$$\dot{\tilde{e}}_\psi = \tilde{\omega} - \kappa(\tilde{s}) \frac{\tilde{v}_x \cos(\tilde{e}_\psi)}{1 - \kappa(\tilde{s})\tilde{e}_y}. \tag{3.31b}$$

In Figure 3.6, the same data as Figure 3.4 is used, but the prediction step is modeled with the error dynamics as described in (3.31). As a comparison, the Root Mean Square Error (RMSE) is

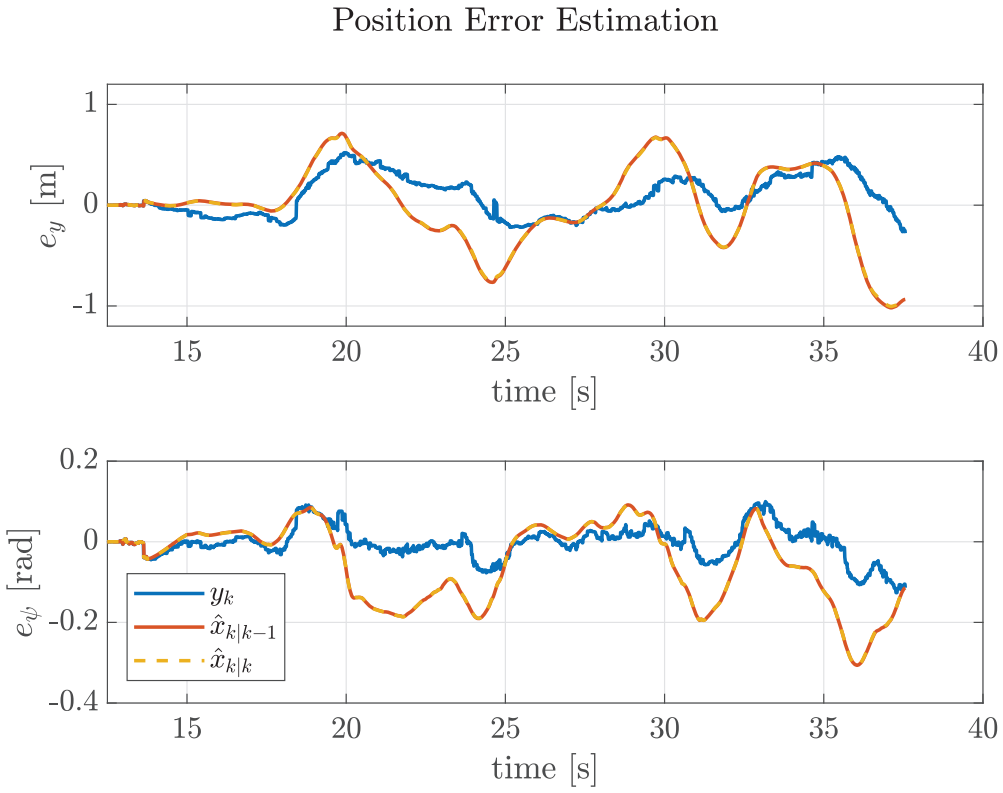

Figure 3.4: Error states estimation for $Q_k << R_k$, using Data Set 2.

computed between the measurement $\hat{y}_{k|k}$ and the estimate $\hat{x}_{k|k}$, providing the results in Table 3.2. As can be seen, the estimation when excluding the lateral velocity is much better, decreasing the RMSE by 57.0 % and 47.2 %. In conclusion, the state covariance on both error states is increased, and the EKF relies more on the input from the measurement.

The model can be improved by performing more tests while measuring the lateral velocity $\tilde{v}_y$, to validate the lateral dynamics model and give more insight into the performance of the EKF. Additionally, tire testing is required to better estimate the cornering stiffness of both the front and rear tires. However, the model is fixed for this thesis work, and the covariance matrices are tuned to get the desired performance.

Table 3.2: Root Mean Square Error data set 1.

RMSE	Including $\tilde{v}_y$	Excluding $\tilde{v}_y$	Difference
$\tilde{e}_y[m]$	0.3525	0.1839	−57.0 %
$\tilde{e}_\psi[rad]$	0.0913	0.0393	−47.2 %

Covariance Matrices

Thus far, the results considered that $Q_k << R_k$, but tuning these provide the desired behavior between the state prediction and measurement. Data Set 1 is used since the car drove much faster and a longer distance than Data Set 2.

To make a fair comparison with the model validation, the output is first analyzed for $Q_k << R_k$ and is displayed in Figure 3.8 and Figure 3.7. The model output is much worse for higher velocities, as the system excites more dynamic behavior. Furthermore, due to a decreased performance in

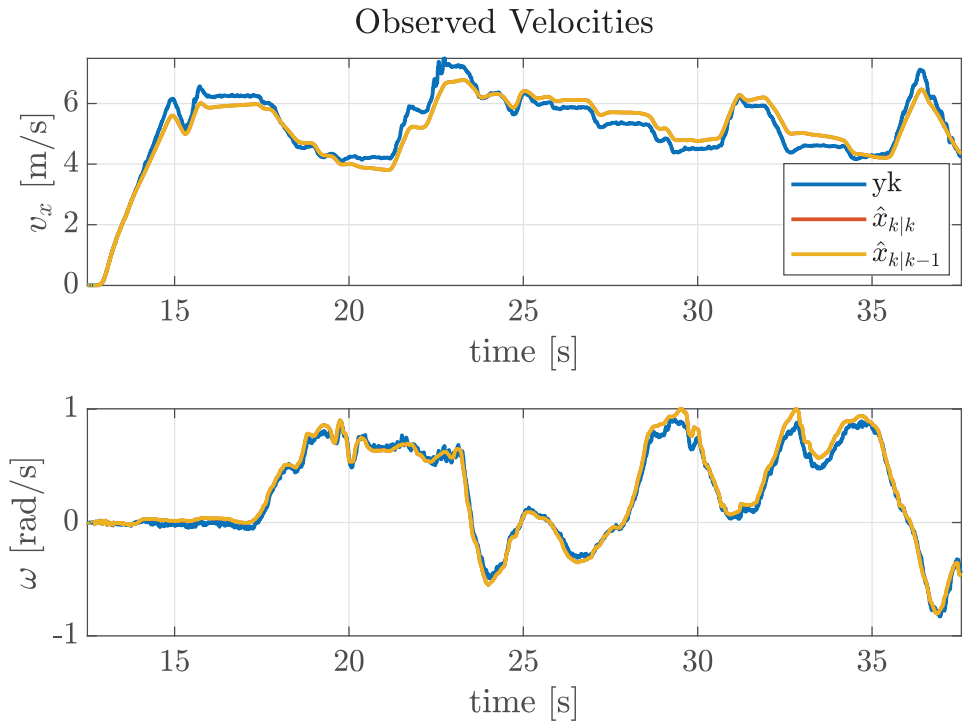

Figure 3.5: $\tilde{v}_x$ and $\tilde{\omega}$ state estimation for $Q_k << R_k$ using Data Set 2.

$\tilde{v}_x$, the estimation for $\tilde{e}_y$ and $\tilde{e}_\psi$ has worsened when comparing it with the results in Figure 3.6 where $\tilde{v}_y$ is excluded from the error dynamics.

Let us define the matrices

$$Q_k = \text{diag}(1\text{e}{-4},\ 1\text{e}{-3},\ 1\text{e}{-4},\ 5\text{e}{-4},\ 3\text{e}{-3},\ 4\text{e}{-3}), \tag{3.32a}$$
$$R_k = \text{diag}(0.1,\ 0.1,\ 0.5\text{e}4,\ 0.15\text{e}4,\ 0.08), \tag{3.32b}$$

where the covariance for the measurements of $\tilde{e}_y$ and $\tilde{e}_\psi$ is still high, but all other states rely now on both the model estimation and the measurement. As a result, the velocity estimates in Figure 3.9 show the improvement made by adjusting the covariance matrices. Analyzing the estimation of $\tilde{v}_x$ shows the trade-off between Q_k and R_k, as the estimation for $t \in [62.5, 65]$ shows an overshoot, where the estimation of $\tilde{v}_x$ might benefit from relying more on the measurement. However, in $t \in [66.5, 70]$, the wheel encoder measurement indicates that the wheels are locking. In contrast, the EKF provides a more realistic vehicle velocity by mitigating the locking wheels and relying more on the prediction. The error estimation in Figure 3.10 also improved due to a better state estimation of both $\tilde{v}_x$ and $\tilde{\omega}$. This highlights the importance of the model validation for $\tilde{v}_y$.

At last, the covariance on the measurement of $\tilde{e}_y$ and $\tilde{e}_\psi$ is defined, resulting in the final Q_k and R_k matrices

$$Q_k = \text{diag}(1\text{e}{-4},\ 1\text{e}{-3},\ 1\text{e}{-4},\ 5\text{e}{-4},\ 3\text{e}{-3},\ 4\text{e}{-3}), \tag{3.33a}$$
$$R_k = \text{diag}(0.1,\ 0.1,\ 0.5,\ 0.15,\ 0.08). \tag{3.33b}$$

In contrary with (3.32) is the EKF output now updated based on the footpoint calculation. Due to the reduced covariance, the influence of the model mismatch is reduced since it now relies more on the measurement data than the prediction model. The result of the updated covariance matrix can be observed in Figure 3.11, where the results show a more reactive behavior towards

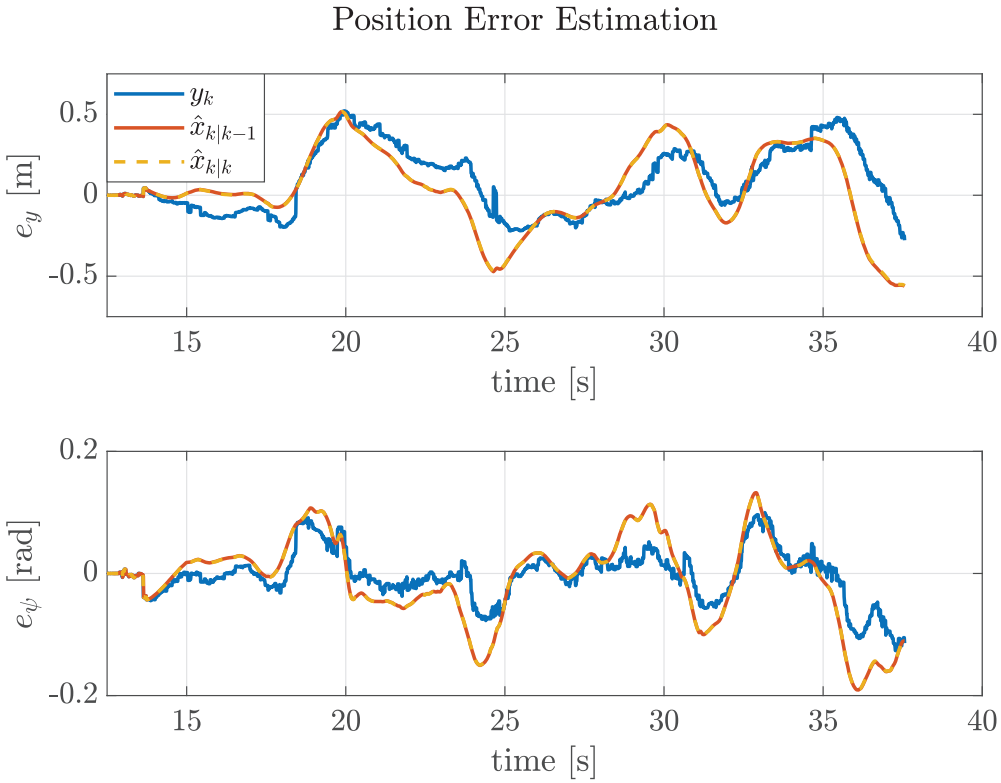

Figure 3.6: Error states estimation for $Q_k << R_k$, using (3.31) excluding $\tilde{v}_y$ and using Data Set 2.

the inputs of $\tilde{e}_y$ and $\tilde{e}_\psi$. Due to the more significant dependency on the measurement, its state estimation changes for every measurement. However, the state estimation does mitigate the issue that is presented initially in section 3.1, as the results show several instantaneous changes in the error measurement in the interval $t \in [35, 40]$, where the EKF does react to the rapid changes but reduces the change in amplitude significantly.

Lower Bound $\tilde{v}_x$

The EKF is based on a single-track model, including linear tire dynamics. When implementing a tire model into the EKF, a singularity is introduced at $\tilde{v}_x = 0$. This singularity affects the accuracy of the EKF output for lower velocities and results in a non-observable system at $\tilde{v}_x = 0$. To mitigate the influence of this singularity, a lower bound $\tilde{v}_{x,\text{switch}}$ is determined for which the EKF turns off and on. Via simulations the lower bound $\tilde{v}_{x,\text{switch}}$ is determined, by analyzing the EKF output for varying $\tilde{v}_{x,\text{switch}}$ using Data Set 1 and Data Set 2.

The implementation of the lower bound is based on the measurement y_k for $\tilde{v}_x$, where the output of the EKF is selected via

$$\hat{x}_{k|k} = \begin{cases} y_k & \text{if } \tilde{v}_x \leq \tilde{v}_{x,\text{switch}}, \\ \hat{x}_{k|k} & \text{if } \tilde{v}_x > \tilde{v}_{x,\text{switch}}. \end{cases} \tag{3.34}$$

When the vehicle velocity is beyond $\tilde{v}_{x,\text{switch}}$, it will pass through the unfiltered measurement data, additionally, when passing through the measurement data, the covariance matrix $P_{k|k}$ and the Kalman gain K_k are not updated. Not updating the covariance matrix and Kalman gain is

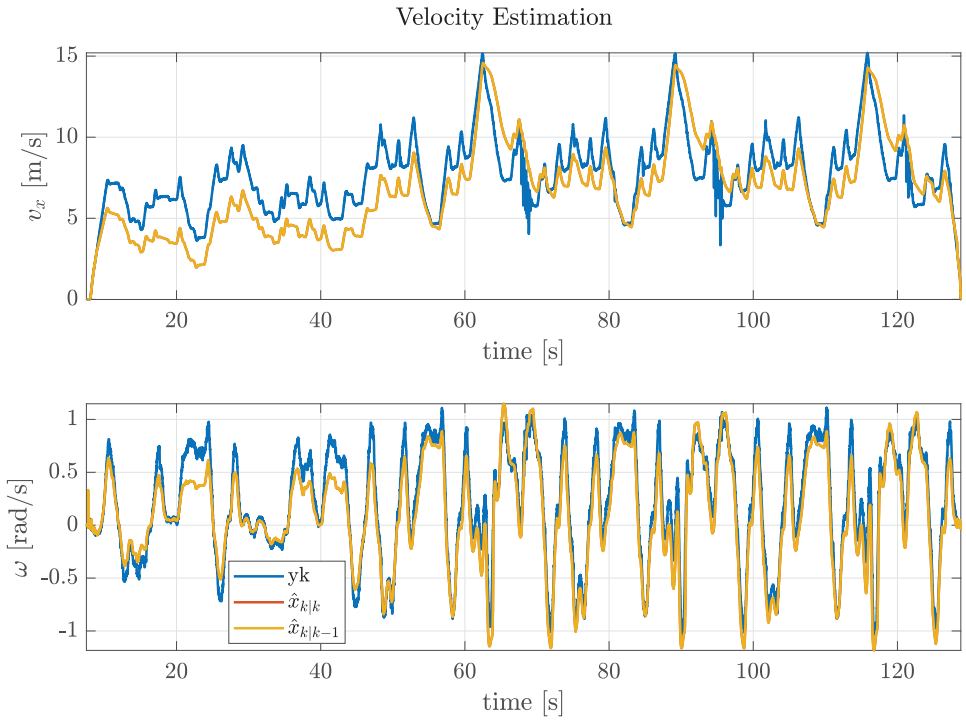

Figure 3.7: Velocity estimation for $Q_k << R_k$ using Data Set 1.

implemented via

$$P_{k|k} = \begin{cases} P_{k-1|k-1} & \text{if } \tilde{v}_x \leq \tilde{v}_{x,\text{switch}}, \\ P_{k|k} & \text{if } \tilde{v}_x > \tilde{v}_{x,\text{switch}}. \end{cases} \tag{3.35a}$$

$$K_k = \begin{cases} K_{k-1} & \text{if } \tilde{v}_x \leq \tilde{v}_{x,\text{switch}}, \\ K_k & \text{if } \tilde{v}_x > \tilde{v}_{x,\text{switch}}. \end{cases} \tag{3.35b}$$

The results of this application can be observed in the figures below. As expected, the effect is mainly observed in the yaw velocity estimation, as Figure 3.12 shows that the model's prediction deviates for $t \in [8.6, 9.2]$ before merging to the same trend. Similar behavior can be observed in Figure 3.13 for $t \in [13.4, 13.8]$ where the results for $\tilde{v}_{x,\text{switch}} \in \{0.1, 0.5\}$ show a more significant deviation before merging to the general solution of the EKF.

In Figure 3.14 and Figure 3.15, the effect on the error estimation can be observed. Where it was expected that the estimation of $\tilde{e}_\psi$ would be affected by the lower bound due to the appearing yaw velocity in its dynamics, it is $\tilde{e}_y$ which shows the more significant difference in output. As is seen in the model validation, are the error dynamics sensitive to a model mismatch in the velocity states. Due to the later transition towards the EKF output, the estimation of the velocity states improves for lower velocities. Therefore a better error estimation is found for increasing $\tilde{v}_{x,\text{switch}}$. The reason that this mainly occurs for $\tilde{e}_y$ is the choice of covariance matrices Q_k and R_k, which are defined in (3.33), where the output for $\tilde{e}_y$ relies more on the model than $\tilde{e}_\psi$, which relies more on the measurement.

Note that for the prediction step a RK4 discretization is used, which provides a better estimate at low speeds and lower frequencies. When applying the EKF with a RK2, or even forward Euler discretization, one should apply a higher $\tilde{v}_{x,\text{switch}}$ than is presented in this thesis. Also, the transition towards a kinematic model at low speed provides a better outcome since a kinematic

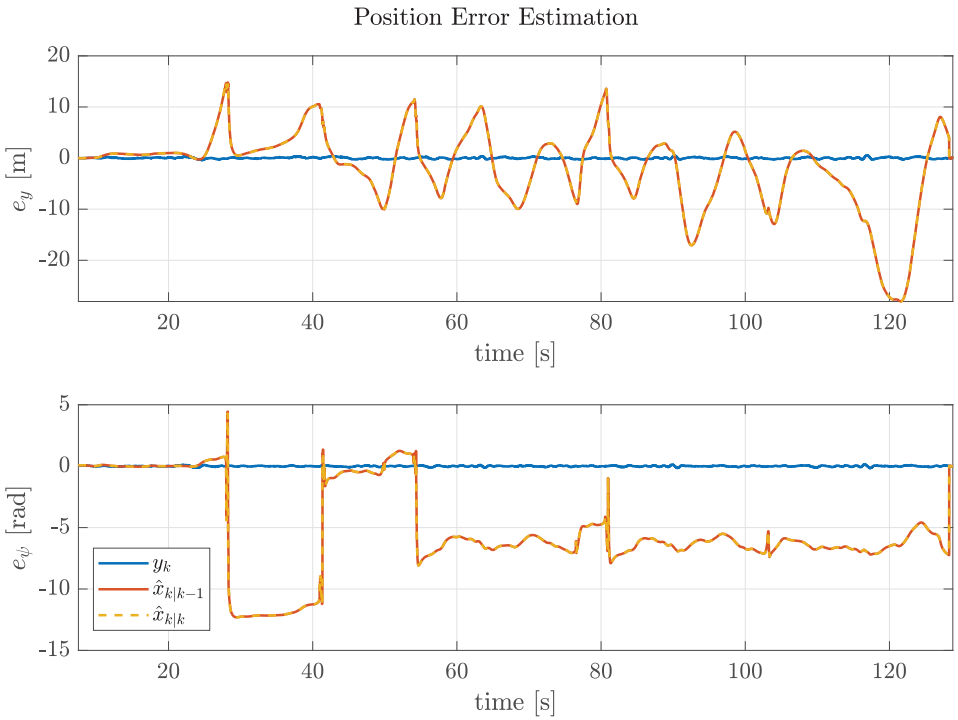

Figure 3.8: Error states estimation for $Q_k << R_k$ using Data Set 1.

model does not suffer from such singularities. Showing the difference in discretization methods and the transition towards a kinematic model for the EKF is beyond the scope of this thesis work.

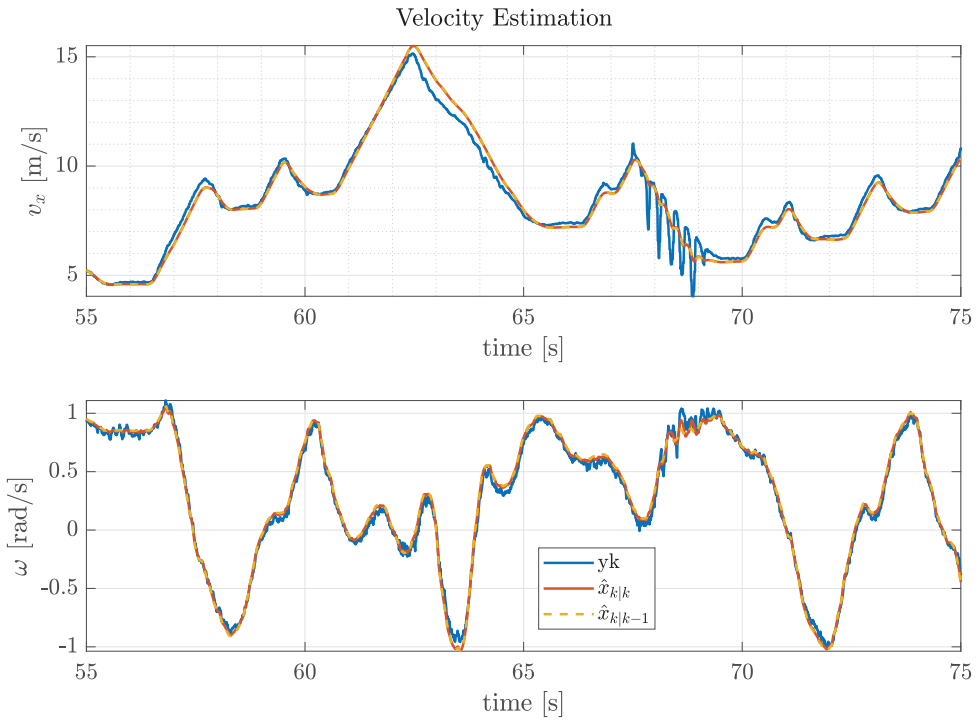

Figure 3.9: Velocity estimation for (3.32), using Data Set 2.

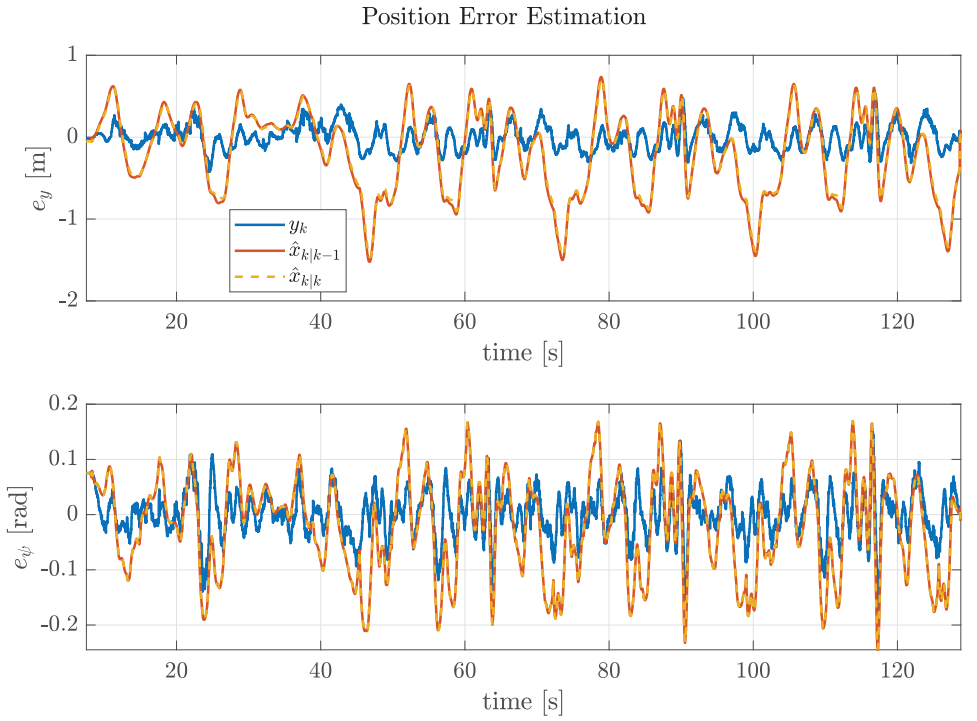

Figure 3.10: Error states estimation Velocity estimation for (3.32), using Data Set 2.

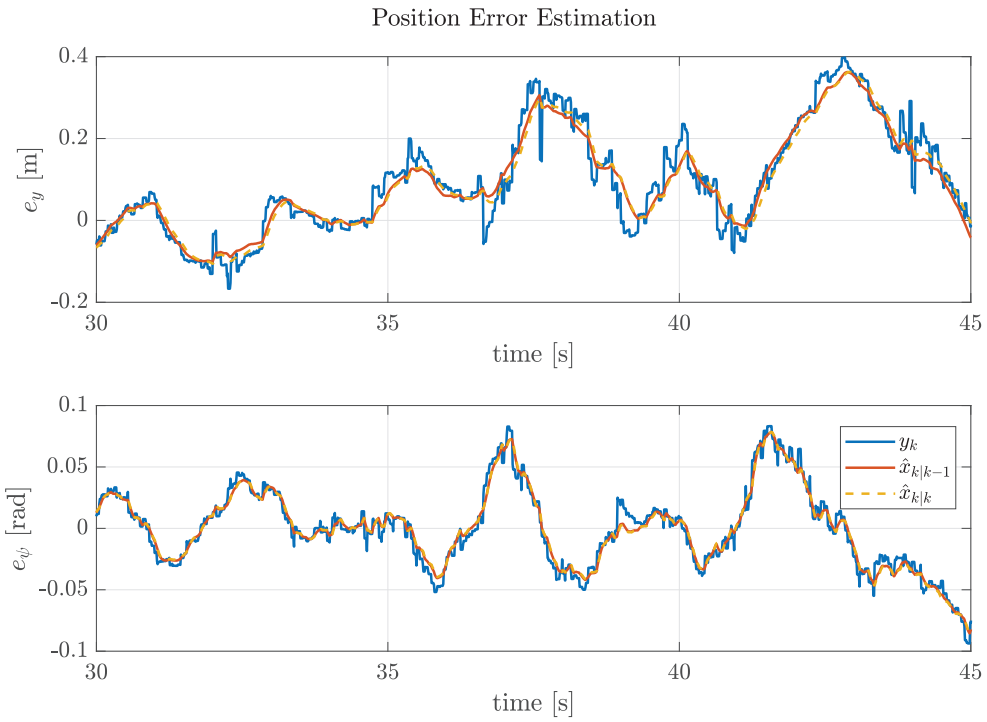

Figure 3.11: Error states estimation (3.33), using Data Set 1.

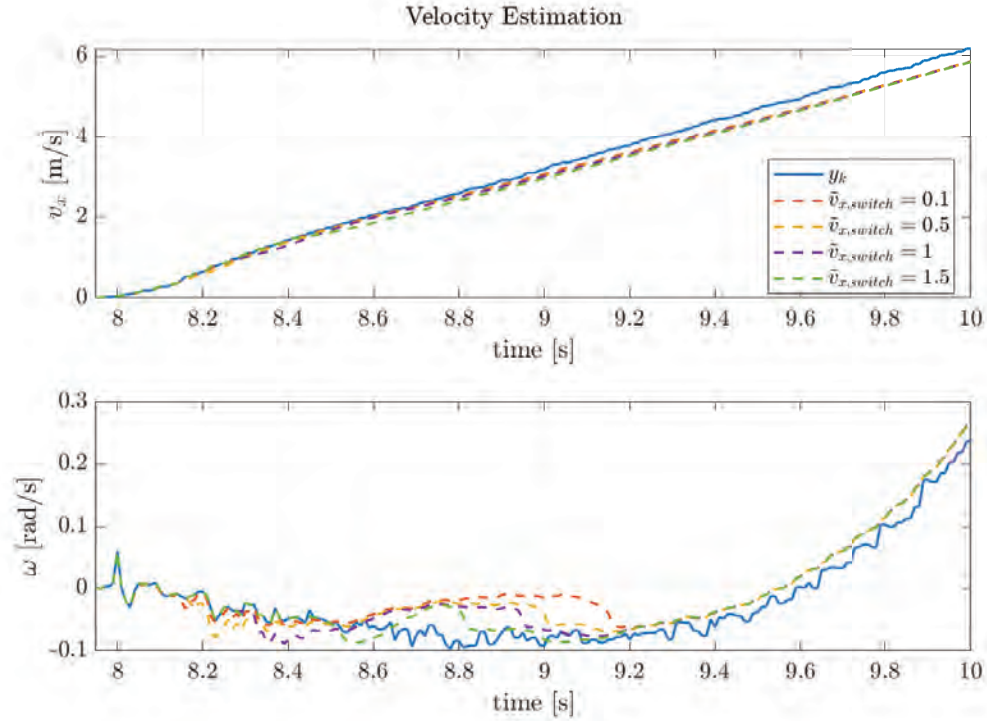

Figure 3.12: Velocity estimation for different $\tilde{v}_{x,\text{switch}}$ using Data Set 1.

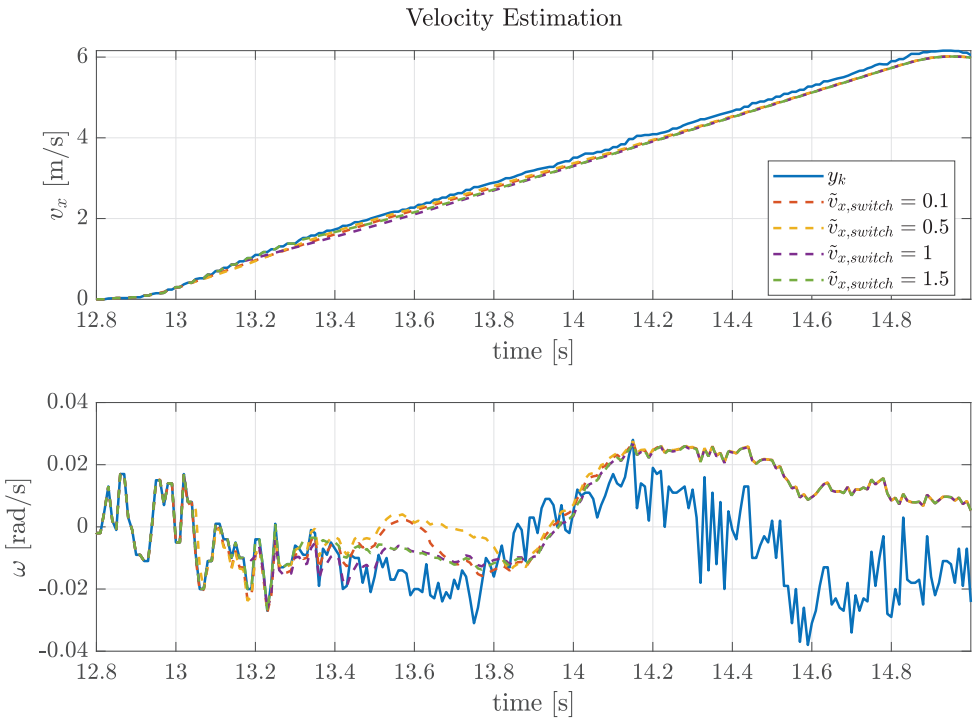

Figure 3.13: Velocity estimation for different $\tilde{v}_{x,\text{switch}}$ using Data Set 2.

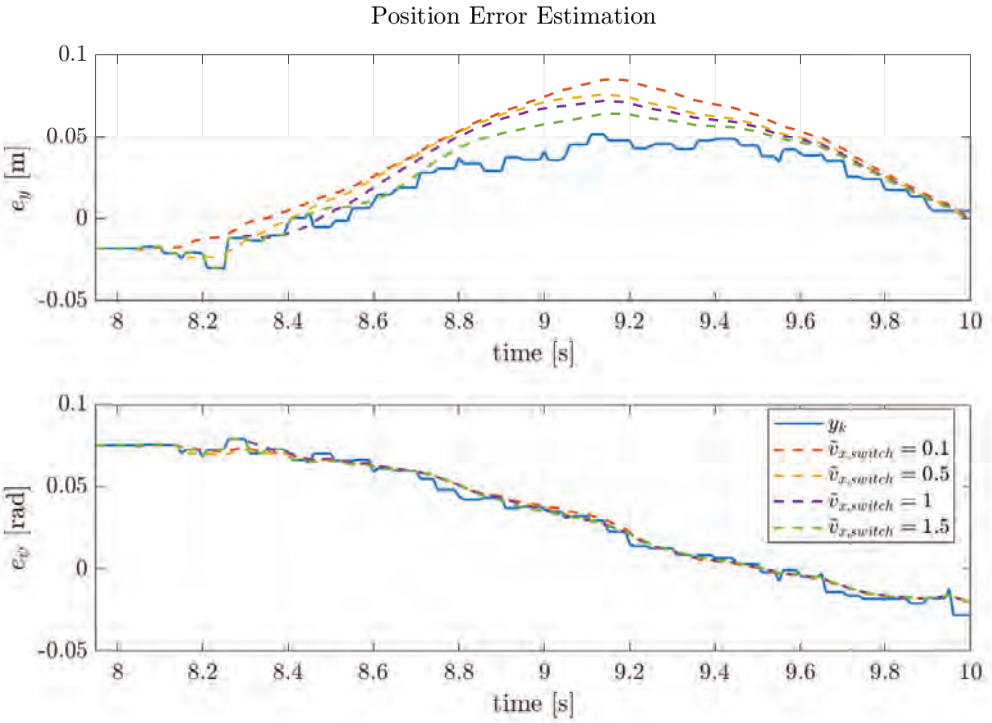

Figure 3.14: Error states estimation for different $\tilde{v}_{x,\text{switch}}$ using Data Set 1.

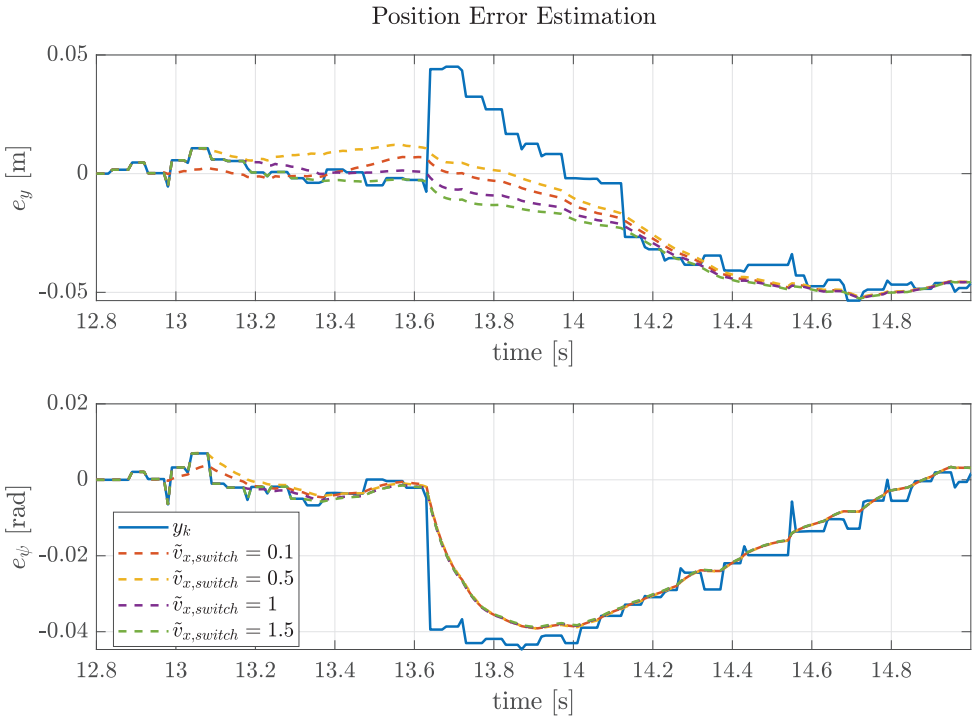

Figure 3.15: Error states estimation using Data Set 2.

3.4 Observation and Conclusion

This chapter discussed the localization of the vehicle with respect to a reference path. First, the principle of the footpoint is introduced, representing the location of the Frenet Coordinate frame T. Then, based on the footpoint, the error states e_y and e_ψ are derived, which then can be used for the MPC.

SLAM provides the reference path, and it is observed that this path can shift, resulting in instantaneous and non-smooth error states. Therefore, an EKF is introduced and applied to solve this problem. A simplified bicycle model is obtained assuming linear tire dynamics and using a small angle approximation. Based on a simplified output mapping, local observability is obtained based on a pragmatic approach where it is shown to be sufficient that $\alpha_f + \tilde{\delta} \neq \tilde{e}_\psi^{-1}$. However, as the results are based on a pragmatic approach, not all scenarios can be included, implying that local observability can not be guaranteed.

Several simulations are performed using two data sets provided by URE. At first, the model and its parameters are validated. These results showed that the error states are sensitive to a model mismatch, and due to lacking the measurement on $\tilde{v}_y$, no good model estimation could be made. Finally, via tuning the covariance matrices Q_k and R_k, a reasonable estimate of $\tilde{v}_x$ and $\tilde{\omega}$ is found, capable of filtering out disturbances, such as wheel slip.

To improve the overall model, the measurement of $\tilde{v}_y$ must be obtained to validate the last velocity state. Also, more testing is required to find a better estimation of the cornering stiffnesses C_f and C_r as including the estimation of $\tilde{v}_y$ in the error dynamics caused the RMSE to increase by 50 %.

The next chapter will discuss the optimization problem to solve the minimum time problem. As this optimization problem is formulated in the Frenet coordinate frame, this chapter worked towards a smoother control input, potentially avoiding infeasibility or undesired behavior in the MPC output.

Chapter 4

Optimization Problem

This chapter discusses Nonlinear Programming (NLP) to solve the online optimization problem. The applied control strategy is MPC, an advanced control technique used to compute control inputs while satisfying a set of constraints. Since MPC is a more advanced control technique, the models used in MPC generally describe the motion of complex dynamic systems. Where a typical optimization problem solves the problem only once, MPC solves the problem repeatedly over a prediction horizon N. The problem formulation for NLP using MPC is formulated as

$$\begin{aligned}
\min_{X_k, U_k} \quad & J(x_{i|k}, u_{i|k}), && \text{(4.1a)}\\
\text{subject to} \quad & x_{i+1|k} = f(x_{i|k}, u_{i|k}), && \text{(4.1b)}\\
& x_{0|k} = x(k), && \text{(4.1c)}\\
& x_{min} \leq x_{i|k} \leq x_{max}, && \text{(4.1d)}\\
& u_{min} \leq u_{i|k} \leq u_{max}, && \text{(4.1e)}\\
& x_{i|k} \in \mathcal{X}, && \text{(4.1f)}\\
& u_{i|k} \in \mathcal{U}, && \text{(4.1g)}
\end{aligned}$$

where $X_k = [x_{1|k}, ..., x_{N+1|k}]$ and $U_k = [u_{1|k}, ..., u_{N|k}]$ represent the predicted states and inputs, respectively, for each step k over the prediction horizon $i \in \{1, ..., N\}$.

MPC requires a model describing the dynamics of the real-time system, which is the two-track model discussed in section 2.2. However, a compromise often must be made between model complexity and computation time to solve the optimization problem. Therefore, in section 4.1, the principle of cascaded vehicle models is used, exploiting the advantages of the single track and point-mass model from section 2.2, resulting in faster computation without decreasing the performance significantly.

The objective function of an optimization problem can be configured to achieve various goals. For this thesis, the main goal of the optimization problem is to minimize lap time for an all-wheel drive autonomous race car while using a finite planning horizon. While pushing the vehicle to the limits, the controller is responsible for making trade-offs in lateral and longitudinal control to stabilize the car. The problem formulation in section 4.2 discusses how the objective function $J(x_{i|k}, u_{i|k})$ is formulated to achieve these control goals.

For the optimal solution to be feasible, it should satisfy a set of constraints in the form of (4.1d) - (4.1g). These constraints are discussed in section 4.1, where the lower and upper bounds from both the states and inputs are defined, and the set of feasible states $\mathcal{X}$ and inputs $\mathcal{U}$ are restricted via several equality and inequality constraints.

4.1 Cascaded Vehicle Model

One of the control objectives of motion planning is stabilizing the car, while in autonomous racing, the car also has to be pushed toward the limits of its performance. A highly non-linear model is required to achieve these goals, describing the longitudinal, stiff lateral, and stiff yaw dynamics. Another control objective is trajectory planning to ensure a safe velocity profile along the reference path and avoid road departure.

As suggested in [28], these control objectives do not necessarily have a constant level of relevance or criticality throughout the planning horizon. Where different motion planning controllers use a hierarchical approach with multiple loops, potentially causing infeasible reference signals between controllers or using conservative reference signals that do not plan to use the system to its full potential, the principle of cascaded vehicle models in a single planning horizon is introduced.

The model for the MPC in this thesis is based on the principle of cascading vehicle models [28]. At first, a two-track model is used to utilize the input from four electric in-wheel motors fully. Where classic control strategies rely on a separate torque vectoring controller, this MPC uses a high-fidelity plant model in the first part of the planning horizon to maximize the vehicle performance without relying on external controllers. However, as the two-track model has a significant computational burden, its prediction horizon should be kept to a minimum while still a feasible and smooth control input can be obtained. To ensure vehicle stability, the second part of the horizon is based on the single-track model, providing high-quality control input and utilizing the system to its full potential. Finally, the last part of the planning horizon is extended with a lower fidelity model, namely a point mass, providing a larger look-ahead distance at a low computational burden. The overall concept is displayed in Figure 4.1, where the total look-ahead distance consists of the three vehicle models combined, yielding that $s_t = s + \tilde{s} + \bar{s}$.

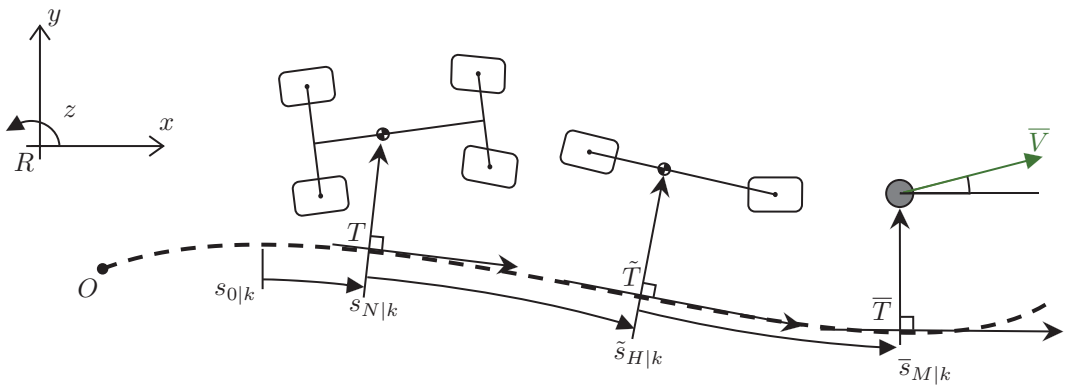

Figure 4.1: Cascaded vehicle model horizon.

The first step in serially cascading different vehicle models is carefully defining a mapping where the final state of one vehicle model propagates towards the initial state of the following vehicle model. Since the same states describe the two-track model and the single-track model, the mapping can be defined relatively simple, where the two-track states at prediction $i = N$ define the initial state

of the single-track model for $z = 0$, yielding

$$\tilde{v}_{x,0|k} = v_{x,N|k}, \tag{4.2a}$$
$$\tilde{v}_{y,0|k} = v_{y,N|k}, \tag{4.2b}$$
$$\tilde{\omega}_{0|k} = \omega_{N|k}, \tag{4.2c}$$
$$\tilde{s}_{0|k} = s_{N|k}, \tag{4.2d}$$
$$\tilde{e}_{y,0|k} = e_{y,N|k}, \tag{4.2e}$$
$$\tilde{e}_{\psi,0|k} = e_{\psi,N|k}, \tag{4.2f}$$
$$\tilde{\delta}_{0|k} = \delta_{N|k}. \tag{4.2g}$$

The mapping between the single track and point-mass model is defined differently, as the point-mass does not consider yaw dynamics and only a single velocity vector. Since $\bar{V}$ represents the resulting velocity vector, the mapping can be defined by determining the resultant velocity vector from the two vectors $\tilde{v}_x$ and $\tilde{v}_y$ at $z = H$.

For the mapping between $\tilde{e}_\psi$ and $\bar{e}_\psi$, it has to be taken into account that both represent a different angle. Where $\tilde{e}_\psi$ represents the error between the path heading and the vehicle chassis, $\bar{e}_\psi$ represents the error between the heading of the path and the resulting velocity vector. Therefore, the vehicle side slip angle from the single-track model has to be determined to compensate for this difference, resulting in the mapping [28]

$$\bar{V}_{0,k} = \sqrt{\tilde{v}_{x,H|k}^2 + \tilde{v}_{y,H|k}^2}, \tag{4.3a}$$
$$\bar{e}_{y,0|k} = \tilde{e}_{y,H|k}, \tag{4.3b}$$
$$\bar{e}_{\psi,0|k} = \arctan\left(\frac{\tilde{v}_{y,H|k}}{\tilde{v}_{x,H|k}}\right) + \tilde{e}_{\psi,H|k}, \tag{4.3c}$$
$$\bar{s}_{0|k} = \tilde{s}_{H|k}, \tag{4.3d}$$

where the mapping for the position states $\bar{e}_y$ and $\bar{s}$ can be set equal to the states of the single track model at $z = H$ since these represent the same position.

The transition of the tire forces between the different vehicle models also has to be considered, but this is done via a penalty in the cost function, which is discussed in the following section. In addition, when developing an MPC with a cascaded model, it is essential to have consistency between state and input mapping, as well as the constraints and cost function. This is also considered in the following sections, where these topics are further discussed.

4.2 Problem Formulation

The goal of the MPC is to minimize the objective function,

$$\min_{X_k\ U_k} \bar{t}_M + J_M + J_U + J_{\Delta U} + J_e + J_{tr} + J_\beta, \tag{4.4}$$

which is the sum of seven different terms. At first, the primary objective is to minimize the lap time, hence minimizing the required time for the point-mass to reach the end of the planning horizon, denoted by $\bar{t}_M$.

Space Domain

As time becomes an optimization variable, a transformation from the time to space domain is suggested. Formulating the equations of motion are naturally done as a function of time, especially when designing controllers where the sampling frequency determines when the input is applied.

However, time becomes an optimization variable when the goal is to minimize lap time. A transformation from time to space domain is performed, such that the dynamics evolve with respect to space rather than time. The transformation can be obtained by applying

$$\dot{x} = \frac{dx}{dt} = \frac{dx}{ds}\frac{ds}{dt}, \tag{4.5}$$

which can be rewritten to find the expression

$$\frac{dx}{ds} = \dot{x}\left(\frac{ds}{dt}\right)^{-1} = \dot{x}\frac{1}{\dot{s}}, \tag{4.6}$$

where $\dot{s}$ is the vehicle's velocity along the reference path. When performing this transformation, the state s becomes redundant since it is linearly increasing with the discretization step in the space domain. Therefore, the state s is replaced with state time t, where the equation of motion in the time domain is formulated as

$$\dot{t} = 1, \tag{4.7}$$

which is in the time domain linearly increasing, similarly to s in the spatial domain. However, the dynamics of time in the spatial domain are described as

$$\dot{t} = \frac{1}{\dot{s}}. \tag{4.8}$$

As a result, time can now be formulated as a function of the vehicle states by solving

$$t = \int_0^{\Delta s} \dot{t}\ ds, \tag{4.9}$$

where Δs is the discretization step size in the spatial domain. This integral is approximated based on the selected discretization method.

Terminal State

The five remaining terms in (4.4) enhance the optimization problem's feasibility and guarantee safe driving conditions. First, let us define the terminal cost J_M, which represents the cost on the final stage in the finite horizon problem. Terminal cost is used to enhance feasibility, where the terminal state is forced towards a state vector which is considered a safe set. When considering autonomous racing with online path planning, being on the centerline of the explored track is set to be a safe position. Therefore, the lateral error $\bar{e}_y$ and the heading error $\bar{e}_\psi$ at the end of the planning horizon is penalized, resulting in

$$J_M = W_{\bar{e}_y,M}\bar{e}^2_{y,M|k} + W_{\bar{e}_\psi,M}\bar{e}^2_{\psi,M|k} + W_{\bar{V}_M}\bar{V}^2_{M|k}, \tag{4.10}$$

where $W_{\bar{e}_y,M}$ and $W_{\bar{e}_\psi,M}$ are the corresponding weights. The last term in the penalty J_M affects the velocity at the terminal state $\bar{V}_{M|k}$. This cost should only apply if the velocity at the end of the horizon exceeds a feasible set of velocities, guaranteeing that the vehicle can stop in time for any corner on the track.

Since the track is unknown at the start of the race, it is also impossible to determine a safe bound $\bar{V}_{M,\max}$ which depends on the progress made along the centerline. Therefore, a general bound is required which guarantees that the NMPC can slow down soon enough to respect the track limits still. The bound $\bar{V}_{M,\max}$ of the feasible set is defined by assuming steady-state cornering, where the lateral acceleration is defined as

$$a_y = \frac{\bar{V}^2}{R}. \tag{4.11}$$

From (4.11), it can be observed that the maximum horizon velocity depends on the smallest radius on the track and the maximum lateral acceleration. The maximum lateral acceleration can be derived from the tire characteristics and the lateral dynamics of the point mass, defined as

$$ma_y = \bar{F}_y, \tag{4.12}$$

substituting (4.11) yields

$$\bar{V} = \sqrt{\frac{\bar{F}_y R}{m}}. \tag{4.13}$$

From (4.13), it can be concluded that the maximum velocity can be achieved by assuming $\bar{F}_y = \bar{D}_y$. However, this assumption does not provide a feasible solution, as the required tire forces to achieve this steady-state condition exceed the friction ellipse. Assuming $\bar{F}_{y,\max} = \bar{D}_y$ excludes the need of any longitudinal force $\bar{F}_x$. However, to drive at a constant velocity, $\bar{F}_x$ must compensate for the occurring drag and rolling resistance forces.

Therefore, the required longitudinal force is estimated to drive the steady-state velocity. Assuming that the tires only generate lateral force, the maximum steady-state velocity can be obtained via

$$\bar{V}_{ss} = \sqrt{\frac{\bar{D}_y R}{m}}. \tag{4.14}$$

Based on this assumption, the required longitudinal force can be determined via

$$\bar{F}_{x,ss} = 0.5\rho_{air} C_D A_s \bar{V}_{ss}^2 + f_r F_z. \tag{4.15}$$

The corrected $\bar{F}_{y,\max}$ to ensure a feasible set for $\bar{V}_M$ can be determined based on the tire friction ellipse

$$\left(\frac{\bar{F}_y}{\bar{D}_y}\right)^2 + \left(\frac{\bar{F}_x}{\bar{D}_x}\right)^2 = 1 \Rightarrow \bar{F}_{y,\max} = \bar{D}_y \sqrt{\left(1 - \left(\frac{\bar{F}_{x,ss}}{\bar{D}_x}\right)^2\right)}. \tag{4.16}$$

Substituting $\bar{F}_{y,\max}$ from (4.16) in (4.13) to determine $V_{M,\max}$ provides a safe set for $\bar{V}$, due to

$$\bar{F}_{y,\max} < \bar{D}_y \Rightarrow V_{M,\max} < V_{ss} \Rightarrow \bar{F}_x < \bar{F}_{x,ss}, \tag{4.17}$$

providing that the upper-bound $V_{M,\max}$ provides a feasible set. This set guarantees that all corners on the track can be driven while staying within the friction ellipse of the tires and without road departures. This directly implies that an increased look-ahead distance will directly increase the performance on the track due to the increased braking distance over the prediction horizon.

The weight on exceeding the feasible set is implemented via

$$W_{\bar{V}_M} = \begin{cases} W_{\bar{V}_{M0}}, & \text{if } \bar{V}_{M|k} \geq \bar{V}_{M,\max}. \\ 0, & \text{Otherwise.} \end{cases}, \tag{4.18}$$

which ensures that the penalty is only applied when $\bar{V}_{M|k}$ exceeds the bound, resulting in a safe trajectory over the prediction horizon.

Input and Input Rate

Using the principle of cascaded vehicle models, where each model has a different control goal, it is essential to scale the objective function accordingly. Since each model can be discretized with different step sizes, the weights W in the cost function are weighted by their respective model

step size Δs. Scaling the objective function with its respective step size enhances the principle of different control objectives over the prediction horizon. Using the product between the step size and the respective weight puts the focus more on the point-mass model due to its significant step size compared to the step size of the two-track and single-track models. The opposite can be said when the weight is divided by its respective step size.

For the cost on input J_U and change of input $J_{\Delta U}$, the scaling is performed by dividing its weight W by the respective step size Δs. Since the complex vehicle models are responsible for stabilizing the vehicle, changing its input and using input are penalized more. Moreover, it allows the point mass to explore varying inputs further away from the car. All three vehicle models combined consist of different inputs, separated into seven different terms for the input cost J_U

$$J_U = J_{\dot{\delta}} + J_{\dot{\tilde{\delta}}} + J_{\tilde{M}_{tv}} + J_{F_x} + J_{\tilde{F}_x} + J_{\bar{F}_x} + J_{\bar{F}_y}, \tag{4.19}$$

to promote the smoothness of the different control inputs. The costs $J_{\dot{\delta}}$ and $J_{\dot{\tilde{\delta}}}$ penalizes steering rate input for the single and two-track model, respectively, $J_{\tilde{M}_{tv}}$ penalizes the torque vectoring moment input in the single track model, J_{F_x}, $J_{\tilde{F}_x}$ and $J_{\bar{F}_x}$ penalizes the total longitudinal input for the two-track, single track and point-mass model, respectively, and $J_{\bar{F}_y}$ penalizes the lateral force input in the point-mass model. All the different costs are defined as

$$J_{\dot{\delta}} = \frac{W_{\dot{\delta}_0}}{\Delta s} \sum_{i=0}^{N-1} \dot{\delta}_{i|k}^2, \tag{4.20a}$$

$$J_{\dot{\tilde{\delta}}} = \frac{W_{\dot{\delta}_0}}{\Delta \tilde{s}} \sum_{z=0}^{H-1} \dot{\tilde{\delta}}_{z|k}^2, \tag{4.20b}$$

$$J_{\tilde{M}_{tv}} = \frac{W_{\tilde{M}_{tv}}}{\Delta \tilde{s}} \sum_{z=0}^{H-1} \tilde{M}_{tv,z|k}^2, \tag{4.20c}$$

$$J_{F_x} = \frac{W_{F_{x0}}}{\Delta s} \sum_{i=0}^{N-1} \sum_{l=1}^{4} F_{x,l,i|k}^2, \tag{4.20d}$$

$$J_{\tilde{F}_x} = \frac{W_{F_{x0}}}{\Delta \tilde{s}} \left(\sum_{z=0}^{H-1} \tilde{F}_{x,f,z|k}^2 + \sum_{z=0}^{H-1} \tilde{F}_{x,r,z|k}^2 \right), \tag{4.20e}$$

$$J_{\bar{F}_x} = \frac{W_{F_{x0}}}{\Delta \bar{s}} \sum_{j=0}^{M-1} \bar{F}_{x,j|k}^2, \tag{4.20f}$$

$$J_{\bar{F}_y} = \frac{W_{\bar{F}_y}}{\Delta \bar{s}} \sum_{j=0}^{M-1} \bar{F}_{x,j|k}^2, \tag{4.20g}$$

where for the steering rate and longitudinal force input the same weight is used for the different models, namely $W_{\dot{\delta}_0}$ and $W_{F_{x0}}$ respectively, to enhance a smooth input trajectory between the cascaded models.

The cost $J_{\Delta U}$ is similarly defined as J_U, namely

$$J_{\Delta U} = J_{\Delta \dot{\delta}} + J_{\Delta \dot{\tilde{\delta}}} + J_{\Delta \tilde{M}_{tv}} + J_{\Delta F_x} + J_{\Delta \tilde{F}_x} + J_{\Delta \bar{F}_x} + J_{\Delta \bar{F}_y}, \tag{4.21}$$

where

$$J_{\Delta\dot{\delta}} = \frac{W_{\Delta\dot{\delta}_0}}{\Delta s}\left(\sum_{i=0}^{N-2}(\dot{\delta}_{i+1|k} - \dot{\delta}_{i|k})^2\right), \tag{4.22a}$$

$$J_{\Delta\dot{\tilde{\delta}}} = \frac{W_{\Delta\dot{\delta}_0}}{\Delta \tilde{s}}\left(\sum_{z=0}^{H-2}(\dot{\tilde{\delta}}_{z+1|k} - \dot{\tilde{\delta}}_{z|k})^2\right), \tag{4.22b}$$

$$J_{\Delta\tilde{M}_{tv}} = \frac{W_{\Delta\tilde{M}_{tv}}}{\Delta \tilde{s}}\left(\sum_{z=0}^{H-2}(\tilde{M}_{tv,z+1|k} - \tilde{M}_{tv,z|k})^2\right), \tag{4.22c}$$

$$J_{\Delta F_x} = \frac{W_{\Delta F_{x0}}}{\Delta s}\left(\sum_{i=0}^{N-2}\sum_{l=1}^{4}(F_{x,l,i+1|k} - F_{x,l,i|k})^2\right), \tag{4.22d}$$

$$J_{\Delta\tilde{F}_x} = \frac{W_{\Delta F_{x0}}}{\Delta \tilde{s}}\left(\sum_{z=0}^{H-2}(\tilde{F}_{x,f,z+1|k} - \tilde{F}_{x,f,z|k})^2 + \sum_{z=0}^{H-2}(\tilde{F}_{x,r,z+1|k} - \tilde{F}_{x,r,z|k})^2\right), \tag{4.22e}$$

$$J_{\Delta\bar{F}_x} = \frac{W_{\Delta F_{x0}}}{\Delta \bar{s}}\left(\sum_{j=0}^{M-2}(\bar{F}_{x,j+1|k} - \bar{F}_{x,j|k})^2\right), \tag{4.22f}$$

$$J_{\Delta\bar{F}_y} = \frac{W_{\Delta\bar{F}_y}}{\Delta \bar{s}}\left(\sum_{j=0}^{M-2}(\bar{F}_{y,j+1|k} - \bar{F}_{y,j|k})^2\right), \tag{4.22g}$$

where the change in longitudinal force and steering rate also is penalized with the same weights $W_{\Delta F_{x0}}$ and $W_{\Delta\dot{\delta}0}$, respectively.

Error States

Since the vehicle models are formulated in the Frenet reference frame it allows the application of a quadratic cost on the error states. The position errors over the planning horizon are penalized via J_e, where tuning its weight can obtain the desired reference tracking behavior. The penalty on the error states is defined via two terms

$$J_e = J_{e_y} + J_{e_\psi}, \tag{4.23}$$

where

$$J_{e_y} = W_{e_y}\left(\Delta s\sum_{i=0}^{N} e_{y,i|k}^2 + \Delta\tilde{s}\sum_{z=0}^{H}\tilde{e}_{y,z|k}^2 + \Delta\bar{s}\sum_{j=0}^{M-1}\bar{e}_{y,j|k}^2 +\right), \tag{4.24a}$$

$$J_{e_\psi} = W_{e_\psi}\left(\Delta s\sum_{i=0}^{N} e_{\psi,i|k}^2 + \Delta\tilde{s}\sum_{z=0}^{H}\tilde{e}_{\psi,z|k}^2 + \Delta\bar{s}\sum_{j=0}^{M-1}(\bar{e}_{\psi,j|k}^2\right). \tag{4.24b}$$

The weights W_{e_y} and W_{e_ψ} are the common weights for all three vehicle models, which are scaled by multiplying the weight with the respective step size. Using the point mass to explore the trajectory has two risks. First, it is a lower-fidelity model with an increased distance between the solutions, resulting in a less accurate prediction. Second, since the point-mass model is responsible for path planning, where the reference trajectory can shift due to external disturbances, the risk of exceeding the track limits increases. Therefore, deviating from the reference path for the point mass is only possible at a higher cost, enforcing the NMPC to find a solution closer to the reference path the further it gets from the vehicle's position.

Vehicle Side Slip Angle

As the car is pushed towards the limits, there is a chance that the MPC tends towards a drifting solution, which is beyond the limits of vehicle handling as the tires operate in the fully saturated region. Since one of the control objectives is stabilizing the vehicle, a penalty on excessive vehicle side slip angle β is applied, stabilizing the car [40]. This cost is only applied when exceeding the point where gripping can not be guaranteed

$$\begin{aligned} J_\beta = W_{\beta_0} \sum_{i=0}^{N} \begin{cases} (|\beta_{i|k}| - \beta_{max})^2, & \text{if } |\beta_{i|k}| \geq \beta_{max} \\ 0, & \text{otherwise.} \end{cases} \\ + W_{\beta_0} \sum_{z=0}^{H} \begin{cases} (|\tilde{\beta}_{z|k}| - \beta_{max})^2, & \text{if } |\tilde{\beta}_{z|k}| \geq \beta_{max} \\ 0, & \text{otherwise.} \end{cases}, \end{aligned} \tag{4.25}$$

where β_{max} depends on many non-linear terms and usually is hard to predict, but a well-educated guess based on the results in [40] indicates that it is between $5° \sim 7.5°$. The vehicle side slip angle is determined via

$$\beta_{i|k} = \arctan\left(\frac{v_{y,i|k}}{v_{x,i|k}}\right) \approx \frac{v_{y,i|k}}{v_{x,i|k}}, \tag{4.26a}$$

$$\tilde{\beta}_{z|k} = \arctan\left(\frac{\tilde{v}_{y,z|k}}{\tilde{v}_{x,z|k}}\right) \approx \frac{\tilde{v}_{y,z|k}}{\tilde{v}_{x,z|k}}, \tag{4.26b}$$

where it is assumed that the vehicle side slip angle remains small.

Force Transition

The last cost term is the penalty on the transition of tire force between the different vehicle models. Due to the cascaded planning horizon, the dynamics change, where J_{tr} penalizes a non-smooth transition when the model changes. This penalty is defined as

$$J_{tr} = J_{F|\tilde{F}} + J_{\tilde{F}|\bar{F}}, \tag{4.27}$$

where $J_{F|\tilde{F}}$ is the penalty on the transition between the two-track and single-track vehicle model, and $J_{\tilde{F}|\bar{F}}$ the penalty on the transition between the single-track and point-mass model. This penalty term is an additional cost on the tire forces to enforce a smooth transition in tire forces, where $J_{F_x}, J_{\tilde{F}_x}, J_{\bar{F}_x},$ and $J_{\bar{F}_y}$ only penalize the input of a tire force and not the transition between them. The cost terms are defined as

$$J_{F|\tilde{F}} = W_{F|\tilde{F}} \left((\tilde{F}_{x,0|k} - \sum_{l=1}^{4} F_{x,l,N-1|k})^2 + (\tilde{F}_{y,f,0|k} + \tilde{F}_{y,r,0|k} - \sum_{l=1}^{4} F_{y,l,N|k})^2 \right), \tag{4.28a}$$

$$J_{\tilde{F}|\bar{F}} = W_{\tilde{F}|\bar{F}} \left((\bar{F}_{x,0|k} - \tilde{F}_{x,H-1|k})^2 + (\bar{F}_{y,0|k} - (\tilde{F}_{y,f,H|k} + \tilde{F}_{y,r,H|k}))^2 \right), \tag{4.28b}$$

where $W_{F|\tilde{F}}$ and $W_{\tilde{F}|\bar{F}}$ are the weights on the transition of tire forces between the two-track and single-track mode, and the single track model and the point mass model, respectively.

4.3 Constraints

After formulating all the different costs, the feasible domain of the NMPC is defined via a set of constraints. The constraints are formulated to enhance both practical and numerical feasibility. First, optimization along the centerline allows the NMPC to explore the track, but the vehicle should always stay on the track. To enforce safe driving on the track, the lateral error is constrained via the track's width so that the optimal solution always lies within track limits. Additionally, the

vehicle's limited steering capabilities should be considered when exploring the track. Therefore, both the steering angle and steering rate are constrained. Second, reverse driving should always be avoided to prevent numerical infeasibility. Therefore, the minimum longitudinal velocity and the heading error are constrained. These constraints provide that the optimal path ensures forward driving. Last, since a dynamic vehicle model is used in the NMPC, the feasible domain is also defined by nonlinear tire dynamics. In the single-track model, a torque vectoring moment is considered system input. The maximum torque vectoring moment the vehicle can produce is limited via the longitudinal peak tire force and is therefore constrained via tire and vehicle parameters. Furthermore, all the vehicle models use longitudinal force as system input, and the point-mass model also considers lateral force as a system input. To ensure that all forces acting on the system as a result of tire forces are within the friction ellipse of the tire, each model is constrained via tire constraints. In the remainder of this section, the formulation of the constraints is discussed in more detail.

State constraints

A constraint is set on the maximum and minimum lateral error to ensure that the vehicle remains on track. As the track width can differ, the constraint is a function of the progress made along the reference path, resulting in

$$e_{y,i|k,\min}(s_{i|k}) \leq e_{y,i|k} \leq e_{y,i|k,\max}(s_{i|k}) \quad \forall i \in \{0,...,N\}, \tag{4.29a}$$
$$\tilde{e}_{y,z|k,\min}(\tilde{s}_{z|k}) \leq \tilde{e}_{y,z|k} \leq \tilde{e}_{y,z|k,\max}(\tilde{s}_{z|k}) \quad \forall z \in \{0,...,H\}, \tag{4.29b}$$
$$\bar{e}_{y,j|k,\min}(\bar{s}_{j|k}) \leq \bar{e}_{y,j|k} \leq \bar{e}_{y,j|k,\max}(\bar{s}_{j|k}) \quad \forall j \in \{0,...,M\}, \tag{4.29c}$$

where the lateral error's lower and upper bound is defined by the track width, which is provided with the reference path by SLAM.

The heading error constraint is defined to avoid uncontrollable vehicle poses and infeasible optimization problems. Allowing for an extensive heading error results in more aggressive corner cutting when encountering a small cornering radius; therefore, it should not be restricted too much. However, an issue occurs when encountering a heading error of 90°, which indicates that the vehicle is perpendicular to the reference path. As the car is perpendicular to the path, the longitudinal velocity and the rear tires do not contribute to the minimization problem. Once the car surpasses the 90°, the vehicle is driving in reverse, and the rear tires want to generate an opposing longitudinal force. Notice that the front tires can still generate a longitudinal force in the path heading if the vehicle heading error does not surpass the limit $e_{\psi,\max} + \delta_{max}$. This situation has to be avoided since reverse driving is not allowed on the track. Therefore, the heading error is constrained via

$$-80\frac{\pi}{180} \leq e_{\psi,i|k} \leq 80\frac{\pi}{180} \quad \forall i \in \{0,...,N\}, \tag{4.30a}$$
$$-80\frac{\pi}{180} \leq \tilde{e}_{\psi,z|k} \leq 80\frac{\pi}{180} \quad \forall z \in \{0,...,H\}, \tag{4.30b}$$
$$-80\frac{\pi}{180} \leq \bar{e}_{\psi,j|k} \leq 80\frac{\pi}{180} \quad \forall j \in \{0,...,M\}, \tag{4.30c}$$

where the limit is set on 80° to also account for any measurement error that can occur during the position estimation, as discussed in Chapter 3. Additionally, as mentioned above, to prevent reverse driving, a lower bound is set on the longitudinal velocity

$$0.1 \leq v_{x,i|k} \quad \forall i \in \{0,...,N\}, \tag{4.31a}$$
$$0.1 \leq \tilde{v}_{x,z|k} \quad \forall z \in \{0,...,H\}, \tag{4.31b}$$
$$0.1 \leq \bar{V}_{j|k} \quad \forall j \in \{0,...,M\}. \tag{4.31c}$$

The constraint on the longitudinal velocity also prevents the vehicle from reaching the model singularity at $v_x = 0$.

The last constrained state is the steering angle, which is limited via mechanical constraints due to the limited steering capacity of the vehicle,

$$\delta_{min} \leq \delta_{i|k} \leq \delta_{max} \quad \forall i \in \{0, ..., N\}, \tag{4.32a}$$

$$\tilde{\delta}_{min} \leq \tilde{\delta}_{z|k} \leq \tilde{\delta}_{max} \quad \forall z \in \{0, ..., H\}. \tag{4.32b}$$

Input constraints

The two-track and single-track model use steering rate as input, which is limited by the steering actuator. Therefore, the steering rate is constrained via

$$\dot{\delta}_{min} \leq \dot{\delta}_{i|k} \leq \dot{\delta}_{max} \quad \forall i \in \{0, ..., N-1\}, \tag{4.33a}$$

$$\dot{\tilde{\delta}}_{min} \leq \dot{\tilde{\delta}}_{z|k} \leq \dot{\tilde{\delta}}_{max} \quad \forall z \in \{0, ..., H-1\}. \tag{4.33b}$$

This constraint depends on the steering actuator that is being used. As the specification of the steering actuator does not always provide the required information, and the steering rate relies on highly nonlinear tire dynamics, the maximum steering rate is determined based on testing data from University Racing Eindhoven.

To model the torque vectoring moment on a single-track model, an additional moment M_{tv} about the CG is implemented, which should not exceed the maximum moment a two-track model can generate via torque vectoring. Therefore, M_{tv} must be constrained based on the maximum moment the tires in a two-track model can generate. To estimate the maximum moment, the sum of the moments about the CG has to be derived, assuming that the tires do not generate any lateral force. This assumption is made since the same lateral tire forces can be generated in both vehicle models, where M_{tv} purely focuses on the additional moment due to braking and accelerating the four tires. The most extreme case is displayed in Figure 4.2, where the vehicle is rotating counter-clockwise, purely relying on torque vectoring. Deriving the sum of all moments about the CG results in

$$\sum M_z = \frac{w_f}{2}(F_{x,1} + F_{x,4}) + \frac{w_r}{2}(F_{x,2} + F_{x,3}). \tag{4.34}$$

The resulting moment can be maximized when all four tires are operating at the limits, in which the longitudinal tire force equals

$$F_{x,l,\max} = \mu_x F_{z,l}, \tag{4.35}$$

substituting this into (4.34) results in

$$\begin{aligned} M_{z,\max} &= \frac{w_f}{2}(F_{x,1,\max} + F_{x,4,\max}) + \frac{w_r}{2}(F_{x,2,\max} + F_{x,3,\max}), \\ &= \frac{w_f}{2}(\mu_x F_{z,1} + \mu_x F_{z,4}) + \frac{w_r}{2}(\mu_x F_{z,2} + \mu_x F_{z,3}). \end{aligned} \tag{4.36}$$

This expression can be further simplified when only static weight distribution is considered, resulting in

$$M_{z,\max} = \frac{\mu_x F_z}{2}(w_f w_{dis} + w_r(1 - w_{dis})), \tag{4.37}$$

where w_{dis} is the weight distribution of the vehicle, and F_z is the gravitational force acting on the CG. Deriving the minimum moment $M_{z,\min}$ yields the same result as $M_{z,\max}$, resulting in a clockwise rotation of the vehicle. The resulting constraint is formulated as

$$\tilde{M}_{tv,\min} \leq \tilde{M}_{tv,z|k} \leq \tilde{M}_{tv,\max} \quad \forall z \in \{0, ..., H-1\}, \tag{4.38}$$

where

$$\tilde{M}_{tv,\min} = -\frac{\mu_x F_z}{2}\left(w_f w_{dis} + w_r(1 - w_{dis})\right), \tag{4.39a}$$

$$\tilde{M}_{tv,\max} = \frac{\mu_x F_z}{2}\left(w_f w_{dis} + w_r(1 - w_{dis})\right). \tag{4.39b}$$

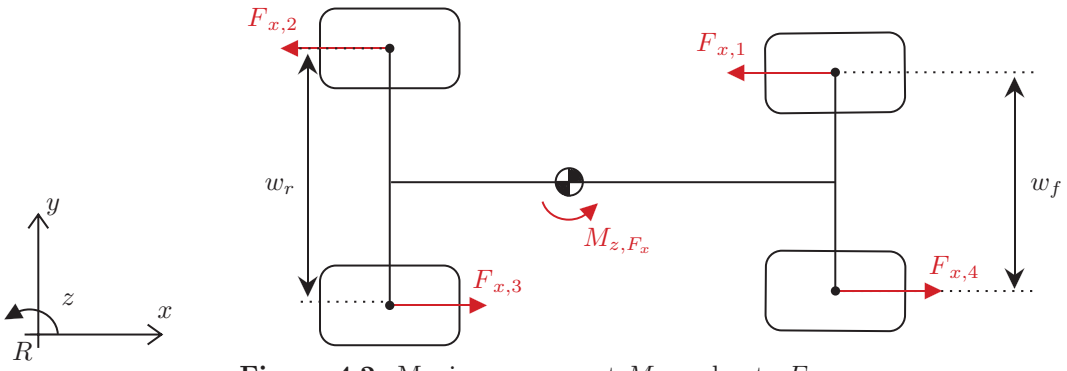

Figure 4.2: Maximum moment M_{z,F_x} due to $F_{x,l}$.

Tire constraints

All the vehicle models presented in Section 2.2 use the simplified Pacejka tire model

$$F_y = D_y \sin\left(C_y \arctan\left(B_y \alpha\right)\right), \tag{4.40}$$

to describe the lateral tire dynamics. The longitudinal tire force F_x is used as model input. Since the tire forces are incorporated in such a manner, the forces are decoupled. However, for the optimal solution to be feasible, the tire forces must be within the tire friction ellipse, which can be formulated as

$$\left(\frac{F_x}{D_x}\right)^2 + \left(\frac{F_y}{D_y}\right)^2 \leq 1. \tag{4.41}$$

The friction ellipse is applied as a constraint to couple the lateral and longitudinal tire force, where the constraints for the two-track model are defined as

$$\left(\frac{F_{x,1}}{D_{x,1}}\right)^2 + \left(\frac{F_{y,1}}{D_{y,1}}\right)^2 - 1 \leq 0, \tag{4.42a}$$

$$\left(\frac{F_{x,2}}{D_{x,2}}\right)^2 + \left(\frac{F_{y,2}}{D_{y,2}}\right)^2 - 1 \leq 0, \tag{4.42b}$$

$$\left(\frac{F_{x,3}}{D_{x,3}}\right)^2 + \left(\frac{F_{y,3}}{D_{y,3}}\right)^2 - 1 \leq 0, \tag{4.42c}$$

$$\left(\frac{F_{x,4}}{D_{x,4}}\right)^2 + \left(\frac{F_{y,4}}{D_{y,4}}\right)^2 - 1 \leq 0, \tag{4.42d}$$

where $D_{x,l} = \mu_x F_{z,l}$ and $D_{y,l} = \mu_y F_{z,l}$, $l \in \{1, 2, 3, 4\}$ and $F_{z,l}$ representing the vertical load acting on the respective tire. However, the friction ellipse is a nonlinear constraint, resulting in a long computation time, especially when considering four tires. Therefore, to reduce the computation time, the nonlinear inequality constraints in (4.42) are replaced with several linear inequality constraints.

The goal is to estimate an ellipse with several linear functions, which in the case of the friction ellipse is defined as

$$F_{y,l} = aF_{x,l} + b, \tag{4.43}$$

where a and b represent the linear function's slope and intercept, respectively. Tires are considered highly nonlinear, where the ellipse of a tire can have varying eccentricity. Depending on the tire characteristics, the required number of linear functions can vary to get a good estimate. The number of linear functions to approximate the ellipse can be selected based on the number of intersection points. An intersection point represents the coordinates where the linear function intersects with the friction ellipse.

The minimum number of linear functions to estimate the friction ellipse is four since each quadrant requires to be constrained for the optimization problem to be well-constrained. These four linear functions are defined based on the maximum tire force in both lateral and longitudinal direction, resulting in the intersection points $\{(D_{x,l},0),(-D_{x,l},0),(0,D_{y,l}),(0,-D_{y,l})\},\ l\in\{1,2,3,4\}$. Figure 4.3 shows the estimation based on these four points. These linear functions can be reformulated as linear inequality constraints, depending on which quadrant the constraint is active, resulting in

$$h_i(F_{x,l},F_{y,l})=\begin{cases}-F_{y,l}+(aF_{x,l}+b)\leq 0 & \text{if } (F_{x,l},F_{y,l})\in\{I,II\},\\ F_{y,l}-(aF_{x,l}+b)\leq 0 & \text{if } (F_{x,l},F_{y,l})\in\{III,IV\},\end{cases} \tag{4.44}$$

where $i\in\{1,...,i_n\}$ and

$$i_n=4(1+P_n), \tag{4.45}$$

is the total number of inequality constraints to approximate (4.41), and P_n is the added number of intersection points per quadrant. The coordinates of the intersection points are determined by dividing each quadrant into P_n+1 equal-sized segments, where the intersection point is located on the line splitting two segments and intersecting with the ellipse. Increasing P_n provides a better estimate, increasing the car's performance since it enlarges the feasible area of the tire constraints, resulting in larger tire forces. In Figure 4.4, the estimation for $P_n=2$ is shown, which shows the enlarged feasible domain for both F_x and F_y, comparing it with the feasible domain shown in Figure 4.3. The results are compared in Chapter 5 based on computation time and vehicle performance.

The tire constraints are also applicable to the single track and point-mass model. So similarly to (4.42) can the friction ellipse for each tire of the single track be determined, now considering the friction ellipse of the axle instead of the tire

$$\left(\frac{\tilde{F}_{x,f}}{\tilde{D}_{x,f}}\right)^2+\left(\frac{\tilde{F}_{y,f}}{\tilde{D}_{y,f}}\right)^2-1\leq 0, \tag{4.46a}$$

$$\left(\frac{\tilde{F}_{x,r}}{\tilde{D}_{x,r}}\right)^2+\left(\frac{\tilde{F}_{y,r}}{\tilde{D}_{y,r}}\right)^2-1\leq 0. \tag{4.46b}$$

Since the single-track model only considers a single force at the CG, the distribution between the front and rear axle has to be defined. The grip of the axle scales with the vertical load acting on the axle since the maximum possible grip is defined as $\tilde{D}_y=\mu_y\tilde{F}_{z,l},\ l\in\{f,r\}$, similarly for $\tilde{D}_x$. Therefore, the fixed longitudinal force distribution between the front and rear axle is defined by the weight distribution, resulting in

$$\tilde{F}_{x,f}=w_{dis}\tilde{F}_x, \tag{4.47a}$$

$$\tilde{F}_{x,r}=(1-w_{dis})\tilde{F}_x. \tag{4.47b}$$

By substituting (4.47) into (4.46) the tire constraints for the single track model are obtained

$$\left(\frac{w_{dis}\tilde{F}_x}{\tilde{D}_{x,f}}\right)^2+\left(\frac{\tilde{F}_{y,f}}{\tilde{D}_{y,f}}\right)^2-1\leq 0, \tag{4.48a}$$

$$\left(\frac{(1-w_{dis})\tilde{F}_x}{\tilde{D}_{x,r}}\right)^2+\left(\frac{\tilde{F}_{y,r}}{\tilde{D}_{y,r}}\right)^2-1\leq 0. \tag{4.48b}$$

These two ellipses are also approximated via several linear constraints, where the four default intersection points are defined as $\{(\tilde{D}_{x,l},0),(-\tilde{D}_{x,l},0),(0,\tilde{D}_{y,l}),(0,-\tilde{D}_{y,l})\},\ l\in\{f,r\}$.

At last, the friction ellipse of the point mass is a constraint. Since the point-mass model uses both the lateral and longitudinal forces as system inputs, the ellipse can be determined as relatively

straightforward as

$$\left(\frac{\bar{F}_x}{\bar{D}_x}\right)^2 + \left(\frac{\bar{F}_y}{\bar{D}_y}\right)^2 - 1 \leq 0, \tag{4.49}$$

which also is approximated via linear constraints. The default intersection points for the linear approximation are defined as $\{(\bar{D}_x, 0), (-\bar{D}_x, 0), (0, \bar{D}_y), (0, -\bar{D}_y)\}$.

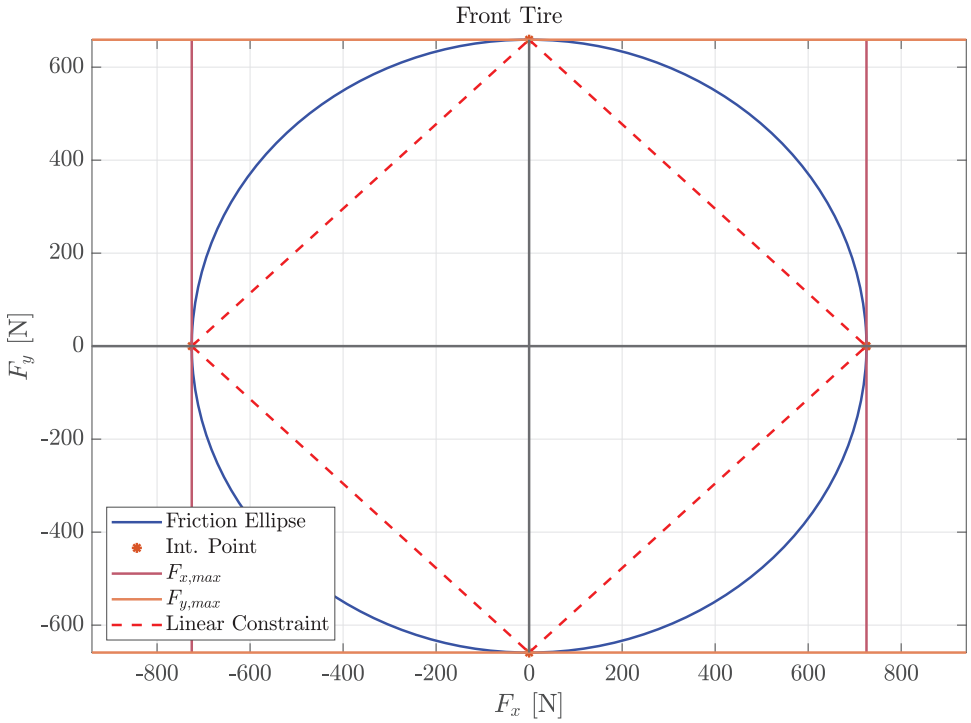

Figure 4.3: The friction ellipse of the front tire with linear tire constraints, based on $P_n = 0$.

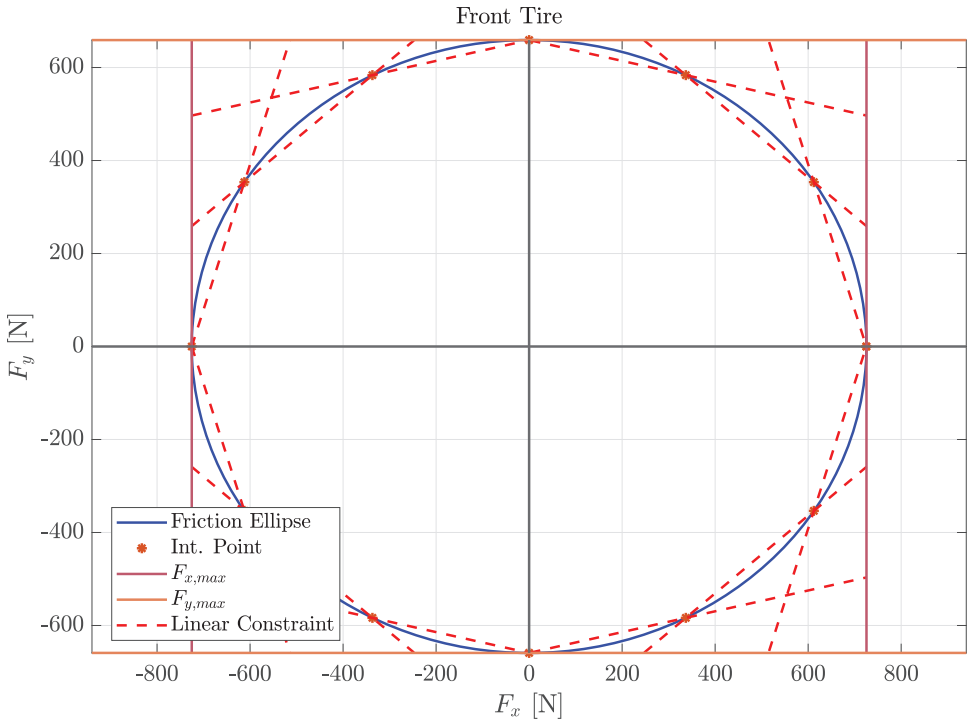

Figure 4.4: The friction ellipse of the front tire with linear tire constraints, based on $P_n = 2$.

4.4 Variable Step Size

The overall problem formulation, the cost to be minimized, and the required set of constraints are all defined in the previous sections. These three sections describe the overall control structure to achieve an online optimization-based motion planning system. However, before solving the NMPC, the different vehicle models are discretized in the spatial domain using the curvilinear approach. Therefore, this section focuses on the discretization of the different vehicle models.

Discretization of the different vehicle models can be done by applying a fixed step size defined in the spatial domain, providing a constant path geometry for each stage in the planning horizon. However, a disadvantage of discretization in the spatial domain is the increased stiffness of the vehicle dynamics with lower vehicle speed, which require a smaller step size to prevent numerical instability. Therefore, a sufficiently small step size is required to guarantee the feasibility of the control problem, with the disadvantage of reducing the look-ahead distance for the same horizon length. Finding the balance between numerical stability and a sufficiently large look-ahead distance is proven difficult due to the difficulty in formulating model stiffness. Therefore, a simulation study is done to determine the accuracy of the discretized vehicle model for different step sizes.

Instead of using a fixed step discretization in the spatial domain, the system's step size is determined based on the current vehicle speed and a fixed time constant h. The resulting step size in the spatial domain is then determined via

$$\Delta s_k = h\, \dot{s}_{0|k}, \tag{4.50a}$$
$$\Delta \tilde{s}_k = \tilde{h}\, \dot{\tilde{s}}_{0|k}, \tag{4.50b}$$
$$\Delta \bar{s}_k = \bar{h}\, \dot{\bar{s}}_{0|k}. \tag{4.50c}$$

An advantage of applying a variable step size is numerical stability at low vehicle speeds due to a smaller step size but an increased look-ahead distance once the vehicle velocity increases.

The disadvantage of such an application is that by planning further ahead over the horizon, the model will accelerate and decelerate, resulting in a velocity profile that is not representative of the velocity at the start of the respective horizon. This disadvantage is mainly applicable to the point-mass model, as these dynamics are less-stiff, it allows for larger discretization step sizes and therefore aggravates this effect.

The two-track model described in section 2.2 is used to analyze the two-track and single-track models. Since the single-track consists of the same system dynamics as the two-track model but considering fewer system inputs, the stiffness is assumed to be equally stiff.

As a reference, the same input is used while simulating the model response with an ODE45 solver from MATLAB. The ODE45 solver from MATLAB is chosen as a reference since it applies variable step sizes to increase model accuracy when states are changing rapidly. ODE45 is selected over the implicit ODE15s, which is considered better for solving stiff dynamics. However, ODE45 provides higher accuracy with the cost of slow computational performance, which is not considered an issue during this analysis [41].

To determine the accuracy of the discretized vehicle model, which is discretized with the RK2 discretization, the error between the RK2 discretization and the output from the ODE45 solver is determined via

$$e = \sqrt{\sum_{k=0}^{L} \left(x_{\mathrm{RK2},k} - x_{\mathrm{ODE45},k}\right)^2}, \tag{4.51}$$

where $x_{\mathrm{RK2},k}$ and $x_{\mathrm{ODE45},k}$ represent the state output from the RK2 and ODE45 solver output at time k, respectively, and $L = \frac{s_{\mathrm{end}}}{\Delta s}$. Note that the absolute values of the states are used since numerical instability also evolves in oscillations around the true solution. These oscillations cause the errors to cancel each other out, resulting in an inaccurate estimation of the error.

At first, the simulation is performed for zero steering input to examine the influence of different time constants h and initial velocities $v_{x|0}$ while driving a straight line of 30 meters. The force input at the four tires remains constant, where the front tires combined deliver 130 N and the rear tires combined 200 N, resulting in a force distribution that is approximately equal to the weight distribution of the vehicle. The results in Figure 4.5 show the resulting error for an $h \in \{0.001, 0.002, ..., 0.04\}$ and $v_{x|0} \in \{1.0, 1.1, ..., 10.0\}$, where the error for each state can be evaluated in detail in Appendix A.

The results clearly show the model's stiffness for low vehicle velocities, indicating that the discrete model does not model the dynamics well for $v_{x|0}$ below 2 m/s for any h. On the other hand, for all the velocities above 2 m/s, there is an h which does provide an equally good estimate as the ODE45 solver. However, this error does not incorporate the model's fast yaw dynamics and requires an additional simulation study.

The second comparison does consider steering input to excite the fast yaw dynamics. As steering rate input, we use

$$\dot{\delta}_{in} = 0.15 \sin\left(s_k \frac{2\pi}{15}\right), \tag{4.52}$$

resulting in a left-hand turn where the steering angle is zero at the start and the beginning.

In Figure 4.6 the result of this simulation is shown, indicating that the error increased for an increased step size h at lower velocities. This is in line with the expectations, as the fast yaw dynamics excite the singularity in the tire dynamics model for α. When considering both results, it can be concluded that the accuracy of the discretization for the region $h \in \{0.01, ..., 0.02\}$ and $v_{x|0} \in \{3, ..., 5\}$ shows small to no error compared with the ODE45 solver. Therefore, the initial longitudinal velocity for the NMPC is set to at least 3 m/s when the vehicle is driving more slowly, which is only expected at the start and finish. A time constant of $h = 0.02s$ can be applied for a

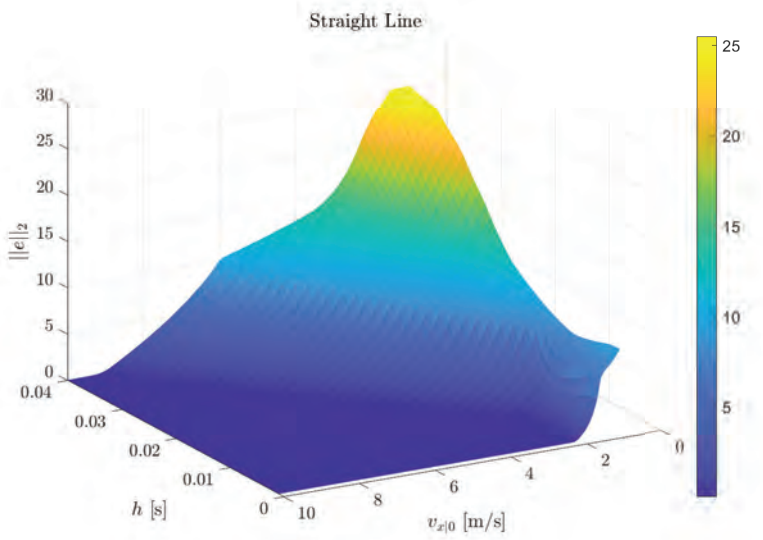

Figure 4.5: The error between RK2 and ODE45 for straight-line behavior using a two-track model.

minimum velocity of 3 m/s, maximizing the look-ahead distance while maintaining the required accuracy.

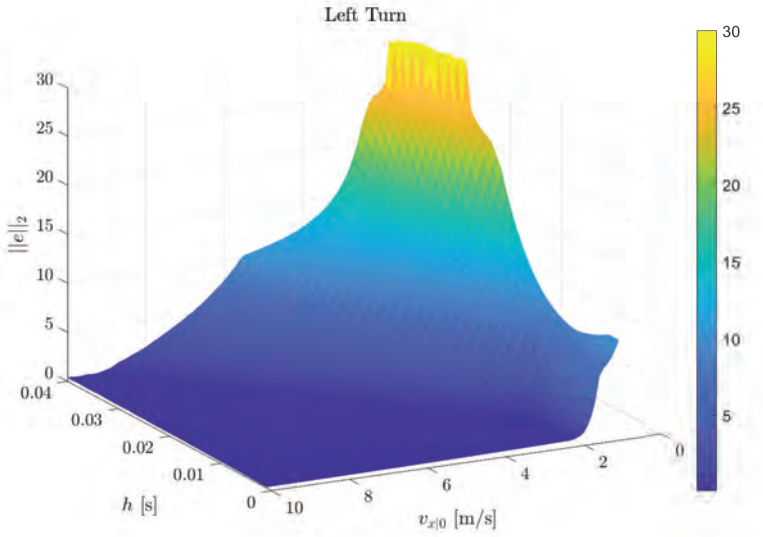

Figure 4.6: The error between RK2 and ODE45 for left turn behavior using a two-track model.

The same simulation study is performed for the point-mass model, where the output is compared for an $\bar{h} \in \{0.001, 0.002, ..., 0.3\}$ and $\bar{V}_0 \in \{1.0, 1.1, ..., 10.0\}$. The same force as the two-track model simulations is applied for the longitudinal force input, namely $\bar{F}_x = 330$ N. The lateral force input is set to zero for the first simulation to analyze the straight-line behavior, after which the lateral force is defined via a sinusoidal input, but now with a force amplitude rather than steering rate amplitude, resulting in

$$\bar{F}_y = 300 \sin\left(\bar{s}_k \frac{2\pi}{15}\right). \tag{4.53}$$

In Figure 4.7 and Figure 4.8, the results for the point mass are shown. The straight line behavior

shows a minimal error and only shows a peak for low velocity and small step sizes. The increase in error for small step sizes is also observed in the other simulations and is caused by a round-off error. In Figure 4.9, the round-off and discretization errors are plotted as a function of the step size h. Choosing a small step size increases accuracy due to a decreasing discretization error, but selecting a small step size will excite a round-off error. Based on all the results, the round-off error occurs for $h < 0.002$ and decreases with increasing velocity.

Concluding from the point-mass results, and considering that the initial vehicle velocity for the two-track model is lower bounded by 3 m/s, a time constant of $\bar{h} = 0.2$s is chosen to have good accuracy and enlarge the look-ahead distance significantly.

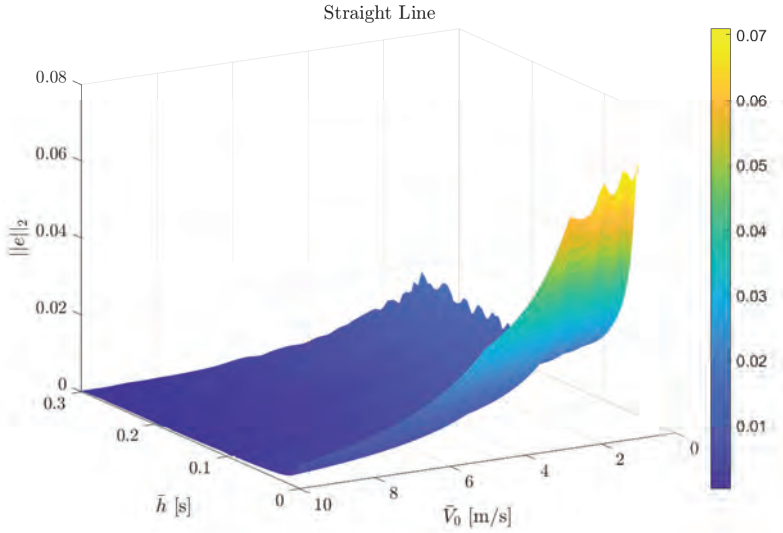

Figure 4.7: The error between RK2 and ODE45 for straight-line behavior using a point mass.

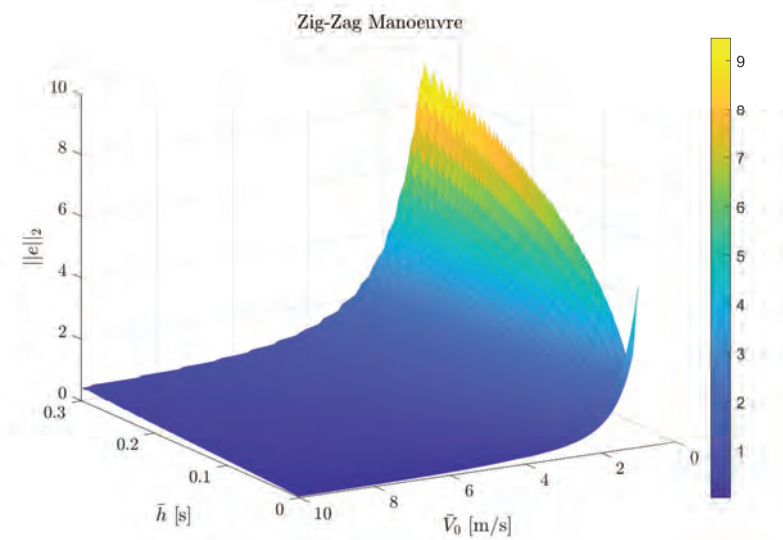

Figure 4.8: The error between RK2 and ODE45 for the zig-zag maneuver using a point mass.

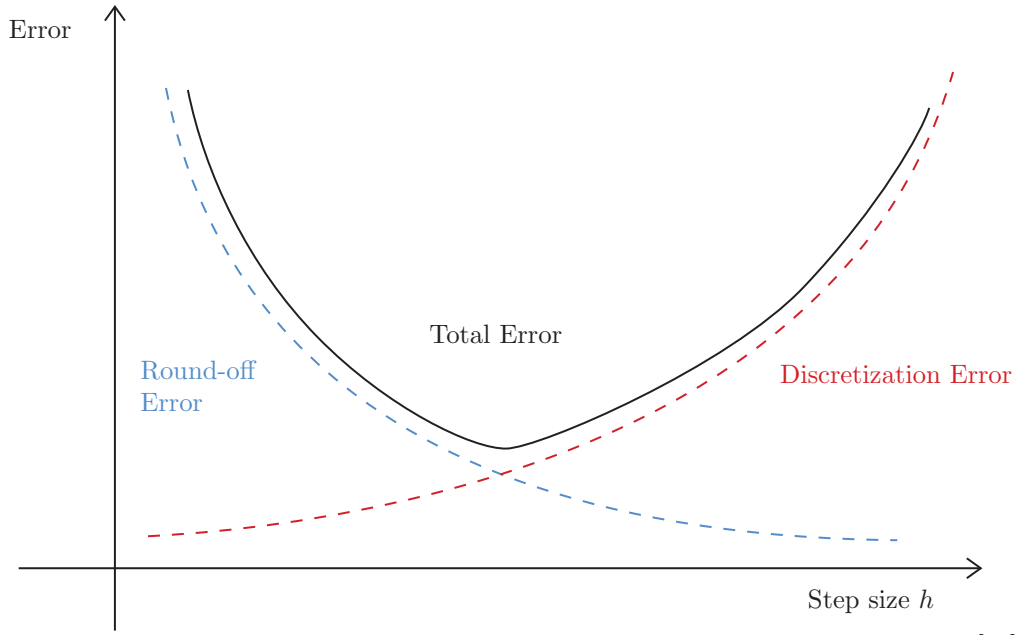

Figure 4.9: The discretization and round-off error as a function of the step size h [42].

4.5 Observations and Conclusions

In this chapter, the NLP to solve the online optimization problem has been proposed. The principle of a cascaded vehicle model is introduced to find the compromise between model complexity and computation time. To this end, the vehicle models are formulated using the curvilinear reference frame, allowing to formulate time as a function of the vehicle states and the reference path. A mapping is found between a two-track, single-track, and point-mass model, each serving its own purpose. The two-track and single-track model stabilize the vehicle, while the point-mass model explores the track at a low computational cost.

The resulting cascaded model is used in an NMPC formulation, where the primary objective is to minimize time at the end of the point-mass horizon. To enhance the feasibility of the optimization problem, the horizon is said to be on the reference path, while the vehicle's velocity is below a so-called horizon velocity which ensures that steady-state cornering can be achieved for the smallest turn on the track. This horizon velocity limits the potential of the NMPC by limiting the velocity but avoids the dependency on offline tools to determine a safe velocity at each point on the track. It is crucial to not depend on pre-knowledge of the track layout, as formula student requires the autonomous system to determine the track layout while driving.

Additionally to minimizing time, several costs are introduced to ensure the durability of the actuators, increase the smoothness between the cascaded model, and penalize undesired vehicle behavior. The different costs are scaled based on the respective discretization step, while the weight on the costs is the same for the different vehicle models. Furthermore, various hard constraints are proposed, guaranteeing that mechanical constraints, such as the steering bounds, are respected and avoiding potential damage to the car. The tire constraints are also modeled via hard constraints, where each friction ellipse is scaled with the respective mass acting on the CG, axle, or tire. Using the non-linear tire model requires access to detailed tire parameters, which can not always be guaranteed. However, the non-linear friction ellipse is essential to push the car to the limits since linear tire dynamics are inaccurate when undergoing excessive tire side slip angles.

The influence on computation time and performance using several linear constraints to estimate the friction ellipse is explored. Due to the quadratic nature of the friction ellipse, long computation times might occur, especially when considering a two-track model, which requires a hard constraint

for each tire. The approximation of the friction ellipse is made via several linear constraints, which potentially are computationally more efficient to solve. To get a reasonable estimate of the ellipse, several constraints are required, which is considered a disadvantage, as the required number of linear constraints might cause an increase in computation time.

At last, a simulation study is performed to determine the maximum discretization step size, which is required to construct an accurate prediction horizon. Applying a variable step size in the spatial domain increases numerical stability for low velocities, while an increased look-ahead distance can be achieved when driving faster, directly resulting in a reduction in lap time.

The proposed controller, including all costs and constraints, is validated in the next chapter via simulation.

Chapter 5

Results

The performance of the NMPC, presented in the previous chapter, is tested via different simulations. This chapter discusses various results that are obtained due to these simulations. All the simulations are performed in MATLAB 2021b [43], using a computer with an AMD Ryzen 9 5900x processor. Also, the NLP is programmed via the open-source tool for nonlinear optimization CasADi [44].

Obtaining results with the suggested problem formulation from Chapter 4 was not possible due an infeasible control problem. Therefore, based on these infeasible results adjustments are made to the original NMPC, increasing the robustness of the system and avoiding infeasible control problems. Consequently, with the suggested adjustments a simulation study is done to compare the performance of a cascaded vehicle model with a single two-track model. Finally, the application of linear tire constraints is explored via simulations. These results are then compared with quadratic tire constraints via lap time and computation time.

5.1 Feasibility

As mentioned in the control challenges, NLP's feasibility is a challenge to achieve for complex problems. For example, the NMPC in Chapter 4 introduced several hard inequality constraints, such as the steering system limits, tire friction ellipse, and heading and lateral error constraints. Hard inequality constraints should be implemented carefully since these limit the numerical feasibility of the NLP. Not only can hard constraints result in infeasible control problems, but also poorly defined terminal states and undesired behavior that is not penalized.

Different simulations are performed with the cascaded vehicle model while driving the Formula Student German (FSG) Driverless track of 2019, shown in Figure 5.1. This race track includes nine corners and has a centerline length of 382.7 m. The NLP is solved for the vehicle parameters in Table 5.1 and the optimization variables in Table 5.2. However, due to a poorly formulated NMPC no feasible solution could be found for different horizon configurations. These problems were due to three different reasons, namely the formulation of the tire constraints, saturation of the tires and numerical inaccuracy in the lateral error. This section suggests adjustments to the originally formulated NMPC with the main focus on achieving feasibility, which is crucial to provide control input to the vehicle persistently.

Tire Constraints

The cascaded vehicle model provides an increased look-ahead distance due to the simplified dynamics of a point mass with a significantly larger discretization. Therefore, as the terminal state

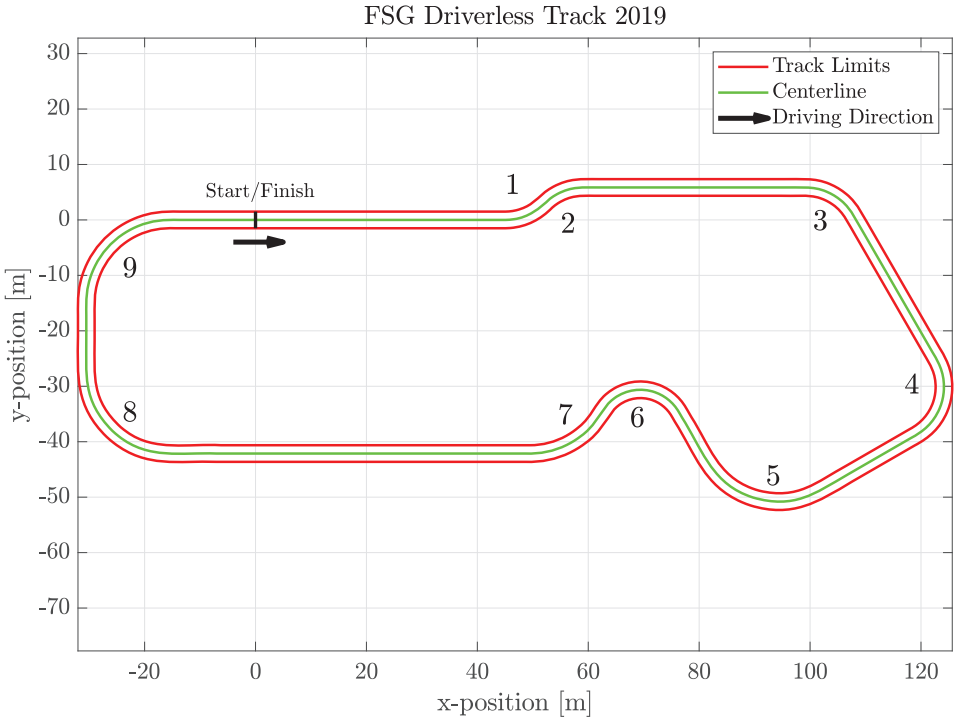

Figure 5.1: Layout FSG driverless track.

is further ahead, the NMPC allows for increased vehicle velocity and also anticipates better on the track's layout. However, the cascaded vehicle model introduces an infeasible control problem caused by each model's difference in potential grip.

The cause of this problem can be seen in Figure 5.2, where the resulting longitudinal and lateral force, acting at the CG, are shown over a cascaded prediction horizon. It is observed that the longitudinal and lateral forces act against each other for the different vehicle models, where for example the point mass wants to accelerate while the two-track model is still heavily braking. These results indicate that the difference in potential grip between the cascaded vehicle model is too large. Moreover, the two-track and single-track models are incapable of following the optimal trajectory of the point mass model.

In Section 4.3 the tire constraints are formulated, only considering the weight of the respective vehicle model to scale the friction ellipses. However, when cascading different vehicle models, each model should consider the constraints of the model it is cascaded with to enhance feasibility. Therefore, are the tire constraints from Section 4.3 reformulated to reduce the difference in potential grip between the cascaded vehicle models. At first, the point-mass longitudinal and lateral force inputs are distributed between the front and rear axle via the vehicle's weight distribution. Based on these assumptions, (4.49) is reformulated as

$$\left(\frac{w_{\text{dis}}\bar{F}_y}{\tilde{D}_{y,f}}\right)^2 + \left(\frac{w_{\text{dis}}\bar{F}_x}{\tilde{D}_{x,f}}\right)^2 - 1 \leq 0, \tag{5.1a}$$

$$\left(\frac{(1-w_{\text{dis}})\bar{F}_y}{\tilde{D}_{y,r}}\right)^2 + \left(\frac{(1-w_{\text{dis}})\bar{F}_x}{\tilde{D}_{x,r}}\right)^2 - 1 \leq 0, \tag{5.1b}$$

Similarly is done for the single-track model, where the forces acting on the system are assumed to be generated by four tires. An equal distribution between the left and rear tires can be assumed

Table 5.1: Vehicle parameters.

Parameter	Symbol	Value	Unit
Vehicle Mass	m	192	kg
Yaw moment of inertia	J	82	$kg\,m^2$
Distance front axle to CG	l_f	0.88	m
Distance rear axle to CG	l_r	0.64	m
Track width front	w_f	1.18	m
Track width rear	w_r	1.39	m
Weight Distribution	w_{dis}	0.4220	-
Rolling friction coefficient	f_r	0.072	-
Aerodynamic drag coefficient	C_d	1.23	$N\,(m/s)^{-2}$
Lateral tire friction coefficient	μ_y	1.4471	-
Longitudinal tire friction coefficient	μ_x	1.5930	-
Steering angle limits	$\delta_{min,max}$	± 24	deg
Steering rate limits	$\dot{\delta}_{min,max}$	± 22.1	$deg\,s^{-1}$

for the longitudinal forces based on the force acting on each axle. As a result, (4.48) is replaced with

$$\left(\frac{0.5\tilde{F}_{x,f}}{D_{x,1}}\right)^2 + \left(\frac{\tilde{F}_{y,1}}{D_{y,1}}\right)^2 - 1 \leq 0, \tag{5.2a}$$

$$\left(\frac{0.5\tilde{F}_{x,f}}{D_{x,2}}\right)^2 + \left(\frac{\tilde{F}_{y,1}}{D_{y,2}}\right)^2 - 1 \leq 0, \tag{5.2b}$$

$$\left(\frac{0.5\tilde{F}_{x,r}}{D_{x,3}}\right)^2 + \left(\frac{\tilde{F}_{y,3}}{D_{y,3}}\right)^2 - 1 \leq 0, \tag{5.2c}$$

$$\left(\frac{0.5\tilde{F}_{x,r}}{D_{x,4}}\right)^2 + \left(\frac{\tilde{F}_{y,4}}{D_{y,4}}\right)^2 - 1 \leq 0. \tag{5.2d}$$

Notice that the friction constraints are now scaled via the corresponding tire peak force $D_{x,i},\ D_{y,i}\,\forall\, i \in \{1,2,3,4\}$.

When assuming that the single-track model has four tires, the lateral tire force is also affected by the rotational velocity of the vehicle. Therefore, the lateral tire forces in (5.2) are determined via the tire side slip angle calculation for two-track models, which can be calculated based on the vehicle states of the single-track model via

$$\tilde{\alpha}_1 = \tilde{\delta} - \arctan\left(\frac{\tilde{v}_{y,1}}{\tilde{v}_{x,1}}\right), \tag{5.3a}$$

$$\tilde{\alpha}_2 = -\arctan\left(\frac{\tilde{v}_{y,2}}{\tilde{v}_{x,2}}\right), \tag{5.3b}$$

$$\tilde{\alpha}_3 = -\arctan\left(\frac{\tilde{v}_{y,3}}{\tilde{v}_{x,3}}\right), \tag{5.3c}$$

$$\tilde{\alpha}_4 = \tilde{\delta} - \arctan\left(\frac{\tilde{v}_{y,4}}{\tilde{v}_{x,4}}\right), \tag{5.3d}$$

Table 5.2: Optimization parameters.

Parameter	Symbol	Value	Unit
Horizon length TTM	N	10	–
Horizon length STM	H	10	–
Horizon length PMM	M	40	–
Terminal lateral error	$W_{\bar{e}_y,M}$	0.35	–
Terminal heading error	$W_{\bar{e}_\psi,M}$	0.85	–
Terminal velocity	$W_{\bar{V}_{M0}}$	0.3	–
Lateral error	W_{e_y}	0.001	–
Heading error	W_{e_ψ}	0.05	–
Steering rate	$W_{\dot{\delta}_0}$	0.015	–
Slew rate steering rate	$W_{\Delta\dot{\delta}_0}$	0.005	–
Longitudinal force	$W_{F_{x0}}$	0.01	–
Slew rate longitudinal force	$W_{\Delta F_{x0}}$	0.01	–
Vehicle side slip angle	W_{β_0}	0.09	–
Torque vectoring moment STM	$W_{\tilde{M}_{tv}}$	0.04	–
Slew rate TV moment STM	$W_{\Delta\tilde{M}_{tv}}$	0.02	–
Lateral force PPM	$W_{\bar{F}_y}$	0.006	–
Slew rate lateral force PMM	$W_{\Delta\bar{F}_y}$	0.01	–
Force transition TTM to STM	$W_{F\|\tilde{F}}$	0.001	–
Force transition STM to PMM	$W_{\tilde{F}\|\bar{F}}$	0.001	–

where

$$\tilde{v}_{x,1} = \tilde{v}_x - \frac{w_f}{2}\tilde{\omega}, \quad \tilde{v}_{y,1} = \tilde{v}_y + l_f\tilde{\omega}, \tag{5.4a}$$

$$\tilde{v}_{x,2} = \tilde{v}_x - \frac{w_r}{2}\tilde{\omega}, \quad \tilde{v}_{y,2} = \tilde{v}_y - l_r\tilde{\omega}, \tag{5.4b}$$

$$\tilde{v}_{x,3} = \tilde{v}_x + \frac{w_r}{2}\tilde{\omega}, \quad \tilde{v}_{y,3} = \tilde{v}_y - l_r\tilde{\omega}, \tag{5.4c}$$

$$\tilde{v}_{x,4} = \tilde{v}_x + \frac{w_f}{2}\tilde{\omega}, \quad \tilde{v}_{y,4} = \tilde{v}_y + l_f\tilde{\omega}. \tag{5.4d}$$

The resulting lateral tire forces $\tilde{F}_{y,l}$ can be calculated with the tire side slip angles in (5.3) via

$$\tilde{F}_{y,l} = D_{y,l}\sin\left(C_{y,l}\arctan\left(B_{y,l}\tilde{\alpha}_l\right)\right) \forall l \in \{1,2,3,4\}. \tag{5.5}$$

In Figure 5.3, the same trajectory as shown in Figure 5.2 is shown to show the influence of the newly formulated tire constraints. It can be observed that a much smoother trajectory is obtained, comparing it with the results in Figure 5.2. That the difference in grip caused the infeasible control problem can be concluded based on the resulting vehicle velocities shown in Figure 5.4. Where the infeasible trajectory still has to slow down, the new tire constraints already slowed the vehicle down to the required velocity to make the turn. Due to the early braking with the new tire constraints, the horizon velocity outperforms the old trajectory by means of top speed, essentially resulting in lower lap time.

Tire Saturation

Additionally to the adjusted tire constraints, a penalty is applied on excessive slip angle to prevent tire saturation beyond the peak slip angle α_{peak}. Surpassing the peak slip angle results in unstable vehicle behavior, as the increasing slip reduces tire grip. The lack of grip causes the vehicle to slide away from the desired trajectory, which it wants to correct by increasing the steering input.

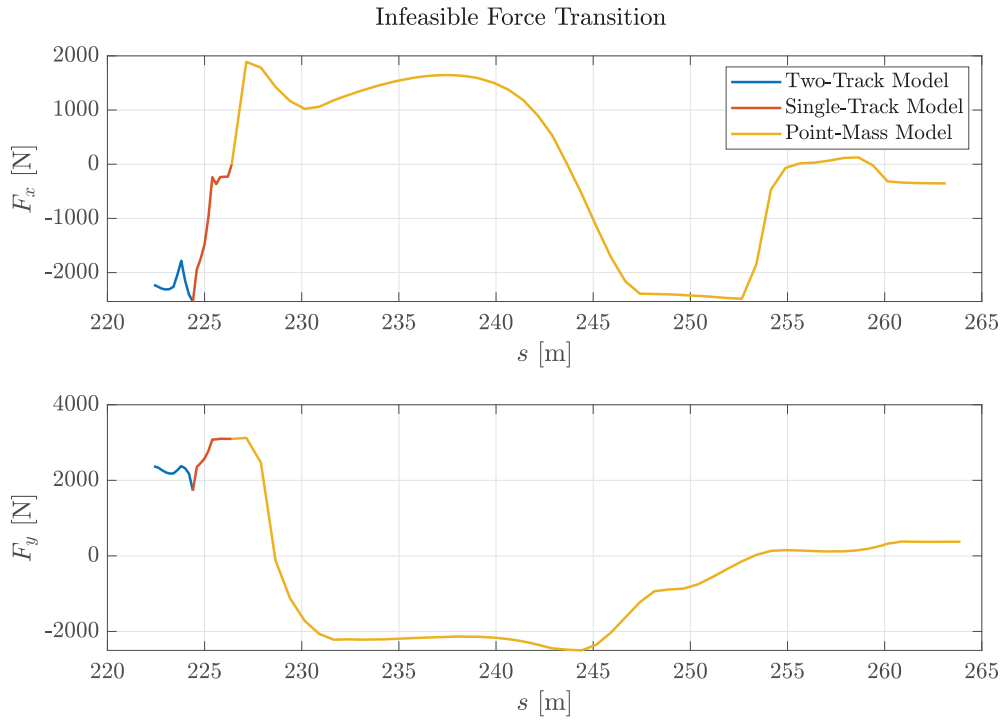

Figure 5.2: Force transition between the vehicle models for the infeasible model.

However, an increase in steering directly increases the slip angle, which reduces the grip further and results in an uncontrollable vehicle.

In Figure 5.5, the lateral tire forces are plotted against the corresponding side slip angle, where the saturation of the front tires is visible.

The penalty for operating beyond the peak slip angle α_{peak} is defined as

$$J_\alpha = W_\alpha \sum_{i=0}^{N} \left(\sum_{l=1}^{4} (\alpha_{l,i|k} - \alpha_{\text{peak}})^2 \right) + W_{\tilde{\alpha}} \sum_{z=0}^{H} \left(\sum_{l=1}^{4} (\tilde{\alpha}_{l,z|k} - \alpha_{\text{peak}})^2 \right), \tag{5.6}$$

where

$$W_\alpha = \begin{cases} W_{\alpha_0}, & \text{if } |\alpha_{l,i|k}| \geq \alpha_{\text{peak}} \\ 0, & \text{otherwise,} \end{cases}, \tag{5.7a}$$

$$W_{\tilde{\alpha}} = \begin{cases} W_{\alpha_0}, & \text{if } |\tilde{\alpha}_{l,z|k}| \geq \alpha_{\text{peak}} \\ 0, & \text{otherwise,} \end{cases}, \tag{5.7b}$$

and the weight W_{α_0} is the generalized weight for excessive tire slip angles for both the two-track and single-track model. In Figure 5.6, the result of this penalty can be observed, where all the tire slip angles for a single lap remain under the peak slip angle.

Lateral Error

It was found that the hard constraint on the lateral error caused infeasibility of the NLP. Minimizing lap timings provide an optimal trajectory close to the track limits and therefore gets close to $e_{y,\max}$. However, when driving close to the set bound, disturbances such as model mismatch,

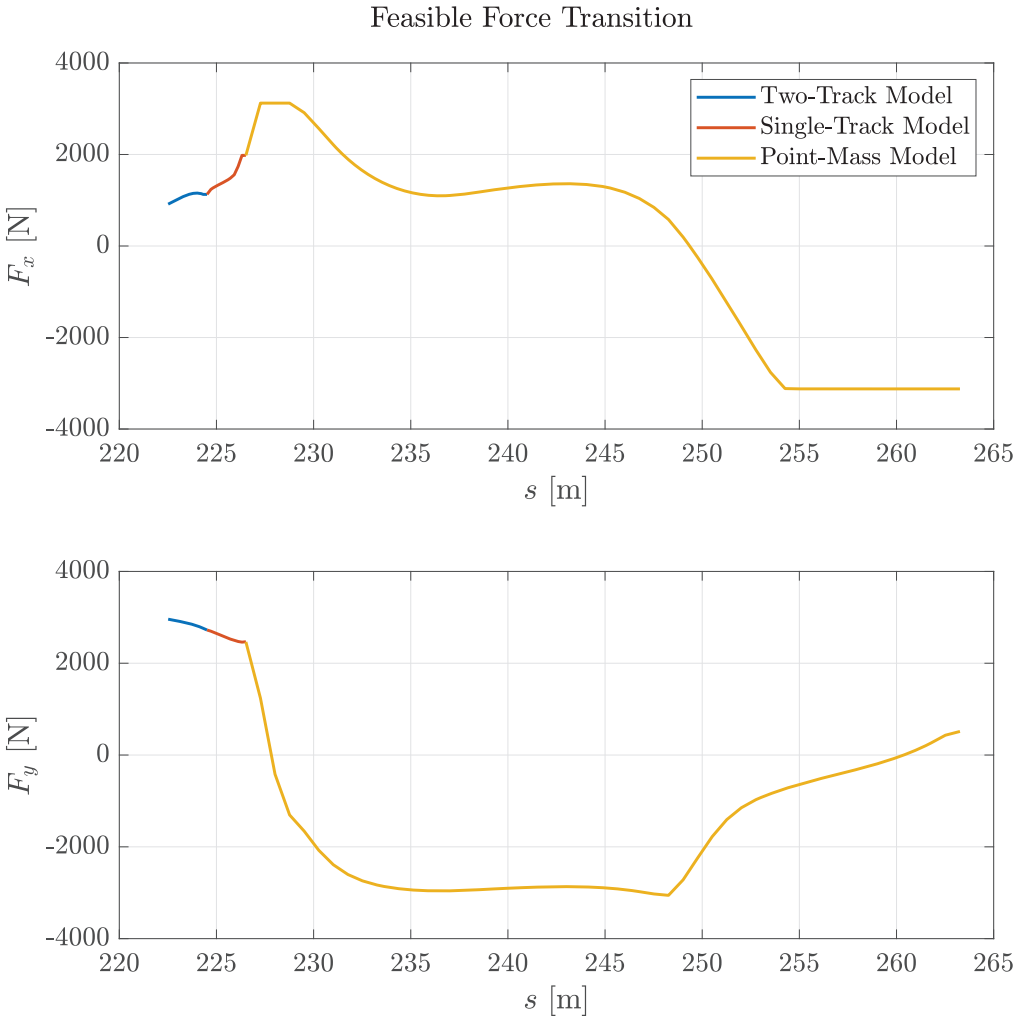

Figure 5.3: Force transition between the cascaded models with the new tire constraints.

numerical errors, and sensor drift can cause the NMPC input to be outside the track limits and yield an infeasible control problem.

Softening the hard constraint potentially causes dangerous situations, considering that there might be walls as track limits. Therefore, driving near the track limit is penalized via J_{e_b}

$$\begin{aligned} J_{e_b} =& W_{e_b} \Delta s \sum_{i=0}^{N} (|e_{y,i|k}| - e_{y,max})^2 + W_{\tilde{e}_b} \Delta \tilde{s} \sum_{z=0}^{H} (|\tilde{e}_{y,z|k}| - e_{y,max})^2 \\ &+ W_{\bar{e}_b} \Delta \bar{s} \sum_{j=0}^{M-1} (|\bar{e}_{y,j|k}| - e_{y,max})^2, \end{aligned} \tag{5.8}$$

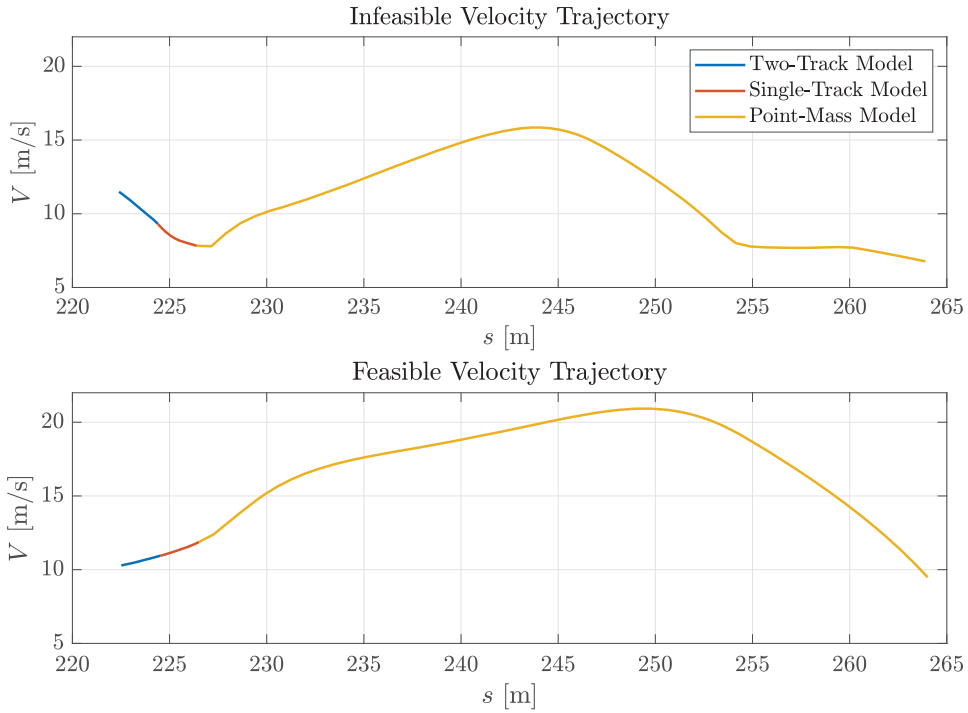

Figure 5.4: Vehicle velocity over the horizon for the infeasible and feasible trajectory.

where the weights are determined via

$$W_{e_b} = \begin{cases} W_{e_{b,0}}, & \text{if } |e_{y,i|k}| \geq e_b \\ 0, & \text{otherwise,} \end{cases} \tag{5.9a}$$

$$W_{\tilde{e}_b} = \begin{cases} W_{e_{b,0}}, & \text{if } |\tilde{e}_{y,z|k}| \geq e_b \\ 0, & \text{otherwise,} \end{cases} \tag{5.9b}$$

$$W_{\bar{e}_b} = \begin{cases} W_{e_{b,0}}, & \text{if } |\bar{e}_{y,j|k}| \geq e_b \\ 0, & \text{otherwise,} \end{cases} \tag{5.9c}$$

where $W_{e_{b,0}}$ represents the generalized cost for all vehicle models, and e_b is the set bound until the NLP is not receiving an additional cost.

The results in Figure 5.7 show the solution for $e_{y,\max} = 0.9$ and the bound $e_b = 0.675$, leaving a conservative minimum distance of 0.225 m between the track limits and e_b. The left plot shows the original result, where the optimal trajectory lays on the track limits of 0.9 m, causing infeasibility once the simulation model reaches this optimal solution. Applying (5.8) results in a more conservative trajectory, shown in the right plot. Note that the new trajectory does exceed e_b, showing it still uses the entire track width if necessary but provides a margin for any external disturbances on the NMPC's input.

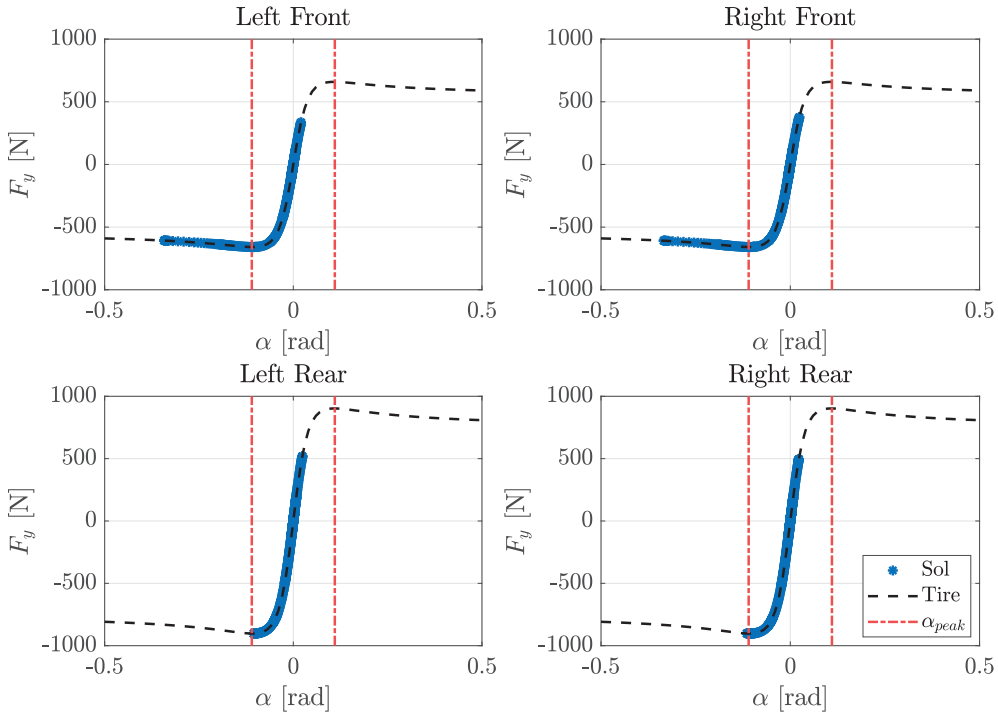

Figure 5.5: Tire forces beyond the peak slip angle α_{peak}.

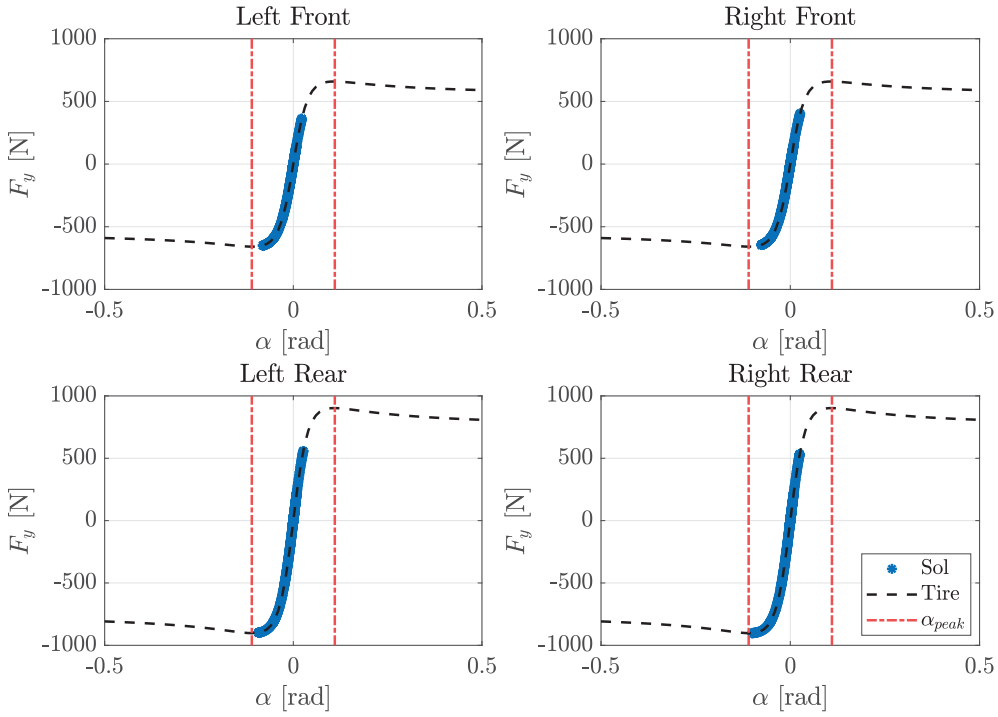

Figure 5.6: The optimal solution when excessive tire usage is penalized.

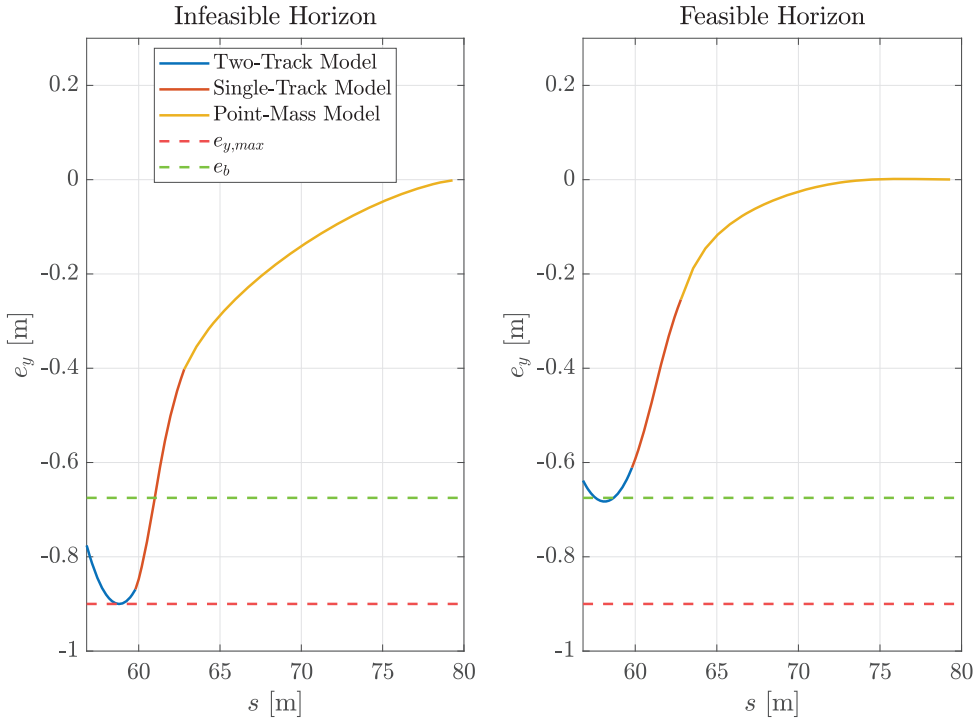

Figure 5.7: Left: Optimal horizon without (5.8), resulting in infeasibility. Right: Optimal horizon including (5.8), resulting in a feasible solution.

5.2 Model Comparison

This thesis aims to develop an online optimization-based motion planning system for all-wheel drive autonomous race cars. For online implementation, the required computation time to solve an NMPC is often proven to be a bottleneck and therefore is considered one of the key performance indicators when comparing the different horizon configurations. Also, since we aim towards autonomous racing, the resulting lap time is also a key performance indicator. Moreover, the resulting lap time is a direct performance indicator of the optimization algorithm since its main goal is the minimization of the resulting time at the end of the prediction horizon.

A simulation study is performed for varying prediction horizon lengths of the cascaded model. The results of these simulations give an indication of the influence of different horizon configurations on computation time and lap time. This simulation study includes all the changes described in Section 5.1 to the original NMPC from Chapter 4. In addition, the optimization is performed on the FSG Driverless track in Figure 5.1, as this track length allows for more simulations in the given time span of this project.

The prediction horizons for the single-track and two-track models are chosen to analyze the influence on lap time and computation time when both N and H are increasing, and what the difference is when $N > H$ or $N < H$. Therefore, the respective horizons are chosen such that $N \in \{10, 20\}$ and $H \in \{1020\}$. Additionally, the point mass model horizon is at first increasing with smaller steps, to highlight the significant benefit of using a point mass and consequently increased with larger steps to analyze the results of large prediction horizons. Therefore, the point mass horizon is chosen such that $M \in \{5, 10, 15, 20, 30, 40, 50\}$.

All different model configurations are compared with the performance of an NMPC only using the two-track model and for $N \in \{10, 20, ..., 150\}$. In Figure 5.8, the resulting lap time and computation times for all different configurations are displayed. A quick observation already answers the question of what the influence is of using a cascaded vehicle model when comparing the two key performance indicators. When comparing the results of the most complex model with $N = 150$ with the results of a more simple cascaded NMPC, i.e. $N = 10, H = 10, M = 40$, a lap time of 22.2 s can be achieved while reducing the computation time from 3.71 s to 0.49 s, which is a reduction of 86.8 %. This performance increase is a result of the simplified dynamics and significant discretization step of the point mass.

However, a more detailed analysis is required to determine the difference in vehicle behavior when using a cascaded prediction horizon. To do so, each cascaded configuration that matches the lap time similar to the complex model with $N = 150$ is chosen to analyze the difference in vehicle behavior while achieving overall similar track performance. Based on the results in Figure 5.8, the five selected cases with their respective lap time are

case 1: $N = 150$, $H = 0$, $M = 0$, $t = 22.20s$

case 2: $N = 10$, $H = 10$, $M = 50$, $t = 21.80s$

case 3: $N = 10$, $H = 20$, $M = 30$, $t = 22.25s$

case 4: $N = 20$, $H = 10$, $M = 30$, $t = 22.25s$

case 5: $N = 20$, $H = 20$, $M = 30$, $t = 21.86s$

Note that for case 2 and 3, where N and H differ but the overall horizon length is the same, yield the exact same lap time. A similar trend is observed in Figure 5.8, where the resulting lap times for $N = 10, H = 20$ and $N = 20, H = 10$ for each M is approximately the same and only result in an additional computation time for larger N.

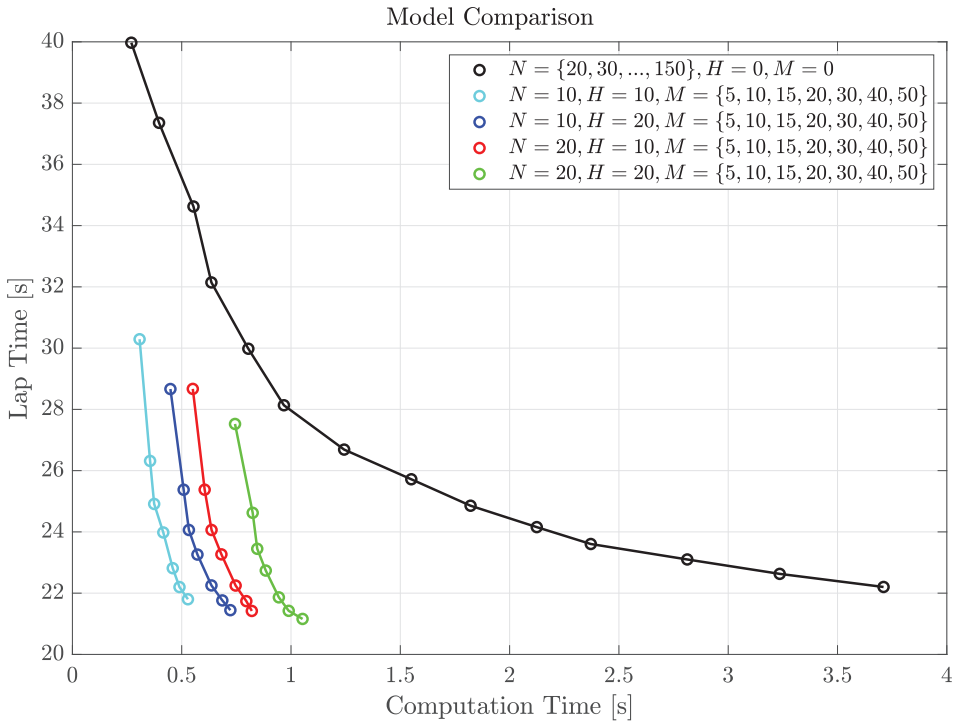

Figure 5.8: Computation time vs lap time for different horizon lengths.

Minimizing lap time is highly dependent on the resulting vehicle velocity. Therefore, in Figure 5.9, the resulting velocity of the simulation is displayed for the five cases mentioned above. To determine where each model distinguishes itself regarding minimization of time, the difference in time Δt with respect to case 2, which has the lowest lap time of the five cases, is shown. These results are obtained via simulating for two consecutive laps, where the results of the second lap are used to compare the different cases, excluding the effects due to initialization.

Analyzing the results in Figure 5.9 indicate that the point-mass model mainly benefits the optimization problem for $s \in \{382.7, ..., 482.8\}$ and $s \in \{622.6, ..., 697\}$, which is the start/finish straight till turn one, and the long straight between turn seven and turn eight. On the other hand, a more complex vehicle model proves to be faster in the remainder of the track, which consists mainly of corners.

The improved vehicle velocity for increasing M at $s \in \{382.7, ..., 482.8\}$ and $s \in \{622.6, ..., 697\}$ is the result of an increased look-ahead distance of the NMPC. By putting the terminal velocity $V_{M,\max}$ further away from the vehicle, the controller can exploit the prediction horizon to achieve a higher top speed. This results in a maximum velocity of $22.20\,\mathrm{m\,s^{-1}}$ for case 2, which is the least complex model, where case 1 only achieves a top-speed of $19.13\,\mathrm{m\,s^{-1}}$, which uses the most complex model. Cases 3, 4, and 5 follow a similar trend, where the increasing point-mass horizon directly translates into increasing vehicle velocity.

Furthermore, the results of Δt in Figure 5.9 show that the more complex vehicle models improve their lap time during cornering. This increased performance is related to the use of an increased prediction horizon of the complex vehicle models, gaining advantages due to corner cutting and exceeding the bound e_b. In (5.8), surpassing the bound is weighted for increasing step sizes, allowing the two-track and single-track model to use the track more extensively. The lateral error, displayed in Figure 5.10, shows the difference in track usage, indicating that case 1 has a maximum lateral error of 1.15 m, where case 2 achieves only 0.71 m.

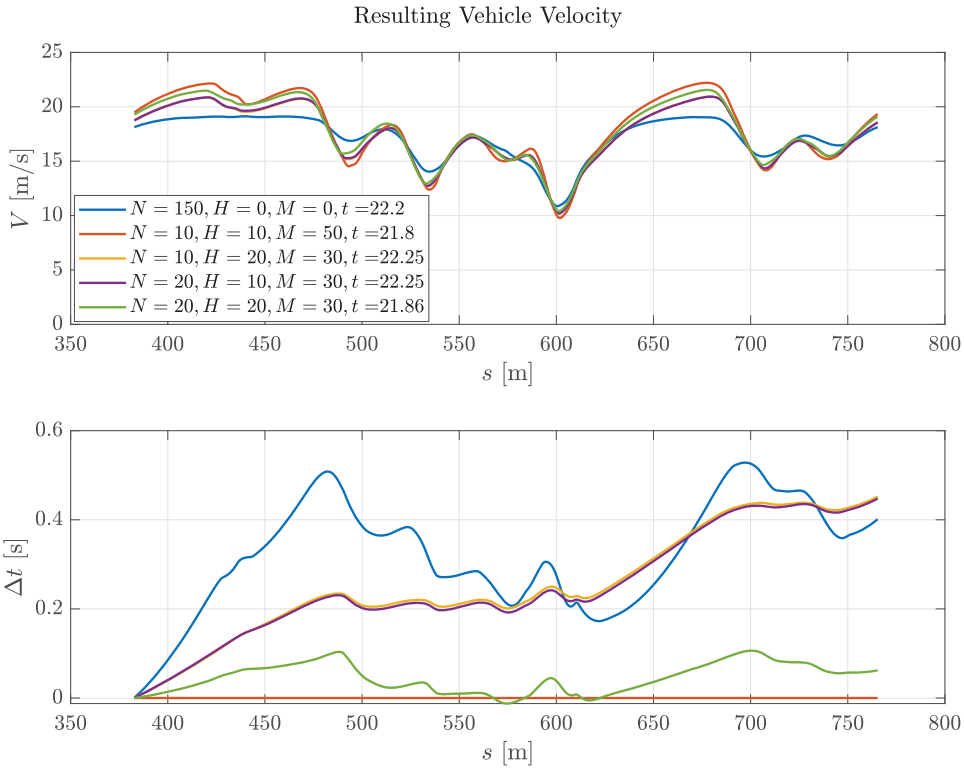

Figure 5.9: Resulting vehicle velocity after an out-lap, and the resulting Δt with respect to the fastest lap time.

The hypothesis that the complex models utilize the track more is observed in Figure 5.11, where the resulting racing line is shown from the entry of corner five till the exit of corner seven. Furthermore, using the width of the track more effectively allows the car to carry more speed through the turns, which is also illustrated in Figure 5.12, where the velocity while cornering of case 1 is higher than all other cases, decreasing the Δt by 0.0845 s from entry corner five till exit corner seven. The same observation is made for case 5, which increases the cornering performance due to the two-track and single-track model while maximizing its top speed due to the point-mass horizon.

Not only is velocity important to minimize lap time, but also the vehicle behavior and stability during cornering. The different steering inputs in Figure 5.13 show that a decreasing model complexity requires more steering, as its path planning is more conservative and not exploring the full track width. As a result, at $s = 601$ case 2 achieves the highest steering amplitude while driving the slowest, explaining the time loss compared to all other cases when looking at Figure 5.9. Moreover, a smoother steering trajectory is obtained by increasing the model complexity. The results in Figure 5.13 show that cascaded vehicle models are correcting their steering behavior while cornering, potentially causing unbalance in a real-time application.

Such behavior can be further analyzed by comparing the resulting yaw-moment, which is graphically shown in Figure 5.14, where a distinction is made in the yaw-moment due to steering, namely F_y, and due to torque vectoring, namely F_x. For visibility reasons, only case 1 and case 2 are shown, but the two cases also clearly distinguish the difference in model complexity and their respective cornering behavior. Case 2 uses torque vectoring more extensively while case 1 relies more on its steering. This behavior can be observed for $s \in \{400, ..., 450\}$, which is from entry turn one till exit turn two. Case 1 shows a smooth trajectory for the resulting yaw-moment through this corner combination, requiring a maximum yaw-moment of approximately $-300\,\mathrm{N\,m}$ at the exit of

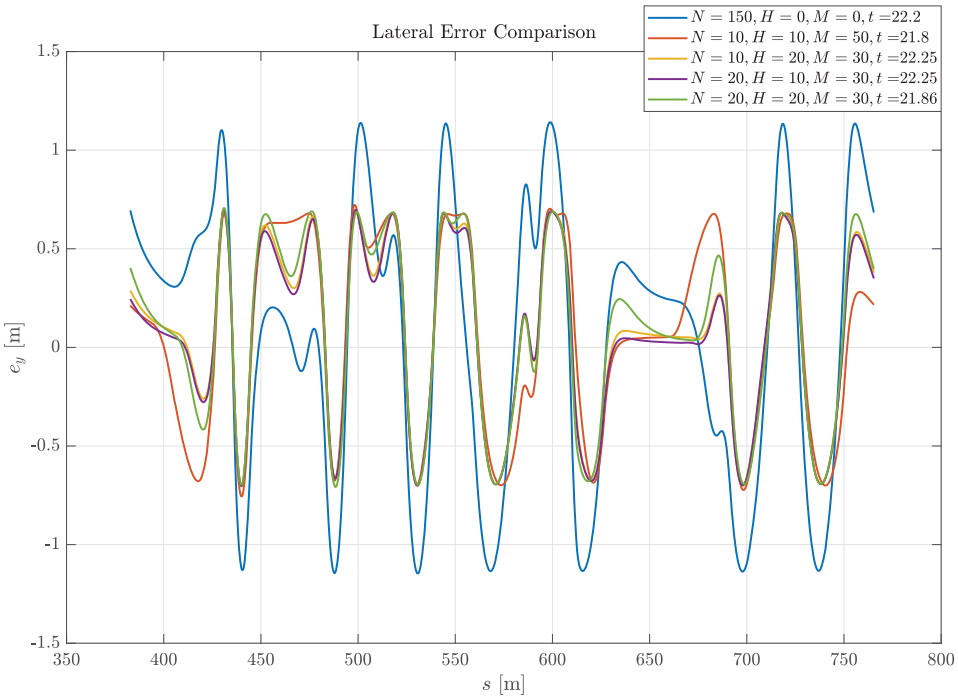

Figure 5.10: Lateral error e_y comparison for the 2^{nd} lap.

turn two to stabilize the vehicle, which is at $s = 436$. Comparing these results with case 2 show that the car requires much more torque vectoring to stabilize the vehicle during cornering, causing the resulting yaw-moment to fluctuate. At the exit of turn 2, it also requires a yaw-moment of $-885\,\mathrm{N\,m}$ for a short duration to stabilize the vehicle. This prominent peak for case 2 is where case 1 requires approximately the same amount of yaw moment but achieves this via less aggressive steering.

The final comparison between the different cases is based on the g-g diagram in Figure 5.15. The longitudinal and lateral acceleration show the amount of grip utilized in a single lap. These results confirm the hypothesis that the point-mass model is mainly responsible for pushing the vehicle toward the limits when looking at longitudinal behavior since it achieves the highest acceleration and deceleration. Also, it shows that the two-track model and single-track exploit the track more such that it can carry more speed through turns, where the mean value of the absolute accelerations are given in Table 5.3, showing an increase in $|a_y|$ for increasing N and H. Notice that these results indicate that case 1 is the slowest during cornering. However, case 1 is mainly limited due to the short look-ahead distance and is still close to the performance of the cascaded vehicle models. Moreover, case 1 is driving almost as fast during cornering as case 2 while using 46.4 % less longitudinal acceleration.

Table 5.3: Mean longitudinal and lateral acceleration.

	$\lvert a_x \rvert$ [g]	$\lvert a_y \rvert$ [g]
Case 1	0.1561	0.0548
Case 2	0.2911	0.0568
Case 3	0.2382	0.0557
Case 4	0.2351	0.0588
Case 5	0.2467	0.0620

Not only is lap time critical, but the required computation time to solve the NLP is of utmost

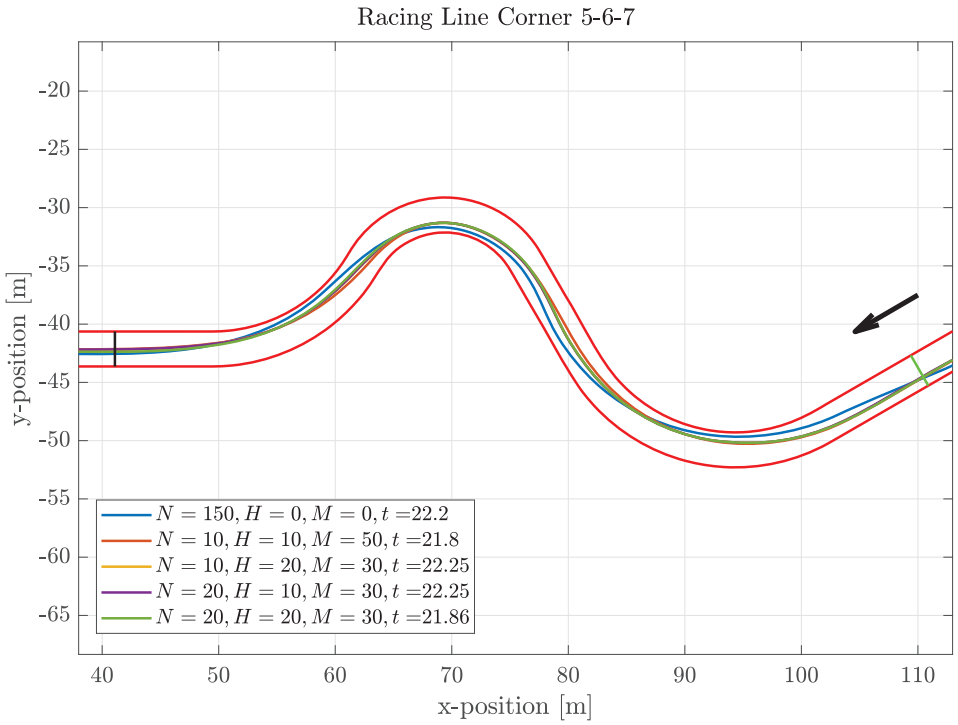

Figure 5.11: The racing line for entry corner five till exit corner seven according to Figure 5.1, where the green and black line indicate the start and end point of the evaluation in Figure 5.12, respectively.

importance. However, the computation time needed in Figure 5.9 indicates that online implementation for such an NMPC application is complex. Different coding languages can significantly boost the required computation time to solve the NLP, but this is not considered within the time span of this project. Nonetheless, the cascaded model substantially improves the needed computation time compared to a single two-track model, reducing the computation time up to 80 %. This result is mainly achieved due to the simplified vehicle dynamics and the reduced number of states, reducing the required time to solve the Hessian of the Lagrangian and the constraint Jacobian. In Figure 5.16, the average computation times can be observed, showing the increase in computation for increasing model complexity.

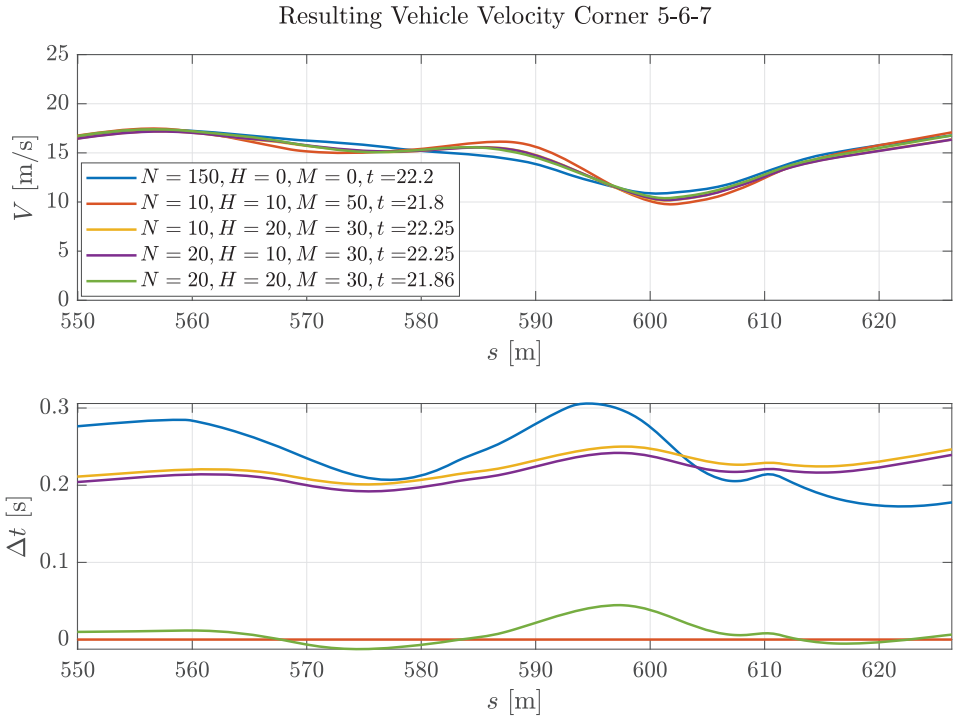

Figure 5.12: Resulting vehicle velocity and Δt for entry corner 5 till exit corner 7 according Figure 5.1.

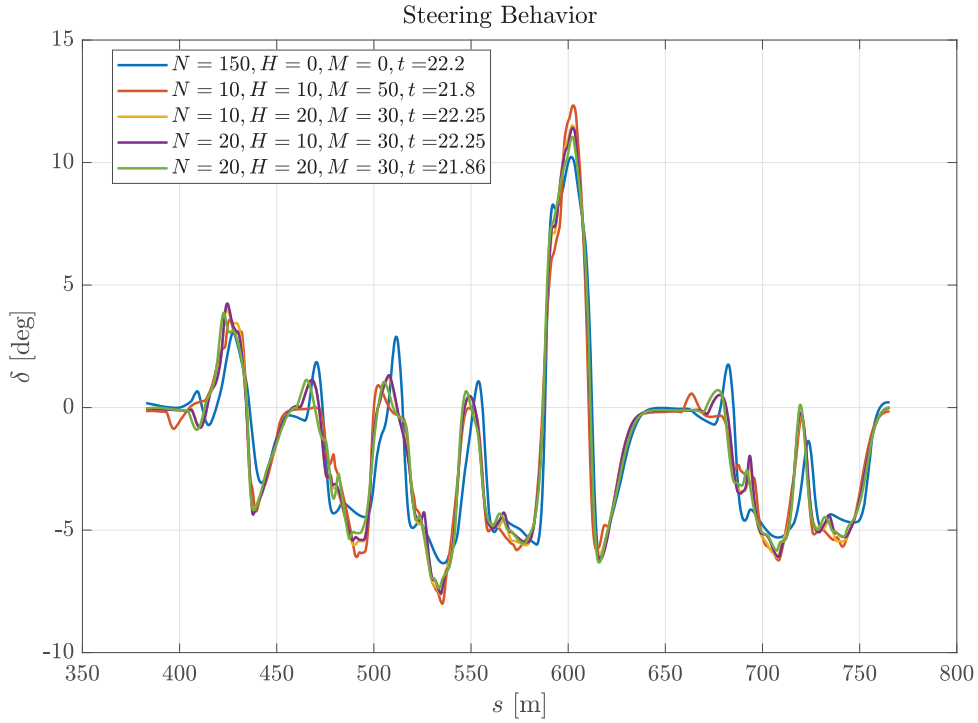

Figure 5.13: Steering behavior for the 2^{nd} lap for case 1 till 5.

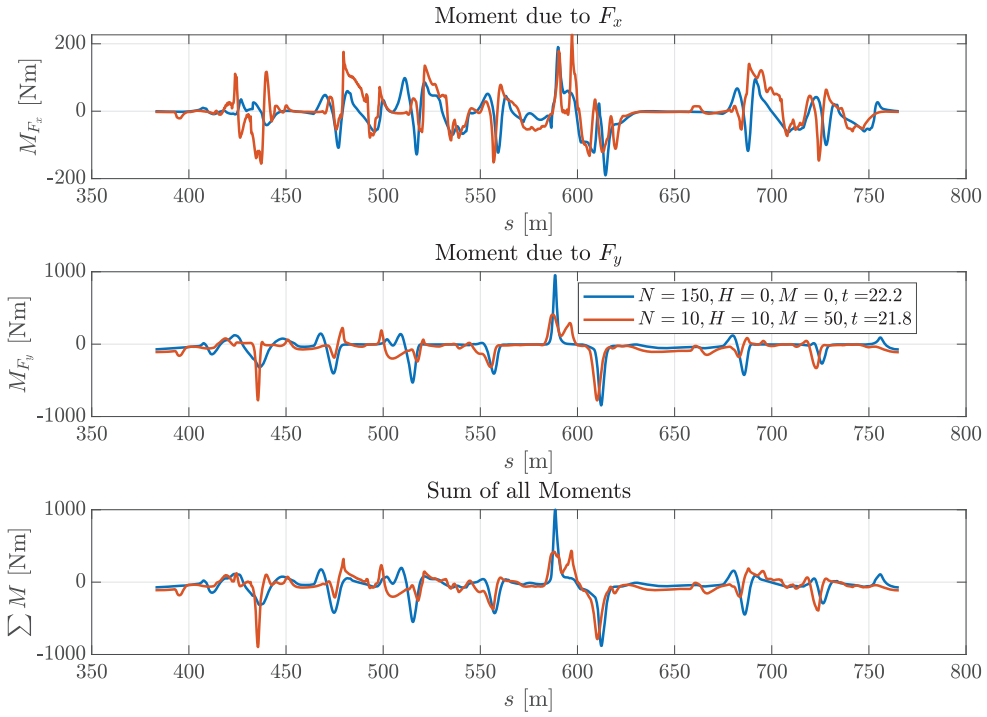

Figure 5.14: The resulting moment due to F_x, F_y, and the sum of both moments combined for cases 1 and 2.

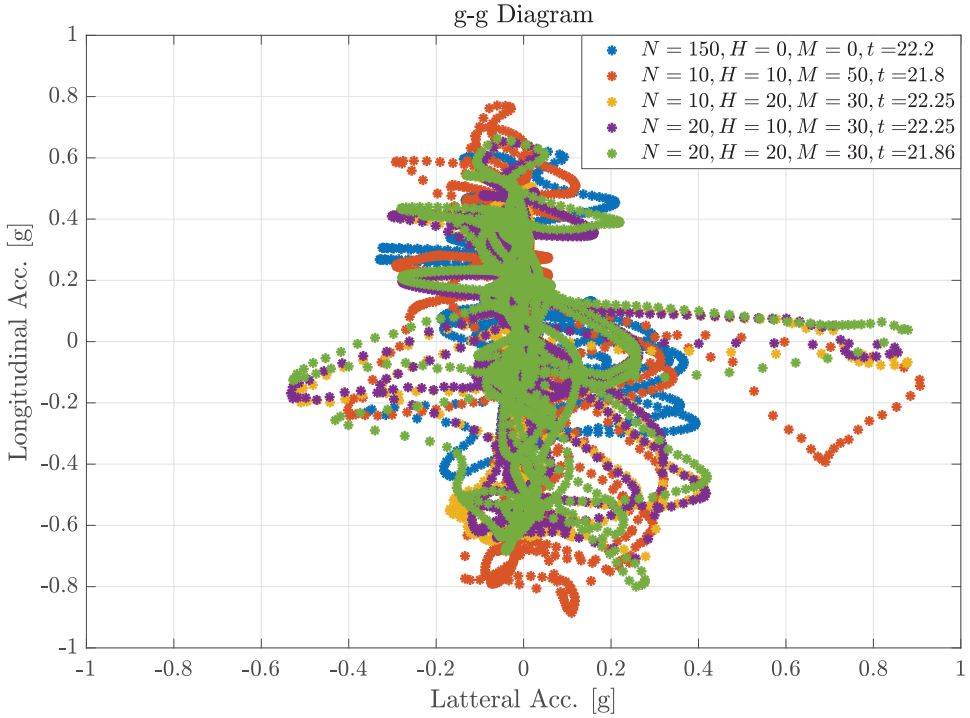

Figure 5.15: g-g Diagram, indicating overall vehicle performance for all five cases.

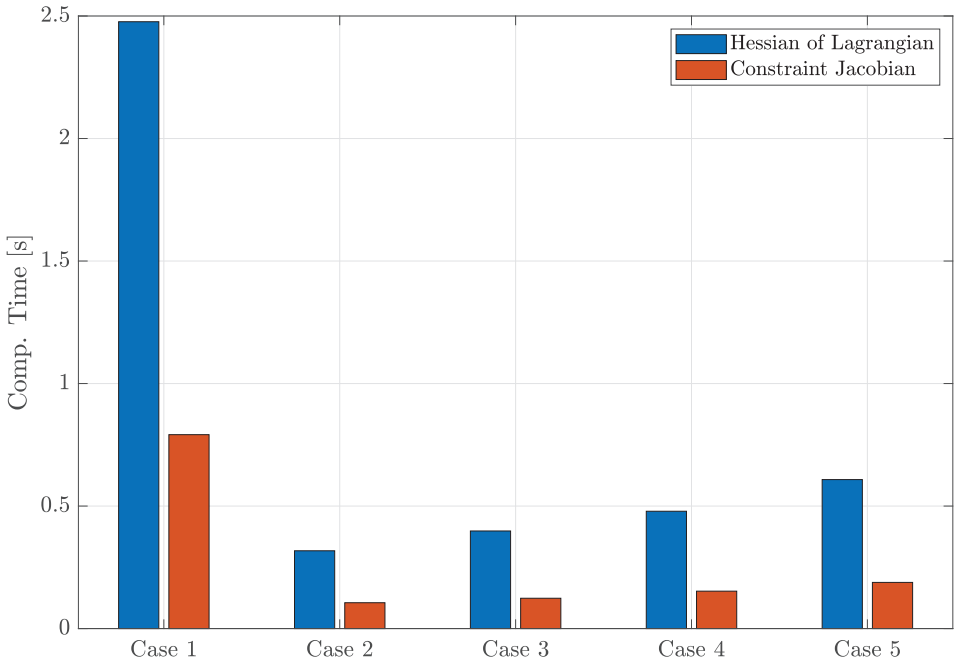

Figure 5.16: Average computation time for solving the Hessian of the Lagrangian and the constraint Jacobian.

5.3 Linear Tyre Constraints

In this section, the hypothesis from Section 4.3 is investigated, where the use of linear tire constraints to approximate the friction ellipse can be used to reduce the computation time of the NLP. Furthermore, several model configurations are used to investigate the influence on the different vehicle models where the number of linear constraints per quadrant varies from one to four.

The different cases considered in this section are

case 1: $N = 10$, $H = 10$, $M = 20$, No. Points $\in \{1, 2, 3, 4\}$,

case 2: $N = 10$, $H = 10$, $M = 30$, No. Points $\in \{1, 2, 3, 4\}$,

case 3: $N = 10$, $H = 20$, $M = 20$, No. Points $\in \{1, 2, 3, 4\}$,

case 4: $N = 10$, $H = 20$, $M = 30$, No. Points $\in \{1, 2, 3, 4\}$,

case 5: $N = 20$, $H = 10$, $M = 20$, No. Points $\in \{1, 2, 3, 4\}$,

case 6: $N = 20$, $H = 10$, $M = 30$, No. Points $\in \{1, 2, 3, 4\}$,

case 7: $N = 20$, $H = 20$, $M = 20$, No. Points $\in \{1, 2, 3, 4\}$,

case 8: $N = 20$, $H = 20$, $M = 30$, No. Points $\in \{1, 2, 3, 4\}$,

while using the parameters from Section 5.2 and simulating the vehicle driving on the FSG Driverless track from Figure 5.1.

At first sight, the results Figure 5.17 show that the system's performance decreases while using linear tire constraints, where it was expected to increase since the tire forces are coupled linearly and not quadratically. Nonetheless, the simulation results indicate that using (4.41) is computationally more efficient, where the results from Figure 5.8 for the same model configuration are shown in Figure 5.17 via the diamonds in the same color.

The required number of linear constraints can explain the decrease in performance for a well-constrained optimization problem. At least four linear constraints per friction ellipse are required, increasing the total number of constraints from four to sixteen for $P_n = 1$. While improving the accuracy of the approximation by increasing P_n, the number of linear constraints increases linearly with a slope of 4 times the number of friction ellipses.

The increasing number of constraints increases the required computation time to solve the Hessian of the Lagrangian, graphically shown in Figure 5.18, and calculate the constraint Jacobian, displayed Figure 5.19. Concluding from these results, it is not beneficial to use linear tire constraints, as both the computation and lap timings decrease compared to a quadratic coupling of the longitudinal and lateral tire forces.

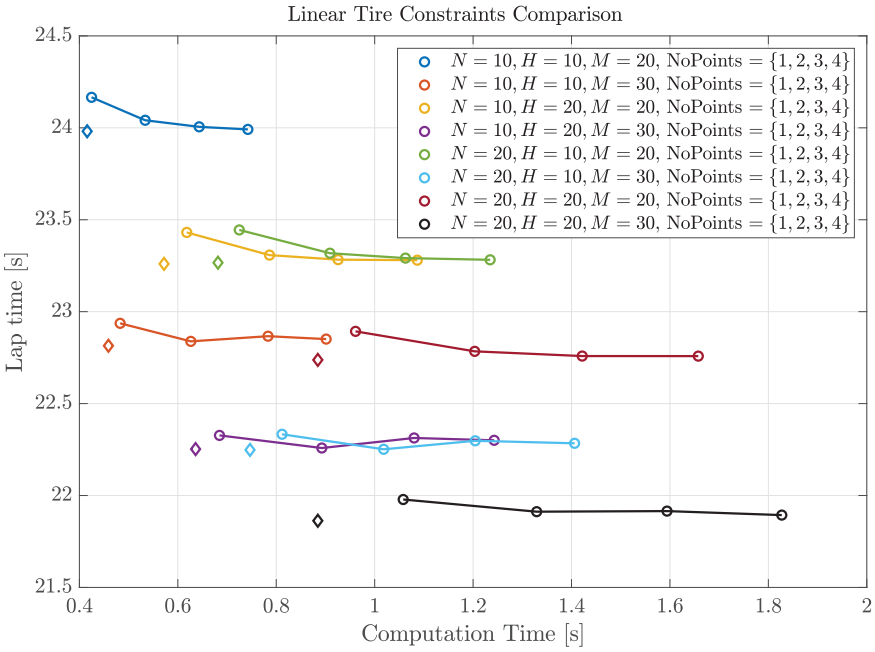

Figure 5.17: Laptime vs computation time comparison for linear tire constraints, where the diamonds indicate the respective nonlinear solution from Section 5.2.

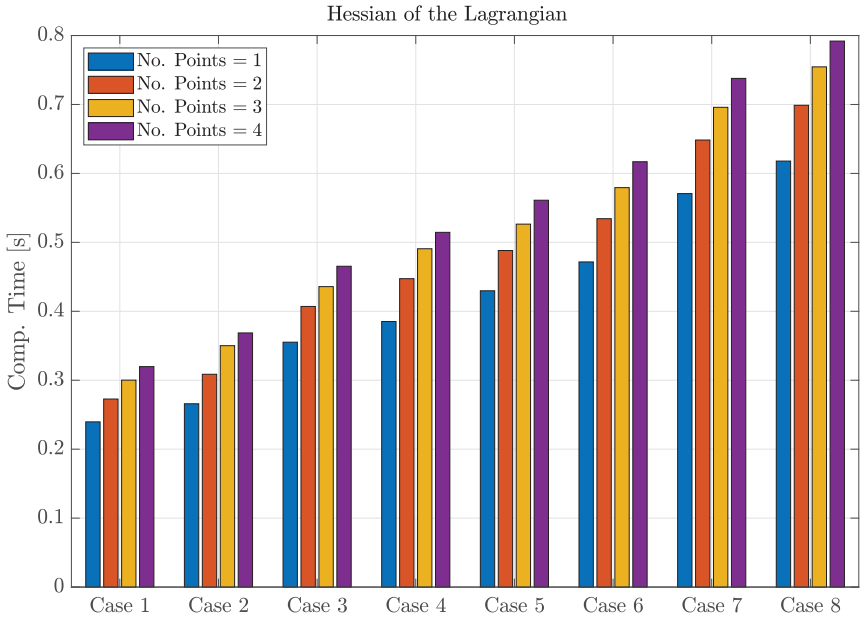

Figure 5.18: Required computation time to solve the Hessian of the Lagrangian with linear tire constraints.

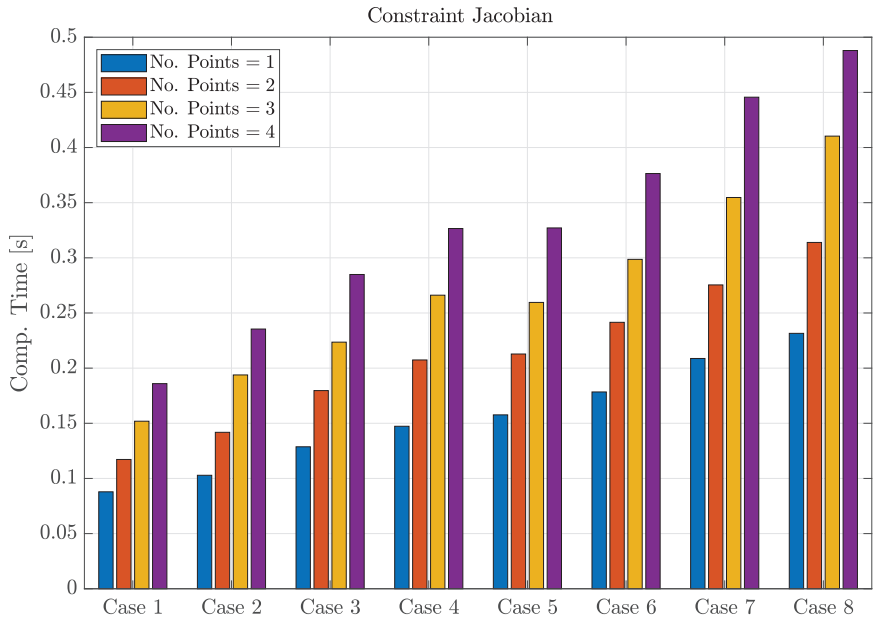

Figure 5.19: Required computation time to solve the constraint Jacobian with linear tire constraints.

5.4 Convergence

In non-convex optimization, there are multiple control challenges. One is the proof of stability, which is often hard due to the complexity of the optimization problem. Second, due to the non-convex optimization problem, local rather than global minima are usually found. In this chapter, a simulation study is performed to analyze the output from the NMPC on its convergence over consecutive laps using a cascaded vehicle model. The convergence of a solution indicates that the NMPC finds the same system input for a reference path for the same controller input. Moreover, driving multiple laps suggests that the controller can drive the car safely around the track while maintaining vehicle and computational stability.

For the simulation study, two different control configurations are selected. The first configuration, case 6, is the configuration achieving the lowest computation time in the simulation study of Section 5.2, where the results are given Figure 5.8. This model configuration is also the most simple configuration, having the smallest prediction horizons of all cascaded vehicle models. The second configuration, case 7, is the most complex model configuration from the results in Figure 5.8, using the largest prediction horizon but achieving the best lap time. For convenience, the model configuration given here

case 6: $N = 10$, $H = 10$, $M = 5$

case 7: $N = 20$, $H = 20$, $M = 50$

The first comparison between the consecutive laps is made based on the resulting lap time and computation time, with the results for both cases 6 and 7 presented in Table 5.4. Both model configurations show a different lap time in the first lap due to the initial state of the vehicle at the start of the simulation. Based on the resulting lap times, it could be said that both cases 6 and 7 converge to the same solution after lap 1. The same conclusion can be drawn by looking at the results in Figure 5.20 and Figure 5.21. Notice that for these results, the modulo operation is used on s to plot the results from start to finish for each lap. These results compare the velocity V, steering input δ, and lateral error e_y for the consecutive laps. The three states describe the overall vehicle behavior since the velocity and steering angle combined describe the longitudinal, lateral and rotational vehicle motion. The lateral error describes the vehicle's position, which, combined with the vehicle motions, can be used to determine if convergence is achieved.

The results in Figure 5.20 and Figure 5.21 show that for all three vehicle states the same solution is found, indicating that the solution converged to a periodic solution for this particular track. Moreover, the NMPC provides stable vehicle behavior and numerical stability is shown for multiple consecutive laps. Comparing the average computation time per lap in Table 5.4 show that required computation time for case 6 deviates slightly between laps, but nothing out of the ordinary. However, case 7 show a decline in required computation time after the first two laps, but no difference is observed between the output of lap 2 and 3. Therefore, it is assumed that other operations consumed the computer's computational power during the simulations.

Table 5.4: Lap time and computation time per lap for case 6 and case 7.

	t_{lap} [s] Case 6	t_{comp} [s] Case 6	t_{lap} [s] Case 7	t_{comp} [s] Case 7
Lap 1	30.8821	0.2990	21.9941	1.0785
Lap 2	30.2869	0.2958	21.1660	1.0716
Lap 3	30.2871	0.2958	21.1665	1.0519
Lap 4	30.2871	0.3074	21.1669	1.0524
Lap 5	30.2869	0.2960	21.1659	1.0514

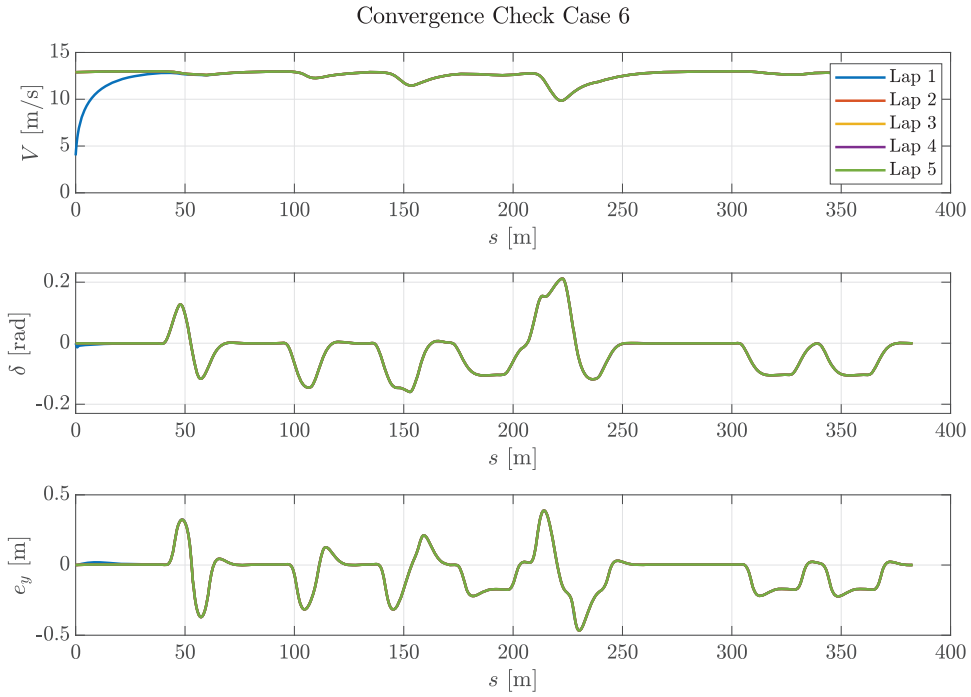

Figure 5.20: Convergence check of the velocity V, steering angle δ and lateral error e_y. Controller configuration: $N = 10$, $H = 10$, $M = 5$.

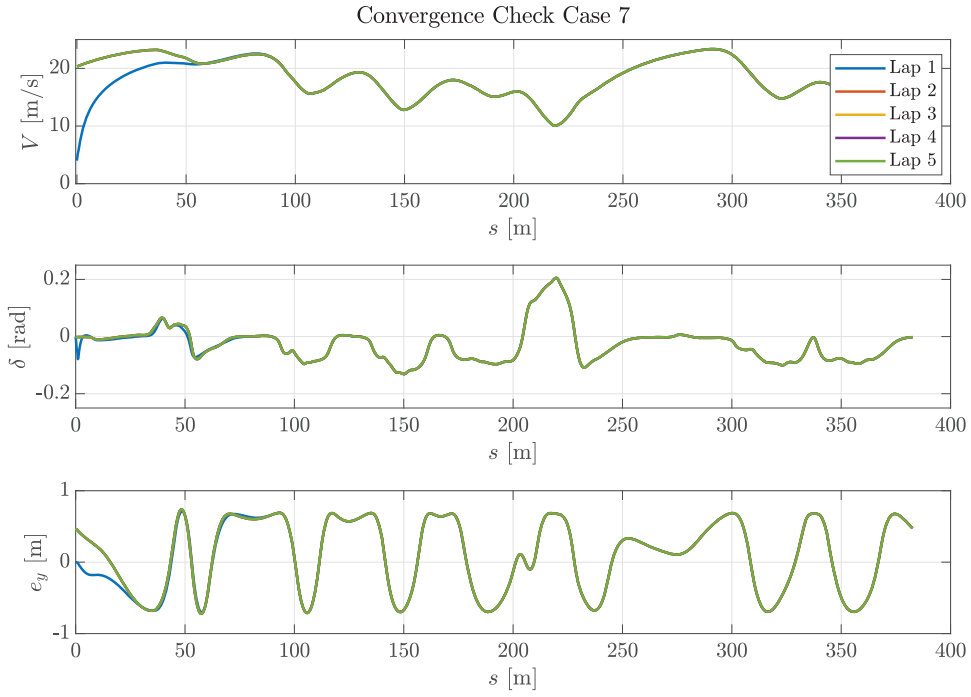

Figure 5.21: Convergence check of the velocity V, steering angle δ and lateral error e_y. Controller configuration: $N = 20$, $H = 20$, $M = 50$.

5.5 Conclusions and Observations

In this chapter, the proposed NMPC from Chapter 4 is tested using the driverless track from FSG 2019, as it is a good representation of a real-time application for URE. The proposed framework from Chapter 4 is adjusted to solve several feasibility issues.

First, the difference in grip between the cascaded vehicle models caused an infeasible prediction horizon. Each vehicle model's friction ellipse is scaled based on its weight, not considering the other vehicle models. Solving the optimization problem resulted in a poor force transition due to the optimal trajectory of the point mass being too optimistic, causing the tire forces of the two-track model to saturate. To bridge the gap between the different vehicle models, the tire constraints are formulated to consider the maximum grip of the vehicle model it is cascaded with. Also, an additional cost is applied when operating beyond the peak slip angle to prevent excessive slip.

Second, the optimal solution is pushed toward the track bounds when the track layout is fully utilized. However, numerical inaccuracy caused the simulation model to surpass the track bounds, causing an infeasible control problem. When an online implementation is considered, disturbances such as sensor drift or external disturbances should be accounted for. Therefore, an additional penalty is applied, resulting in a more conservative racing line, but providing robustness to the motion planning algorithm.

The resulting controller is tested for varying horizon lengths and is compared with a single two-track model based on two key performance indicators, computation time and lap time. Analysis shows that using a cascaded vehicle model is capable of minimizing lap time while stabilizing the vehicle at the same time. An increased point-mass horizon directly translates into better longitudinal vehicle behavior, but a high-fidelity model is required to achieve the desired cornering behavior. However, the simulation results indicate that no reduction in lap time is obtained by choosing a longer prediction horizon for the two-track model over a longer prediction horizon for the single-track model. Moreover, the same lap time is obtained while an increase in computation time is observed.

Implementing linear tire constraints to decrease the computation time is tested via several simulations. However, where it was excepted to increase, the system's performance decreased significantly compared with the quadratic tire constraints. Furthermore, the analysis showed that the lap time and computation time increased; therefore, the linear tire constraints are not included in the proposed controller.

One drawback of the proposed controller is the required computation time. Using a two-track model to utilize all four in-wheel motors fully requires complex constraints and vehicle models with many states. Using a cascaded vehicle model reduced the overall computation time by 80 % compared to a single two-track model. Unfortunately, solving a cascaded model's computation time is still too high to consider real-time experiments. This thesis did not consider using a different programming language, which should boost the overall performance, but one could say that even considering another computer language will not suffice.

Chapter 6

Conclusion and Recommendations

The automotive industry has evolved in the past few years, trying to achieve the ultimate goal of fully self-driving vehicles. Many future challenges lay within the scope of motion planning to bridge the gap between autonomous driving and achieving the performance of an expert driver. This thesis aims to contribute by developing a motion planning algorithm for all-wheel drive autonomous racing cars to participate in a formula student race. These competitions do not consider head-to-head racing, providing a safe environment to bring online motion planning a step closer to the performance of an actual racing driver. In this chapter, a summary of the thesis and its primary conclusions and recommendations for future research are stated.

6.1 Conclusion

Several dynamic models have been derived to describe all-wheel-driven vehicles' motion and dynamic behavior. These vehicle models describe the behavior concerning a reference path, which SLAM provides. The vehicle's position is then determined via the footpoint, representing the location of the Frenet coordinate frame. Based on the location of the Frenet coordinate frame, a measurement for the lateral and heading error of the vehicle is obtained. An Extended Kalman Filter is proposed to mitigate external disturbances on the controller input using a single-track model with linear tire dynamics. Local observability of the Extended Kalman Filter is proven via a pragmatic approach, which means that it is not proven reliable for all conditions. The localization algorithm is validated on the autonomous racing car of University Racing Eindhoven. The results indicated that the observer is very sensitive to a model mismatch, especially the lateral and heading error prediction. Moreover, the prediction step was inaccurate due to incorrect tire parameters, causing a significant error in estimating the lateral dynamics. The lateral dynamics of the model could not be validated since the tests were performed without the possibility of measuring the lateral velocity. Nonetheless, the desired output is obtained by tuning the covariance matrices such that the observer output depends more on the footpoint measurement.

Based on the vehicle's position and the track's layout, a desired input for the car is determined. An online optimization algorithm is proposed, using nonlinear model predictive control to determine the desired longitudinal force at each of the four wheels and the desired steering rate. The primary objective of the optimization problem is to minimize the time at the end of the prediction horizon, where time is an optimization variable due to the transformation to the spatial domain. While pushing the vehicle to the limits of performance, several costs are introduced to ensure the durability of the actuators, increase the smoothness over the prediction horizon and penalize undesired vehicle behavior. Using a nonlinear two-track model to solve a nonlinear optimization problem requires significant computational power. Therefore, the principle of cascading vehicle models is introduced to find a balance between model complexity and computation time. The

resulting controller is a single-layer nonlinear model predictive controller responsible for path planning and reference tracking. This controller is tested via simulations, showing that a cascaded vehicle model achieves similar lap timings while reducing the computation time up to 80 % when comparing the cascaded horizon configuration with a long two-track model horizon. A high-fidelity model is required to stabilize the vehicle and maximize its cornering behavior. The point mass is allowed to maximize the longitudinal performance by increasing the look-ahead distance of the controller. The convergence of the solution is analyzed by performing an additional simulation study on the most simple and most complex model configuration while using a cascaded prediction horizon. It is observed that the NMPC finds periodic convergence for the considered racing track, resulting in the same stable vehicle behavior after lap 1.

An attempt is made to approximate the friction ellipse via several linear constraints, primarily to decrease the required computation time. However, the implementation of linear tire constraints resulted in a significant decrease in overall controller performance, increasing both the lap time and computation time. Unfortunately, the hypothesis is not valid, and one is therefore limited to the quadratic friction ellipse.

In conclusion, a motion planning system that respects general state, input, track, and tire constraints have been developed. It employs a two-track model, fully utilizing the torque vectoring possibilities for all-wheel drive vehicles. However, the computational and numerical complexity of the controller prevents real-time applications. Using the cascaded vehicle model reduces the computational complexity of the optimization problem, but unfortunately, not enough. The objective function can maximize the performance on track while preventing undesired vehicle behavior. Incorporating a nonlinear tire model allows the control to push the vehicle, but access to the required parameters can not always be guaranteed.

6.2 Recommendations

The proposed motion planning system showed promising results in both localization and optimization. However, various improvements and topics for the future are provided based on the results obtained in this thesis.

- **Observer Validation**
  Predicting future states with the Extended Kalman Filter depends on accurate tire parameters. Unfortunately, a model mismatch was observed, which could not be validated since no lateral velocity measurement was available. Therefore, further implementation of the proposed localization system in Chapter 3 should start with validating the lateral dynamics and a more accurate tire model.

- **Vehicle Dimensions**
  The proposed cost function includes a penalty on the lateral error, tuning the weight on the lateral error can achieve the desired reference tracking behavior. However, the lateral error is defined with respect to the center of gravity of the vehicle and therefore excludes the vehicle's dimensions. During racing, the entire track is used, resulting in the tires exceeding the track limits. Therefore, a penalty should be included, which accounts for the heading of the vehicle and dimensions of the vehicle. In [9] a hard constraint is used to account for the vehicle's dimensions, which can be used as inspiration.

- **Linear Parameter Varying System**
  The work in [24] and [25] showed the computational advantages that can be gained by implementing a linear parameter-varying system. Therefore, it should be considered to implement the proposed control structure from this thesis using a linear parameter-varying system, potentially bridging the gap toward real-time implementation.

- **Robustness**
  In chapter 3, it is observed that the reference trajectory can shift, potentially causing infeasible control problems. An Extended Kalman Filter is used to solve non-smooth control inputs, but the NMPC is not validated including arbitrary errors and time-varying reference trajectories. Therefore, the robustness of the optimization problem should be investigated by using testing data as input to the controller rather than a predefined track with a limited look-ahead distance.

- **Powertrain Constraints**
  In [28] the power limit of the engine is considered a hard constraint. This thesis does not consider the power limits of the electric motors, potentially resulting in an infeasible solution for low-level powertrain control. The first step towards implementation of the powertrain constraints could be similar to [28], but then defined for all four electric motors.

- **Discontinuous look-ahead distance**
  Online implementation has the difficulty of a discontinuous look-ahead distance, which results from driving on a track unknown to SLAM. During the first lap, the look-ahead distance can shift due to the track layout still being constructed, shifting the terminal state and causing a change in desired vehicle input. Simulations should be performed to measure the impact of such discontinuous control input and to define the minimum required horizon length to maintain feasibility.

- **Model Mismatch**
  The results are obtained using a simulation model with vehicle parameters similar to the proposed controller. While performing real-life experiments will cause a model mismatch between the controller and the vehicle. Good track performance is obtained in this thesis, but how the proposed controller reacts to such behavior is not yet examined.

Bibliography

[1] A. Hawkins, *Tesla's 'full self-driving' software is starting to roll out to select customers*, Oct. 2020. [Online]. Available: https://www.theverge.com/2020/10/21/21527577/tesla-full-self-driving-autopilot-beta-software-update.

[2] L. Hermansdorfer, J. Betz and M. Lienkamp, 'Benchmarking of a software stack for autonomous racing against a professional human race driver,' *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020. DOI: 10.1109/ever48776.2020.9242926.

[3] N. Sariff and N. Buniyamin, 'An overview of autonomous mobile robot path planning algorithms,' in *2006 4th Student Conference on Research and Development*, 2006, pp. 183–188. DOI: 10.1109/SCORED.2006.4339335.

[4] L. Claussmann, M. Revilloud, D. Gruyer and S. Glaser, 'A review of motion planning for highway autonomous driving,' *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2020. DOI: 10.1109/TITS.2019.2913998.

[5] J.-P. Skeete, 'The obscure link between motorsport and energy efficient, low-carbon innovation: Evidence from the UK and European Union,' *Journal of Cleaner Production*, vol. 214, pp. 674–684, 2019, ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2019.01.048. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959652619300551.

[6] *How consumers benefit from formula 1 technology*, Oct. 2021. [Online]. Available: https://www.alpha-sense.com/blog/formula-1-technology-benefits/.

[7] *World's first extreme competition of teams developing self-driving ai*. [Online]. Available: https://roborace.com/.

[8] [Online]. Available: https://www.formulastudent.de/fsg/.

[9] J. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan and J. Lygeros, 'Optimization-based hierarchical motion planning for autonomous racing,' 2020. [Online]. Available: https://arxiv.org/abs/2003.04882.

[10] D. Mayne, 'Nonlinear model predictive control: Challenges and opportunities,' *Nonlinear Model Predictive Control*, Allgöwer, Frank, Zheng and Alex, Eds., pp. 23–44, 2000. DOI: 10.1007/978-3-0348-8407-5_2.

[11] J. Sawma, F. Khatounian, E. Monmasson, R. Ghosn and L. Idkhajine, 'The effect of prediction horizons in mpc for first order linear systems,' *2018 IEEE International Conference on Industrial Technology (ICIT)*, pp. 316–321, 2018. DOI: 10.1109/ICIT.2018.8352196.

[12] R. Findeisen and F. Allgöwer, 'An introduction to nonlinear model predictive control,' Jan. 2002.

[13] K. Worthmann, 'Estimates on the prediction horizon length in MPC,' Jan. 2012.

[14] M.-W. Park, S.-W. Lee and W.-Y. Han, 'Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm,' pp. 1443–1447, 2014. DOI: 10.1109/ICCAS.2014.6987787.

[15] J. Yang, H. Bao, N. Ma and Z. Xuan, 'An algorithm of curved path tracking with prediction model for autonomous vehicle,' pp. 405–408, 2017. DOI: 10.1109/CIS.2017.00094.

[16] Q. Yao, Y. Tian, Q. Wang and S. Wang, 'Control strategies on path tracking for autonomous vehicle: State of the art and future challenges,' *IEEE Access*, vol. 8, pp. 161 211–161 222, 2020. DOI: 10.1109/ACCESS.2020.3020075.

[17] K. Kritayakirana and J. Gerdes, 'Using the centre of percussion to design a steering controller for an autonomous race car,' *Vehicle System Dynamics*, vol. 50, no. sup1, pp. 33–51, 2012. DOI: 10.1080/00423114.2012.672842. eprint: https://doi.org/10.1080/00423114.2012.672842. [Online]. Available: https://doi.org/10.1080/00423114.2012.672842.

[18] S. Bagheri, T. Jafarov, L. Freidovich and N. Sepehri, 'Beneficially combining lqr and pid to control longitudinal dynamics of a smartfly uav,' in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2016, pp. 1–6. DOI: 10.1109/IEMCON.2016.7746309.

[19] S. Xu and H. Peng, 'Design, analysis, and experiments of preview path tracking control for autonomous vehicles,' *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48–58, 2020. DOI: 10.1109/TITS.2019.2892926.

[20] M. Schwenzer, 'Review on model predictive control: An engineering perspective,' *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5, pp. 1327–1349, 2021. DOI: 10.1007/s00170-021-07682-3.

[21] S. Srinivasan, S. Nicolas Giles and A. Liniger, 'A holistic motion planning and control solution to challenge a professional racecar driver,' *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7854–7860, 2021. DOI: 10.1109/LRA.2021.3101244.

[22] H. de Carvalho Pinheiro, A. Messana, L. Sisca, A. Ferraris, A. Airale and M. Carello, 'Torque vectoring in electric vehicles with in-wheel motors,' in Jun. 2019, pp. 3127–3136, ISBN: 978-3-030-20130-2. DOI: 10.1007/978-3-030-20131-9_308.

[23] H. Pacejka and I. Besselink, *Tire and Vehicle Dynamics*. Elsevier/BH, 2012.

[24] E. Alcalá, V. Puig and J. Quevedo, 'LPV-MP planning for autonomous racing vehicles considering obstacles,' *Robotics and Autonomous Systems*, vol. 124, p. 103 392, 2020, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2019.103392. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889019304877.

[25] E. Alcalá, V. Puig, J. Quevedo and U. Rosolia, 'Autonomous racing using linear parameter varying-model predictive control (LPV-MPC),' *Control Engineering Practice*, vol. 95, p. 104 270, 2020, ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2019.104270. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066119302187.

[26] S. Abdulsalam, D. Lakomski, Q. Gu, T. Jin and Z. Zong, 'Program energy efficiency: The impact of language, compiler and implementation choices,' in *International Green Computing Conference*, 2014, pp. 1–6. DOI: 10.1109/IGCC.2014.7039169.

[27] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Prison and M. Diehl, 'Nmpc for racing using a singularity-free path-parametric model with obstacle avoidance,' *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 324–14 329, 2020, 21st IFAC World Congress, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.1376. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320317845.

[28] V. Laurense and C. Gerdes, 'Long-horizon vehicle motion planning and control through serially cascaded model complexity,' *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 166–179, 2022. DOI: 10.1109/TCST.2021.3056315.

[29] D. Lam, C. Manzie and M. Good, 'Model predictive contouring control,' in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 6137–6142. DOI: 10.1109/CDC.2010.5717042.

[30] A. Liniger, A. Domahidi and M. Morari, 'Optimization-based autonomous racing of 1:43 scale RC cars,' *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015. DOI: 10.1002/oca.2123.

[31] B. Brito, B. Floor, L. Ferranti and J. Alonso-Mora, 'Model predictive contouring control for collision avoidance in unstructured dynamic environments,' *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, Jul. 2019. DOI: 10.1109/LRA.2019.2929976.

[32] L. Ferranti, B. Brito, E. Pool *et al.*, 'Safevru: A research platform for the interaction of self-driving vehicles with vulnerable road users,' in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1660–1666. DOI: 10.1109/IVS.2019.8813899.

[33] J. Kabzan, M. de la Iglesia Valls, V. Reijgwart *et al.*, 'AMZ driverless: The full autonomous racing system,' *Journal of Field Robotics*, vol. 37/7, pp. 1267–1294, 2020. DOI: 10.1002/rob.21977.

[34] M. Švec, K. Hrvatinić, Š. Ileš and J. Matuško, 'Predictive torque vectoring vehicle control based on a linear time varying model,' in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 960–965. DOI: 10.23919/MIPRO.2019.8756680.

[35] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems*. Springer Science Business Media, 2016.

[36] S. M. LaValle, *Planning Algorithms*. Cambridge Universtiy Press, 2006, http://planning.cs.uiuc.edu/.

[37] *478578 chapter 3*. [Online]. Available: http://www.math.iit.edu/~fass/478578_Chapter_3.pdf.

[38] L.-H. Lyu, *Numerical simulation of space plasmas(1)*, http://www.ss.ncu.edu.tw/~lyu/lecture_files_en/lyu_NSSP_Notes/Lyu_NSSP_AppendixC.pdf, 2016.

[39] *Formula student rules*. [Online]. Available: https://www.formulastudent.de/fsg/rules/.

[40] C. Voser, R. Hindiyeh and J. Gerdes, 'Analysis and control of high sideslip manoeuvres,' *Vehicle System Dynamics*, vol. 48, no. sup1, pp. 317–336, 2010. DOI: 10.1080/00423111003746140.

[41] *Odeset*. [Online]. Available: https://nl.mathworks.com/help/matlab/math/choose-an-ode-solver.html.

[42] R. Butt, *Introduction to numerical analysis using matlab®*. [Online]. Available: https://books.google.nl/books?id=QWub-UVGxqkC&pg=PA11&redir_esc=y#v=onepage&q&f=false.

[43] MATLAB, *version 4.0 (R2021b)*. Natick, Massachusetts: The MathWorks Inc., 2022.

[44] J. Andersson, J. Gillis, G. Horn, J. Rawlings and M. Diehl, 'CasADi – A software framework for nonlinear optimization and optimal control,' *Mathematical Programming Computation*, 2018.

Appendix A

Variable Step Size

In Section 4.4 a simulation study is performed to determine the total error between a RK2 discretization and an ODE45 MatLab solver. In this appendix the error for each state is given, which sums up to the overall error which is given Section 4.4. In Figure A.1 and Figure A.2 the results for the vehicle states of the two-track model are given. In Figure A.3 and Figure A.4 the results for the vehicle states of the point-mass model are given.

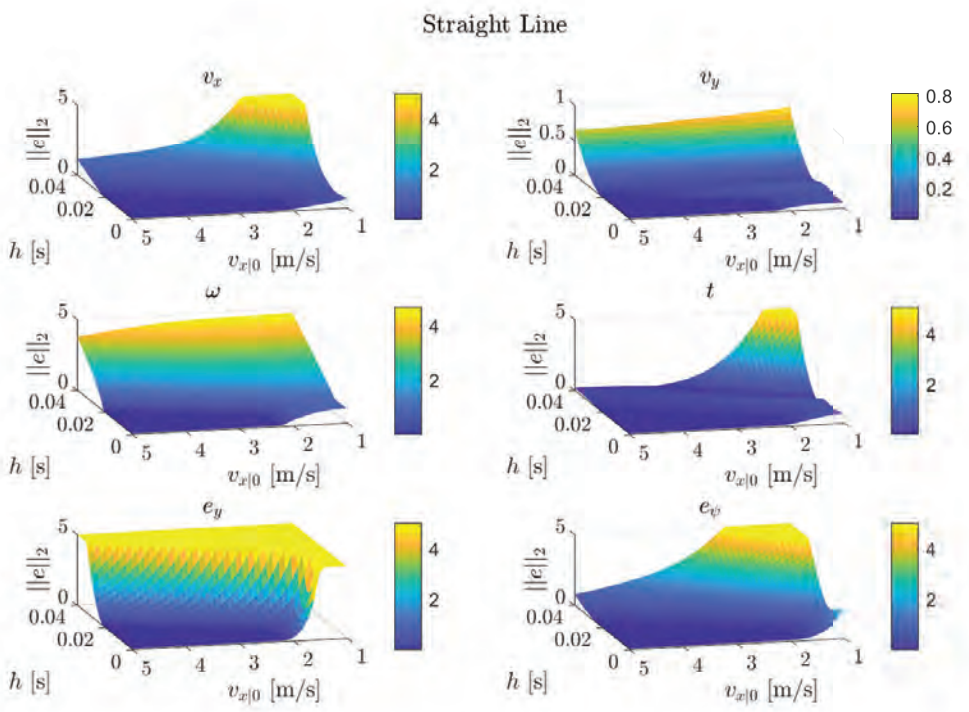

Figure A.1: Error between RK2 and ODE45 for a straight line using a two-track model.

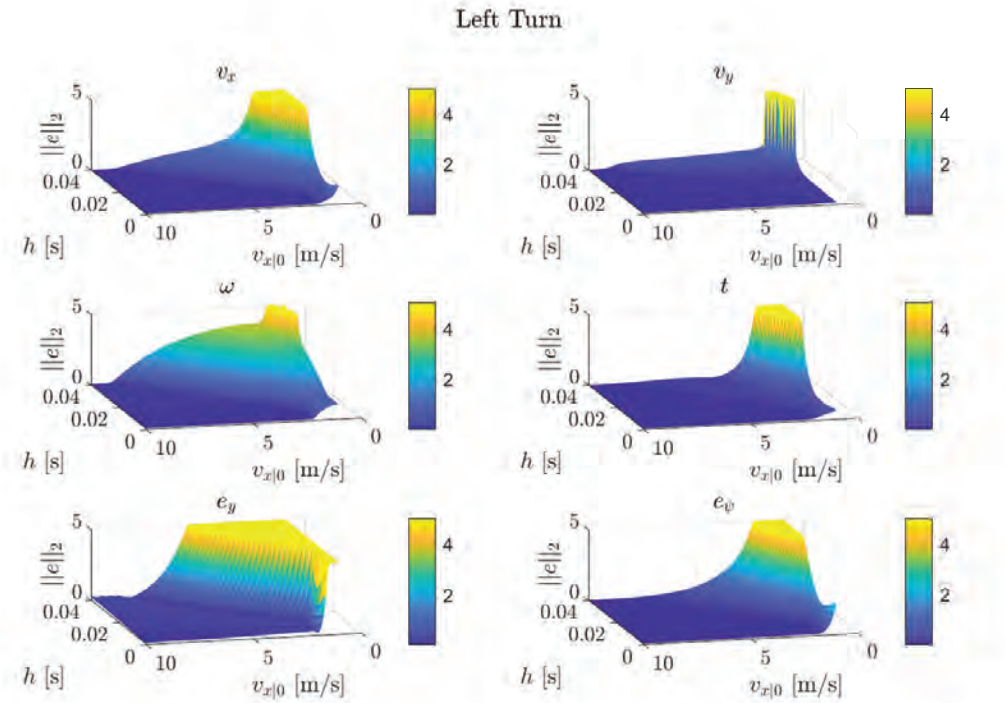

Figure A.2: Error between RK2 and ODE45 for a left turn using a two-track model.

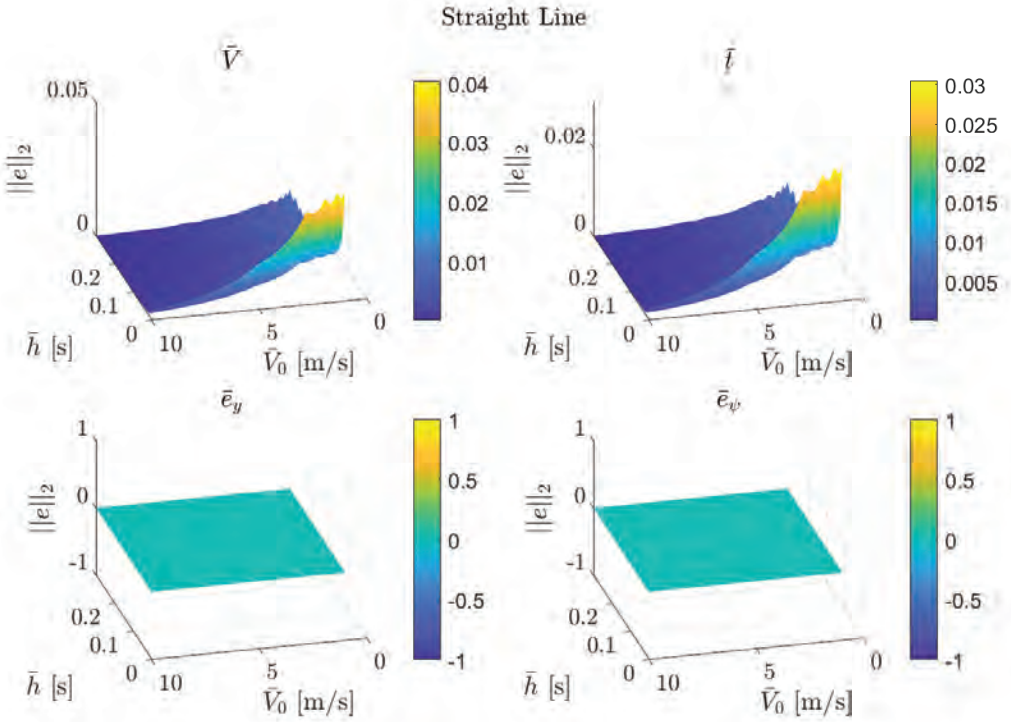

Figure A.3: Error between RK2 and ODE45 for a straight line using a point-mass.

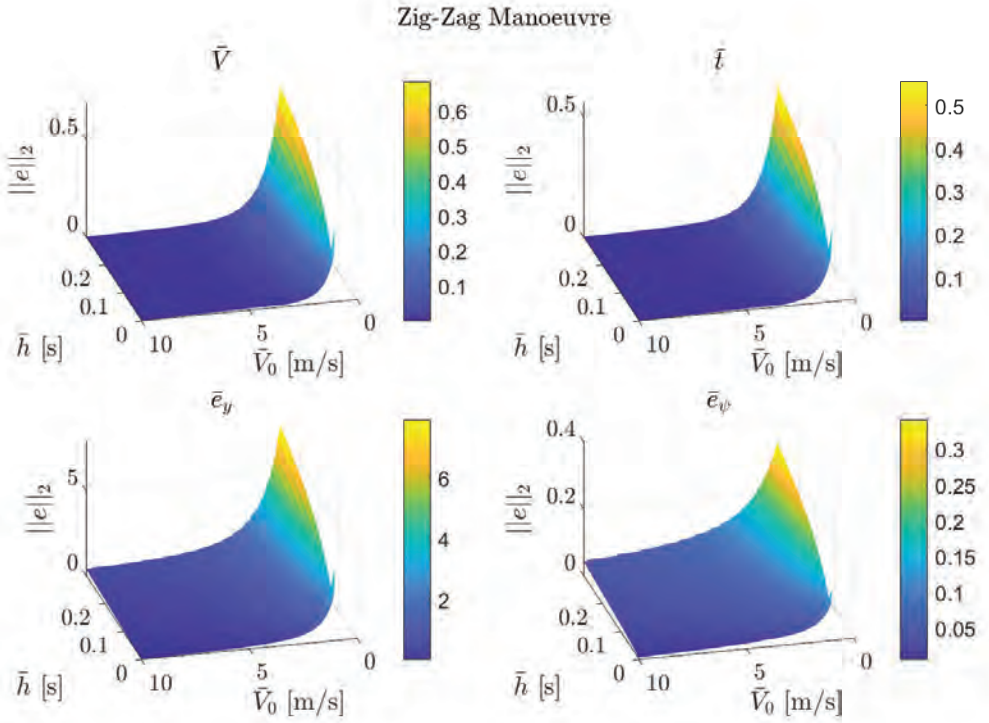

Figure A.4: Error between RK2 and ODE45 for zig-zag manoeuvre using a point-mass.

Appendix B

Extended Kalman Filter Implementation

The Extended Kalman Filter from Chapter 3 is tested via online implementation on the autonomous racing car of University Racing Eindhoven. To provide an idea of how this is implemented, Figure B.1 shows the structure of how the EKF is implemented. The system runs at 100 Hz.

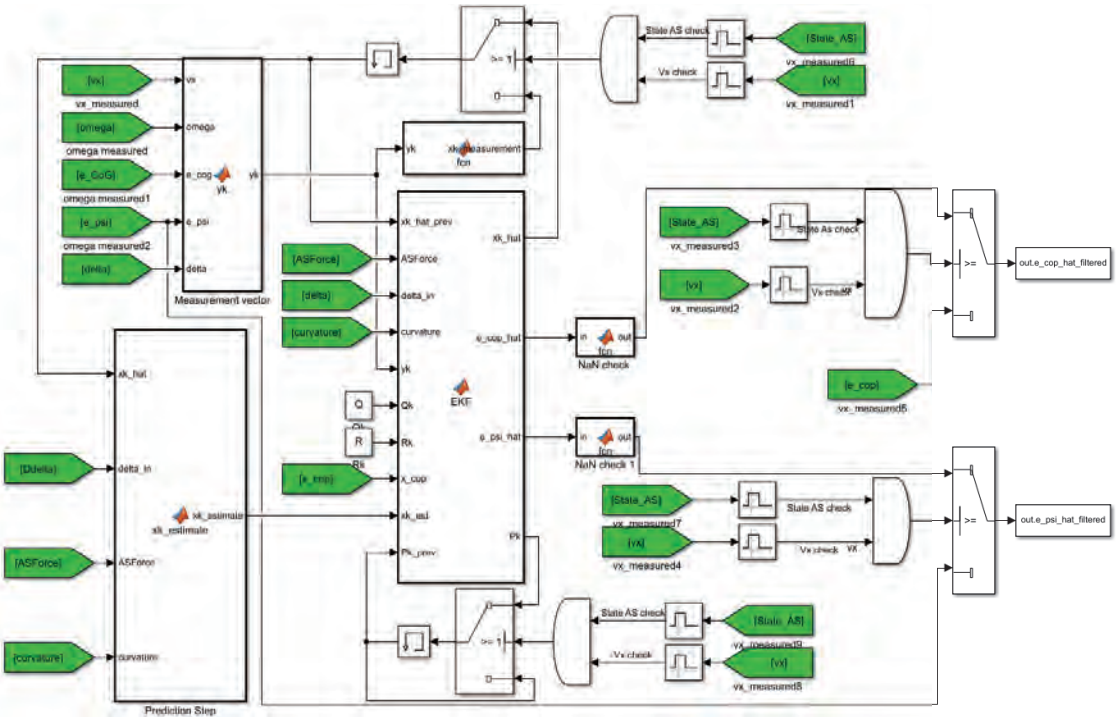

Figure B.1: Extended Kalman Filter implementation in Simulink MatLab 2021b.

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

08/11/2022

Name

Mischa Huisman

ID-number

0971152

Signature

Mischa Huisman

Submit the signed declaration to the student administration of your department.

[i] See: https://www.tue.nl/en/our-university/about-the-university/organization/integrity/scientific-integrity/
The Netherlands Code of Conduct for Scientific Integrity, endorsed by 6 umbrella organizations, including the VSNU, can be found here also. More information about scientific integrity is published on the websites of TU/e and VSNU