

Model Predictive Control for Autonomous and Semiautonomous Vehicles

by

Yiqi Gao

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Francesco Borrelli, Chair
Professor J. Karl Hedrick
Professor Laurent El Ghaoui

Spring 2014

Model Predictive Control for Autonomous and Semiautonomous Vehicles

Copyright 2014
by
Yiqi Gao

Abstract

Model Predictive Control for Autonomous and Semiautonomous Vehicles

by

Yiqi Gao

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Chair

In this thesis we consider the problem of designing and implementing Model Predictive Controllers (MPC) for lane keeping and obstacle avoidance of autonomous or semi-autonomous ground vehicles. Vehicle nonlinear dynamics, fast sampling time and limited computational resources of embedded automotive hardware make it a challenging control design problem. MPC is chosen because of its capability of systematically taking into account nonlinearities, future predictions and operating constraints during the control design stage.

We start from comparing two different MPC based control architectures. With a given trajectory representing the driver intent, the controller has to autonomously avoid obstacles on the road while trying to track the desired trajectory by controlling front steering angle and differential braking. The first approach solves a single nonlinear MPC problem for both replanning and following of the obstacle free trajectories. While the second approach uses a hierarchical scheme. At the high-level, new trajectories are computed on-line, in a receding horizon fashion, based on a simplified point-mass vehicle model in order to avoid the obstacle. At the low-level an MPC controller computes the vehicle inputs in order to best follow the high level trajectory based on a higher fidelity nonlinear vehicle model. Experimental results of both approaches on icy roads are shown. The experimental as well as simulation results are used to compare the two approaches. We conclude that the hierarchical approach is more promising for real-time implementation and yields better performance due to its ability of having longer prediction horizon and faster sampling time at the same time.

Based on the hierarchical approach for autonomous drive, we propose a hierarchical MPC framework for semi-autonomous obstacle avoidance, which decides the necessity of control intervention based on the aggressiveness of the evasive maneuver necessary to avoid collisions. The high level path planner plans obstacle avoiding maneuvers using a special kind of curve, the clothoid. The usage of clothoids have a long history in highway design and robotics control. By optimizing over a small number of parameters, the optimal clothoids satisfying the safety constraints can be determined. The same parameters also indicate the aggressiveness of the avoiding maneuver and thus can be used to decide whether a control intervention is needed before its too late to avoid the obstacle. In the case of control

intervention, the low level MPC with a nonlinear vehicle model will follow the planned avoiding maneuver by taking over control of the steering and braking. The controller is validated by both simulations and experimental tests on an icy track.

In the proposed autonomous hierarchical MPC where the point mass vehicle model is used for high level path replanning, despite of its successful avoidance of the obstacle, the controller's performance can be largely improved. In the test, we observed big deviations of the actual vehicle trajectory from the high level planned path. This is because the point mass model is overly simplified and results in planned paths that are infeasible for the real vehicle to track. To address this problem, we propose an improved hierarchical MPC framework based on a special coordinate transformation in the high level MPC. The high level uses a nonlinear bicycle vehicle model and utilizes a coordinate transformation which uses vehicle position along a path as the independent variable. That produces high level planned paths with smaller tracking error for the real vehicle while maintaining real-time feasibility. The low level still uses an MPC with higher fidelity model to track the planned path. Simulations show the method's ability to safely avoid multiple obstacles while tracking the lane centerline. Experimental tests on an autonomous passenger vehicle driving at high speed on an icy track show the effectiveness of the approach.

In the last part, we propose a robust control framework which systematically handles the system uncertainties, including the model mismatch, state estimation error, external disturbances and etc. The framework enforces robust constraint satisfaction under the presence of the aforementioned uncertainties. The actual system is modeled by a nominal system with an additive disturbance term which includes all the uncertainties. A “Tube-MPC” approach is used, where a robust positively invariant set is used to contain all the possible tracking errors of the real system to the planned path (called the “nominal path”). Thus all the possible actual state trajectories in time lie in a tube centered at the nominal path. A nominal NMPC controls the tube center to ensure constraint satisfaction for the whole tube. A force-input nonlinear bicycle vehicle model is developed and used in the RNMPC control design. The robust invariant set of the error system (nominal system vs. real system) is computed based on the developed model, the associated uncertainties and a predefined disturbance feedback gain. The computed invariant set is used to tighten the constraints in the nominal NMPC to ensure robust constraint satisfaction. Simulations and experiments on a test vehicle show the effectiveness of the proposed framework.

This dissertation is lovingly dedicated to my parents for their eternal and unconditional love, support and encouragement.

Contents

Contents	ii
List of Figures	iv
List of Tables	viii
1 Introduction	1
1.1 Motivation and Background	1
1.2 Thesis Layout	4
2 Vehicle Models	7
2.1 Introduction	7
2.2 Nonlinear Four Wheel Model	10
2.3 Nonlinear Bicycle Model	13
2.4 Tire Models	16
2.5 Linear Bicycle Model	22
2.6 Point Mass Model	24
2.7 Concluding Remarks	24
3 Hierarchical Model Predictive Control	26
3.1 Introduction	26
3.2 Vehicle Model Summary	27
3.3 MPC Control Architectures for An Obstacle Avoiding Path Replanner on Slippery Surface	28
3.4 MPC Controller Formulation	30
3.5 Simulation and Experimental Results	34
3.6 Concluding Remarks on Control Architecture Comparison	41
3.7 Clothoid Based Semi-Autonomous Hierarchical MPC for Obstacle Avoidance	41
3.8 Concluding Remarks	47
4 Spatial Model Predictive Control	49
4.1 Introduction	49
4.2 Vehicle model	50

4.3	MPC control architecture with obstacle avoidance	52
4.4	Simulation and experimental results	54
4.5	Concluding Remarks	58
5	Robust Nonlinear Model Predictive Control	61
5.1	Introduction	61
5.2	Invariant set and robust MPC	62
5.3	Modeling	66
5.4	Invariant set computation	70
5.5	Safety constraints	73
5.6	Robust predictive control design	73
5.7	Simulation results	74
5.8	Experimental results	77
5.9	Conclusions	88
	Bibliography	89

List of Figures

2.1	Vehicle body fixed frame of reference.	8
2.2	Two commonly used frameworks for modeling of vehicle's position and orientation.	9
2.3	Four wheel model notation.	10
2.4	Tire model notations.	12
2.5	Modelling notation depicting the forces in the vehicle body-fixed frame ($F_{x\star}$ and $F_{y\star}$), the forces in the tire-fixed frame ($F_{l\star}$ and $F_{c\star}$), and the rotational and translational velocities.	14
2.6	Lateral and longitudinal tire forces at different values of friction coefficients.	17
2.7	Lateral and longitudinal tire forces in combined cornering and braking/driving, with $\mu = 0.9$	19
2.8	Lateral and longitudinal tire forces as functions of slip ratio and slip angle in combined cornering and braking/driving, with $\mu = 0.9$	20
2.9	Lateral tire forces from the modified Fiala tire model at different levels of braking/driving.	21
2.10	Linearized lateral tire forces in small slip angle region compared to Pacejka model, with $s = 0$ and $\mu = 0.9$	22
3.1	Reference state trajectory of the double lane change path to be followed.	29
3.2	Two different architectures of controller design.	30
3.3	Graphical representation of $d_{k,t,j}$, $p_{x_{t,j}}$ and $p_{y_{t,j}}$	32
3.4	The test vehicle.	36
3.5	Experimental result: Test results of the one-level MPC at vehicle speed 40 Kph. The red box represents the obstacle.	36
3.6	Experimental result: Test results of the two-level MPC at vehicle speed 45 Kph. The red box represents the obstacle.	36
3.7	Experimental result: Test results of the two-level MPC at vehicle speed 55 Kph. The red box represents the obstacle.	37
3.8	Experimental result: Replanned paths from the high-level path replanner in two-level MPC. Vehicle speed is 55 Kph.	37
3.9	Experimental result: Replanned paths from high-level when the communication from high-level to low-level is cut off. Vehicle speed is 55 Kph.	38

3.10 Simulation results: At speed of 70Kph, the one-level MPC becomes unstable even if given enough computation time. While the two-level MPC is able to stabilize the vehicle in the same situation.	40
3.11 Lane change paths with different aggressiveness. Each is generated by connecting four pieces of clothoids. The upper figure shows the shapes of the paths in global frame. The lower figure shows the piecewise affine relation between curvature and curve length for each path.	42
3.12 Ad-hoc Clothoid parameters setup for obstacle constraints.	44
3.13 Simulation result: Various lane change maneuvers are compared. As the vehicle approaches the obstacle the planned paths become more aggressive (high curvatures).	45
3.14 Simulation result: In the upper plot an attentive driver is assumed. The low-level control takes over when the planned path becomes aggressive. In the lower plot the low-level control takes over for a distracted driver and the result is a smoother and safer path.	46
3.15 Experimental results: Vehicle successfully avoids the obstacle using maneuvers based on clothoids.	46
3.16 Experimental result: Actual path of the vehicle deviates from the planned path due to model mismatch and caused infeasibility of tracking. Braking was invoked to enlarge the feasible region in that situation.	47
 4.1 The curvilinear coordinate system. The dynamics are derived about a curve defining the center-line of a track. The coordinate s defines the arc-length along the track. The relative spatial coordinates e_y and e_ψ are shown.	51
4.2 Architecture of the two-level MPC. A spatial vehicle model is used for high-level path planning. A four-wheel vehicle model is used for low-level path tracking.	52
4.3 Simulated result: The vehicle entered the maneuver at 50 kph. The green lines are planned paths from the high-level which are updated every 200 ms. The black line is the actual trajectory the vehicle traveled.	56
4.4 Experimental result: The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the obstacle and continued to track the lane center. The green lines are planned paths from the high-level which were updated every 200 ms. The black line is the actual trajectory the vehicle traveled.	57
4.5 Experimental result: The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the two obstacles and continued to track the lane center.	57
4.6 Experimental result: The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the first obstacle and continued to track the lane center until the second obstacle came into sight. It then turned left to avoid the second obstacle.	58

4.7	Simulation result comparison with the controller proposed in [23]. In both cases the high levels replan every 200ms. The same low level path follower is used, which uses a nonlinear four wheel model and runs every 50ms.	59
5.1	Modeling notation depicting the forces in the vehicle body-fixed frame (F_{x*} and F_{y*}), the forces in the tire-fixed frame (F_{l*} and F_{c*}), and the rotational and translational velocities. The relative coordinates e_y and e_ψ are illustrated on the sketch of the road as well as the road tangent ψ_d	66
5.2	Illustration of center of percussion. p denotes the distance from CoG to CoP . .	68
5.3	Lateral tire force. μ is 0.5 in Pacejka tire model. When β_r varies moderately, F_c behaves similar within the linear region.	69
5.4	One-step model prediction error (equation (5.31)) of 47 testing trials.	72
5.5	Simulation result: The vehicle enters the maneuver at 50Kph. The dashed black line and blue line are the nominal and actual vehicle trajectories respectively. The green dot-dashed lines indicate the robust bounds around the nominal trajectory. The actual vehicle path is seen to be very close to the nominal one and within the robust bounds.	75
5.6	Simulation result: Control inputs during the simulation in figure 5.5. β_f is the braking/throttle ratio for the front wheels and β_r for the rear wheels. We enable both braking and throttle on the wheels in simulation.	76
5.7	Simulation result: Trajectories of nominal MPC controlled system under a random external disturbance.	77
5.8	Simulation result: Trajectories of RNMPC controlled system under a random external disturbance.	77
5.9	Simulation result: Trajectories of RNMPC controlled system under unknown and constantly changing tire road friction coefficients.	78
5.10	The test vehicle.	79
5.11	Experimental result: The vehicle enters the maneuver at 50Kph. The dashed black line and blue line are the nominal and actual vehicle trajectories respectively. The green dot-dashed lines indicate the robust bounds around the nominal trajectory. The actual vehicle path is seen to be very close to the nominal one and within the robust bounds. $\mu \simeq 0.3$	80
5.12	Experimental result: A plot of 4 states of the vehicle, $[\dot{y}, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.11. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.	80
5.13	Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.11. The braking is applied by the controller at the beginning of the maneuver to reduce the speed so that the avoiding maneuver can be more smooth.	81

5.14 Experimental result: The vehicle enters the maneuver at 80Kph. The dashed black line and blue line are the nominal and actual vehicle trajectories respectively. The green dot-dashed lines indicate the robust bounds around the nominal trajectory. The actual vehicle path is seen to be very close to the nominal one and within the robust bounds. $\mu \simeq 0.3$.	81
5.15 Experimental result: A plot of 4 states of the vehicle, $[\dot{y}, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.14. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.	82
5.16 Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.14.	82
5.17 Experimental result: Avoiding a moving obstacle. Vehicle speed is 50 Kph, obstacle moves at 18 Kph. The obstacle position at time t_0 , t_1 and t_3 during the test are shown in dash red. The vehicle position at the corresponding times are marked with blue circles. $\mu \simeq 0.3$.	83
5.18 Experimental result: A plot of 4 states of the vehicle, $[\dot{y}, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.19. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.	84
5.19 Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.19.	84
5.20 Experimental result: Vehicle trajectory. The vehicle enters the maneuver at 50Kph. $\mu \simeq 0.3$.	85
5.21 Experimental resultA plot of 4 states of the vehicle, $[\dot{y}, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.20. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.	85
5.22 Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.20.	86
5.23 Experimental result: Vehicle trajectory on an ice track. The vehicle enters the maneuver at 35Kph. The actual μ on the track is 0.1, while the controller is set up for $\bar{\mu} = 0.3$ on snow track.	86
5.24 Experimental result: A plot of 4 states of the vehicle, $[\dot{y}, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.23. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.	87
5.25 Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.23.	87

List of Tables

1.1	U.S.A. Traffic Fatality Data Report	1
3.1	Comutation Time of One-level and Two-level MPC at Vehicle Speed of 40 Kph.	39
5.1	Simulation Parameters	75
5.2	Real-Time Design Parameters for Experimental Test	78

Acknowledgments

My greatest thanks go to Prof. Francesco Borrelli, my adviser and mentor. Without his help and guidance, I wouldn't have accomplished this work. More importantly, his passion, meticulousity, diligence and wisdom always inspire and inspirit me. These influences are the greater gifts for me to take into my life and work.

I want to express my most sincere thanks to Eric Tseng for his valuable suggestions in my research and tremendous help in the testing. I want to thank all the Professors from inside and outside the Mechanical Engineering Department who shared their knowledge and experience with me, especially, Prof. Karl Hedrick. A lot of thanks also go to Mitch McConnell and Vladimir Ivanovic from Ford Motor company for their valuable help in conducting the tests.

My special thanks go to my colleagues and friends Theresa Lin, Andrew Gray and Ashwin Carvalho for their accompany through the hardest and happiest time. I also want to thank Miroslav Baric and Yudong Ma for their sharing as seniors through the starting years of my Ph.D. life. Thanks to all the friends in or used to be in MPC lab, the birthday cake is sweet!

I thank all the friends that walked into my everyday life for sharing the happiness and sadness and bringing the feeling of home when far from it.

Chapter 1

Introduction

1.1 Motivation and Background

Despite the development of numerous vehicular safety features over the past decades, the traffic accidents still claim tens of thousands of lives on the roads in USA. Worldwide, there was approximately 1.24 million deaths occurred on the roads in 2010. Table 1.1 reports the motorists and non-motorists killed in traffic crashes in the US each year from 2009 to 2011 posted by the National Highway Traffic Safety Administration (NHTSA) under the Fatality Analysis Reporting System (FARS) [50]. FARS is a nationwide census who provides NHTSA, Congress and the American public yearly reports on fatal injuries suffered in motor vehicle traffic crashes. As shown in Table 1.1, despite the small decrement over the past years, the numbers are still impressive. By analyzing data from vehicles instrumented with cameras, several studies have estimated that driver distraction contributes to 22 – 50% of all crashes [33, 58].

Table 1.1: U.S.A. Traffic Fatality Data Report

Year	Fatal Crashes	Fatality	Vehicle Miles Traveled	Fatality Rate ¹
2011	29,757	32,367	2946 Billions	1.10
2009	30,296	32,999	2967 Billions	1.11
2008	30,862	33,883	2957 Billions	1.15

As an effort of making driving safer, the development of vehicle active safety systems dates back to the eighties. Examples include the Anti-lock Braking System (ABS) [10], the traction control (TC) system [42, 6], and the Electronic Stability Programm [39, 60] (ESP, also known under different acronyms such as VSC for Vehicle Stability Control and IVD for Interactive Vehicle Dynamics). These “standard active safety systems” aim at assisting and

¹Per 100 Million Vehicle Miles of Travel

improving the driver's control over the vehicle by avoiding undesired situations, such as wheel locking in braking (ABS), tire slipping (TC), and loss of steering control (ESP). While these systems' effectiveness in general has been widely acknowledged, they can offer little help when the driver is inattentive, which unfortunately contributes to 22 – 50% of all crashes according to some studies[33, 58]. More recent developments in “advanced active safety systems” introduce additional sensors such as onboard cameras, radars, infrared sensors etc., and additional actuators such as active steering or active suspensions. Vehicles equipped with these systems are able to identify obstacles on the road such as pedestrians as well as the lane markers. They can perform emergency maneuvers (mostly emergency braking) to avoid collisions or apply assist steering to avoid lane departures. An example of vehicle with advanced active safety systems is the Volvo S60 equipped with cameras and radars in order to implement a fully autonomous braking [11, 40]. An even more ambitious approach towards driving safety is the development of autonomous driving systems. These systems aim at driving the vehicle fully autonomously by controlling the steering, braking and throttling, examples include the Darpa Grant Challenge Car [3, 48] and the Google Car [29, 38].

This thesis focuses on Model Predictive Control (MPC) of autonomous and semiautonomous vehicles. MPC is the only control technology that can systematically take into account the future predictions and system operating constraints in design stage [46, 49]. That makes it a suitable choice for autonomous drive where the system faces dynamically changing environment and has to satisfy crucial safety constraints (such as obstacle avoidance) as well as actuator constraints.

In MPC, a model of the plant is used to predict the future evolution of the system. Based on this prediction, at each time step t , a performance index is optimized under operating constraints with respect to a sequence of future input moves. The first of such optimal moves is the control action applied to the plant at time t . At time $t + 1$, a new optimization is solved over a shifted prediction horizon. In general, the following finite horizon optimization problem is solved at each time step t :

$$\min_{U_t} J_N(\bar{\xi}_t, U_t) = \sum_{k=t}^{t+H_{p,hl}-1} Cost(\xi_{k,t}, u_{k,t}) \quad (1.1a)$$

$$\text{subj. to } \xi_{k+1,t} = f(\xi_{k,t}, u_{k,t}) \quad k = t, \dots, t + N - 1, \quad (1.1b)$$

$$\xi_{k,t} \in \Xi_t \quad k = t, \dots, t + N, \quad (1.1c)$$

$$u_{k,t} \in \mathcal{U} \quad k = t, \dots, t + N - 1, \quad (1.1d)$$

$$\xi_{t,t} = \xi(t), \quad (1.1e)$$

where the symbol $v_{k,t}$ stands for “the variable v at time k predicted at time t ”, N is the prediction horizon. ξ is the state of the system and in this thesis includes the lateral and longitudinal vehicle velocities, yaw and yaw rate, and vehicle position. u is the control input to the system and in this thesis includes the steering angle, braking torque and throttle. The function $f(\xi, u)$ in (1.1b) allows to predict the future system states based on the current

states and control inputs. (1.1c) and (1.1d) are the state and input constraints the controller has to respect. The cost in (1.1a) can be any performance index and in this thesis can include penalties on state tracking error, penalties on inputs, penalties on input change rate and etc.

The earliest implementations of MPC were on “slow” plants such as those in process industry [52, 12, 5], where the plant dynamics are sufficiently “slow” for the on-line computation needed in implementation. Parallel advances in theory and computing systems have enlarged the range of applications where real-time MPC can be applied [7, 8, 31, 22, 64, 13]. Yet, for a wide class of fast applications, the computational burden of Nonlinear MPC (NMPC) is still a barrier. As an example, in [9] an NMPC has been implemented on a passenger vehicle for path following via an Active Front Steering (AFS) system at 20 Hz, by using the state of the art optimization solvers and rapid prototyping systems. It is shown that the real-time execution is limited to low vehicle speeds on icy roads, because of its computational complexity.

This thesis focuses on developing nonlinear model predictive controllers for autonomous and semiautonomous driving, particularly for obstacle avoidance and lane keeping. It will show that by proper modeling and formulation, the autonomous and semiautonomous driving can be achieved with NMPC on standard rapid prototyping platform with limited computational resource. In particular, the main contributions of this thesis include:

1. The design of a hierarchical MPC scheme for autonomous obstacle avoidance and lane keeping. The autonomous driving problem is decomposed into two levels, a high level path planner/replanner which uses an MPC with point mass vehicle model to plan obstacle free paths with long prediction horizons, and a low level path follower which uses an MPC with a higher fidelity vehicle model to follow the planned path with short prediction horizons. The decomposition of the problem reduces the computational complexity compared to a single level MPC approach and results in improved performance in real-time implementation.
2. The design of a hierarchical MPC scheme for a semi-autonomous active safety system that is able to determine the time of intervention and avoid upcoming collisions with obstacles. The high level MPC uses a special kind of curve, the clothoid, to plan obstacle avoiding maneuvers. The use of clothoid makes it possible to plane feasible vehicle trajectories with few parameters and thus with low computational burden. The parameters also give a measure to the necessary aggressiveness of the avoiding maneuver and thus can be used to determine whether a control intervention is necessary. In the case of intervention, the low level MPC with a higher fidelity vehicle model will follow the high level planned path by controlling the steering and braking.
3. The design of a spatial predictive controller for autonomous obstacle avoidance and lane keeping. It is an improved approach over the point mass model hierarchical MPC. The hierarchical scheme as well as the low level path follower are kept the same. The high level uses a nonlinear bicycle vehicle model and utilizes a coordinate transformation which uses vehicle position along a path as the independent variable. That results in

high level planned paths that are more trackable for the real vehicle while maintaining real-time feasibility and thus improves the performance of overall system.

4. The design of a robust nonlinear predictive control framework based on the “Tube-MPC” approach. The framework presents a systematic way of enforcing robust constraint satisfaction under the presence of model mismatch and disturbances during the MPC design stage. A force-input nonlinear bicycle vehicle model is developed and used in the RNMPC control design. A robust invariant set is used to tighten the constraints in order to guarantee that state and input constraints are satisfied in the presence of disturbances and model error.

1.2 Thesis Layout

The remainder of this thesis is structured as follows:

- Chapter 2 describes the vehicle models used in control design and simulations for autonomous and semiautonomous vehicles. The tire ground reaction forces are the only external forces acting on the vehicle, if exclude the gravity force and the much less significant aerodynamic forces. Unfortunately, the tire forces are highly nonlinear in part of the operational region and the lateral and longitudinal tire dynamics are highly coupled in the same region. That is the most important source of the nonlinear behavior of vehicle dynamics and coupling of lateral and longitudinal vehicle dynamics. Simplifications can be made at a cost of greater model mismatch and restrictions on operation range. The proper choice over various models is an important aspect in the MPC design.

Three tire models are introduced: 1, the Pacejka model, a complex nonlinear semi-empirical tire model being able to describe the nonlinear and coupling behavior of tire forces under wide operation range; 2, the modified Fiala model, a simplified nonlinear tire model which also captures the nonlinear and coupling behavior; 3, the linear tire model only for pure cornering or pure braking/throttling.

Using rigid body dynamics and the tire models, four vehicle models with various fidelities and complexities are introduced: 1, the nonlinear four wheel vehicle model which captures both lateral and longitudinal vehicle dynamics and models the tire forces from each of the four tires using the Pacejka model; 2, the nonlinear bicycle model which is simplified from the nonlinear four wheel model by lumping the left and right tires together at the front and rear wheel axels; 3, the linear bicycle model which is further simplified from the nonlinear bicycle model by assuming constant longitudinal velocity and the use of linear tire model for pure cornering; 4, the simplest point mass model, where the vehicle is treated as a point with a given mass and no tire model is used except a constraint that the tire forces being inside the friction circle.

- Chapter 3 compares two approaches to the autonomous obstacle avoidance problem, the one-level MPC approach and the hierarchical two-level MPC approach. In the first approach, the problem is formulated in a single nonlinear MPC problem. In the second approach, the controller design is decomposed into two levels. The high-level NMPC uses a simplified point-mass vehicle model to replan the obstacle free path with long prediction horizon. The planned trajectory is fed to the low-level NMPC which uses the higher fidelity nonlinear four wheel model to follow the planned paths with shorter prediction horizon. Simulation and experimental results on an icy-snow handling track is used to compare the performance and computation time of both approaches. Conclusions are made that the hierarchical MPC is more promising for real-time implementation and has better performance due to its lower computational complexity.

Chapter 3 also proposes a hierarchical approach for semi-autonomous obstacle avoidance. Where the high level path planner utilizes a special kind of curve, the clothoid, to plan the smoothest path avoiding the obstacle. Very few parameters are needed to determine the planned path, which is obstacle free and feasible for the vehicle to track. That results in low computational complexity for the high level path planner. The parameters also give a measure of the aggressiveness of the avoiding maneuver, providing a suitable indication of whether an intervention from the controller is needed to keep the vehicle safe. When intervention is decided, the low level NMPC path follower takes control of the steering and braking to follow the planned avoiding maneuver.

- Chapter 4 proposes a spatial predictive control framework for autonomous obstacle avoidance and lane keeping. It is an improved approach over the point mass model hierarchical MPC. The hierarchical scheme as well as the low level path follower are kept the same. The high level uses a nonlinear bicycle vehicle model and utilizes a coordinate transformation which uses vehicle position along a path as the independent variable. That results in high level planned paths that are more trackable for the real vehicle while maintaining real-time feasibility and thus improves the performance of overall system. The proposed frameworks is validated by experiments on the same icy-snow handling track.
- Chapter 5 proposes a robust nonlinear predictive control framework based on the “Tube-MPC” approach. The framework presents a systematic way of enforcing robust constraint satisfaction under the presence of model mismatch and disturbances during the MPC design stage. The basic idea is to use a control law of the form $u = \bar{u} + K(\xi - \bar{\xi})$ where \bar{u} and $\bar{\xi}$ are the nominal control input and system states trajectories determined by a nominal NMPC. For a given linear controller K , a robust invariant set is computed for the error system. The invariant is used to bound the maximum deviation of the actual states from the nominal states under the linear control action K . The nominal MPC optimises \bar{u} and $\bar{\xi}$ with tightened state and input constraints. The tightening is computed as a function of the bounds derived from

the robust invariant set. A force-input nonlinear bicycle vehicle model is developed and used in the RNMPC control design. The proposed robust NMPC framework is validated by both simulation and experiments.

Chapter 2

Vehicle Models

2.1 Introduction

In this chapter, we present several dynamic vehicle models used for control design and simulation in this thesis. The modeling of vehicle dynamics has been extensively studied in the past decades. A wide spectrum of vehicle models has appeared in the rich literature [53] [32]. We present a set of models which is suitable for the proposed model based control designs.

Figure 2.1 shows a sketch of vehicle body fixed frame of reference and the forces acting on vehicle's Center of Gravity (CoG). x , y and z are the vehicle's longitudinal, lateral and vertical axes, respectively. \dot{x} , \dot{y} and $\dot{\psi}$ are the longitudinal, lateral velocities and yaw rate. F_x , F_y and M_z are the longitudinal, lateral forces acting on the vehicle CoG and the rotating moment about z axes.

The rigid body dynamic equations are:

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_x, \quad (2.1a)$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_y, \quad (2.1b)$$

$$I_z\ddot{\psi} = M_z, \quad (2.1c)$$

where m is the vehicle mass and I is the vehicle's moment of inertia about z axis.

We use different choices of modeling vehicle's position and orientation in inertial frames. Figure 2.2 shows two commonly used setups for modeling the vehicle's planner motion in inertial frames.

Setup one (Figure 2.2(a)) uses a cartesian coordinate system (X-Y coordinate) for the inertial frame of reference. In Figure 2.2(a), x and y axes are the body fixed coordinate system same as in Figure 2.1. The angle ψ is the vehicle's heading angle, or yaw angle, in the inertial frame $X - Y$. ψ is defined positive counterclockwise with $\psi = 0$ when x axis is aligned with X axis. The vehicle's 2-D motion in $X - Y$ coordinate can be modeled as follows:

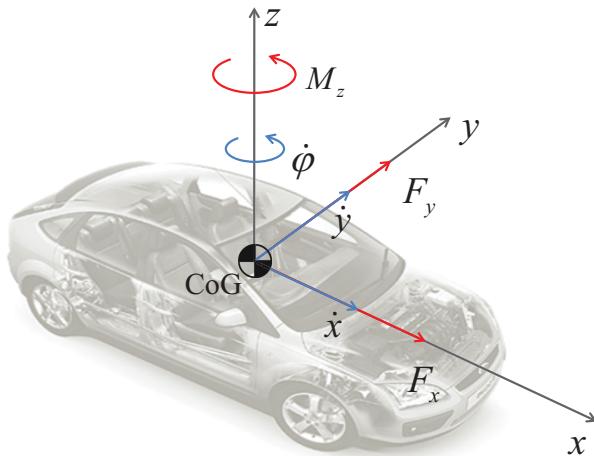


Figure 2.1: Vehicle body fixed frame of reference.

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi, \quad (2.2a)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi. \quad (2.2b)$$

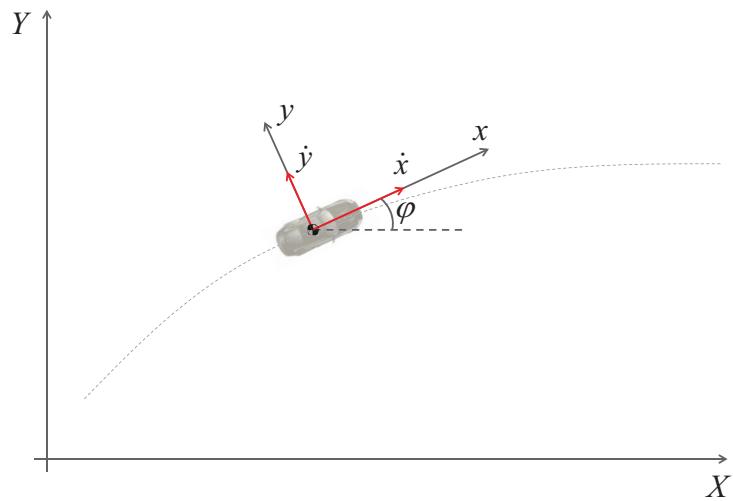
Setup two (Figure 2.2(b)) uses a curve linear coordinate system $s - e_y$ to present the vehicle's position with respect to a given curve, usually a lane the vehicle is following. This setup is particularly useful in environments such as highways and city streets [53]. As shown in Figure 2.2(b), e_y is the vehicle's lateral deviation from the lane center and s is the vehicle's longitudinal position along the lane center. The angle e_ψ is the vehicle's heading angle with respect to the lane direction. e_ψ is defined positive counterclockwise with $e_\psi = 0$ when x axis is aligned with the tangent of lane center. The vehicle's 2-D motion in this setup is modeled as follows:

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \quad (2.3a)$$

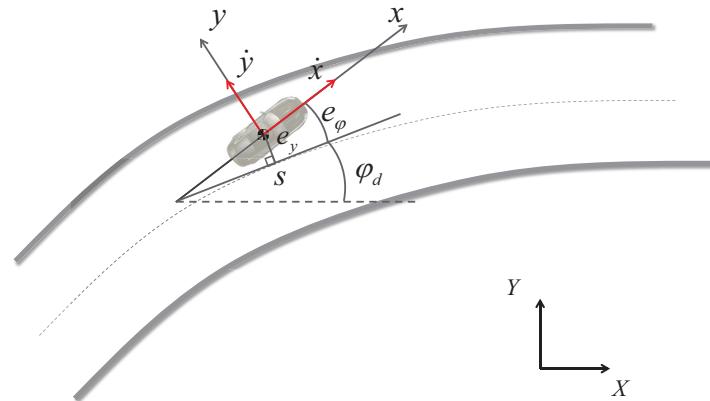
$$\dot{s} = \frac{1}{1 - \kappa e_y} \cdot (\dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi)), \quad (2.3b)$$

$$\dot{e}_y = \dot{x} \sin(e_\psi) + \dot{y} \cos(e_\psi), \quad (2.3c)$$

where $\dot{\psi}$ is the vehicle's yaw rate. ψ_d is the angle of the tangent to the lane centerline in a fixed coordinate frame. κ is the curvature of the lane and is defined positive counterclockwise.



(a) Setup one: Inertial frame of reference with cartesian coordinate system.



(b) Setup two: Track based coordinate system.

Figure 2.2: Two commonly used frameworks for modeling of vehicle's position and orientation.

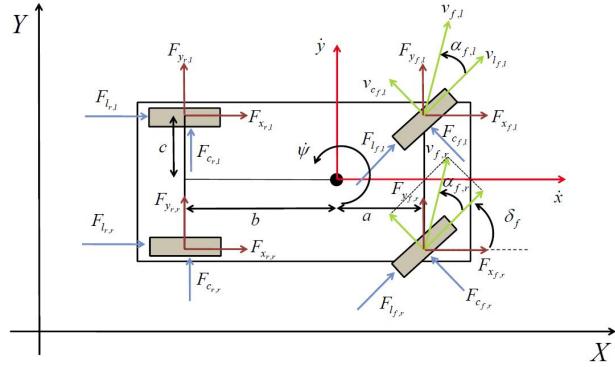


Figure 2.3: Four wheel model notation.

In the following sections, we particularize the equations (2.1) to (2.3) to define different vehicle models with various levels of details. In particular, section 2.2 presents a nonlinear four wheel vehicle model where the tire forces from each of the four tires are modeled as nonlinear functions of the vehicle states; Section 2.3 presents a nonlinear bicycle model which is simplified from the nonlinear four wheel model by lumping the left and right tires together at the front and rear wheel axles; Section 2.5 presents a linear bicycle model which is further simplified from the nonlinear bicycle; Section 2.6 presents the simplest point mass model, where the vehicle is treated as a point with a given mass and no tire model is used with the exception of the tire forces being constrained inside the friction circle.

2.2 Nonlinear Four Wheel Model

In this section, we present a four wheel vehicle model whose dynamic equation is compactly written as:

$$\dot{\xi}(t) = f^{4w}(\xi(t), u(t)) \quad (2.4)$$

where $\xi(t) \in \mathbb{R}^n$ is the state of the system and $u(t) \in \mathbb{R}^{m_r}$ is the input, $n = 10$ is the number of states and $m_r = 5$ is the number of inputs. The ten states are lateral and longitudinal velocities in the body frame, the yaw angle, yaw rate, lateral and longitudinal vehicle coordinates in the inertial frame and the angular velocity on the four wheels. These are denoted respectively as $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X, \omega_{f,l}, \omega_{f,r}, \omega_{r,l}, \omega_{r,r}]'$. Where $\omega_{f,l}, \omega_{f,r}, \omega_{r,l}$ and $\omega_{r,r}$ denotes the angular velocity on the front left, front right, rear left and rear right wheels, respectively. For the simplicity of notation, later we will use the subscript $\star \in \{f, r\}$ to denote the front or rear axles and $\bullet \in \{l, r\}$ to denote the left or right side of the vehicle. The five inputs are $u = [T_f, T_{f,l}, T_{f,r}, T_{r,l}, T_{r,r}]'$ where δ_f is the front steering angle and $T_{\star,\bullet}$ are the braking and tractive torque on the four wheels. Positive $T_{\star,\bullet}$ denotes driving torque while negative $T_{\star,\bullet}$ denotes braking torque.

Figure 2.3 defines the notations of the four wheel model. In particular, $F_{c_{\star,\bullet}}$ and $F_{l_{\star,\bullet}}$ are the lateral (cornering) and longitudinal tire forces in tire frame. $F_{y_{\star,\bullet}}$ and $F_{x_{\star,\bullet}}$ are the components of the tire forces along the lateral and longitudinal vehicle axes. $\alpha_{\star,\bullet}$ are the wheel slip angles which are defined later in this section. δ_f is the front wheel steering angle. a and b are the distances from the CoG to the front and rear axles and c is the distance from the CoG to the left/right side at the wheels.

The dynamics in (2.4) can be derived by using the equations of motion about the vehicles Center of Gravity (CoG) and coordinate transformations between the inertial frame and the vehicle body frame:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}}, \quad (2.5a)$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}}, \quad (2.5b)$$

$$\begin{aligned} I\ddot{\psi} = & a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + \\ & + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}), \end{aligned} \quad (2.5c)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi, \quad (2.5d)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi, \quad (2.5e)$$

$$I_w \dot{\omega}_{f,l} = -F_{l_{f,l}} r_w + T_{f,l} - b_w \dot{\omega}_{f,l}, \quad (2.5f)$$

$$I_w \dot{\omega}_{f,r} = -F_{l_{f,r}} r_w + T_{f,r} - b_w \dot{\omega}_{f,r}, \quad (2.5g)$$

$$I_w \dot{\omega}_{r,l} = -F_{l_{r,l}} r_w + T_{r,l} - b_w \dot{\omega}_{r,l}, \quad (2.5h)$$

$$I_w \dot{\omega}_{r,r} = -F_{l_{r,r}} r_w + T_{r,r} - b_w \dot{\omega}_{r,r}, \quad (2.5i)$$

where the constant m is the vehicle's mass. I is the vehicle's rotational inertia about the z axis. I_w includes the wheel and driveline rotational inertias. r_w is the radius of the wheel. b_w is the damping coefficient.

The x and y components of tire forces, $F_{x_{\star,\bullet}}$ and $F_{y_{\star,\bullet}}$, are computed as follows:

$$F_{y_{\star,\bullet}} = F_{l_{\star,\bullet}} \sin \delta_{\star} + F_{c_{\star,\bullet}} \cos \delta_{\star}, \quad (2.6a)$$

$$F_{x_{\star,\bullet}} = F_{l_{\star,\bullet}} \cos \delta_{\star} - F_{c_{\star,\bullet}} \sin \delta_{\star}. \quad (2.6b)$$

Assumption 1 Only the steering angle at the front wheels can be controlled. Moreover, the front left and front right wheel steering angles are assumed to be the same. i.e., $\delta_{f,l} = \delta_{f,r} = \delta_f$ and $\delta_{r,\bullet} = 0$.

The lateral and longitudinal tire forces $F_{l_{\star,\bullet}}$ and $F_{c_{\star,\bullet}}$, as shown in Figure 2.4, are given by:

$$F_{c_{\star,\bullet}} = f_c(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu, F_{z_{\star,\bullet}}), \quad (2.7a)$$

$$F_{l_{\star,\bullet}} = f_l(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu, F_{z_{\star,\bullet}}), \quad (2.7b)$$

where α is the slip angle of the tire, s is the slip ratio, μ is the friction coefficient and F_z is the normal force. We use a Pacejka tire model [51] to model F_c and F_l in (2.7) at the

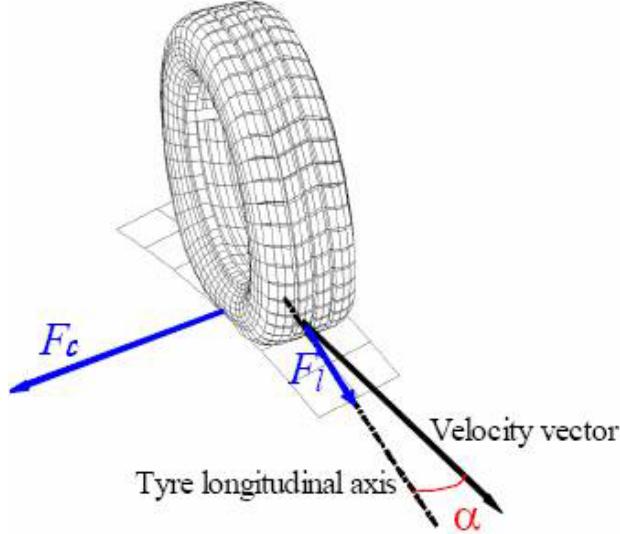


Figure 2.4: Tire model notations.

four tires. This complex, semi-empirical model is able to describe the tire behavior over the linear and nonlinear operating ranges of slip ratio, tire slip angle and friction coefficient.

As shown in Figure 2.4, the tire slip angle $\alpha_{*,\bullet}$ in (2.7) denotes the angle between the tire velocity and its longitudinal direction. It can be expressed as:

$$\alpha_{*,\bullet} = \arctan \frac{v_{c_{*,\bullet}}}{v_{l_{*,\bullet}}}, \quad (2.8)$$

where $v_{c_{*,\bullet}}$ and $v_{l_{*,\bullet}}$ are the lateral and longitudinal wheel velocities computed from:

$$v_{c_{*,\bullet}} = v_{y_{*,\bullet}} \cos \delta_{*} - v_{x_{*,\bullet}} \sin \delta_{*}, \quad (2.9a)$$

$$v_{l_{*,\bullet}} = v_{y_{*,\bullet}} \sin \delta_{*} + v_{x_{*,\bullet}} \cos \delta_{*}, \quad (2.9b)$$

$$v_{y_{f,\bullet}} = \dot{y} + a\dot{\psi} \quad v_{x_{*,l}} = \dot{x} - c\dot{\psi}, \quad (2.9c)$$

$$v_{y_{r,\bullet}} = \dot{y} - b\dot{\psi} \quad v_{x_{*,r}} = \dot{x} + c\dot{\psi}. \quad (2.9d)$$

The slip ratio is defined as:

$$s_{*,\bullet} = \begin{cases} \frac{r\omega_{*,\bullet}}{v_{l_{*,\bullet}}} - 1 & \text{if } v_{l_{*,\bullet}} > r\omega_{*,\bullet}, v \neq 0 \text{ for braking,} \\ 1 - \frac{v_{l_{*,\bullet}}}{r\omega_{*,\bullet}} & \text{if } v_{l_{*,\bullet}} < r\omega_{*,\bullet}, \omega \neq 0 \text{ for driving.} \end{cases} \quad (2.10)$$

The friction coefficient μ is assumed to be a known constant and is the same at all four wheels.

We use the static weight distribution to estimate the normal force on the wheel, $F_{z_{\star,\bullet}}$ in Equation (2.7). They are approximated as:

$$F_{z_{f,\bullet}} = \frac{bmg}{2(a+b)}, \quad F_{z_{r,\bullet}} = \frac{amg}{2(a+b)}, \quad (2.11)$$

where g is the gravitational acceleration.

Remark 1 For control purposes, the wheel dynamics (2.5f)-(2.5i) can be neglected. In this case the wheel speed can be assumed to be measured at each sampling time and is kept constant until the next available update.

Remark 2 The model (2.5) uses reference framework one (2.2) for position representation. Note one can use the reference framework two (2.3) without affecting the dynamics. i.e. (2.5d)-(2.5e) in (2.5) can be replaced by (2.3a)-(2.3c).

2.3 Nonlinear Bicycle Model

In this section, we present a reduced order model from the four wheel model (2.4), the single track model or "bicycle model" [41]. It is based on the following simplification:

Simplification 1 At the vehicle front and rear axles, the left and right wheels are lumped together.

The dynamic equations of the bicycle model is compactly written as:

$$\dot{\xi}(t) = f^{nb}(\xi(t), u(t)) \quad (2.12)$$

where the superscript "nb" stands for "nonlinear bicycle". $\xi(t) \in \mathbb{R}^n$ is the state of the system and $u(t) \in \mathbb{R}^{m_r}$ is the input, $n = 6$ is the number of states and $m_r = 3$ is the number of inputs. The six states are lateral and longitudinal velocities in the body frame, the yaw angle, yaw rate, lateral and longitudinal vehicle coordinates in the inertial frame. These are denoted respectively as $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X]'$. The three inputs are $u = [\delta_f, T_f, T_r]'$ where δ_f is the front steering angle and T_\star is the braking and tractive torque on the wheel. Positive T_\star denotes tractive torque while negative T_\star denotes braking torque.

Figure 2.5 shows the modeling notation of the bicycle model. In particular, F_{c_\star} and F_{l_\star} are the lateral (cornering) and longitudinal tire forces in the tire frame. F_{y_\star} and F_{x_\star} are the components of the tire forces along the lateral and longitudinal vehicle axes. δ_f is the front wheel steering angle, a and b are the distances from the CoG to the front and rear axles.

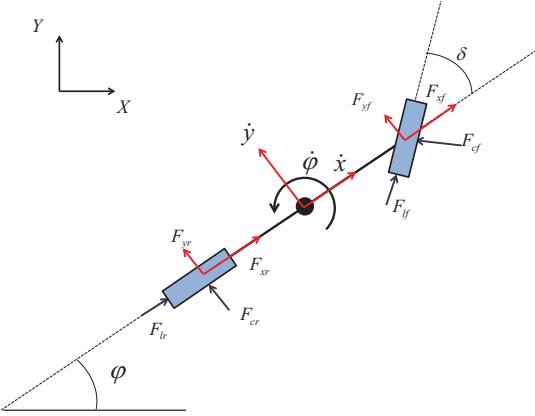


Figure 2.5: Modelling notation depicting the forces in the vehicle body-fixed frame (F_{x*} and F_{y*}), the forces in the tire-fixed frame (F_{l*} and F_{c*}), and the rotational and translational velocities.

With Simplification 1, the dynamics in Equation (2.12) can be written as follows:

$$m\ddot{x} = my\dot{\psi} + 2F_{xf} + 2F_{xr}, \quad (2.13a)$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr}, \quad (2.13b)$$

$$I\ddot{\psi} = 2aF_{yf} - 2bF_{yr}, \quad (2.13c)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi, \quad (2.13d)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi, \quad (2.13e)$$

where m and I denote the vehicle mass and yaw inertia, respectively. \dot{x} and \dot{y} denote the vehicle longitudinal and lateral velocities, respectively, and $\dot{\psi}$ is the turning rate around a vertical axis at the vehicle's centre of gravity. s is the vehicle longitudinal position along the desired path. F_{yf} and F_{yr} are front and rear tire forces acting along the vehicle lateral axis, F_{xf} and F_{xr} forces acting along the vehicle longitudinal axis.

The longitudinal and lateral tire force components F_{x*} and F_{y*} in the vehicle body frame are modeled as follows:

$$F_{x*} = F_{l*} \cos(\delta_*) - F_{c*} \sin(\delta_*), \quad (2.14a)$$

$$F_{y*} = F_{l*} \sin(\delta_*) + F_{c*} \cos(\delta_*), \quad * \in \{f, r\}, \quad (2.14b)$$

where $*$ denotes either f or r for front and rear tire, δ_* is the steering angle at wheel. We introduce the following assumption on the steering angles.

Assumption 2 Only the steering angle at the front wheels can be controlled. i.e., $\delta_f = \delta$ and $\delta_r = 0$.

The longitudinal and lateral tire forces $F_{l\star}$ and $F_{c\star}$ are given by Pacejka's model [51]. They are nonlinear functions of the tire slip angles α_\star , slip ratios σ_\star , normal forces $F_{z\star}$ and friction coefficient between the tire and road μ_\star :

$$F_{l\star} = f_l(\alpha_\star, \sigma_\star, F_{z\star}, \mu_\star), \quad (2.15a)$$

$$F_{c\star} = f_c(\alpha_\star, \sigma_\star, F_{z\star}, \mu_\star). \quad (2.15b)$$

The slip angles are defined as follows:

$$\alpha_\star = \arctan \frac{v_{c\star}}{v_{l\star}}, \quad (2.16)$$

where the tire lateral and longitudinal velocity components $v_{c\star}$ and $v_{l\star}$ are computed from,

$$v_{c\star} = v_{y\star} \cos \delta_\star - v_{x\star} \sin \delta_\star, \quad (2.17a)$$

$$v_{l\star} = v_{y\star} \sin \delta_\star + v_{x\star} \cos \delta_\star, \quad (2.17b)$$

where the tire y and x velocity components $v_{y\star}$ and $v_{x\star}$ can be computed from,

$$v_{y_f} = \dot{y} + a\dot{\psi}, \quad v_{x_f} = \dot{x}, \quad (2.18a)$$

$$v_{y_r} = \dot{y} - b\dot{\psi}, \quad v_{x_r} = \dot{x}. \quad (2.18b)$$

The slip ratios s_\star are approximated as follows:

$$s_\star = \begin{cases} \frac{r\omega_\star}{v_{l\star}} - 1 & \text{if } v_{l\star} > r\omega_\star, v \neq 0 \text{ for braking,} \\ 1 - \frac{v_{l\star}}{r\omega_\star} & \text{if } v_{l\star} < r\omega_\star, \omega \neq 0 \text{ for driving.} \end{cases} \quad (2.19)$$

The friction coefficient μ is assumed to be a known constant and is the same at all four wheels. We use the static weight distribution to estimate the normal force on the wheel, $F_{z\star}$ in Equation (2.15). They are approximated as,

$$F_{z_{f\star}} = \frac{bm}{2(a+b)}, \quad F_{z_{r\star}} = \frac{am}{2(a+b)}, \quad (2.20)$$

where g is the gravitational acceleration.

Remark 3 *The tire forces $F_{x\star}$ and $F_{y\star}$ ($F_{l\star}$ and $F_{c\star}$) are the forces generated by the contact of a single wheel with the ground. Thus the normal forces $F_{z\star}$ used in (2.15) are those acting on one single tire instead of the lumped forces on the front and rear wheel axles.*

2.4 Tire Models

With the exception of aerodynamics force and gravity, tire ground reaction forces are the only external force acting on the vehicle. The tire forces have highly nonlinear behavior when slip ratio or slip angle is large. Thus it is of extreme importance to have a realistic nonlinear tire force model for the vehicle dynamics when operating the vehicle in the tire nonlinear region. In such situations, large slip ratio and slip angle can happen simultaneously and the longitudinal and lateral dynamics of the vehicle is highly coupled and nonlinear due to the nature of the tire forces. Similar situation can occur even with small inputs when the surface friction coefficient μ is small.

When the slip ratio and slip angle are both small, both longitudinal and lateral tire forces show linear behavior and are less coupled. This situation holds true when the vehicle operates with moderate inputs on high μ surfaces. In such situations, linearized tire models might serve well in control design, with proper constraints on the slip ratio and slip angle.

In this section, we will present three tire models that will be used in this thesis. From a complex nonlinear semi-empirical model capturing the nonlinear and coupling behavior of the tire forces, to a simplest linear model with no coupling between longitudinal and lateral tire forces.

Pacejka tire model

This subsection gives more details on the Pacejka tire model [51] used in Section 2.2 and 2.3. It is a complex nonlinear semi-empirical model being able to describe the nonlinear and coupled behavior of tire forces under wide operation range. Pacejka model describes the tire forces as functions of the tire normal force, slip ratio, slip angle and surface friction coefficient.

It uses a function of following form to fit the experiment data:

$$Y(X) = D \sin(C \arctan(B\Theta(X))) + S_v, \quad (2.21)$$

where Y is either the longitudinal or lateral tire force. X is the slip ratio when Y is the longitudinal force and the slip angle when Y is the lateral force. B , C , D and S_v are the parameters fit from the experimental data. $\Theta(X)$ is defined as follows:

$$\Theta(X) = (1 - E)(X + S_h) + (E/B) \arctan(B(X + S_h)), \quad (2.22)$$

where E and S_h are parameters fit from the experimental data.

Equation (2.21) and (2.22) only describe the tire forces at pure cornering ($s = 0$) or pure braking/driving ($\alpha = 0$). They are then modified to include the combined slip conditions as presented in [51]. We will skip that part in this thesis.

We show some typical plots of tire forces from Pacejka tire model in Figure 2.7(a) to 2.8(b).

In particular, Figure 2.6 shows lateral and longitudinal tire forces at different values of friction coefficient. 2.6(a) shows the lateral tire forces as functions of slip angle (α) in pure

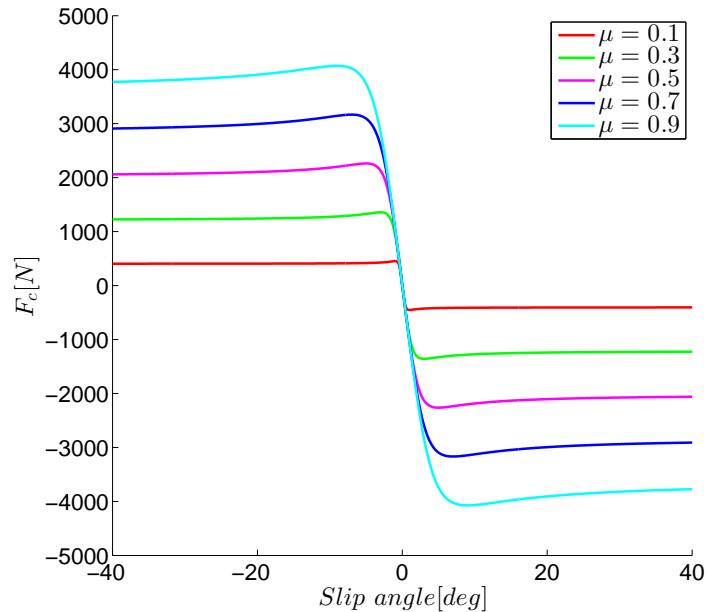
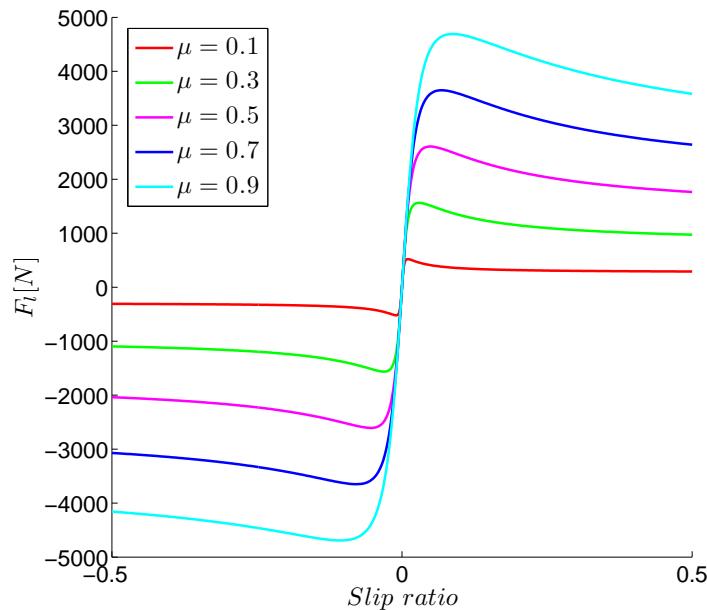
(a) Lateral tire forces in pure cornering ($s = 0$).(b) Longitudinal tire forces in pure braking/driving ($\alpha = 0$).

Figure 2.6: Lateral and longitudinal tire forces at different values of friction coefficients.

cornering, i.e. $s = 0$. 2.6(b) shows the longitudinal tire forces as functions of slip ratio (s) in pure braking/driving, i.e. $\alpha = 0$. We observe that the lateral and longitudinal tire forces are linear functions of the slip angle and slip ratio, respectively, at small values of slip angle and slip ratio, as we mentioned previously in this section. As the slip angle and slip ratio increases, the tire forces become saturated at some point and the magnitude start to decrease when the slip angle and slip ratio keep increasing. We refer the region before saturation as the tire linear region.

Note that besides the saturation value of tire forces, the friction coefficient also affects the width of the tire linear region. For instance, when the friction coefficient decreases from 0.9 (asphalt) to 0.1 (ice surface), the saturation slip angle for the lateral tire force decreases from 8° to 0.9° .

Figure 2.7 shows the longitudinal and lateral tire forces in combined cornering and braking/driving, with a friction coefficient of $\mu = 0.9$. In particular, Figure 2.7(a) shows the lateral forces as functions of slip angle at different values of slip ratio, while Figure 2.7(b) shows the longitudinal forces as functions of slip ratio at different values of slip angle.

From Figure 2.7(a), we observe that when slip ratio increases, the maximum magnitude of lateral force decreases. We also observe decreased slop in the linear region with increased slip ratio. The similar can be observed in the longitudinal forces as shown in Figure 2.7(b), i.e., the bigger the slip angle is, the smaller the maximum magnitude of longitudinal force as well as the slop of the curve in the linear region are.

Figure 2.8 shows the lateral and longitudinal forces in 3-D plots, as functions of both slip angle and slip ratio.

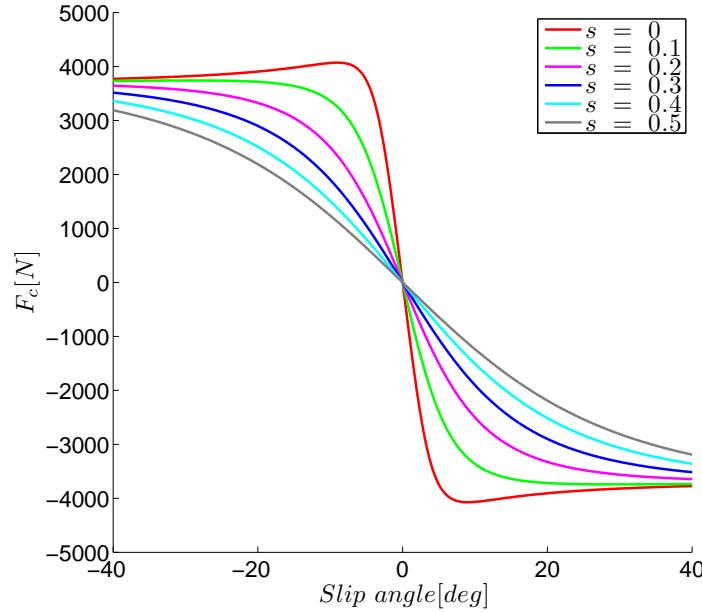
A modified Fiala tire model for combined cornering and braking/driving

This subsection presents a simpler version of nonlinear tire model, a modified fiala model [28], which can be used for modeling the tire lateral forces in combined cornering and braking/driving condition.

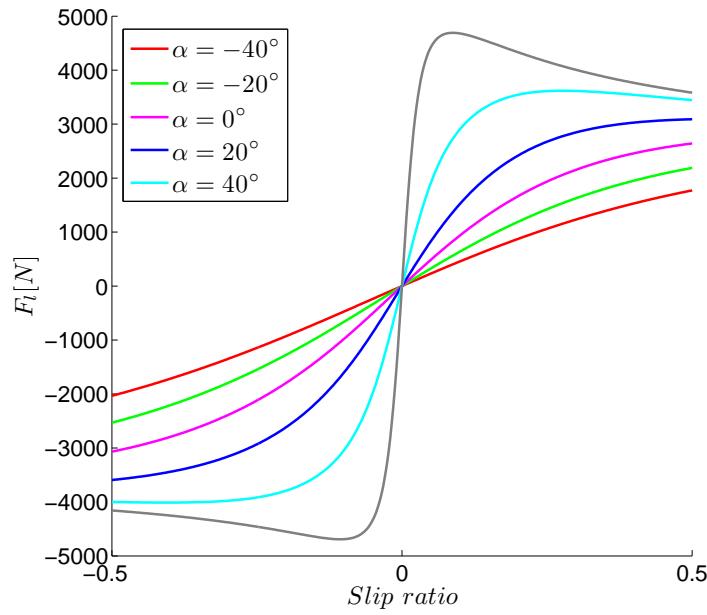
This model uses α_{sl} to denote the saturation slip angle for the lateral force and assumes the lateral force keeps the same beyond the saturation point. The lateral force is modeled as:

$$f_c = \begin{cases} -C_\alpha \tan(\alpha) + \frac{C_\alpha^2}{3\eta\mu F_z} |\tan(\alpha)| \tan(\alpha) - \frac{C_\alpha^3}{27\eta^2\mu^2 F_z^2} \tan^2(\alpha), & \text{if } |\alpha| < \alpha_{sl} \\ -\eta\mu F_z sgn(\alpha), & \text{if } |\alpha| \geq \alpha_{sl} \end{cases} \quad (2.23)$$

where α denotes the slip angle, μ denotes the friction coefficient, and F_z denotes the tire normal force. C_α is a constant presenting the slop of the curve at zero slip angle. $\eta = \sqrt{\mu^2 F_z^2 - f_l^2}/(\mu F_z)$, where f_l is the longitudinal force and is treated as one of the input to the model. The saturation slip angle is defined as follows:

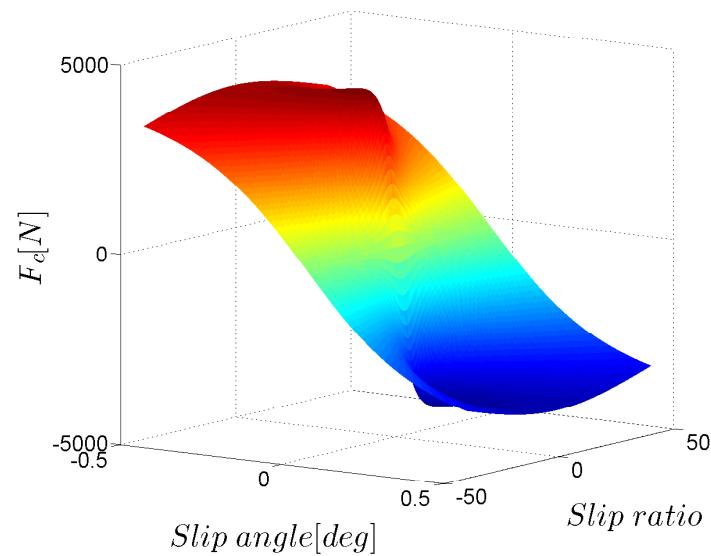


(a) Lateral tire forces under different values of slip ratio.

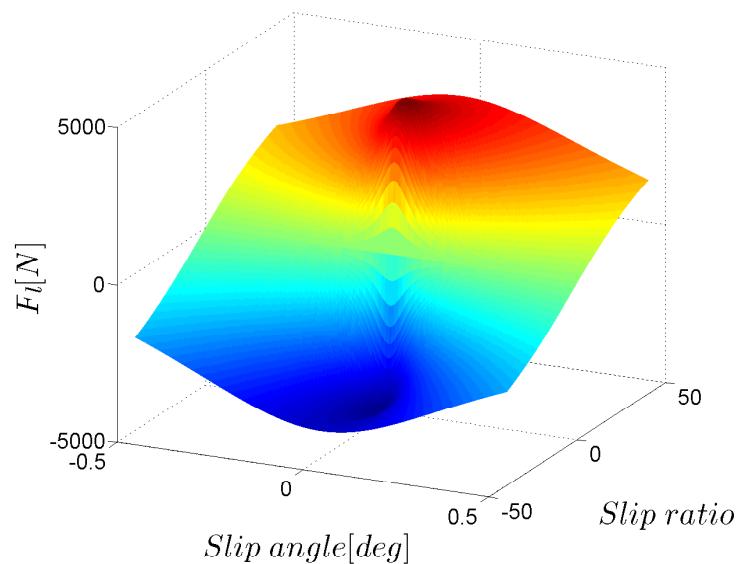


(b) Longitudinal tire forces under different values of slip angles.

Figure 2.7: Lateral and longitudinal tire forces in combined cornering and braking/driving, with $\mu = 0.9$.



(a) Lateral tire forces as a function of slip ratio and slip angle.



(b) Longitudinal tire forces as a function of slip ratio and slip angle.

Figure 2.8: Lateral and longitudinal tire forces as functions of slip ratio and slip angle in combined cornering and braking/driving, with $\mu = 0.9$.

$$\alpha_{sl} = \tan^{-1}\left(\frac{3\eta\mu F_z}{C_\alpha}\right). \quad (2.24)$$

We use $\beta = f_l/F_z$ to denote the level of braking/driving, with $\beta = 1$ denoting the maximum driving force and $\beta = -1$ denoting the maximum braking. Figure 2.9 shows the lateral forces from this model at different levels of braking/driving. Similar to the Pacejka model, the lateral forces show linear behavior at small values of slip angle and the maximum lateral force decreases as the braking/driving level increases. The Fiala model is less accurate than the Pacejka model, nevertheless it captures one of the most important feature of the lateral tire dynamics, the saturation of lateral force.

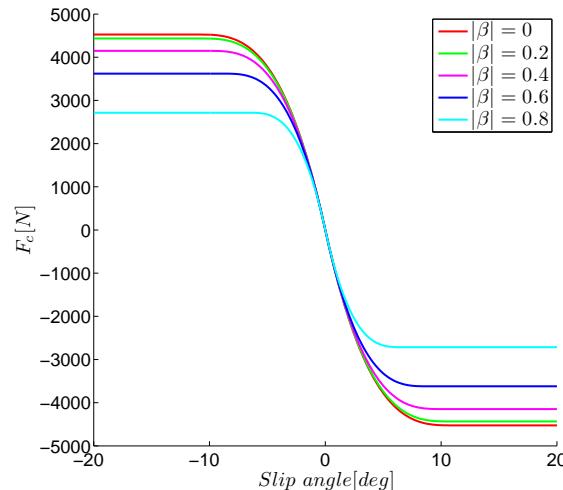


Figure 2.9: Lateral tire forces from the modified Fiala tire model at different levels of braking/driving.

Linear tire model

This subsection presents a linear tire model, which can be used for modeling lateral tire force inside the linear region. This model is only valid when both slip angle and slip ratio are restricted to have small values.

With small α and s assumption, the lateral tire force is modeled as:

$$F_c = C_\alpha(\mu, F_z)\alpha, \quad (2.25)$$

where C_α is called the tire's cornering stiffness coefficient and is a function of the friction coefficient μ and normal force F_z .

Figure 2.10 compares the lateral tire forces computed from the linear tire model and the Pacejka model. The linear tire model is the simplest tire model we can possibly get and should be implemented only with small slip angles, i.e. inside the tire linear region.

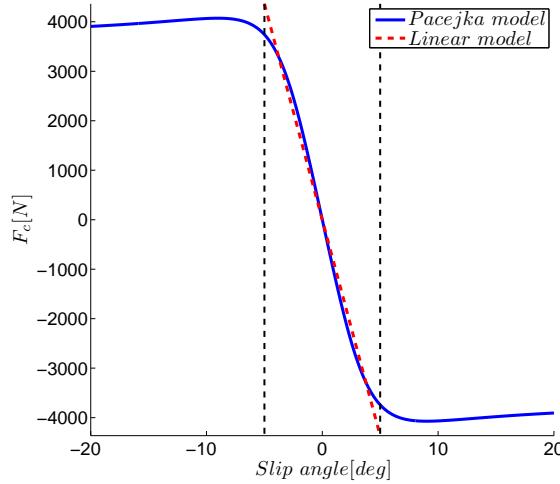


Figure 2.10: Linearized lateral tire forces in small slip angle region compared to Pacejka model, with $s = 0$ and $\mu = 0.9$.

2.5 Linear Bicycle Model

In this section we present a linear bicycle model derived from the model (2.13). The model is based on a linear tire model, small angle assumption and constant velocity assumption. This model is useful for some cases of lateral dynamics control during which the changes of the vehicle's longitudinal velocity is small and not critical. We first make the following assumption:

Assumption 3 *The vehicle travels at a constant longitudinal velocity. i.e. $\ddot{x} = 0$.*

The linear tire model presented in the previous section is used to model the lateral tire force:

$$F_{c\star} = C_\star(\mu, F_{z\star}) \cdot \alpha_\star, \quad (2.26)$$

where C_\star is the tire's cornering stiffness coefficient and is a function of the friction coefficient μ and normal force $F_{z\star}$. We assume μ is a known constant and use (2.20) to estimate the normal force. The constant C_\star then can be estimated according to the Pacejka tire model with the given μ and normal force.

To derive a linear model of the y component of tire force F_y , consider the Equations (2.14b) (2.16) and (2.18). With small angle assumption ($\delta_\star < 10^\circ$), we approximate $\sin \delta_\star$ and $\cos \delta_\star$ using their first order Taylor expansion:

$$\sin \delta_\star \simeq \delta_\star, \cos \delta_\star \simeq 1. \quad (2.27)$$

Combining Equations (2.16) (2.17) and (2.27), we can express the slip angle α_\star as:

$$\alpha_* \simeq \frac{v_{y_*} - v_{x_*}\delta_*}{v_{y_*}\delta_* + v_{x_*}}, \quad (2.28)$$

Assuming $v_y \ll v_x$ (i.e. small side slip angle), Equation (2.28) can be approximated as:

$$\alpha_* \simeq \frac{v_{y_*}}{v_{x_*}} - \delta_*. \quad (2.29)$$

With the assumption $\delta_r = 0$ and equation (2.18), we have:

$$\alpha_f \simeq \frac{v_y + a\dot{\psi}}{v_x} - \delta_f, \quad (2.30a)$$

$$\alpha_r \simeq \frac{v_y - b\dot{\psi}}{v_x}. \quad (2.30b)$$

Combining Equations (2.26) and (2.30), the lateral tire forces are computed as:

$$F_{c_f} \simeq C_f \cdot \left(\frac{v_y + a\dot{\psi}}{v_x} - \delta_f \right), \quad (2.31a)$$

$$F_{c_r} \simeq C_r \cdot \left(\frac{v_y - b\dot{\psi}}{v_x} \right). \quad (2.31b)$$

With the linear tire model (2.26) and small angle assumption, we approximate Equation (2.14b) as:

$$F_{y_*} = F_l_* \delta_* + F_{c_*} \simeq F_{c_*}. \quad (2.32a)$$

Substitute (2.32) into (2.13) and neglect longitudinal dynamics \ddot{x} , we have the following linear dynamic equations:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2[C_f(\frac{v_y + a\dot{\psi}}{v_x} - \delta_f) + C_r(\frac{v_y - b\dot{\psi}}{v_x})], \quad (2.33a)$$

$$I\ddot{\psi} = 2aC_f(\frac{v_y + a\dot{\psi}}{v_x} - \delta_f) - 2bC_r(\frac{v_y - b\dot{\psi}}{v_x}), \quad (2.33b)$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \quad (2.33c)$$

$$\dot{s} = \dot{x}, \quad (2.33d)$$

$$\dot{e}_y = \dot{x}(e_\psi) + \dot{y}. \quad (2.33e)$$

Note that we used the reference framework two as in (2.3) and assumed \dot{x} to be a constant. This model is compactly written as:

$$\dot{\xi}(t) = f^{lin}(\xi(t), u(t)) \quad (2.34)$$

where the state and input vectors are $\xi = [\dot{y}, \dot{\psi}, e_\psi, s, e_y]'$ and $u = [\delta_f]$, respectively.

2.6 Point Mass Model

This section presents the most simplified model for the vehicle dynamics, the point mass model. It is based on the following simplification:

Simplification 2 *The vehicle is treated as a point with a given mass. The yaw dynamics are neglected since there is no vehicle orientation can be defined. x and y direction are set to be the same as X and Y direction. The front and rear tire forces are treated together and presented by forces in x and y directions, F_x and F_y .*

With Simplification 2, the point mass model dynamic equations can be given as follows:

$$m\ddot{x} = F_x, \quad (2.35a)$$

$$m\ddot{y} = F_y, \quad (2.35b)$$

$$\dot{X} = \dot{x}, \quad (2.35c)$$

$$\dot{Y} = \dot{y}. \quad (2.35d)$$

The maximum tire forces can be constrained by the friction circle:

$$F_x^2 + F_y^2 \leq (\mu mg)^2, \quad (2.36a)$$

where μ is assumed a known constant, m is the vehicle's mass and g is the gravitational acceleration.

The dynamics in (2.35) is compactly written as:

$$\dot{\xi}(t) = f^{PM}(\xi(t), u(t)) \quad (2.37)$$

where the state and input vectors are $\xi = [\dot{y}, \dot{x}, Y, X]'$ and $u = [F_x, F_y]',$ respectively.

The point mass model is oversimplified. Besides its incapability of capturing the yaw dynamics and vehicle orientation, it also fails to capture the state dependant nature of the tire forces. For instance, when starting a turn from zero yaw rate and side slip, the rear tire lateral forces need time to build up with the increase of the yaw rate and side slip. This coupling of tire forces and vehicle states is not modeled in the point mass model.

Remark 4 *The point mass model given by (2.35) and (2.36) is an oversimplification of the models presented previously in this Chapter. Nevertheless, the constraint (2.36) captures one of the most important features of the tire forces that can be generated with a given surface and normal force.*

2.7 Concluding Remarks

Different vehicle models and tire models are presented in this Chapter, from a relatively complex four wheel model with Pacejka tire model to the simplest point mass model with

no tire model but only a friction circle constraint on tire forces. These models vary in their accuracy and complexity and poses one important trade off in the model based vehicle controller design as will be shown later in this thesis. Higher fidelity models usually limit the prediction horizon and sampling time of the MPC controller which limits the controller performance. Low fidelity models, on the other hand, allow large prediction horizon with small sampling time but can exhibit poor controller performance due to their large model mismatch.

Chapter 3

Hierarchical Model Predictive Control

3.1 Introduction

Parallel advances in theory and computing systems have enlarged the range of applications where real-time MPC can be applied [7, 8, 31, 22, 64, 13]. Yet, for a wide class of fast applications, the computational burden of Nonlinear MPC (NMPC) is still a barrier. In [9] an NMPC has been implemented on a passenger vehicle for path following via an Active Front Steering (AFS) system at 20 Hz, by using the state of the art optimization solvers and rapid prototyping systems. It is shown that the real-time execution is limited to low vehicle speeds on icy roads, because of its computational complexity. In order to decrease the computational complexity, in [15, 20] a Linear Time Varying MPC approach is presented to tackle the same problem. Experimental results [20, 21, 14, 16] demonstrated the capability of the controller to stabilize the vehicle and follow a reference trajectory at higher speeds.

Following the previous works, in this chapter, we will first compare two approaches to the autonomous obstacle avoidance problem. In the first approach, we use the formulation in [19] and modify the cost function by adding a distance-based cost term which increases as the vehicle gets closer to the obstacle. In the second approach, the controller design is decomposed into two levels. The high-level uses a simplified point-mass vehicle model and replans the path based on information about the current position of the obstacle. This is done in a receding horizon fashion with an NMPC controller. The trajectory is fed to the low-level controller which is formulated as the one in [19]. We compare performance and computation time of both approaches by using simulation and experimental results.

After concluding that the hierarchical approach is more promising and suitable for real-time implementation, we then propose a hierarchical approach for semi-autonomous obstacle avoidance. The high level path planner utilizes a special kind of curve, the clothoid, to plan the smoothest path avoiding the obstacle. A few parameters is used to determine the planned path, which is obstacle free and feasible for the vehicle to track. That results in low computational complexity for the high level path planner. The parameters also gives a measure of the aggressiveness of the avoiding maneuver, providing a suitable indication of

whether an intervention from the controller is needed to keep the vehicle safe.

This chapter is structured as follows. First, we briefly summarize the dynamics models used in the controller, then outline the control architecture of the two approaches. We then discuss about the MPC formulation of the two approaches and explain how we formulate the cost term associated with obstacle avoidance. Simulation and experimental results are then presented, followed by discussion on the comparisons between the two approaches. At the last part, the clothoid based semi-autonomous hierarchical controller is proposed, including the introduction of clothoid, the controller formulation and the simulation and experimental results.

3.2 Vehicle Model Summary

In this chapter, two vehicle models from Chapter 2 will be used: the **Four Wheel Model** and the **Point-Mass Model**. The ten state nonlinear four wheel model (2.4) which includes the longitudinal and lateral vehicle dynamics as well as wheel rotating dynamics is used for simulating the vehicle dynamics in close loop simulations. In controller design, a six state four wheel vehicle model is used instead, where the wheel dynamics are neglected.

Four wheel model

We use the six state four wheel vehicle model:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}} \quad (3.1a)$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}} \quad (3.1b)$$

$$\begin{aligned} I\ddot{\psi} = & a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + \\ & + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}) \end{aligned} \quad (3.1c)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi \quad (3.1d)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi \quad (3.1e)$$

Details of the model can be found in Chapter 2. Note that the wheel dynamics are neglected as discussed in Remark 1. The model is compactly written as:

$$\dot{\xi}(t) = f^{4ws}(\xi(t), u(t)) \quad (3.2)$$

where the superscript $4ws$ stands for “four wheel simplified”. The state and input vectors are $\xi = [\dot{y}, \dot{x}, \dot{\psi}, \psi, Y, X]$ and $u = [\delta_f, T_{f,l}, T_{f,r}, T_{r,l}, T_{r,r}]$.

Point-mass model

Consider the point mass model (2.37) and assume that the longitudinal velocity of the vehicle to be constant. We define the point mass model as follows:

$$\ddot{y} = a_y \quad (3.3a)$$

$$\ddot{x} = 0 \quad (3.3b)$$

$$\dot{\psi} = \frac{a_y}{\dot{x}} \quad (3.3c)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi \quad (3.3d)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi \quad (3.3e)$$

$$|a_y| < \mu g \quad (3.3f)$$

where ψ is defined as the direction of vehicle speed and the maximum lateral acceleration a_y is bounded by μg . The dynamics of the point-mass model are compactly written as,

$$\dot{\xi}(t) = f^{pm}(\xi(t), u(t)) \quad (3.4a)$$

$$|u(t)| < \mu g \quad (3.4b)$$

where $\xi = [\dot{y}, \dot{x}, \psi, Y, X]'$ and $u = a_y$.

3.3 MPC Control Architectures for An Obstacle Avoiding Path Replanner on Slippery Surface

Control goal

In the next few sections we explore two different control architectures for an obstacle avoiding path replan controller. The controller follows a predefined double lane change maneuver and at the same time avoids a pop up obstacle on its way by replanning the path around the obstacle. Figure 3.3 shows the reference trajectory for Y and ψ . The goal is to replan and execute the obstacle avoiding maneuver at high speed on low friction snow tracks.

The Y and ψ references in Figure 3.3 is generated by the following equations as functions of X :

$$Y_{ref}(X) = \frac{d_{y_1}}{2}(1 + \tanh(z_1)) - \frac{d_{y_2}}{2}(1 + \tanh(z_2)), \quad (3.5a)$$

$$\psi_{ref}(X) = \arctan(d_{y_1}\left(\frac{1}{\cosh(z_1)}\right)^2\left(\frac{1.2}{d_{x_1}}\right) - d_{y_2}\left(\frac{1}{\cosh(z_2)}\right)^2\left(\frac{1.2}{d_{x_2}}\right)), \quad (3.5b)$$

where $z_1 = \frac{2.4}{25}(X - 27.19) - 1.2$, $z_2 = \frac{2.4}{21.95}(X - 56.46) - 1.2$, $d_{x_1} = 25$, $d_{x_2} = 21.95$, $d_{y_1} = 4.05$, $d_{y_2} = 5.7$.

Control architectures

We compare two control architectures in this chapter. The first one is shown in Figure 3.2(a), and is referred to as **one-level MPC**. It uses the four wheel model presented in previous section. The desired trajectory $[\dot{x}_{ref}, \psi_{ref}(X), \dot{\psi}_{ref}(X), Y_{ref}(X)]$ along with the obstacle position are fed to the control algorithm. The predictive control scheme computes the

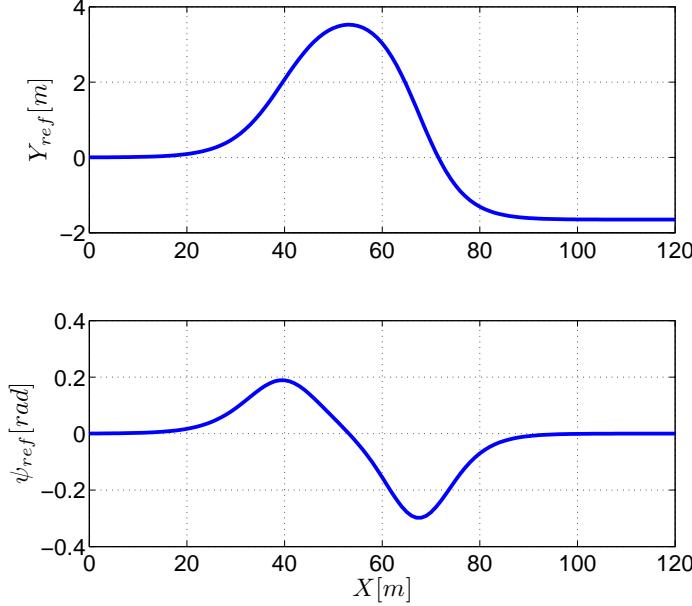
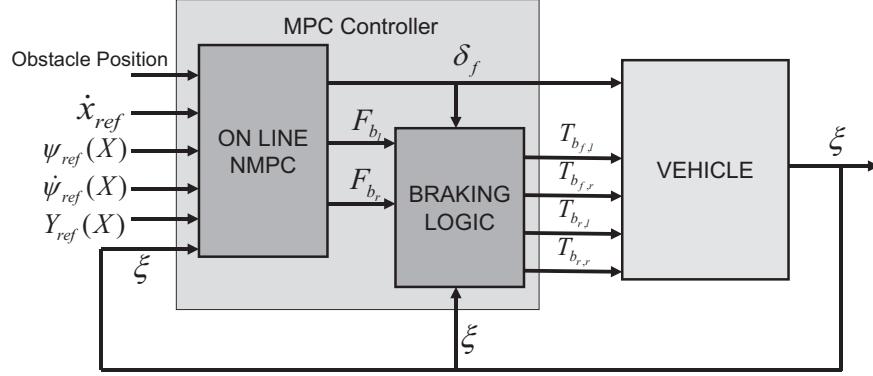


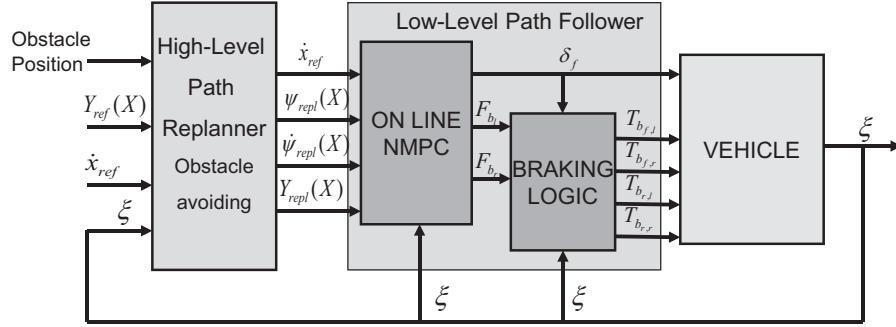
Figure 3.1: Reference state trajectory of the double lane change path to be followed.

optimal inputs to safely avoid the obstacle while trying to track the desired trajectory. Note that the forces T_{b_l} and T_{b_r} computed by the MPC are distributed to the braking torques, $T_{b_{*,\bullet}}$, on the four wheels by using a braking logic. Details about the braking logic can be found in [19].

The second approach is shown in Figure 3.2(b), and is referred to as **two-level MPC**. It is based on a hierarchical decomposition to the control problem. The desired trajectory $[Y_{ref}(X), \dot{x}_{ref}]$ and obstacle position are fed to the high-level path replanner. The path replanner replans a path which avoids the obstacle and tracks the reference, using the point-mass model presented in Section **Point-Mass Model**. The replanned path $[\dot{x}_{ref}, \psi_{repl}(X), \dot{\psi}_{repl}(X), Y_{repl}(X)]$ is then passed to a low-level path follower. The path follower computes the optimal input to follow the new trajectory, using the four wheel vehicle model presented in section **Four Wheel Model**. The same braking logic as in the one-level MPC is used here. Both controllers are NMPCs and are described in detail in the next section.



(a) Architecture of one-level MPC. The four wheel vehicle model is used.



(b) Architecture of two-level MPC. The point-mass vehicle model is used for high-level path replanner. While the four wheel vehicle model is used for the low-level path follower.

Figure 3.2: Two different architectures of controller design.

3.4 MPC Controller Formulation

General NMPC formulation

To formulate the MPC controller as a finite dimensional optimization problem, we discretize the vehicle dynamics $\dot{\xi} = f^*(\xi, u)$ with a sampling time T_s :

$$\xi(k+1) = f_d^*(\xi(k), u(k)), \quad (3.6a)$$

$$u(k) = u(k-1) + \Delta u(k), \quad (3.6b)$$

where f^* can be any continuous vehicle dynamics. Note we will use the change rate of inputs Δu as optimization variable.

The MPC controller can be formulated as a general optimization problem:

$$\min_{\Delta U_t} J_N(\bar{\xi}_t, U_t, \Delta U_t) \quad (3.7a)$$

$$\text{subj. to } \xi_{k+1,t} = f(\xi_{k,t}, u_{k,t}) \quad k = t, \dots, t + H_p - 1, \quad (3.7b)$$

$$\Delta u_{k+1,t} = u_{k+1,t} - u_{k,t} \quad k = t, \dots, t + H_c - 1, \quad (3.7c)$$

$$\Delta u(k) = 0 \quad k = t + H_c + 1, \dots, t + H_p - 1, \quad (3.7d)$$

$$\xi_{k,t} \in \Xi \quad k = t + 1, \dots, t + H_p - 1, \quad (3.7e)$$

$$u_{k,t} \in \mathcal{U} \quad k = t, \dots, t + H_p - 1, \quad (3.7f)$$

$$\Delta u_{k,t} \in \Delta \mathcal{U} \quad k = t + 1, \dots, t + H_p - 1, \quad (3.7g)$$

$$\xi_{t,t} = \xi(t), \quad (3.7h)$$

where $\bar{\xi}_t = [\xi_{t,t}, \xi_{t+1,t}, \dots, \xi_{t+H_p-1,t}]$ is the sequence of states $\bar{\xi}_t \in \mathbb{R}^{nH_p}$ over the prediction horizon H_p predicted at time t , and updated according to the discretized dynamics of the vehicle model (3.7b), and $u_{k,t}$ and $\Delta u_{k,t} \in \mathbb{R}^{m_r}$ is the k^{th} vector of the input sequence $U_t \in \mathbb{R}^{m_r H_p}$ and $\Delta U_t \in \mathbb{R}^{m_r (H_p-1)}$ respectively,

$$U_t = [u'_{t,t}, u'_{t+1,t}, \dots, u'_{t+H_c-1,t}, u'_{t+H_c,t}, \dots, u'_{t+H_p-1,t}]' \quad (3.8a)$$

$$\Delta U_t = [\Delta u'_{t+1,t}, \dots, \Delta u'_{t+H_c-1,t}, \Delta u'_{t+H_c,t}, \dots, \Delta u'_{t+H_p-1,t}]' \quad (3.8b)$$

We reduce the computational complexity of the MPC problem by holding the last $H_p - H_c$ input vectors in U_t constant and equal to the vector $u_{t+H_c-1,t}$. With this assumption, only the first H_c input vectors constitutes the optimization variables. We refer to H_p as the prediction horizon and H_c as the control horizon.

At each time step t , the performance index $J_N(\bar{\xi}_t, U_t, \Delta U_t)$ is optimized under the constraints (3.7c)-(3.7g) starting from the state $\xi_{t,t} = \xi(t)$ to obtain an optimal control sequence, $U_t^* = [u_{t,t}^*, \dots, u_{t+H_p-1,t}^*]'$. The first of such optimal moves $u_{t,t}^*$ is the control action applied to the vehicle at time t . At time $t+1$, a new optimization is solved over a shifted prediction horizon starting from the new measured state $\xi_{t+1,t+1} = \xi(t+1)$. The time interval between time step $t+1$ and time step t is the sampling time T_s .

Obstacle avoidance cost

The cost for obstacle avoidance is based on the distance between the front of the vehicle and the obstacle [62]. Assuming information on the position of the obstacle is given at time t as a collection of N discretized points, the position of the j^{th} point is denoted as $P_{t,j} = (p_{X_{t,j}}, p_{Y_{t,j}})$, $j = 1, 2, \dots, N$. If these points are given in the inertial frame, they are transformed to the vehicle body frame by,

$$p_{x_{k,t,j}} = (p_{Y_{t,j}} - Y_{k,t}) \sin \psi_{k,t} + (p_{X_{t,j}} - X_{k,t}) \cos \psi_{k,t} \quad (3.9a)$$

$$p_{y_{k,t,j}} = (p_{Y_{t,j}} - Y_{k,t}) \cos \psi_{k,t} - (p_{X_{t,j}} - X_{k,t}) \sin \psi_{k,t} \quad (3.9b)$$

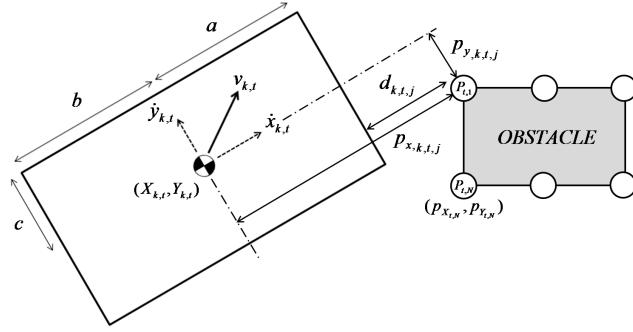


Figure 3.3: Graphical representation of $d_{k,t,j}$, $p_{x,t,j}$ and $p_{y,t,j}$.

The cost at time k associated to the distance predicted at time t between the vehicle and an obstacle is denoted as $J_{obs_{k,t}}$:

$$J_{obs_{k,t}} = \frac{K_{obs} v_{k,t}}{d_{min_{k,t}} + \epsilon} \quad (3.10)$$

where $v_{k,t}^2 = \dot{x}_{k,t}^2 + \dot{y}_{k,t}^2$ is the speed of the vehicle at time k predicted at time t , ϵ is a small number, K_{obs} is the collision weight and $d_{min_{k,t}} = \min_j d_{k,t,j}$ is the minimum distance to all N obstacle points, with $d_{k,t,j}$ defined as,

$$d_{k,t,j} = \begin{cases} p_{x_{k,t,j}} - a & \text{if } p_{y_{k,t,j}} \in [-c, c] \text{ and } p_{x_{k,t,j}} > a \\ 0 & \text{if } p_{y_{k,t,j}} \in [-c, c] \text{ and } p_{x_{k,t,j}} \in [-b, a] \\ M & \text{otherwise} \end{cases} \quad (3.11)$$

$$\forall j = 1, 2, \dots, N$$

where a, b and c are the vehicle dimensions, $d_{k,t,j} = 0$ indicates a collision occurrence, and M is a constant big enough to disregard obstacles that do not lie within the vehicle's line of sight.

Note, the obstacle is discretized at a resolution which ensures that the vehicle cannot drive right through the obstacle without containing at least one point within its frame. The obstacle bounds are also enlarged to conservatively consider the width and length of the vehicle since the controller only controls the vehicle's center of gravity. More sophisticated constraints can be introduced to consider the vehicle size if the constraints are tight.

One-Level MPC

The one-level MPC uses the four wheel vehicle mode, i.e., equation (3.7b) is a discrete time version of the model 3.2. It is formulated as follows:

$$\min_{\Delta U_t} J_N(\bar{\xi}_t, U_t, \Delta U_t) = \sum_{k=t}^{t+H_{tr}-1} (\|\eta_{k,t} - \eta_{ref_{k,t}}\|_Q^2 + \|u_{k,t}\|_R^2 + \|\Delta u_{k,t}\|_S^2) + \sum_{k=t}^{t+H_p-1} J_{obs_{k,t}} \quad (3.12a)$$

$$\text{subj. to } \xi_{k+1,t} = f_d^{4ws}(\xi_{k,t}, u_{k,t}) \quad k = t, \dots, t + H_p - 1, \quad (3.12b)$$

$$\Delta u_{k+1,t} = u_{k+1,t} - u_{k,t} \quad k = t, \dots, t + H_c - 1, \quad (3.12c)$$

$$\Delta u(k) = 0 \quad k = t + H_c + 1, \dots, t + H_p - 1, \quad (3.12d)$$

$$u_{k,t} \in \mathcal{U} \quad k = t, \dots, t + H_p - 1, \quad (3.12e)$$

$$\Delta u_{k,t} \in \Delta \mathcal{U} \quad k = t + 1, \dots, t + H_p - 1, \quad (3.12f)$$

$$\xi_{t,t} = \xi(t), \quad (3.12g)$$

The cost function J_N considers both the reference tracking and obstacle avoidance, where the first term penalizes the deviations of the tracking state $\eta_{k,t} = [\dot{x}_{k,t}, \psi_{k,t}, \dot{\psi}_{k,t}, Y_{k,t}]'$ from the state reference, $\eta_{ref_{k,t}} = [\dot{x}_{ref_{k,t}}, \psi_{ref_{k,t}}, \dot{\psi}_{ref_{k,t}}, Y_{ref_{k,t}}]'$, with weighting Q ; the second term penalizes the control input with weighting R ; the third term penalizes the change in control input with weighting S ; the fourth term gives the obstacle avoidance cost.

Notice that predicted state trajectories are penalized over the interval $k \in [t, t + H_{tr} - 1]$, where H_{tr} is the tracking horizon, $H_{tr} < H_p$. The obstacle avoidance cost term is penalized over the entire prediction horizon. In addition we use two different sampling times: T_{s1} for $k \in [t, t + H_{tr} - 1]$ and T_{s2} for $k \in [t + H_{tr}, t + H_p - 1]$ with $T_{s1} < T_{s2}$. Both constructions are fundamental for obtaining a scheme that is real-time implementable on the vehicle hardware.

The inputs vector $u_{k,t} = [\delta_f, T_{b_l}, T_{b_r}]'$ consists of the steering angle δ_f , left braking force F_{r_l} and right braking force F_{b_r} . The braking logic in [19] is used to distribute the corresponding torques at the four wheels.

Two-Level MPC

Two MPC problems are formulated for the high-level path replanning and for the low-level path following. They can have different setup in sampling time, prediction and etc.

The low-level MPC is similar to the one-level MPC: the vehicle model, the inputs and their constraints are the same. The difference is that the cost function neglects the obstacle avoidance term and the tracking reference is the obstacle free replanned path from the high level, $\eta_{repl_{k,t}} = [\dot{x}_{repl_{k,t}}, \psi_{repl_{k,t}}, \dot{\psi}_{repl_{k,t}}, Y_{repl_{k,t}}]'$. The cost function of low level MPC is given as:

$$J_N^{LL}(\bar{\xi}_t, U_t, \Delta U_t) = \sum_{k=t}^{t+H_{p,ll}-1} \|\eta_{k,t} - \eta_{repl_{k,t}}\|_Q^2 + \|u_{k,t}\|_R^2 + \|\Delta u_{k,t}\|_S^2 \quad (3.13)$$

The sampling time and prediction horizon of the low level are denoted as $T_{s,ll}$ and $H_{p,ll}$ respectively.

The high-level MPC uses the point-mass model (3.4), i.e. equation (3.7b) is a discrete time version of the model (3.4): $\dot{\xi}(t) = f^{pm}(\xi(t), u(t))$. The optimization problem is formulated as follows:

$$\min_{\Delta U_t} J_N^{HL}(\bar{\xi}_t, U_t) = \sum_{k=t}^{t+H_{p,hl}-1} \|Y_{k,t} - Y_{ref_{k,t}}\|_Q^2 + \|a_{y_{k,t}}\|_R^2 + J_{obs_{k,t}} \quad (3.14a)$$

$$\text{subj. to } \xi_{k+1,t} = f_d^{pm}(\xi_{k,t}, u_{k,t}) \quad k = t, \dots, t + H_p - 1, \quad (3.14b)$$

$$u_{k,t} = u_{t+H_c,t} \quad k = t + H_c + 1, \dots, t + H_p - 1, \quad (3.14c)$$

$$|u_{k,t}| < \mu g \quad k = t, \dots, t + H_p - 1, \quad (3.14d)$$

$$\xi_{t,t} = \xi(t), \quad (3.14e)$$

The cost function J_N^{HL} weights the deviations from the lateral reference $Y_{ref_{k,t}}$, the lateral acceleration input $a_{y_{k,t}}$ and the obstacle avoidance term:

$$J_N^{LH}(\bar{\xi}_t, U_t) = \sum_{k=t}^{t+H_{p,hl}-1} \|Y_{k,t} - Y_{ref_{k,t}}\|_Q^2 + \|a_{y_{k,t}}\|_R^2 + J_{obs_{k,t}} \quad (3.15)$$

The sampling time and prediction horizon of the high level are denoted as $T_{s,hl}$ and $H_{p,hl}$ respectively.

3.5 Simulation and Experimental Results

To test the performance of the two approaches, a series of double lane change maneuvers at different entry speeds have been simulated and experimentally tested. The goal is to follow the double lane change as closely as possible while avoiding obstacles. The desired path is described in terms of lateral position Y_{ref} and yaw angle ψ_{ref} as a function of the longitudinal position X , as described in section 3.3.

Simulation setup description

We use Matlab® to simulate the close loop systems. The MPC optimization problem has been implemented as a C-coded s-Function. The commercial NPSOL software package [25] is used for solving the nonlinear programming problem (3.7). The first element of the optimized control sequence is passed to another external block which uses the ten state four wheel model (2.4) to simulate the dynamics of the vehicle, and feeds the current state of the vehicle back to the controller. We simulate results using this procedure to compare the computation loads for the two different approaches.

Experimental setup description

The two approaches presented before have been tested through experiments on slippery surfaces. The experiments have been performed at a test center equipped with icy and

snowy handling tracks. The MPC controllers have been tested on a passenger car, with a mass of 2050 Kg and an inertia of 3344 Kg/m^2 . The controllers were run in a dSPACE Autobox system, equipped with a DS1005 processor board and a DS2210 I/O board.

We used an Oxford Technical Solution (OTS) RT3002 sensing system to measure the position and the orientation of the vehicle in the inertial frame and the vehicle velocities in the vehicle body frame. The OTS RT3002, is housed in a small package that contains a Differential Global Positioning System (DGPS) receiver, Inertial Measurement Unit (IMU), and a Digital Signal Processor (DSP). It is equipped with a single antenna to receive GPS information. The IMU includes three accelerometers and three angular rate sensors. The DSP receives both the measurements from the IMU and the DGPS, utilizes a Kalman filter for sensor fusion, and calculate the position, orientation and other states of the vehicle such as longitudinal and lateral velocities. Compared to a dual antenna setup, a single antenna system has to learn the vehicle orientation and/or coordinate during vehicle motion. This might cause the presence of a small offset in the orientation measurement. In dual antenna systems the vehicle orientation can be easily set even if the vehicle is not moving.

The car was equipped with an Active Front Steering (AFS) and Differential Braking system which utilizes an electric drive motor to change the relation between the hand steering wheel and road wheel angles. This is done independently from the hand wheel position, thus the front road wheel angle is obtained by summing the driver hand wheel position and the actuator angular movement. Both the hand wheel position and the angular relation between hand and road wheels are measured.

This steering actuator can operate in two modes: with and without the intervention of the controller. In the first mode the steering commands of the controller and the driver overlap and the overall steering action is the results of the contributions from the driver and the controller. In the second mode there is no controller intervention and the steering angle is completely decided by the driver through the steering wheel. We used the second mode, and the driver action represents an input disturbance. The sensor, the dSPACE Autobox and the actuators communicate through a CAN bus.

The test is initiated by the driver with a button. When the button is pushed, the inertial frame in Figure 2.2(a) is initialized as follows: the origin is the current vehicle position, the axes X and Y are directed as the current longitudinal and lateral vehicle axes, respectively. Such inertial frame becomes also the desired path coordinate system. Once the initialization procedure is concluded, the vehicle executes the double lane change and obstacle avoidance maneuver.

Note that, noise may affect the yaw angle measurement due to the single antenna sensor setup. Compared to a dual antenna setup, a single antenna system has to learn the vehicle orientation and/or coordinate during vehicle motion. When the vehicle stands still the yaw angle is computed by integrating the yaw rate measurement from the IMU. This might cause the presence of a small offset in the orientation measurement, while traveling at low speed or being still.

A photo of the test vehicle is shown in Figure 3.4.



Figure 3.4: The test vehicle.

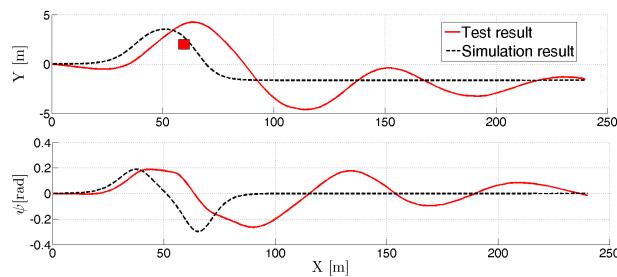


Figure 3.5: Experimental result: Test results of the one-level MPC at vehicle speed 40 Kph. The red box represents the obstacle.

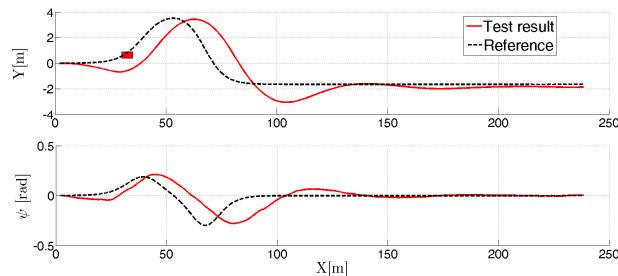


Figure 3.6: Experimental result: Test results of the two-level MPC at vehicle speed 45 Kph. The red box represents the obstacle.

Experimental results

In this section, we compare the two approaches presented before, the one-level and the two-level MPC controller. Shown in Figure 3.5, 3.6 and 3.7 are the experimental results obtained with the vehicle travelling on icy roads ($\mu \simeq 0.3$) at different entry speeds. The controllers parameters are reported next.

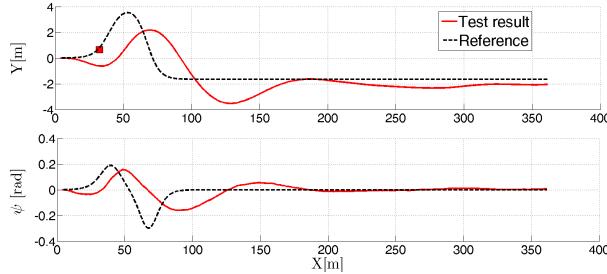


Figure 3.7: Experimental result: Test results of the two-level MPC at vehicle speed 55 Kph. The red box represents the obstacle.

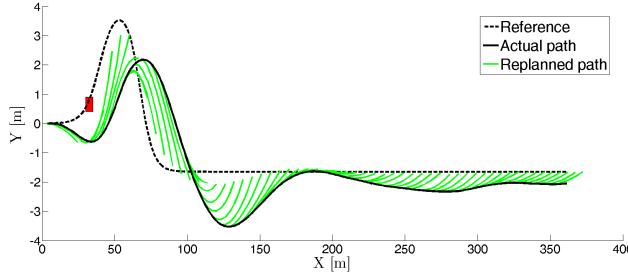


Figure 3.8: Experimental result: Replanned paths from the high-level path replanner in two-level MPC. Vehicle speed is 55 Kph.

Tuning 1A One-level MPC: Nonlinear MPC (3.12) with model (3.2) and cost (3.12a) and with the following parameters

- $T_{s1} = 0.1\text{s}$, $T_{s2} = 0.3\text{s}$, $H_p = 10$, $H_c = 2$, $H_{tr} = 5$
- $\delta_f \in [-10^\circ, 10^\circ]$, $\Delta\delta_f \in [-17^\circ, 17^\circ] \times T_{s1}$
- $F_{b_\bullet} \in [-1500, 0]$, $\Delta F_{b_\bullet} \in [-1000, 1000] \times T_{s1}$
- $Q = \text{diag}(10^{-2}, 1, 1, 30)$, $R = \text{diag}(10, 1, 1)$
 $S = \text{diag}(10^{-2}, 1, 1)$, $K_{obs} = 1$

Tuning 2HL Hierarchical MPC High-level: Nonlinear MPC (3.14) with model (3.4) and cost (3.14a) and with the following parameters.

- $T_{s,hl} = 0.1\text{s}$, $H_{p,hl} = 15$; $H_c = 2$
- $a_y \in [-\mu g, \mu g]$
- $Q = 10$, $R = 12$, $K_{obs} = 1$

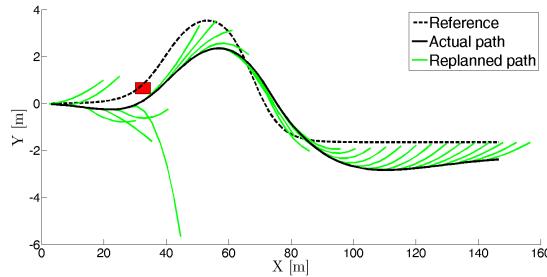


Figure 3.9: Experimental result: Replanned paths from high-level when the communication from high-level to low-level is cut off. Vehicle speed is 55 Kph.

Tuning 2LL Hierarchical MPC Low-level: Nonlinear MPC (3.7) with model (3.2) and cost (3.13) and with the following parameters

- $T_{s,ll} = 0.05\text{s}$, $H_{p,ll} = 15$, $H_c = 2$
- $\delta_f \in [-10^\circ, 10^\circ]$, $\Delta\delta_f \in [-17^\circ, 17^\circ] \times T_{s,ll}$
- $F_{b\bullet} \in [-1500, 0]$, $\Delta F_{b\bullet} \in [-1000, 1000] \times T_{s,ll}$
- $Q = \text{diag}(0, 10, 1, 30)$, $R = \text{diag}(1, 10, 10)$
 $S = \text{diag}(1, 4, 4)$

The road friction coefficient μ is set to be a constant 0.3 even if the road conditions were changing during each experiment depending on the weather conditions (ice, snow, fresh snow on top of packed snow). Also, during the experiments, the maximum iteration number in NPSOL has been limited in order to guarantee real-time computation.

For the one-level MPC, successful tests were obtained only at 40kph (Figure 3.5). The two-level MPC was able to run at higher speeds of 45 and 55kph (Figure 3.6 and 3.7). In Figure 3.5 to 3.7, the dotted black line represents the double-lane reference trajectory for Y_{ref} and ψ_{ref} and the solid red line represents the test results. There is also a red rectangular box drawn to illustrate the position of the obstacle.

In Figure 3.8 the replanned trajectories at the high-level are shown along with the actual vehicle trajectory. Substantial deviation of the actual trajectory from the replanned paths can be observed. This is due to the model mismatch of the point mass model used in the high level path replanner. The over simplification of the vehicle dynamics causes infeasible replanned paths for the real vehicle.

In order to better understand the role of the path replanner, in Figure 3.9 we present a test where the communication from the high-level to the low-level controller is cut off. Therefore the low-level controller tries to follow the desired path regardless of the obstacle position and the replanned trajectory. We observe that before passing the obstacle, the high-level replans paths that avoid the obstacle by turning left at beginning and then by

Table 3.1: Computation Time of One-level and Two-level MPC at Vehicle Speed of 40 Kph.

	One-lvl MPC Comp.time		Two-lvl MPC Comp.time	
	[ms]		[ms]	
	Tuning 1A	Tuning 1B	High-lvl	Low-lvl
Mean	10 ± 1	23 ± 3	0.5 ± 0.2	3.6 ± 1.4
Max	15	36	1.8	8.2

turning right when driver insists staying to the right. After passing the obstacle, the path replanner plans paths that converge to the reference.

Computation time

The average and maximum computation times for the one-level and the two-level MPC are summarized in Table 3.1. In order to test the one-level MPC for longer horizons (not implementable in real-time on the experimental platform), the results have been obtained in simulation with a 3.0 GHz Intel® Core(TM)2Duo desktop running Matlab 7.5, with the vehicle speed set to 40kph, and obstacle position as shown in Figure 3.7.

The computation load is measured in terms of Floating Point Operations (FLOP), N_{comp} , NPSOL takes at each iteration. Using the conversion, $T_{comp} = N_{comp}/(3 \times 10^6)$, the results summarized in Table 3.1 are the actual running times on the given processor.

For the two-level MPC, the computation time in Table 3.1 is obtained with **Tuning 2HL** and **Tuning 2LL**. For the one-level MPC, **Tuning 1A** and **Tuning 1B** is used, where Tuning 1B is similar to Tuning 1A with these parameters changed: $T_{s1} = 0.05s$, $T_{s2} = 0.1s$, $H_p = 23$, $H_c = 2$, $H_{tr} = 16$.

Notice that, with the one-level MPC set at Tuning 1B, the prediction horizon is now comparable to the two-level MPC. The tracking horizon is, $H_{p,ll}T_{s,ll} \simeq H_{tr}T_{s1} = 0.8s$ and the prediction horizon with $J_{obs_{k,t}}$ is, $H_{p,hl}T_{s,hl} = H_{tr}T_{s1} + (H_p - H_{tr})T_{s2} = 1.5s$.

With Tuning 1B, the one-level MPC is not able to run in real-time on the experimental platform. As shown in Table 3.1, the computation time of two-level MPC is 39% less than that of one-level MPC at (Tuning 1A). The one-level MPC suffers from the heavy computational burden and is only able to run in real time with compromised horizon and sampling time, which limits the performance (Figure 3.5). The two-level MPC, on the other hand, reduces the computation complexity by a hierarchical decomposition to the problem and is able to obtain good performance (Figure 3.6 and 3.7).

Stability and obstacle avoidance

Consider the one-level MPC. From Figure 3.10, we observe that once the vehicle deviates too far from the reference, the controller can no longer pull the system back to the reference

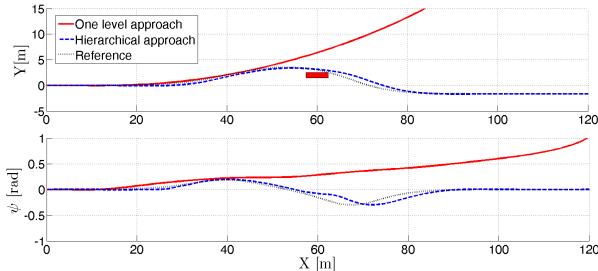


Figure 3.10: Simulation results: At speed of 70Kph, the one-level MPC becomes unstable even if given enough computation time. While the two-level MPC is able to stabilize the vehicle in the same situation.

trajectory and the system becomes uncontrollable. This means that the vehicle state is outside the region of attraction of the equilibrium trajectory associated to the desired reference. When the vehicle is travelling at high speeds, this region of attraction becomes smaller, and the reference trajectory becomes harder to follow. In addition, when obstacle avoidance is combined with trajectory following, the controller will force the vehicle to deviate from the reference in order to avoid the obstacle. This often causes the system to become unstable for the one-level controller running at high speeds. We remark that this type of behavior is induced by tire saturation and cannot be observed with a simple point-mass model.

However, for the two-level controller, this issue is significantly resolved, since the path replanner always replans a path starting from the current state of the vehicle, and therefore, ensures that the low-level reference is close to current state. This explains why the performance of the two-level approach is better than the one-level approach.

Figure 3.10 shows the comparison between simulations for the two approaches at 70kph. Both controllers tried to avoid the obstacle by steering to the right, forcing the vehicle away from the reference trajectory. The one-level MPC becomes unstable while the two-level MPC avoids the obstacle and is still able to return to the reference trajectory. We can improve the performance of the one-level MPC by extending the prediction horizon and control horizon, however, this is done at the expenses of computation time, and generates runtime error on the described experimental platform.

If the one-level MPC is properly designed (for example, by using an invariant set as a terminal constraint [46]), then persistent feasibility is guaranteed for all initial feasible states (note that the obstacle position becomes a state of the system). For the current two-level MPC design, even if the high-level path replanner computes a path that avoids the obstacle, collision may still occur due to errors in the path following at the low-level. By adding a safety distance to the obstacle (i.e. enlarge the obstacle) and a constraint to the maximum tracking error in the low-level controller, obstacle avoidance can be guaranteed (for all initial conditions for which the low-level controller is persistently feasible).

3.6 Concluding Remarks on Control Architecture Comparison

We have presented two MPC approaches for trajectory following for autonomous ground vehicles with obstacle avoidance. The one-level approach combines both requirements into one large finite horizon optimization problem, while the two-level approach breaks this problem into a high-level path replanner and a low-level path follower. The experiments show successful results from the two level approach with entering speeds up to 55kph on icy surfaces.

For the one-level approach, the computation time increases at higher speeds and therefore real-time implementation presents an issue. The high computational complexity also limits the prediction horizon and sampling time in the one-level approach, which limits the controller performance. The two-level approach shows promise in this aspect with its much lower computational burden. However, the point mass model used in the high level is over simplified. That results in infeasible replanned path for the real vehicle. As can be seen in Figure 3.8, the actual vehicle trajectory is not able to follow the replanned path closely.

To address the model over simplification problem in the two-level approach, we will propose an improved approach in the next chapter.

3.7 Clothoid Based Semi-Autonomous Hierarchical MPC for Obstacle Avoidance

In this section, we propose a hierarchical MPC with the same structure of the **two-level MPC** in section 3.3. The low level is the same NMPC as in the point mass two-level controller. The high level is a path planner using a special curve, clothoid, to plan obstacle free paths. Clothoids are parameterized curves suitable for constructing lane change trajectories. Later we will show its usefulness in obtaining a semi-autonomous vehicle with adjustable degrees of autonomy.

Parameterized clothoids

A clothoid is a curve with a curvature that changes linearly with the curve length. It is widely used in railway and highway engineering for curvature transition. Also, its use in road design and robot path planning has a long history. Relevant references include [57, 56, 37]. In a clothoid, at a given curve length s the curvature $K(s)$ is determined by the linear function $K(s) = 2a^2 \cdot s$, where a is the curvature change rate. The clothoid is expressed by the Fresnel integral [57] as:

$$X(s) = \frac{1}{a} \int_0^{aL} \cos(s^2) ds, \quad (3.16a)$$

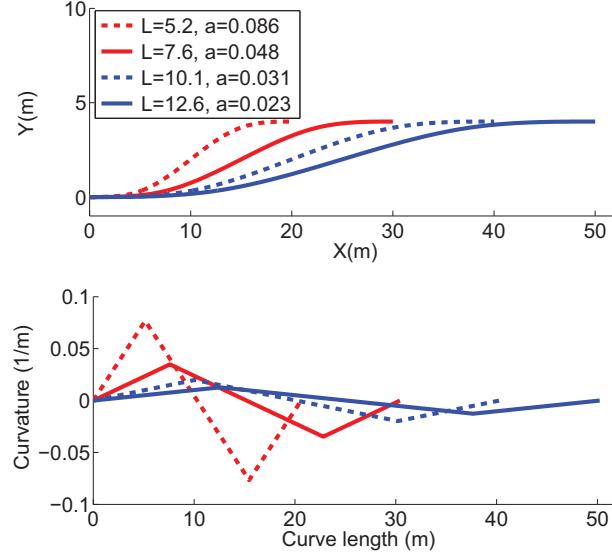


Figure 3.11: Lane change paths with different aggressiveness. Each is generated by connecting four pieces of clothoids. The upper figure shows the shapes of the paths in global frame. The lower figure shows the piecewise affine relation between curvature and curve length for each path.

$$Y(s) = \frac{1}{a} \int_0^{aL} \sin(s^2) ds, \quad (3.16b)$$

where X and Y are the global coordinates and L is the maximum curve length. In numerical computation schemes, the integrals in Equation (3.16) are approximated by their Taylor expansion:

$$\int_0^{aL} \cos(s^2) ds = aL - \frac{(aL)^5}{5 \times 2!} + \frac{(aL)^9}{9 \times 4!} - \frac{(aL)^{13}}{13 \times 6!} + \dots, \quad (3.17a)$$

$$\int_0^{aL} \sin(s^2) ds = \frac{(aL)^3}{3} - \frac{(aL)^7}{7 \times 3!} + \frac{(aL)^{11}}{11 \times 5!} - \frac{(aL)^{15}}{15 \times 7!} + \dots, \quad (3.17b)$$

By varying the parameters a and L , different clothoids are generated. A lane-change maneuver is built by connecting 4 pieces of clothoids together, as shown in Figure 3.11. The maximum curvature of the clothoid defined by Equation (3.16) is $2a^2L$. Thus, the maximum lateral force needed to track the clothoid is $m\dot{x}^2 \cdot 2a^2L$. Thus, by constraining a and L we can guarantee that the required lateral force is feasible.

For a point-mass model, the upper bound on lateral force is μmg . Thus, the constraint on curvature for a point-mass model is $2a^2L \leq \frac{\mu g}{\dot{x}^2}$. For the four-wheel model described in

section 3.2, the maximum achievable lateral force depends on the current vehicle states. At the beginning of a turn where lateral velocity \dot{y} and yaw rate $\dot{\psi}$ are both zero, the lateral force at the rear tires are zero, therefore the only source of lateral force at that instant are the front tires. We use sampling and extensive simulations of our vehicle model to determine the maximum available lateral force at the beginning of a turn ($0.46\mu mg$). After the turn begins, \dot{y} and $\dot{\psi}$ increase and more lateral force will become available. For a conservative approximation, the value $0.46\mu mg$ is taken as the upper bound of lateral force along the turn. The constraint on a and L for the clothoid becomes:

$$2a^2L \leq \frac{0.46\mu g}{\dot{x}^2} \quad (3.18a)$$

where μ is the friction coefficient and \dot{x} is the body-fixed longitudinal velocity.

Obstacle and safety constraints

We use scenario based ad-hoc constraints for the obstacle avoidance. Figure 3.12 shows one scenario where the vehicle takes a left lane change to avoid the obstacle which is presented by the red box. The lane change maneuver is composed of four clothoids, indicated by 1, 2, 3 and 4 in the figure. We assume 1 and 2 have the same parameters $[a_1, L_1]$ and 3, 4 share the same parameters $[a_2, L_2]$. The final heading angle at point B is assumed to be zero for simplification. The optimization variables for the high level are chosen to be $P = [X_a, X_b, Y_b]$. With a given P and the initial point $[X_0, Y_0, \psi_0]$, the parameters $[a_1, L_1, a_2, L_2]$ as well as the whole planned curve can be uniquely determined, including the coordinates (X_a, X_b, ψ_a) and (Y_a, Y_b, ψ_b) .

To guarantee obstacle avoidance, we apply the following constraints on P for the scenario in Figure 3.12:

$$X_a \leq X_c, \quad (3.19a)$$

$$X_b \geq X_c, \quad (3.19b)$$

$$Y_b \geq Y_c, \quad (3.19c)$$

$$(Y_c - Y_a) \leq (Y_b - Y_a) \cdot \frac{X_c - X_a}{X_b - X_a}, \quad (3.19d)$$

where (3.19a) ensures point A is at the left side of the obstacle; (3.19b) ensures point B has passed the left side of the obstacle; (3.19c) ensures point B is above the obstacle; (3.19d) ensures point C is at the bottom right side of the line \overline{AB} . Similar constraints can be formulated under other scenarios.

The following constraints are also applied for safety:

$$Y_{LB} \leq Y_b \leq Y_{UB}, \quad (3.20a)$$

$$|\psi_a| \leq \pi/4, \quad (3.20b)$$

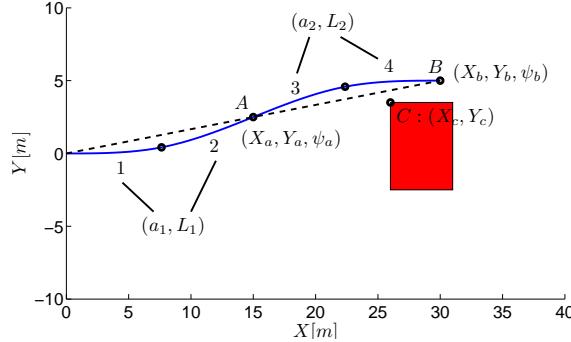


Figure 3.12: Ad-hoc Clothoid parameters setup for obstacle constraints.

$$K_{1_{max}} \leq \frac{0.46\mu g}{\dot{x}}, \quad (3.20c)$$

$$K_{2_{max}} \leq \frac{0.46\mu g}{\dot{x}}, \quad (3.20d)$$

where (3.20a) represents the road bound constraint, with Y_{LB} and Y_{UB} being the lower and upper bound of Y for the road bound, respectively; (3.20b) limits the maximum yaw angle of the vehicle along the planed path; (3.20c) and (3.20d) constraints the maximum curvature of the clothoids. $K_{1_{max}}$ and $K_{2_{max}}$ are functions of the optimization variable $P = [X_a, X_b, Y_b]$.

Equations (3.19) and (3.20) are compactly written as:

$$h(\xi_0, P) \leq \mathbf{0}, \quad (3.21a)$$

where $\mathbf{0}$ is the zero vector with appropriate dimension.

MPC formulation

The optimization problem for the high level path planner is formulated as follows:

$$\min_{P_t} J_N^{HL}(\xi_{t,t}, P_t) = K_{1_{max}}^2 + K_{2_{max}}^2 \quad (3.22a)$$

$$\text{subj. to } h(\xi_{t,t}, P_t) \leq \mathbf{0}, \quad (3.22b)$$

$$\xi_{t,t} = \xi(t), \quad (3.22c)$$

Once the optimal parameter vector P^* is determined, the optimal clothoid curves can be computed and fed to the low level. The values $K_{1_{max}}$ and $K_{2_{max}}$ tells how aggressive the maneuver has to be in order to avoid the obstacle. Based on that, The low level MPC can determine whether or not to intervene by taking over control and following the planned path from the high level.

A similar MPC is set up to take the vehicle back into the original lane.

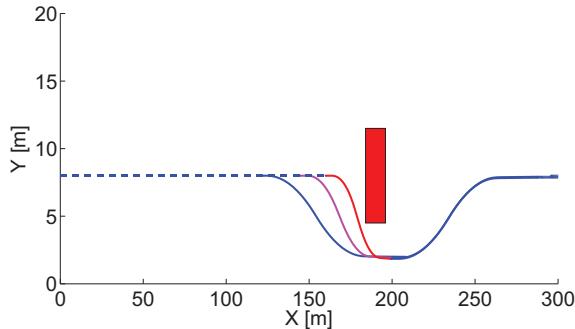


Figure 3.13: Simulation result: Various lane change maneuvers are compared. As the vehicle approaches the obstacle the planned paths become more aggressive (high curvatures).

Simulation and experimental results

In this subsection we show the simulation and experimental results of the proposed hierarchical MPC. The simulation and experimental setups are the same as the ones described in section 3.5. The experiments have been performed at a test center equipped with icy and snowy handling tracks.

Figure 3.13 shows three different lane change maneuvers. They are generated from the high-level path planner starting from $X = 120, 140$, and $160m$. As the vehicle approaches the obstacle, the planned paths become more aggressive, that is, the maximum curvature of the planned clothoids approaches the limit at the given speed and friction coefficient. The maneuver in red is the most aggressive maneuver the vehicle can handle at the given speed. The simulations presented in Figure 3.14 use a binary driver model. The driver is either attentive, as in the upper plot of Figure 3.14, and the low-level controller does not assume control until the planned path becomes aggressive, or the driver is distracted, as in the lower plot of Figure 3.14, and the low-level control begins tracking the planned path with a lower degree of curvature.

Figure 3.15 and 3.16 show the experimental results of the semi-autonomous driving on a snow track. In the experiments, X_b in (3.19b) is set to be equal to the X coordinate of the obstacle center, for simplicity. In the first test shown in Figure 3.15 the driver was assumed to be distracted and the path tracker assumed control early enough to track a smoothly planned trajectory. In the second experiment in Figure 3.16 the driver was assumed to be attentive and the controller did not take over until the planned path became aggressive. Both experiments successfully avoided the obstacle.

Note that the low-level path follower might not be able to track the planned path perfectly because of model mismatch and external disturbances. In some cases, the tracking error is large enough that the maneuver becomes infeasible to track. The second experiment in Figure 3.16 shows this interesting behavior. In the test, when infeasibility appeared, a braking maneuver was invoked to reduce the velocity and thus enlarge the feasible region.

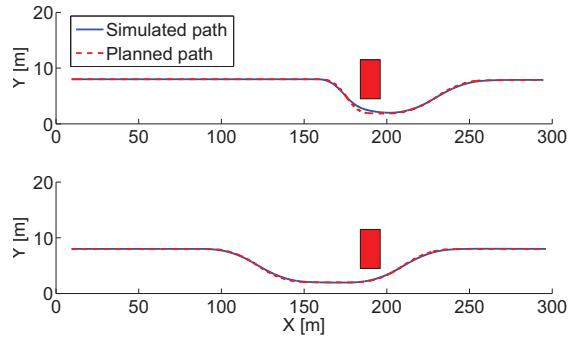


Figure 3.14: Simulation result: In the upper plot an attentive driver is assumed. The low-level control takes over when the planned path becomes aggressive. In the lower plot the low-level control takes over for a distracted driver and the result is a smoother and safer path.

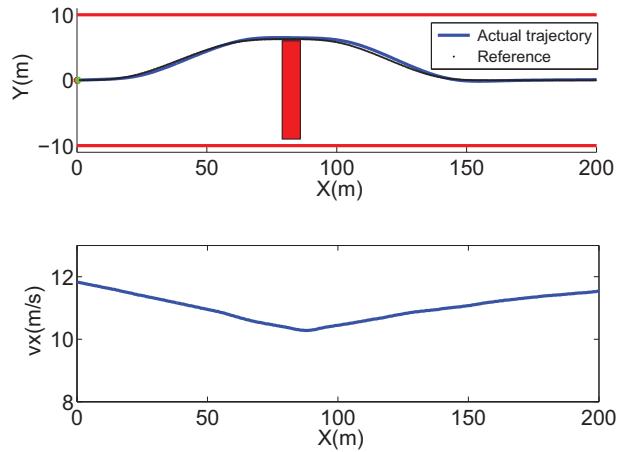


Figure 3.15: Experimental results: Vehicle successfully avoids the obstacle using maneuvers based on clothoids.

In Fig 3.16 we can see that after a few seconds of braking the path planner is able to find a feasible solution around the obstacle, braking is interrupted and the obstacle is successfully avoided.

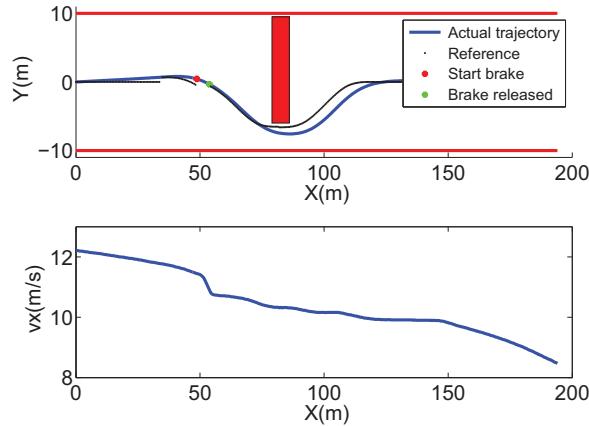


Figure 3.16: Experimental result: Actual path of the vehicle deviates from the planned path due to model mismatch and caused infeasibility of tracking. Braking was invoked to enlarge the feasible region in that situation.

3.8 Concluding Remarks

In this Chapter, we first compared two control architectures for autonomous obstacle avoidance, the **one-level** approach and the two-level hierarchical approach. The **one-level** approach is limited by its high computational complexity and is only able to control the vehicle at low speeds with compromised performances. While the **two-level** approach is more promising thanks to its low complexity due to the decomposition of the problem into a high level path replanner and low level path follower. With a simple vehicle model, the high level is able to plan over long prediction horizon at small sampling time. The higher fidelity vehicle model used in the low level allows the low level path follower to stabilize the vehicle around the planned path. However, due to the oversimplification of the point mass model, the high level planned path is sometimes infeasible for the real vehicle to track. Thus in the next chapter, We will propose an improved approach with the same two level hierarchical architecture.

In the last part of this chapter, We proposed a semi-autonomous obstacle avoidance hierarchical MPC based on a special kind of curve, the clothoid. Thanks to the properties of clothoid, we are able to plan feasible paths with only few parameters. We can determine the aggressiveness of the planned path from the optimal parameters and based on that determine the right time to intervene the vehicle control. The clothoid based hierarchical controller is fast for real-time computation and performs well. However, the use of ad-hoc obstacle constraints makes it hard to be extended to more generic scenarios. There is some study on the generic formulation utilizing the clothoids [26]. However, a mixed integer nonlinear programming is required in that approach, which is not suitable for fast real-time

implementation.

Chapter 4

Spatial Model Predictive Control

4.1 Introduction

Real-time generation and tracking of feasible trajectories is a major barrier in autonomous guidance systems when the vehicle travels at the limits of its handling capability. Trajectories generated by using oversimplified models violate system constraints, while computing trajectories using high-fidelity vehicle models and nonlinear optimization is computationally demanding. Moreover, the presence of external disturbances and model uncertainties might still prevent the vehicle from following the desired path.

The previous chapter proposed a hierarchical two-level NMPC scheme for the autonomous and semiautonomous driving problem. The high-level path planner uses a simplified point-mass vehicle model to generate an obstacle avoiding trajectory by using a NMPC controller. The trajectory is fed to the low-level path follower designed by using a NMPC based on a higher fidelity vehicle model. In [23] the proposed hierarchical framework has been implemented on an autonomous ground vehicle driving at high speeds on an icy road. Although this decomposition allows for real-time implementation, the trajectories generated by the point-mass path planner are not always feasible. The lower level tracking performance deteriorates and obstacle collisions can be observed in conditions where the obstacle could have been avoided.

In order to overcome this issue and still maintain real-time feasibility, the work in [27] uses a motion primitive path planner at the high level. Although the high level trajectories from the motion primitive planner are feasible for the high-fidelity model, the optimal trajectory requires the online solution of a mixed-integer program or the offline computation of a large look-up table. This prevents the use of such approach on current electronic control units. In order to plan a feasible path in real-time this paper studies the use of spatial predictive control at the high level. We follow the approach presented in [61, 30] and transform time-dependent vehicle dynamics into spatial-dependent dynamics. By using this approach obstacle constraints are translated into spatial constraints on the state vector. Simulation and experimental results show the controller's ability to avoid multiple obstacles

while tracking a reference in the center of the lane. Real-time tests have been conducted on a dSPACE platform in hardware-in-the-loop simulations. Furthermore, the hierarchical controller has been implemented on a semi-autonomous vehicle driving high-speed on ice.

This chapter is structured as follows. Section 4.2 describes the vehicle models used in the controller, including the nonlinear bicycle vehicle model and a spatial bicycle vehicle model. Section 4.3 outlines the hierarchical architecture of the controller and the MPC formulation. The simulation and experimental results are reported in section 4.4.

4.2 Vehicle model

This section introduces the vehicle models used for the control design. Section 4.2 describes the six state nonlinear vehicle model. Section 4.2 explains the transformation from a time-dependent bicycle model to a spatial-dependent model.

Four wheel model

As mentioned in Remark 1, we omit the wheel dynamics in the ten state four wheel model for simplicity in control design. This results in the six state four wheel vehicle model:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}} \quad (4.1a)$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}} \quad (4.1b)$$

$$\begin{aligned} I\ddot{\psi} = & a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + \\ & + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}) \end{aligned} \quad (4.1c)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi \quad (4.1d)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi \quad (4.1e)$$

Details of the model can be found in Chapter 2. This model is compactly written as:

$$\dot{\xi}(t) = f^{4ws}(\xi(t), u(t)) \quad (4.2)$$

where the superscript $4ws$ stands for “four wheel simplified”. The state and input vectors are $\xi = [\dot{y}, \dot{x}, \dot{\psi}, \psi, Y, X]$ and $u = [\delta_f, T_{f,l}, T_{f,r}, T_{r,l}, T_{r,r}]$.

Spatial bicycle model

Figure 4.1 shows the curvilinear coordinate system used in the spatial model as well as the states of the model. The solid curve represents the center line of the lane. The states of the spatial vehicle model are defined as $\xi^s = [\dot{y}, \dot{x}, \dot{\psi}, e_\psi, e_y]^T$. Where \dot{y} , \dot{x} and $\dot{\psi}$ are body frame velocities, e_ψ and e_y are the error of heading angle and lateral position with respect to the center line of the lane.

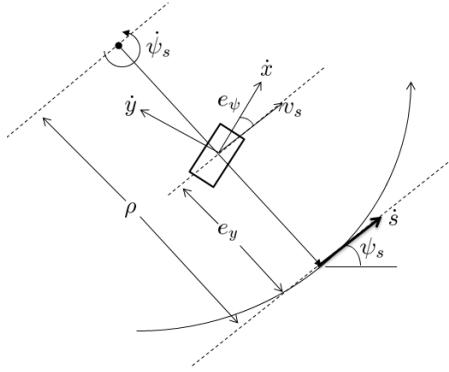


Figure 4.1: The curvilinear coordinate system. The dynamics are derived about a curve defining the center-line of a track. The coordinate s defines the arc-length along the track. The relative spatial coordinates e_y and e_ψ are shown.

The following kinematic equations can be derived from Figure 4.1:

$$v_s = (\rho - e_y) \cdot \dot{\psi}_s; \quad v_s = \dot{x} \cdot \cos(e_\psi) - \dot{y} \cdot \sin(e_\psi), \quad (4.3)$$

where v_s is the projected vehicle speed along direction of the lane center line, ρ and ψ_s are the radius of curvature and the heading of the lane center line. $\dot{\psi}_s$ is the time derivative of ψ_s . The vehicle's velocity along the path $\dot{s} = \frac{ds}{dt}$ is then given by

$$\dot{s} = \rho \cdot \dot{\psi}_s = \frac{\rho}{\rho - e_y} \cdot (\dot{x} \cdot \cos(e_\psi) - \dot{y} \cdot \sin(e_\psi)) \quad (4.4)$$

where s is the projected vehicle position along the lane center line. Using simple relationships in the new curvilinear coordinate system and the fact that $\frac{d\xi^s}{ds} = \frac{d\xi^s}{dt} \cdot \frac{dt}{ds}$ we can calculate the derivative of ξ^s with respect to s as follows ((.)' represents the derivative with respect to s):

$$\ddot{y}' = \ddot{y}/\dot{s}; \quad \ddot{x}' = \ddot{x}/\dot{s}; \quad \ddot{\psi}' = \ddot{\psi}/\dot{s}; \quad (4.5a)$$

$$e'_\psi = (\psi - \psi_s)' = \dot{\psi}/\dot{s} - \dot{\psi}_s'; \quad (4.5b)$$

$$e'_y = \dot{e}_y/\dot{s} = (\dot{x} \cdot \sin(e_\psi) + \dot{y} \cdot \cos(e_\psi))/\dot{s} \quad (4.5c)$$

where \ddot{y} , \ddot{x} and $\ddot{\psi}$ are computed from the nonlinear bicycle model (2.12), a simplified version of the four wheel model (4.1). The road information ψ'_s is assumed to be known. The spatial vehicle dynamics then can be formulated as:

$$\xi^{s'}(s) = f^s(\xi^s(s), u^s(s)). \quad (4.6)$$

where the inputs are the front steering angle δ_f and the braking or throttle effort $\beta_r \in [-1, 1]$ with -1 corresponding to maximum braking and 1 corresponding to maximum throttle ($u^s = [\delta_f, \beta_r]$). Note the time as function of s , $t(s)$, can be retrieved by integrating $t' = 1/\dot{s}$.

4.3 MPC control architecture with obstacle avoidance

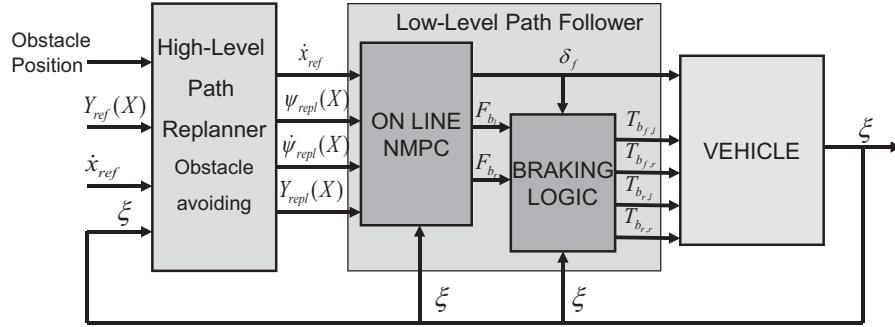


Figure 4.2: Architecture of the two-level MPC. A spatial vehicle model is used for high-level path planning. A four-wheel vehicle model is used for low-level path tracking.

A two level hierarchical control scheme [23] is used in this paper. As illustrated in Fig 4.2, the obstacle avoiding control problem is decomposed into two parts: a high level path planner and a low level path follower. The road information is fed into the high level. The high level plans an obstacle-free path using the nonlinear spatial vehicle model in section 2.24.2. The planned path (in spatial coordinates) is then fed to the low level path follower, which uses the four-wheel vehicle model (section 2.14.2) to compute the optimal control inputs in order to track the planned path. Each level solves a nonlinear MPC problem to plan or follow the path.

MPC formulation

The MPC controller is formulated as a general optimization problem

$$\min_{U_t} J_N(\tilde{\xi}_t, U_t, \Delta U_t) \quad (4.7a)$$

$$\text{subj. to } \xi_{k+1,t} = f(\xi_{k,t}, u_{k,t}) \quad k = t, \dots, t + H_p - 1 \quad (4.7b)$$

$$\Delta u_{k+1,t} = u_{k+1,t} - u_{k,t} \quad k = t, \dots, t + H_p - 1 \quad (4.7c)$$

$$\xi_{k,t} \in \mathcal{X} \quad k = t + 1, \dots, t + H_p - 1 \quad (4.7d)$$

$$u_{k,t} \in \mathcal{U} \quad k = t, \dots, t + H_p - 1 \quad (4.7e)$$

$$\Delta u_{k,t} \in \Delta \mathcal{U} \quad k = t + 1, \dots, t + H_p - 1 \quad (4.7f)$$

$$\xi_{t,t} = \xi(t) \quad (4.7g)$$

where $\tilde{\xi}_t = [\xi_{t,t}, \xi_{t+1,t}, \dots, \xi_{t+H_p-1,t}]$ is the sequence of states $\tilde{\xi}_t \in \mathbb{R}^{nH_p}$ over the prediction horizon H_p predicted at time t . Elements in $\tilde{\xi}_t$ are updated according to the discretized

dynamics of the vehicle model (4.7b). $u_{k,t}$ and $\Delta u_{k,t} \in \mathbb{R}^{m_r}$ is the k^{th} vector of the input sequence $U_t \in \mathbb{R}^{m_r H_p}$ and $\Delta U_t \in \mathbb{R}^{m_r (H_p - 1)}$ respectively,

$$U_t = [u_{t,t}^T, u_{t+1,t}^T, \dots, u_{t+H_u-1,t}^T, u_{t+H_u,t}^T, \dots, u_{t+H_p-1,t}^T]^T \quad (4.8a)$$

$$\Delta U_t = [\Delta u_{t+1,t}^T, \dots, \Delta u_{t+H_u-1,t}^T, \Delta u_{t+H_u,t}^T, \dots, \Delta u_{t+H_p-1,t}^T]^T \quad (4.8b)$$

Note the time “ t ” in this section denotes the **generalized time**, the independent variable with respect to which the system states are differentiated. Specifically, in the high level it is the spatial coordinate “ s ” while in the low level it is the time t .

The computational complexity of the MPC problem is reduced by two means. One is to hold the last $H_p - H_u$ input vectors in U_t constant and equal to the vector $u_{t+H_u-1,t}$. With this assumption, only the first H_u input vectors constitutes the optimization variables. The other one is to keep the input vectors constant for every iH_u steps in the first H_u input vectors (i.e., set $u_{t+n \cdot iH_u+k,t} = u_{t+n \cdot iH_u,t}, \forall k = 1, 2, \dots, iH_u$). With this assumption, the number of optimization variables is further reduced. We refer to H_p as the prediction horizon, H_u as the control horizon and iH_u as the input blocking factor.

At each time step t , the performance index $J_N(\tilde{\xi}_t, U_t, \Delta U_t)$ is optimized under the constraints (4.7c)-(4.7f) starting from the state $\xi_{t,t} = \xi(t)$ to obtain an optimal control sequence, $U_t^* = [u_{t,t}^*, \dots, u_{t+H_p-1,t}^*]^T$. The first of such optimal moves $u_{t,t}^*$ is the control action applied to the vehicle at time t . At time $t + 1$, a new optimization is solved over a shifted prediction horizon starting from the new measured state $\xi_{t+1,t+1} = \xi(t + 1)$. The time interval between time step $t + 1$ and time step t is the sampling time T_s .

High level path planner

The High level MPC uses the spatial bicycle vehicle model. Specifically, in equation (4.7b) we use a discretized version of the equation 4.6.

The cost function considers the deviation of the tracking states $\eta_{k,s}^{hl} = [\dot{x}_{k,s}, \dot{\psi}_{k,s}, e_{\psi_{k,s}}, e_{y_{k,s}}]^T$ with respect to the reference $\eta_{ref,k,s}^{hl} = [\dot{x}_{k,s}, \dot{\psi}_{k,s}, e_{\psi_{k,s}}, e_{y_{k,s}}]^T$, as well as the input and input rate.

$$J_N(\tilde{\xi}_s, U_s, \Delta U_s) = \sum_{k=s}^{s+H_{p,hl}-1} \|\eta_{k,s}^{hl} - \eta_{ref,k,s}^{hl}\|_{Q_{hl}}^2 + \|\dot{u}_{k,s}\|_{R_{hl}}^2 + \|\Delta u_{k,s}\|_{S_{hl}}^2 \quad (4.9)$$

Where the reference $\dot{x}_{ref,k,s}$ is a given constant, $e_{\psi_{ref,k,s}}$ and $e_{y_{ref,k,s}}$ are zero, $\dot{\psi}_{ref,k,s}$ is defined as $\psi'_s \cdot \dot{x}_{ref,k,s}$. Q_{hl} , R_{hl} and S_{hl} are weighting matrices with proper dimensions. The index for **general time** “ t ” is replaced by “ s ” here. The distance interval between step $s + 1$ and s is the sampling interval ds .

A spatial horizon allows one to formulate obstacle constraints as simple bounds on e_y and include them in the state constraints (4.7d). At each prediction step, the vehicle position along the lane center is known to be $(s + k \cdot ds)$. According to the position and width of the obstacle, one can determine the bounds on e_y . With one obstacle, there are two

disconnected regions of feasible e_y , respectively corresponding to passing the obstacle from left and right. In this paper, a simple heuristic based on the vehicle position and the size of each feasible region is used to determine which side should the vehicle pass through. In the case of multiple obstacles at the same coordinate “ s ”, a similar approach can still be used.

The optimal trajectory $[\dot{x}^*(s), \dot{\psi}^*(s), e_\psi^*(s), e_y^*(s), t^*(s)]$ is computed by simulating the vehicle model with the optimal inputs from the MPC problem and then passed to the low level path follower. Note $t^*(s)$ can be easily retrieved as described in section 2.24.2. In the low level, this spatial trajectory is transformed back to a time-dependent trajectory $[\dot{x}^*(t), \dot{\psi}^*(t), \psi^*(t), Y^*(t), X^*(t)]$ by coordinate transformation and interpolation.

Low level path follower

The low level MPC uses the four wheel vehicle model from section 4.2. At each time step t , the system dynamics are linearized around the equilibrium trajectory $[\xi_{k,t}, u_{k,t}]$, with $u_{k,t} = u_{t,t} \forall k = t, \dots, t + H_{p,ll}$ and $\xi_{k+1,t} = f_d^{4w}(\xi_{k,t}, u_{k,t})$, where f_d^{4w} is the discrete version of the equation $\dot{\xi}(t) = f^{4w}(\xi(t), u(t))$. The details of the linearizing process can be found in [17].

The cost function again consists of the deviation of the tracking states $\eta_{k,t}^{ll} = [\dot{x}_{k,t}, \psi_{k,t}, \dot{\psi}_{k,t}, Y_{k,t}]^T$ from the reference $\eta_{ref_{k,t}}^{ll} = [\dot{x}_{ref_{k,t}}, \psi_{ref_{k,t}}, \dot{\psi}_{ref_{k,t}}, Y_{ref_{k,t}}]^T$ as well as the input and input rate penalty.

$$J_N(\tilde{\xi}_t, U_t, \Delta U_t) = \sum_{k=t}^{s+H_{p,ll}-1} \|\eta_{k,t}^{ll} - \eta_{ref_{k,t}}^{ll}\|_{Q_{ll}}^2 + \|\dot{u}_{k,t}\|_{R_{ll}}^2 + \|\Delta u_{k,t}\|_{S_{ll}}^2 \quad (4.10)$$

The inputs vector $u_{k,t} = [\delta_f, F_{b_l}, F_{b_r}]^T$ consists of the steering angle δ_f , left braking force F_{b_l} and right braking force F_{b_r} . The braking logic in [19] is used to distribute the corresponding torques at the four wheels.

4.4 Simulation and experimental results

Simulation and experimental tests are conducted to evaluate the proposed controller. The MPC problem has been implemented as a C-coded s-function where NPSOL [25] is used to solve the high level optimization. At the low-level, the nonlinear program is solved by using a sequential quadratic programming approach [18] and the quadratic program is solved using the QP solver routine available in [59], which implements the Dantzig-Wolfe’s algorithm.

Simulation setup description

Real-time simulation of the controller is tested on a dSPACE rapid prototyping system consisting of a MicroAutoBox and a DS1006 processor board with a DS2211 I/O board.

The controller runs on the MicroAutoBox. The first element of the optimal control sequence is passed to DS1006 board, which simulates the vehicle dynamics using a four wheel vehicle model and Pacejka tire model [1], and then feeds the current vehicle state back to the controller. The two components communicate through a CAN bus.

Experimental setup description

The experiments have been performed at a test center equipped with icy and snowy handling tracks. The MPC controller has been tested on a passenger car, with a mass of 2050 Kg and a yaw inertia of 3344 Kg/m^2 . The controllers were run in a dSPACE Autobox system, equipped with a DS1005 processor board and a DS2210 I/O board.

We used an Oxford Technical Solution (OTS) RT3002 sensing system to measure the vehicle position and orientation in the inertial frame and the vehicle velocities in the vehicle body frame. The OTS RT3002 is housed in a small package that contains a differential GPS receiver, Inertial Measurement Unit (IMU) and a DSP. The IMU includes three accelerometers and three angular rate sensors. The DSP receives both the measurements from the IMU and the GPS, utilizes a Kalman filter for sensor fusion, and calculates the position, orientation and other states of the vehicle.

The car was equipped with an Active Front Steering (AFS) and Differential Braking system which utilizes an electric drive motor to change the relation between the hand steering wheel and road wheel angles. This is done independently from the hand wheel position, thus the front road wheel angle is obtained by summing the driver hand wheel position and the actuator angular movement.

The sensor, the dSPACE Autobox and the actuators communicate through a CAN bus.

Results and discussions

The controller is tested with the following tuning: **Tuning HL** Hierarchical MPC High-level: Nonlinear MPC (4.7) with model (4.6) and cost (4.9) and with the following parameters.

- $ds = 1.5\text{m}$, $H_{p,hl} = 15$, $H_{u,hl} = 12$, $iH_{u,hl} = 3$
- $\delta_f \in [-10^\circ, 10^\circ]$, $\Delta\delta_f \in [-17^\circ, 17^\circ] \times ds$
- $\beta_r \in [-1, 1]$, $\Delta\beta_r \in [-10, 10] \times ds$
- $Q_{hl} = \text{diag}(1, 1, 20, 1)$, $R_{hl} = \text{diag}(50, 50)$
 $S_{hl} = \text{diag}(0.1, 0.1)$

Tuning LL Hierarchical MPC Low-level: Nonlinear MPC (4.7) with model (4.2) and cost (4.10) and with the following parameters

- $T_{s,ll} = 0.05\text{s}$, $H_{p,ll} = 5$, $H_{u,ll} = 3$, $iH_{u,ll} = 1$
- $\delta_f \in [-10^\circ, 10^\circ]$, $\Delta\delta_f \in [-17^\circ, 17^\circ] \times T_{s,ll}$

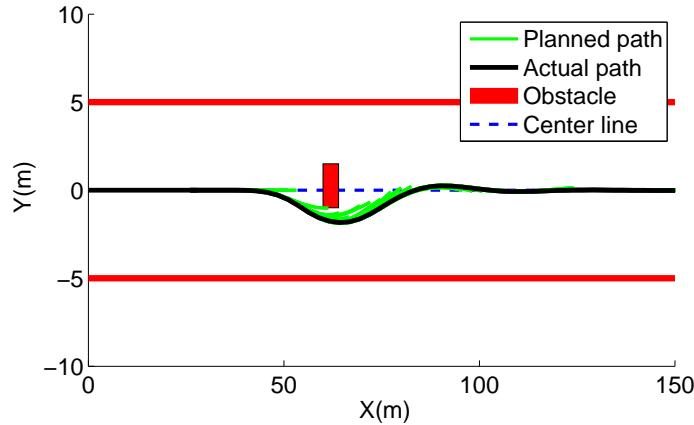


Figure 4.3: Simulated result: The vehicle entered the maneuver at 50 kph. The green lines are planned paths from the high-level which are updated every 200 ms. The black line is the actual trajectory the vehicle traveled.

- $F_{b\bullet} \in [-1500, 0]$, $\Delta F_{b\bullet} \in [-1000, 1000] \times T_{s,ll}$
- $Q = \text{diag}(10, 20, 10, 50)$, $R = \text{diag}(1, 0.5, 0.5)$
 $S = \text{diag}(1, 0.1, 0.1)$

The simulation and experimental results are summarized in Figures 4.3 to 4.6. In all tests, the road friction coefficient is approximately 0.3. The high level path planner is invoked every 200ms, and the low level every 50ms.

Since the obstacle constraint is formulated as hard constraints on the states, the high level path planner will always plan a tight path passing the obstacle due to the tracking error and input penalties, see the green line right below the obstacle in figure 4.3 as an example. To avoid this problem, an unsafe zone around the obstacle is added in the tests. The unsafe zones are shown in gray in figure 4.4 to 4.6.

Figure 4.3 and 4.4 are the simulation and experimental results of the vehicle avoiding one obstacle on a slippery road. The two tests show consistent performance. Figure 4.5 and 4.6 show the experimental results for avoiding two obstacles with different distances between them. In both cases the vehicle was able to avoid both obstacles and get back to the lane center afterwards. In figure 4.6 when the distance between the two obstacles was large, the vehicle was already trying to get back to the lane center before the second obstacle came into the planning horizon.

Figure 4.7 compares the controller's performance with a previously proposed controller in [23], which uses a time-dependent point mass model at the high level. The blue dots in the figures are the reference for the low level at each sampling time. We observe that the use of a nonlinear bicycle model at the high level greatly improves the tracking performance.

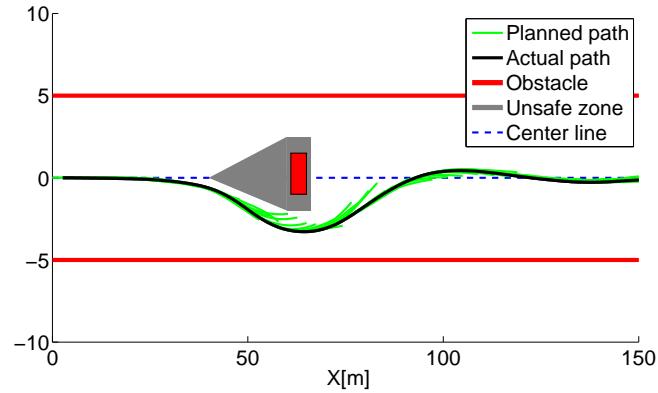


Figure 4.4: Experimental result: The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the obstacle and continued to track the lane center. The green lines are planned paths from the high-level which were updated every 200 ms. The black line is the actual trajectory the vehicle traveled.

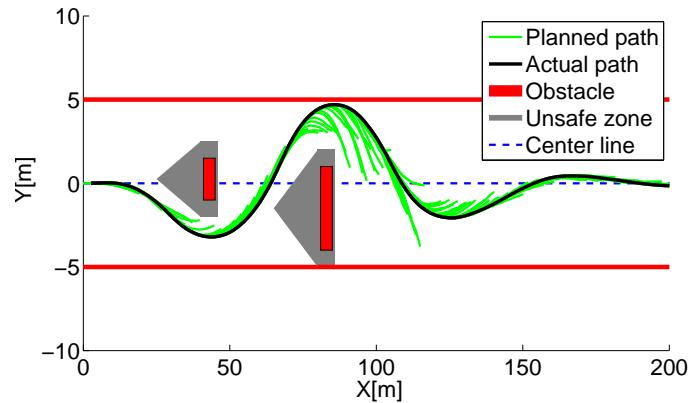


Figure 4.5: Experimental result: The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the two obstacles and continued to track the lane center.

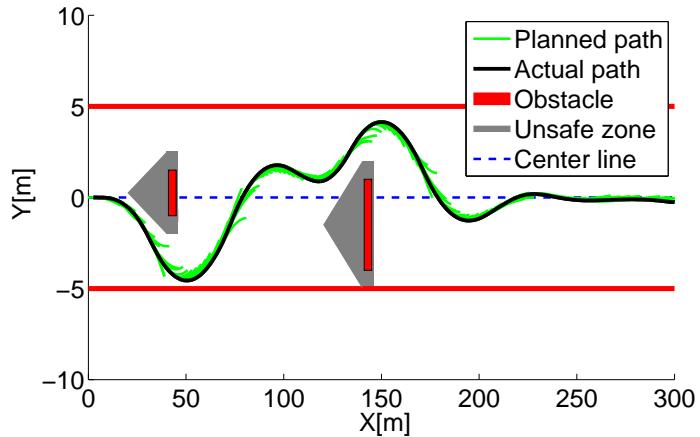


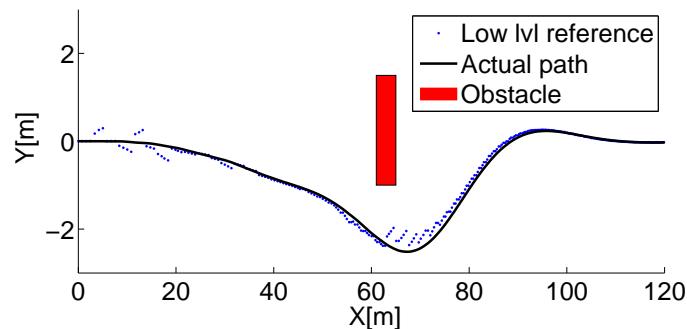
Figure 4.6: Experimental result: The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the first obstacle and continued to track the lane center until the second obstacle came into sight. It then turned left to avoid the second obstacle.

The use of a spatial model at the high level makes the construction of obstacle constraints straight forward as described in section 3.1 4.3. On the other hand, the hard constraint formulation for obstacle is not trivial for general shaped obstacles in MPC formulation with time-dependent models. Potential field approaches are a commonly used solution approach. Of course, it is possible to find indicating functions whose level sets can be used to form the obstacle constraints. This is usually not straight forward, especially if one wants the indicating function to have nice properties, such as differentiability, in order to speed up the optimization problem.

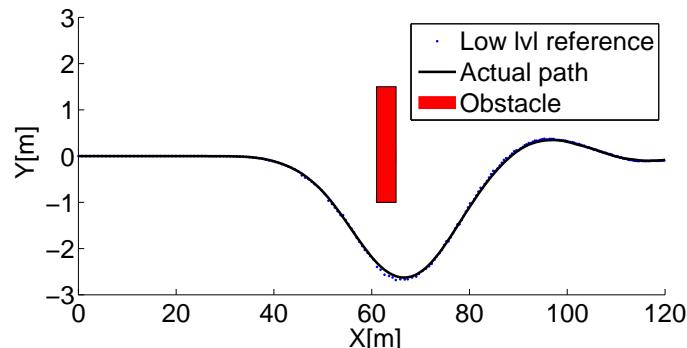
4.5 Concluding Remarks

In this chapter, a hierarchical design of a spatial MPC controller is presented for the obstacle avoiding and lane keeping problem of a ground vehicle on a slippery road. The control problem is decomposed into a high level path planner and a low level path follower. The high level uses a simplified spatial model while the low level uses a higher fidelity model. Experimental results show the controller is real-time implementable and able to avoid multiple obstacles at high speeds on slippery road.

Compared to the point mass model hierarchical MPC in the previous chapter, the tracking of the vehicle trajectory to the planed trajectory is greatly improved by the use of the spatial bicycle model. However, the high level planned path will never be tracked perfectly due to the model mismatch, state estimation error and external disturbances. Thus collision avoidance is not guaranteed even when the high level planned path is obstacle free. The



(a) Simulation result of the controller in [23]. A time-dependent point mass model is used for the high level. The vehicle turns early because the high level uses a potential field approach for obstacle avoidance.



(b) Simulation result of the new controller. A spatial-dependent bicycle model is used for the high level.

Figure 4.7: Simulation result comparison with the controller proposed in [23]. In both cases the high levels replan every 200ms. The same low level path follower is used, which uses a nonlinear four wheel model and runs every 50ms.

next chapter approaches this issue by proposing a robust NMPC framework which enforces robust constraint satisfaction under the presence of uncertainties.

Chapter 5

Robust Nonlinear Model Predictive Control

5.1 Introduction

Previous chapters have addressed the problem of autonomous and semiautonomous obstacle avoidance and lane keeping, by an approach of hierarchical MPC scheme. Successful implementations of such MPC scheme on autonomous vehicle driving at high speeds on slippery icy-snow track have been reported in [23, 24, 27]. However, robust constraint satisfaction is not guaranteed. Because uncertainties from a variety of sources might still prevent the vehicle from following the desired path and satisfying the safety constraints. Such uncertainties include measurement errors, friction coefficient estimation, driver behavior and model mismatch.

This chapter presents a systematic way of enforcing robustness under the presence of uncertainties during the MPC design stage. A robust nonlinear MPC framework [36, 43, 44, 55, 65, 2, 45, 63] is used in this chapter. In particular we follow the “Tube” approach presented in [63] for nonlinear system where the inputs enter linearly in the state update equations. The basic idea is to use a control law of the form $u = \bar{u} + K(\xi - \bar{\xi})$ where \bar{u} and $\bar{\xi}$ are the nominal control input and system states trajectories. For a given linear controller K , a robust invariant set is computed for the error system. The invariant is used to bound the maximum deviation of the actual states from the nominal states under the linear control action K . A nominal MPC optimizes \bar{u} and $\bar{\xi}$ with tightened state and input constraints. The tightening is computed as a function of the bounds derived from the robust invariant set.

In the first part of the chapter we propose a force-input nonlinear bicycle vehicle model. The model captures the main nonlinear vehicle dynamics and the input forces enter linearly in the system equations. In the second part we show how to compute the robust invariant for the nonlinear system controlled in closed-loop by a linear state-feedback controller. The invariant set is computed utilizing the Lipschitz constant of the nonlinear systems. In the

third part we use the robust invariant set computed off-line to tighten the state and input constraints for a nominal nonlinear MPC problem which computes obstacle-free trajectories and corresponding input sequences. Simulations and experiments in different scenarios show the effectiveness of the proposed approach. In particular, in the last part of the chapter, nominal MPC is compared with the proposed approach in terms of performance and constraint satisfaction.

This chapter is structured as follows. In section 5.2 the definitions of invariant sets are presented, the Robust MPC framework is outlined and the method used for computing the robust invariant set is introduced. In section 5.3 the force-input nonlinear vehicle model with additive uncertainty is developed. Section 5.4 details the robust invariant set computation. Section 5.5 presents the safety constraints and section 5.6 formulates the Robust MPC problem. Finally, in section 5.7 and 5.8 we present the simulation and experimental results showing the behavior of the proposed controller.

5.2 Invariant set and robust MPC

Background on set invariance theory

In this section several definitions are provided that will be important in developing the Robust MPC later in this chapter.

We denote by f_a the constrained, discrete time linear autonomous system perturbed by a bounded, additive disturbance

$$\xi_{k+1} = f_a(\xi_k) + w_k, \quad (5.1)$$

where ξ_k and w_k denote the state and the disturbance vectors. System (5.1) is subject to the constraints

$$\xi_k \in \Xi \subseteq \mathbb{R}^n, \quad w_k \in \mathcal{W} \subseteq \mathbb{R}^d, \quad (5.2)$$

where Ξ and \mathcal{W} are polyhedra that contain the origin in their interiors.

Definition 1 (*Reachable Set for autonomous systems*) Consider the autonomous system (5.1)-(5.2). The one-step robust reachable set from a given set of states \mathcal{S} is

$$\begin{aligned} \text{Reach}_{f_a}(\mathcal{S}, \mathcal{W}) &\stackrel{\Delta}{=} \{\xi \in \mathbb{R}^n | \\ &\exists \xi_0 \in \mathcal{S}, \exists w \in \mathcal{W} : \xi = f_a(\xi_0, w)\}. \end{aligned} \quad (5.3)$$

Similarly, for systems with inputs

$$\xi_{k+1} = f(\xi_k, u_k) + w_k, \quad (5.4)$$

subject to the constraints

$$\xi_k \in \Xi, \quad u_k \in \mathcal{U} \subseteq \mathbb{R}^m, \quad w_k \in \mathcal{W}, \quad (5.5)$$

we use the following definition of the one-step robust reachable set.

Definition 2 (*Reachable set for systems with external inputs*) Consider the system (5.4)-(5.5). The one-step robust reachable set from a given set of states \mathcal{S} is

$$\begin{aligned} \text{Reach}_f(\mathcal{S}, \mathcal{W}) &\stackrel{\Delta}{=} \{\xi \in \mathbb{R}^n \mid \\ &\exists \xi_0 \in \mathcal{S}, \exists u \in \mathcal{U}, \exists w \in \mathcal{W} : \xi = f(\xi_0, u, w)\}. \end{aligned} \quad (5.6)$$

All states contained in \mathcal{S} are mapped into the reachable set Reach_{f_a} under the map f_a for all disturbances $w \in \mathcal{W}$, and under the map f for all inputs $u \in \mathcal{U}$ and all disturbances $w \in \mathcal{W}$. We will next define *robust invariant sets*.

Definition 3 (*Robust Positively Invariant Set*) A set $\mathcal{Z} \subseteq \Xi$ is said to be a robust positively invariant set for the autonomous system (5.1) subject to the constraints in (5.2), if

$$\xi_0 \in \mathcal{Z} \Rightarrow \xi_k \in \mathcal{Z}, \quad \forall w_k \in \mathcal{W}, \quad \forall k \in \mathbb{N}^+. \quad (5.7)$$

Definition 4 (*Maximal Robust Positively Invariant Set*) The set $\mathcal{Z}_\infty \subseteq \Xi$ is the maximal robust invariant set for the autonomous system (5.1) subject to the constraints in (5.2), if \mathcal{Z}_∞ is a robust positively invariant set and \mathcal{Z}_∞ contains all robust positively invariant sets contained in Ξ that contain the origin.

Two set operations will be used in this article: the Pontryagin difference and the Minkowski sum. The Pontryagin difference of two polytopes \mathcal{P} and \mathcal{Q} is a polytope

$$\mathcal{P} \ominus \mathcal{Q} := \{x \in \mathbb{R}^n : x + q \in \mathcal{P}, \forall q \in \mathcal{Q}\}, \quad (5.8)$$

and the Minkowski sum of two polytopes \mathcal{P} and \mathcal{Q} is a polytope

$$\mathcal{P} \oplus \mathcal{Q} := \{x + q \in \mathbb{R}^n : x \in \mathcal{P}, q \in \mathcal{Q}\}. \quad (5.9)$$

The Multi-Parametric Toolbox (MPT) [35] can be used for the set computations.

Background on robust MPC

In this section we outline the framework used to develop the robust model predictive controller in section 5.6. We follow a notation similar to [43]. Consider the system:

$$\xi_{k+1} = A\xi_k + g(\xi_k) + Bu_k + w_k. \quad (5.10a)$$

$$s.t. \quad \xi_k \in \Xi \quad (5.10b)$$

$$u_k \in \mathcal{U} \quad (5.10c)$$

Where $g(\cdot) : \mathcal{R}^n \mapsto \mathcal{R}^n$ is a nonlinear Lipschitz function and $w_k \in \mathcal{W} \subset \mathcal{R}^n$ is an additive disturbance. We assume that the matrix pair (A, B) is controllable.

The control problem is divided into two parts. First a feedforward control input is computed for the system (5.10). System (5.10) with the computed feedforward input and

a zero disturbance sequence will be called the “nominal system”. Then, we design a state feedback controller which acts on the error between the actual state of system (5.10) and the predicted state of the nominal system. Next we formalize the aforementioned ideas.

We denote the N -step control sequence and the N -step disturbance sequence for system (5.10) as $\mathbf{u} = \{u_0, u_1, \dots, u_{N-1}\}$ and $\mathbf{w} = \{w_0, w_1, \dots, w_{N-1}\}$, respectively. Let $\Phi(k, x, \mathbf{u}, \mathbf{w})$ denote the solution of (5.10) at time k controlled by \mathbf{u} when $x_0 = x$. Furthermore, let $\bar{\Phi}(k, x, \bar{\mathbf{u}})$ denote the solution of the *nominal* system:

$$\bar{\xi}_{k+i+1} = A\bar{\xi}_{k+i} + g(\bar{\xi}_{k+i}) + B\bar{u}_{k+i}, \quad \forall i = 0, 1, \dots, N-1. \quad (5.11)$$

Let $e_k = \xi_k - \bar{\xi}_k$ be the error between the states of system (5.10) and the nominal system. Let the input u_k to system (5.10) be:

$$u_k = \bar{u}_k + \hat{u}(e_k), \quad (5.12)$$

where $\hat{u}(e_k) : \mathbb{R}^n \mapsto \mathbb{R}^m$. We define the error dynamics as:

$$e_{k+1} = Ae_k + B\hat{u}(e_k) + (g(\xi_k) - g(\bar{\xi}_k)) + w_k. \quad (5.13)$$

The authors of [43] proved the following result.

Proposition 1 Suppose that \mathcal{Z} is a Robust Positively Invariant Set of the error system (5.13) with control law $\hat{u}(\cdot)$. If $\xi_k \in \{\bar{\xi}_k\} \oplus \mathcal{Z}$, then $\xi_{k+i} \in \{\bar{\xi}_{k+i}\} \oplus \mathcal{Z}$ for all $i > 0$ and all admissible disturbance sequences $w_{k+i} \in \mathcal{W}$.

Proposition 1 states that if the state ξ_0 of system (5.10) starts close to the nominal state $\bar{\xi}_0$, the control law (5.12) will keep the state trajectory $\Phi(k, x, \mathbf{u}, \mathbf{w})$ within the robust positively invariant set \mathcal{Z} centred at the predicted nominal states $\bar{\Phi}(k, x, \bar{\mathbf{u}})$ for all admissible disturbance sequence \mathbf{w} :

$$\xi_0 \in \{\bar{\xi}_0\} \oplus \mathcal{Z} \Rightarrow \xi_k \in \{\bar{\xi}_k\} \oplus \mathcal{Z} \quad \forall w_k \in \mathcal{W}, \quad \forall k \geq 0. \quad (5.14)$$

Proposition 1 also suggests that if a feasible solution can always be found for the nominal system (5.11) subject to the tightened constraints

$$\bar{\Xi} = \Xi \ominus \mathcal{Z}, \quad \bar{\mathcal{U}} = \mathcal{U} \ominus \hat{u}(\mathcal{Z}), \quad (5.15)$$

then the control law (5.12) will ensure constraint satisfaction for the controlled *uncertain* system (5.10) [2].

Remark 5 For general nonlinear systems, the controller and invariant set pair $(\hat{u}(\cdot), \mathcal{Z})$ is usually hard to find. In the following section we will propose an algorithm for computing $(\hat{u}(\cdot), \mathcal{Z})$ for system in the form of (5.10), whose dynamics consist of a linear term $A\xi + Bu$ and a “small” nonlinear term $g(\xi)$.

Computation of the robust invariant set

In [63], the authors present a test criterion to determine if a given linear feedback controller, $u = Ke$, and an ellipsoidal set of the system states form a valid controller-invariant set pair. In this section we use a similar approach. In particular we use the Lipschitz constant of the nonlinear term to compute a robust invariant set.

The main idea is to bound the nonlinear term $(g(\xi) - g(\bar{\xi}))$ in (5.13) by using the Lipschitz constant of $g(\cdot)$, and treat it as part of the disturbance. We use the 2-norm over \mathbb{R}^n . The nonlinear function $g(\xi)$ is Lipschitz in the set Ξ with respect to ξ if $\exists L > 0$ such that

$$\|g(\xi_1) - g(\xi_2)\|_2 \leq L\|\xi_1 - \xi_2\|_2, \quad \forall \xi_1, \xi_2 \in \Xi. \quad (5.16)$$

The smallest constant L satisfying (5.16) is called the Lipschitz constant.

Let $L(\Xi)$ be the Lipschitz constant of $g(\cdot)$ over Ξ . From (5.16) we obtain

$$\forall \xi, \bar{\xi} \in \Xi \text{ and } e = (\xi - \bar{\xi}) \in \mathcal{E}, \|g(\xi) - g(\bar{\xi})\|_\infty \leq L(\Xi) \max_{e \in \mathcal{E}} \|e\|_2, \quad (5.17)$$

where \mathcal{E} is a subset of Ξ containing $\{0\}$. The inequality in (5.17) defines a box $\mathcal{B}(\mathcal{E})$:

$$\mathcal{B}(\mathcal{E}) = \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq L(\Xi) \max_{e \in \mathcal{E}} \|e\|_2\} \quad (5.18)$$

$\mathcal{B}(\mathcal{E})$ is used to bound the term $(g(\xi_k) - g(\bar{\xi}_k))$ in (5.13). Equation (5.13) is then written as:

$$e_{k+1} = Ae_k + B\hat{u}(e_k) + \tilde{w}_k, \quad (5.19)$$

where $\tilde{w}_k \in \tilde{\mathcal{W}} = \mathcal{W} \oplus \mathcal{B}(\mathcal{E})$.

Proposition 2 *For $\bar{\xi}, \xi \in \Xi$, If \mathcal{Z} is a robust positively invariant set for system (5.19) with a control law $\hat{u}(\cdot)$, then \mathcal{Z} is a robust positively invariant set for system (5.13) with the same control law $\hat{u}(\cdot)$.*

Proposition 2 is true because all possible values of $(g(\xi) - g(\bar{\xi}))$ in (5.13) lies inside $\mathcal{B}(\mathcal{E})$.

Since we assumed the matrix pair (A, B) to be controllable, there exists a stabilizing linear feedback gain K such that $(A + BK)$ is Hurwitz. Algorithm 1 is used to compute the minimal robust positively invariant set \mathcal{Z} associated with the gain K .

If Algorithm 1 terminates in finitely many iterations i_t , then $\mathcal{Z} = \Omega_{i_t}$ is the minimal positively invariant set for the uncertain system (5.19) [54]. The choice of K will determine the size of \mathcal{Z} , and also the size of $K\mathcal{Z}$. Later in this chapter we will show that $K\mathcal{Z}$ is used to tighten the input constraints for ensuring robust constraint satisfaction. A larger K will result in a smaller \mathcal{Z} , in general, but may lead to larger $K\mathcal{Z}$ and thus smaller input actions.

Algorithm 1 Computation of the minimal positively invariant set \mathcal{Z}

- 1: $\Omega_0 \leftarrow \{\mathbf{0}\}$
 - 2: $\tilde{\mathcal{W}} \leftarrow \mathcal{W}$
 - 3: $i \leftarrow 0;$
 - 4: **repeat**
 - 5: $i = i+1$
 - 6: $\tilde{\mathcal{W}} = \mathcal{W} \oplus \mathcal{B}(\Omega_i);$
 - 7: $\Omega_i = \text{Reach}_{f_a}(\Omega_{i-1}, \tilde{\mathcal{W}}) \cup \Omega_{i-1}.$
 - 8: **until** $\Omega_i == \Omega_{i-1}$
 - 9: $\mathcal{Z} \leftarrow \Omega_i$
-

5.3 Modeling

In this section we present the mathematical models used for control design. The model is based on a nonlinear bicycle model [13]. It is then simplified using the center of percussion and linearizing of the rear tire model. The first subsection introduces the vehicle dynamics, the second introduces the concept of the center of percussion, and the third subsection introduces a further model simplification.

Bicycle vehicle model

Consider the vehicle sketch in Figure 5.1.

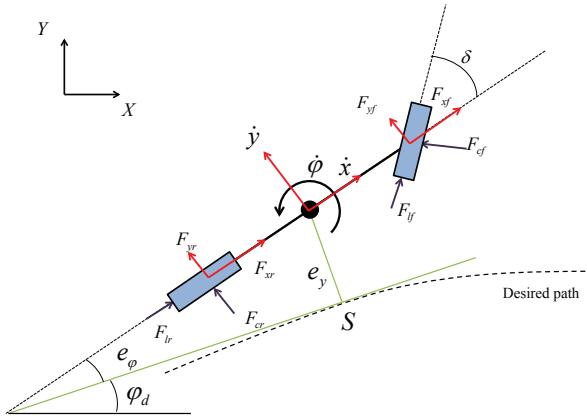


Figure 5.1: Modeling notation depicting the forces in the vehicle body-fixed frame (F_{x*} and F_{y*}), the forces in the tire-fixed frame (F_{l*} and F_{c*}), and the rotational and translational velocities. The relative coordinates e_y and e_ϕ are illustrated on the sketch of the road as well as the road tangent ψ_d .

We use the following set of differential equations to describe the vehicle motion within the lane,

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{xf} + 2F_{xr}, \quad (5.20a)$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr}, \quad (5.20b)$$

$$I_z\ddot{\psi} = 2aF_{yf} - 2bF_{yr}, \quad (5.20c)$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \quad (5.20d)$$

$$\dot{e}_y = \dot{y} \cos(e_\psi) + \dot{x} \sin(e_\psi), \quad (5.20e)$$

$$\dot{s} = \dot{x} \cos(e_\psi) - \dot{y} \sin(e_\psi), \quad (5.20f)$$

where m and I_z denote the vehicle mass and yaw inertia, respectively, a and b denote the distances from the vehicle center of gravity to the front and rear axles, respectively. \dot{x} and \dot{y} denote the vehicle longitudinal and lateral velocities, respectively, and $\dot{\psi}$ is the turning rate around a vertical axis at the vehicle's center of gravity. e_ψ and e_y in Figure 5.1 denote the vehicle orientation and lateral position, respectively, in a road aligned coordinate frame and ψ_d is the angle of the tangent to the road centerline in a fix coordinate frame. s is the vehicle longitudinal position along the desired path. F_{yf} and F_{yr} are front and rear tire forces acting along the vehicle lateral axis, F_{xf} and F_{xr} forces acting along the vehicle longitudinal axis

The longitudinal and lateral tire force components in the vehicle body frame are modeled as

$$F_{x\star} = F_{l\star} \cos(\delta_\star) - F_{c\star} \sin(\delta_\star), \quad (5.21a)$$

$$F_{y\star} = F_{l\star} \sin(\delta_\star) + F_{c\star} \cos(\delta_\star), \quad \star \in \{f, r\}, \quad (5.21b)$$

where \star denotes either f or r for front and rear tire, δ_\star is the steering angle at wheel. We introduce the following assumption on the steering angles.

Assumption 4 Only the steering angle at the front wheels can be controlled. i.e., $\delta_f = \delta$ and $\delta_r = 0$.

The longitudinal and lateral tire forces $F_{l\star}$ and $F_{c\star}$ are given by Pacejka's model [51]. They are nonlinear functions of the tire slip angles α_\star , slip ratios σ_\star , normal forces $F_{z\star}$ and friction coefficient between the tire and road μ_\star :

$$F_{l\star} = f_l(\alpha_\star, \sigma_\star, F_{z\star}, \mu_\star) \quad (5.22a)$$

$$F_{c\star} = f_c(\alpha_\star, \sigma_\star, F_{z\star}, \mu_\star) \quad (5.22b)$$

The slip ratios α_\star are approximated as follows:

$$\alpha_f = \frac{\dot{y} + a\dot{\psi}}{\dot{x}} - \delta, \quad \alpha_r = \frac{\dot{y} - b\dot{\psi}}{\dot{x}}. \quad (5.23)$$

The nonlinear tire force model (5.21)-(5.22) increases the computational burden of real-time model-based control techniques. We will approach this issue as follows. For the front wheels, since both slip angle and slip ratio can be controlled, we will assume that any force within the friction circle can be achieved. For this reason, we will use a model where the front tire forces F_{xf} and F_{yf} are the inputs. A lower level controller will achieve the desired forces. For the rear wheels, however, the slip angles are not directly controllable due to Assumption 4 and are functions of the vehicle states. Thus the rear tire forces cannot be treated in the same way as the front tire forces. In the following subsections we will discuss how to simplify the rear tire force model.

Center of percussion

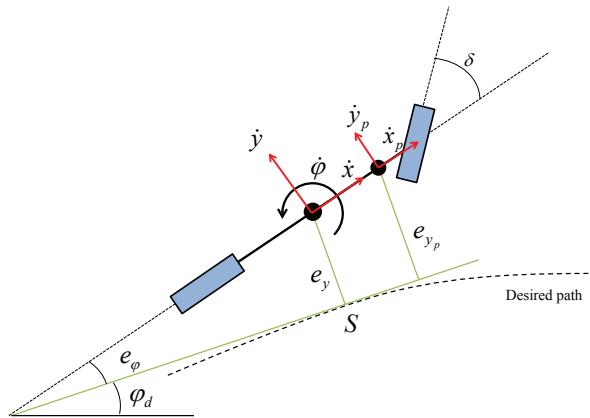


Figure 5.2: Illustration of center of percussion. p denotes the distance from CoG to CoP

In this section, the center of percussion (CoP) is introduced to eliminate the rear tire lateral forces, F_{yr} , in (5.20b). The center of percussion (CoP) is a point along the vehicle longitudinal axis, where the rear tire lateral forces do not affect its dynamic evolution ([47, 34]). As discussed in [34], the rear tire lateral forces have two effects on dynamics, a constant lateral acceleration on the vehicle body and an angular acceleration around the CoG. At the CoP these two effects cancel each other and thus \dot{y}_p is not influenced by the rear tire forces. Figure 5.2 illustrates the concept of CoP. Note that $\dot{x}_p = \dot{x}$. To find the position of the CoP, we refer to the following dynamic equations

$$\dot{y}_p = \dot{y} + p\dot{\psi} \quad (5.24a)$$

$$\ddot{y}_p = \ddot{y} + p\ddot{\psi} \quad (5.24b)$$

By substituting (5.20b) and (5.20c) into (5.24b) we obtain

$$\ddot{y}_p = -\dot{x}\dot{\psi} + \left(\frac{2}{m} + \frac{2ap}{I_z}\right)F_{yf} + \left(\frac{2}{m} - \frac{2bp}{I_z}\right)F_{yr}. \quad (5.25)$$

When $p = \frac{I_z}{mb}$, the F_{yr} term in (5.25) vanishes.

Simplifying rear tire forces

The use of the CoP eliminates the rear tire force in \ddot{y}_p . However, the rear tire forces are still present in the longitudinal and yaw dynamics \ddot{x}_p and $\dot{\psi}$. We will use a linearized tire model for these terms.

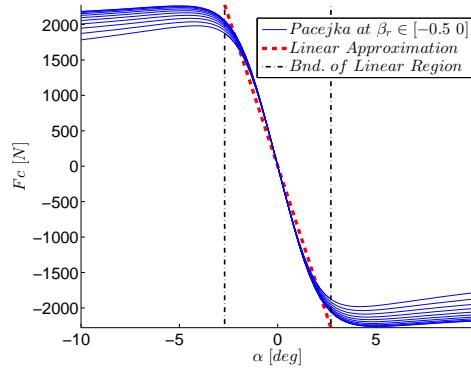


Figure 5.3: Lateral tire force. μ is 0.5 in Pacejka tire model. When β_r varies moderately, F_c behaves similar within the linear region.

Let β_r denote the braking/throttle ratio of rear tire ($F_{lr} = \beta_r \mu F z_r$), with $\beta_r=1$ corresponding to max throttle and -1 max braking. In moderate braking, the resultant lateral tire force F_c as a function of slip angle α has small sensitivity with respect to β_r , as shown in Fig 5.3. Thus a constant linear gain can be used over $\beta \in [-0.5 0]$ and over $\beta \in [0 0.5]$. The remaining rear tire forces in (5.20a) and (5.20c) are modeled as:

$$F_{lr} = \beta_r \mu F z_r \quad (5.26a)$$

$$F_{cr} = C_r \alpha_r \quad (5.26b)$$

Remark 6 *The same linearization could be applied to the F_{yr} term in (5.20b). We decided to use CoP to eliminate it (Section 5.3) to reduce the error introduced by the linearization.*

Force-input nonlinear bicycle model

In summary, the nominal vehicle model (5.20) used for control design is rewritten as:

$$\ddot{y}_p = \frac{2(a+b)\mu F_{zf}\beta_{xf}}{mb} - \dot{\psi}\dot{x}_p, \quad (5.27a)$$

$$\ddot{x}_p = \frac{2\mu F_{zf}\beta_{xf}}{m} + \frac{2\mu F_{zr}\beta_r}{m} + \dot{\psi}\dot{y}_p - \dot{\psi}^2 p, \quad (5.27b)$$

$$\ddot{\psi} = \frac{2a\mu F_{zf}\beta_{yf}}{I_z} - \frac{2bC_r\dot{y}_p}{I_z\dot{x}_p} + \frac{2bC_r(b+p)\dot{\psi}}{I_z\dot{x}_p}, \quad (5.27c)$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \quad (5.27d)$$

$$\dot{e}_y = \dot{y}_p + \dot{x}_p e_\psi, \quad (5.27e)$$

$$\dot{s} = \dot{x}_p \quad (5.27f)$$

where the state is $\xi = [\dot{y}_p, \dot{x}_p, \dot{\psi}, e_\psi, e_y, s]^T \in \mathbb{R}^{6 \times 1}$, and the input is $u = [\beta_{xf}, \beta_{yf}, \beta_r]^T \in \mathbb{R}^3$. β_{xf} and β_{yf} are the normalized longitudinal and lateral tire forces on the front wheels.

The following assumptions are used in (5.27):

Assumption 5 *The vehicle orientation error with respect to the road orientation, e_ψ , is small and thus $\sin(e_\psi) \simeq e_\psi$, $\cos(e_\psi) \simeq 1$. The vehicle side slip angle is assumed to be small, i.e. $\dot{y} \ll \dot{x}$ and $\dot{y}_p \ll \dot{x}_p$.*

Assumption 6 *The normal forces F_{z*} are assumed constant and determined by the vehicle's steady state weight distribution when no acceleration is applied*

Assumption 7 *The signal $\dot{\psi}_d$ is assumed to be known and at every time instant an estimate of $\dot{\psi}_d$ is available over a finite time horizon. See [4] for an overview of sensing technologies that can be used to obtain this signal.*

5.4 Invariant set computation

The objective of the robust MPC is to safely navigate the vehicle to avoid obstacles and stay on the road, in the presence of uncertainties. Uncertainties include measurement errors, friction coefficient estimation, driver behavior and model mismatch. In this section we derive the robust control law defined in (5.12) for the system (5.28), and provide the details of computing the robust positively invariant set used to guarantee robust constraint satisfactions.

Model reformulation

Equation (5.27) can be written in a similar form of (5.10):

$$\dot{\xi} = A\xi + g(\xi) + Bu + G + w, \quad (5.28)$$

where all nonlinear terms are collected in $g(\cdot) : \mathcal{R}^n \mapsto \mathcal{R}^n$. The matrices in (5.28) are

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -\bar{\dot{x}}_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{2bC_r}{I\ddot{x}_p} & 0 & \frac{2bC_r(p+b)}{I\dot{x}_p} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & \dot{\bar{x}}_p & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{g}(\xi) = \begin{bmatrix} -\Delta\dot{x}_p\dot{\psi} \\ \dot{\psi}\dot{y}_p - \dot{\psi}^2 p \\ \frac{2bC_r(\dot{y}_p\Delta\dot{x}_p - (p+b)\dot{\psi}\Delta\dot{x}_p)}{I_z\dot{x}_p(\dot{x}_p + \Delta\dot{x}_p)} \\ 0 \\ \Delta\dot{x}_p e_\psi \\ 0 \end{bmatrix}, \quad (5.29a)$$

$$\mathbf{B} = \begin{bmatrix} 0 & \frac{2\mu F_{zf}}{m} + \frac{2a\mu F_{zf}}{mb} & 0 \\ \frac{2\mu F_{zf}}{m} & 0 & \frac{2\mu F_{zr}}{m} \\ 0 & \frac{2a\mu F_{zf}}{I_z} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\dot{\psi}_d \\ 0 \\ 0 \end{bmatrix}, \quad (5.29b)$$

Note \dot{x}_p is rewritten as $\dot{x}_p = \bar{\dot{x}}_p + \Delta\dot{x}_p$. The signal w represents an additive disturbance. We follow the approach described in section 5.2 to compute the minimal positively invariant set \mathcal{Z} (also called the tube). The algorithm requires an error model, a linear feedback gain and a description of potential disturbance bounds. We use the model (5.28) and (5.29) to compute the error model and the lipschitz constant as described in section 5.2. The selection of the gain is described in section 5.4, and the computation of the disturbance bounds in section 5.4.

Choice of linear stabilising gain

We choose the stabilising state feedback gain K for the error system as the infinite horizon LQR solution K_{LQR}^∞ for system (A, B) . Therefore the controller (5.12) becomes:

$$u_k = \bar{u}_k + K_{LQR}^\infty(\xi_k - \bar{\xi}_k). \quad (5.30)$$

The controller K is used in the computation of the robust invariant set \mathcal{Z} . \mathcal{Z} is computed off-line using Algorithm 1 and a sampled version of (5.28) with a sampling time of 50ms. Specifically, for our system described in (5.28), the linear part of the dynamics are separated into two parts: the longitudinal part includes the states $[\dot{x}_p, s]$ and inputs $[\beta_{xf}, \beta_r]$; and the lateral part includes the states $[\dot{y}_p, \dot{\psi}, e_\psi, e_y]$ and the input β_{yf} . The corresponding

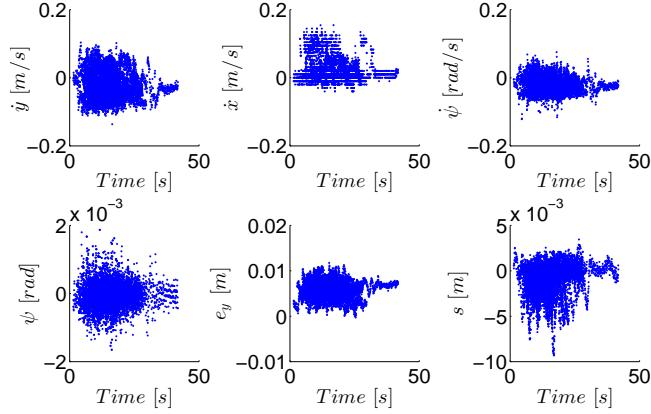


Figure 5.4: One-step model prediction error (equation (5.31)) of 47 testing trials.

invariant sets $\mathcal{Z}_{longitudinal}$ and $\mathcal{Z}_{lateral}$ in the reduced state spaces are computed separately. This reduces the computational complexity of the invariant set computation.

Disturbance bound

The bounded disturbance w is identified using experimental data from a test vehicle. The test vehicle is described in detail in section 5.8. We compare the one-step state prediction of model (5.28), discretized with a sampling time of 50 ms, with the measured vehicle states:

$$e_k^p = \xi_{k+1} - f^d(u_k, \xi_k), \quad (5.31)$$

where f^d denotes the discretized model, ξ_k denotes the measured vehicle state at time step k , and u_k the measured input command. Figure 5.4 shows e^p for 47 testing trials. In each trial, the vehicle takes maneuvers that typically appears in normal and evasive driving situations, such as driving straight, double lane change and slalom. We use a simple bound $|w| \leq [0.2, 0.2, 0.2, 0.005, 0.05, 0.05]$ to conservatively estimate the disturbance bound.

Remark 7 One can notice biases in some of the state errors. They arise both from the simplified model and from state estimation. For instance the bias in the state s is caused by the approximation $\cos(e_\psi) \simeq 1$; the bias in the state e_y is caused by a negative error on vehicle yaw angle estimation; the bias of state error on \dot{x} derives from the driver's throttle input during the test.

Remark 8 The model mismatch and disturbance identification is not the focus of this thesis and it is simple and very conservative here. More sophisticated method may be worth investigating in the future.

5.5 Safety constraints

The main objective of the safety system proposed in this chapter is to keep the vehicle on the road while avoiding obstacles. In this section, we detail the constraints imposed on the vehicle states and inputs to guarantees safe maneuvers.

Road boundary constraint: we constrain the CoG of the vehicle and shrink the road bounds to take the vehicle width into account. The road boundary constraints are written as

$$\bar{e}_{y_{min}} \leq e_y \leq \bar{e}_{y_{max}}, \quad (5.32)$$

where $\bar{e}_{y_{min}}$ and $\bar{e}_{y_{max}}$ are obtained from the tightened state constraint $\bar{\Xi}$.

Obstacle avoiding constraint: consider ellipsoidal obstacles in the form $(\frac{s-s_{obs}}{a_{obs}})^2 + (\frac{e_y-y_{obs}}{b_{obs}})^2 \leq 1$. The obstacle avoiding constraints are written as

$$(\frac{|s - s_{obs}| - s_{\mathcal{Z}}}{a_{obs}})^2 + (\frac{|e_y - y_{obs}| - y_{\mathcal{Z}}}{b_{obs}})^2 \geq 1, \quad (5.33)$$

where $s_{\mathcal{Z}}$ and $y_{\mathcal{Z}}$ are the projection of the invariant set onto s and e_y axes.

Slip angle constraint: we constraint the vehicle to operates in a space where a non-skilled driver can easily handle the vehicle. In particular, we constrain the tire slip angles to belong to the linear region of the tire forces.

$$\alpha_{min} \leq \alpha_{\star} \leq \alpha_{max}. \quad (5.34)$$

The constraints (5.32)-(5.34) are compactly written as,

$$h(\xi, u, w) \leq \mathbf{0}, \quad (5.35)$$

where $\mathbf{0}$ is the zero vector with appropriate dimension.

5.6 Robust predictive control design

In this section we formulate the lane keeping and obstacle avoiding problem as a Model Predictive Control Problem (MPC). At each sampling time instant an optimal input sequence is computed by solving a constrained finite time optimal control problem. The computed optimal control input sequence as well as the corresponding predicted vehicle state trajectory are stored as the nominal input and state trajectories, $\bar{\xi}$ and $\bar{\mathbf{u}}$. At the next time step the optimal control problem is solved staring from new state measurements. An algorithm computes the augmented input $u = \bar{u} + K e$ at sampling time. The MPC and the computation of u can be executed at different sampling times.

We discretize the system (5.28) with a fixed sampling time T_s to obtain,

$$\bar{x}_{k+1} = f_n(\bar{x}_k, \bar{u}_k), \quad (5.36)$$

and formulate the nominal optimization problem with tightened constraints as

$$\min_{\Delta \bar{\mathbf{u}}, \varepsilon} \sum_{k=0}^{H_p-1} \|\bar{\xi}_{t+k,t} - \xi_{ref}\|_Q^2 + \|\bar{u}_{t+k,t}\|_R^2 + \|\Delta \bar{u}_{t+k,t}\|_S^2 + \lambda \varepsilon \quad (5.37a)$$

$$s.t. \bar{\xi}_{t+k+1,t} = f_n(\bar{\xi}_{t+k,t}, \bar{u}_{t+k,t}), \quad (5.37b)$$

$$k = 0, 1, \dots, H_p - 1$$

$$h_t(\bar{\xi}_{t+k,t}, \bar{u}_{t+k,t}) \leq \mathbf{1}\varepsilon, \quad k = 1, \dots, H_p \quad (5.37c)$$

$$\varepsilon \geq 0, \quad (5.37d)$$

$$\bar{u}_{t+k,t} = \Delta \bar{u}_{t+k,t} + \bar{u}_{t+k-1,t}, \quad (5.37e)$$

$$\bar{u}_{t+k,t} \in \bar{\mathcal{U}}, \quad k = 0, 1, \dots, H_c - 1 \quad (5.37f)$$

$$\Delta \bar{u}_{t+k,t} \in \Delta \bar{\mathcal{U}}, \quad k = 0, 1, \dots, H_c - 1 \quad (5.37g)$$

$$\Delta \bar{u}_{t+k,t} = 0, \quad k = H_c, \dots, H_p \quad (5.37h)$$

$$\bar{u}_{t-1,t} = \bar{u}(t-1), \quad (5.37i)$$

$$\bar{\xi}_{t,t} = \bar{\xi}(t), \quad (5.37j)$$

where t denotes the current time instant and $\bar{\xi}_{t+k,t}$ denotes the predicted state at time $t+k$ obtained by applying the control sequence $\bar{\mathbf{u}} = \{\bar{u}_{t,t}, \dots, \bar{u}_{t+k,t}\}$ to the system (5.36) with $\bar{\xi}_{t,t} = \bar{\xi}(t)$. H_p and H_c denote the prediction horizon and control horizon, respectively. We denote by H_i the input blocking factor to hold the $\Delta \bar{u}_k$ constant for a length of H_i . This reduces the number of optimization variable which is useful in real-time implementation. The safety constraints (5.35) have been imposed as soft constraints, by introducing the slack variable ε in (5.37a) and (5.37c). Q , R , S and λ are weights of appropriate dimension penalizing state tracking error, control action, change rate of control, and violation of the soft constraints, respectively.

Remark 9 *Because of the system nonlinearities and time varying constraints, it is difficult to guarantee the feasibility of the nominal nonlinear MPC under all possible scenarios. However, if the nominal MPC is feasible, the vehicle is guaranteed to keep the nominal trajectory within the robust bounds under any admissible disturbance.*

5.7 Simulation results

In this section simulations results are presented. The model predictive control problem is solved using NPSOL [25] at each time step. The off-line computation of robust invariant sets through Algorithm 1 was run in Matlab® using the MPT Toolbox [35]. Table 5.1 lists the parameters used in the simulations. The controller is connected in closed loop with a higher fidelity nonlinear four-wheel vehicle model.

Figure 5.5 shows a simulation where the vehicle is controlled to successfully avoid two obstacles on the road and return to the road center. The actual trajectory is very close to the

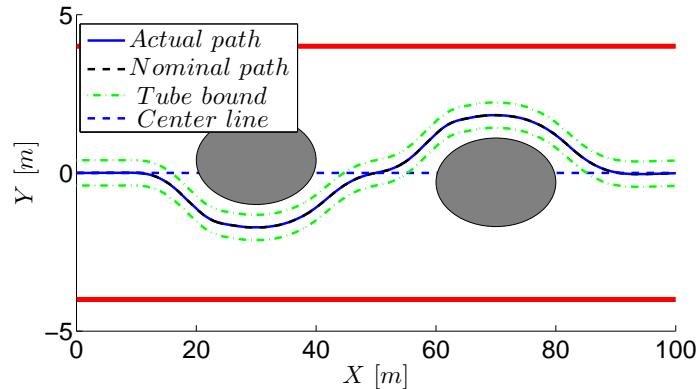


Figure 5.5: Simulation result: The vehicle enters the maneuver at 50Kph. The dashed black line and blue line are the nominal and actual vehicle trajectories respectively. The green dot-dashed lines indicate the robust bounds around the nominal trajectory. The actual vehicle path is seen to be very close to the nominal one and within the robust bounds.

nominal trajectory due to fast replanning. The dot-dashed lines show the projection of the e_y bounds of robust invariant set, $\text{Proj}_{e_y}(\mathcal{Z})$. The actual trajectory is within the robust bounds. Figure 5.6 shows the control input to the vehicle during the simulation. Note these inputs are converted from the desired force input from the MPC by a force controller. The vehicle brakes when encountering the obstacles to reduce the speed and hence the aggressiveness of the maneuver, it then throttles to get back to the reference speed.

Table 5.1: Simulation Parameters

Param.	Value	Units	Param.	Value	Units
m	2050	kg	\bar{u}_{max}	[0.5, 0.5, 0.5]	-
Iz	3344	kgm^2	\bar{u}_{min}	-[0.5, 0.5, 0.5]	-
C_α	65000	N	$\Delta\bar{u}_{max}$	[1, 1, 1]	-
μ	0.5	-	$\Delta\bar{u}_{min}$	-[1, 1, 1]	-
T_s	50	ms	α_{max}	4	deg
H_p	16	-	α_{min}	-4	deg
H_c	16	-	Q	[0, 0.1, 1, 1, 5, 0]	-
H_i	2	-	R	[1, 1, 1]	-
$\bar{e}_{y_{max}}$	3.5	m	S	[1, 1, 1]	-
$\bar{e}_{y_{min}}$	-3.5	m	λ	10000	-

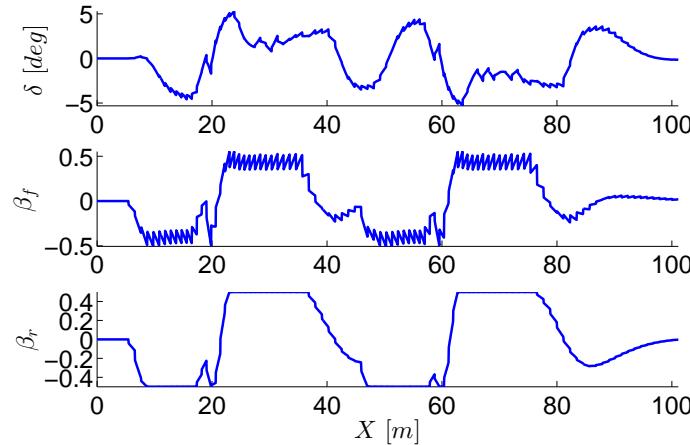


Figure 5.6: Simulation result: Control inputs during the simulation in figure 5.5. β_f is the braking/throttle ratio for the front wheels and β_r for the rear wheels. We enable both braking and throttle on the wheels in simulation.

Robustness in the presence of disturbance

In this section we show the effectiveness of the controller under large disturbances. The approach in this chapter does not use information on the disturbance distribution. It only needs the knowledge of the bounded support in order to enforce robust constraint satisfaction for all admissible disturbances. In this section we use a random disturbance with a uniform distribution over the bound $|w|_{max} = [0.2, 0.2, 0.2, 0.005, 0.05, 0.05]^T$ and add it in the simulation model at each sampling time. The result is compared to a nominal MPC with the same parameters but without robustness guarantees, i.e. $u = \bar{u}$ instead of $u = \bar{u} + Ke$. 50 trials are run for each controller with random realizations of disturbances. Figure 5.7 shows the performance of the nominal MPC. The trajectories are scattered over a large area and four trials fail to avoid the obstacle. In Figure 5.8 the robust MPC is implemented under the same realizations of disturbances. One can notice that the range of the closed-loop trajectories is much tighter and all of the trials successfully avoid the obstacle.

Performance with uncertain friction coefficients

The robust MPC formulation can incorporate disturbances from various sources, as long as they can be formulated as a bounded additive disturbance in Equation (5.28). In this section we simulate the scenario where the tire road friction coefficient is uncertain and changing constantly. This simulates the typical situation on a snow-ice track, where the surface condition changes constantly along the track. The friction coefficient μ is set to a random value from a uniform distribution over $[0.3, 0.7]$ and changes every 0.1 seconds. The controller is only using the nominal friction coefficient $\bar{\mu} = 0.5$ with bounds $|\Delta\mu|_{max} = 0.2$.

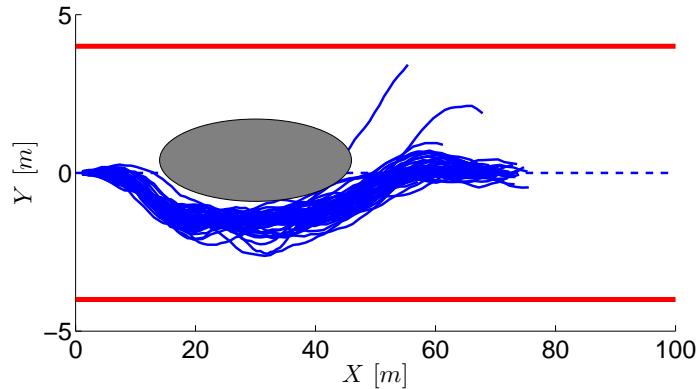


Figure 5.7: Simulation result: Trajectories of nominal MPC controlled system under a random external disturbance.

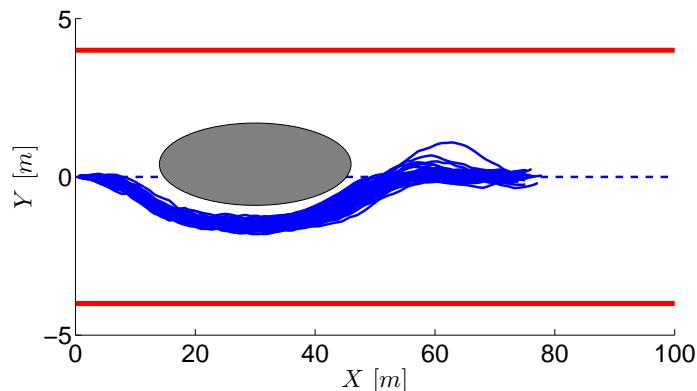


Figure 5.8: Simulation result: Trajectories of RNMPC controlled system under a random external disturbance.

Figure 5.9 shows 20 trials of such simulation. In all trials the vehicle successfully avoids the obstacles and remain on the road.

5.8 Experimental results

The experiments have been performed at a test center equipped with icy and snowy handling tracks. The MPC controller has been tested on a passenger car, with a mass of 2050 Kg and a yaw inertia of 3344 Kg/m². The controllers were run in a dSPACE® Autobox® system, equipped with a DS1005 processor board and a DS2210 I/O board.

We used an Oxford Technical Solution® (OTS) RT3002 sensing system to measure the vehicle position and orientation in the inertial frame and the vehicle velocities in the vehicle

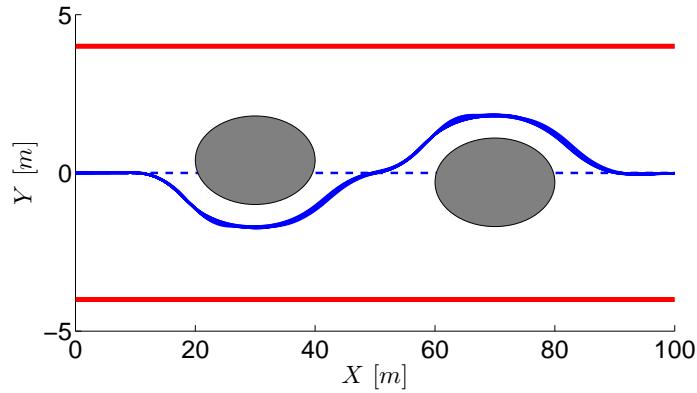


Figure 5.9: Simulation result: Trajectories of RNMPC controlled system under unknown and constantly changing tire road friction coefficients.

body frame. The OTS RT3002 is housed in a small package that contains a differential GPS receiver, Inertial Measurement Unit (IMU) and a DSP. The IMU includes three accelerometers and three angular rate sensors. The DSP receives both the measurements from the IMU and the GPS, utilizes a Kalman filter for sensor fusion, and calculates the position, orientation and other states of the vehicle.

The car was equipped with an Active Front Steering (AFS) and Differential Braking system which utilizes an electric drive motor to change the relation between the hand steering wheel and road wheel angles. This is done independently from the hand wheel position, thus the front road wheel angle is obtained by summing the driver hand wheel position and the actuator angular movement. The sensor, the dSPACE® Autobox®, and the actuators communicate through a CAN bus. Figure 5.10 shows the test vehicle on the test track.

The parameters used in the experiments are summarized in Table 5.2.

Table 5.2: Real-Time Design Parameters for Experimental Test

Param.	Value	Units	Param.	Value	Units
u_{max}	[0, 0.5, 0]	[−]	H_p	12	-
u_{min}	−[0.5, 0.5, 0]	[−]	H_c	8	-
Δu_{max}	[20, 20, 20]	[−]/sec	H_i	2	-
Δu_{min}	−[20, 20, 20]	[−]/sec	T_s	100	ms
Q	(1, 20, 5, 10)	-	$e_{y_{max}}$	4.5	m
R	(1, 1, 1)	-	$e_{y_{min}}$	−4.5	m
S	(1, 1, 1)	-			

Note the controller does not have access to the throttle of the test vehicle, the upper limit for inputs β_{xf} and β_r are set to zero. During the test, the driver uses throttle to keep



Figure 5.10: The test vehicle.

the vehicle at high speed.

Avoiding still and moving obstacle at high speed

In this section we show the controller's ability to control the vehicle to avoid an obstacle on the snow track at high speeds. Figure 5.11 show the trajectory of the vehicle, where the entering speed is 50kph. The dot-dashed lines denotes the bounds of the invariant set. The vehicle successfully avoids the obstacle and returns back to the lane center. The actual vehicle trajectory shown in blue closely follows the nominal trajectory shown in dashed black. Figure 5.12 shows the tracked states together with the corresponding tube bounds projected from the invariant set. The actual states are observed to be inside the tube all the time. Figure 5.13 shows the input from the controller in the same trial. The steering input is saturated between the time 20 an 24 seconds. The small oscillations in the steering command during saturation are due to the augmentation input \bar{u} . We also observe a brief braking at the beginning of the maneuver to decrease the vehicle speed.

Figure 5.14 shows a trial where the vehicle enters at 80 kph, and still avoids the obstacle. It takes longer for the vehicle to return to the center due to the high speed. Figure 5.15 shows all tracked states and their projected robust bounds. We see that all the actual states stay inside the invariant set. The inputs from the controller are reported in Figure 5.16. Compared to the inputs at lower speed in the previous test (Figure 5.13), smaller steering inputs are used to avoid tire slipping at the high speed.

Figure 5.17 to 5.19 show a trail where the test vehicle avoids a moving obstacle ahead. The test vehicle enters at a speed around 50 Kph while the obstacle moves at a speed of 18 Kph in X direction, started around 30 meters ahead of the test vehicle. Three screen shots of the obstacle position during the test are shown in Figure 5.17, along with the test vehicle position a the corresponding time. There is an aborted attempt of returning to the lane-center occurred around $X = 70m$, where the test vehicle turns left for a brief time before it quickly realizes that the obstacle is not clear yet. This is due to the controller's limited

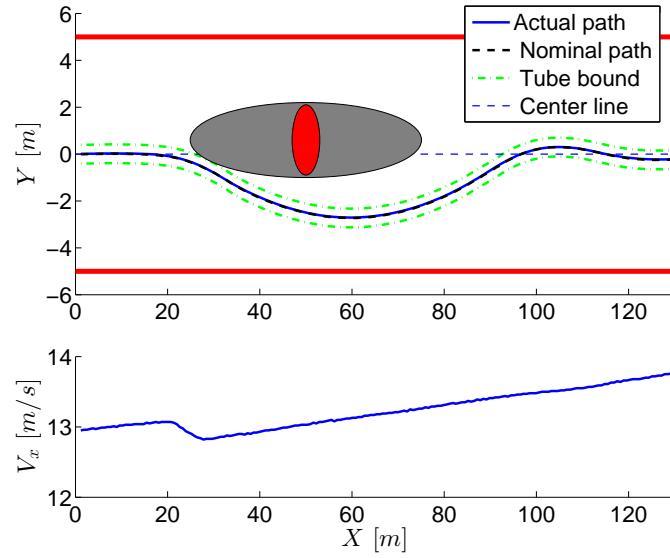


Figure 5.11: Experimental result: The vehicle enters the maneuver at 50Kph. The dashed black line and blue line are the nominal and actual vehicle trajectories respectively. The green dot-dashed lines indicate the robust bounds around the nominal trajectory. The actual vehicle path is seen to be very close to the nominal one and within the robust bounds. $\mu \simeq 0.3$.

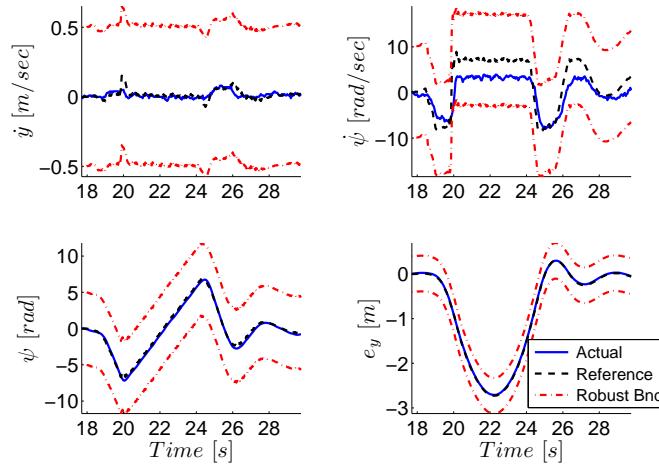


Figure 5.12: Experimental result: A plot of 4 states of the vehicle, $[y, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.11. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.

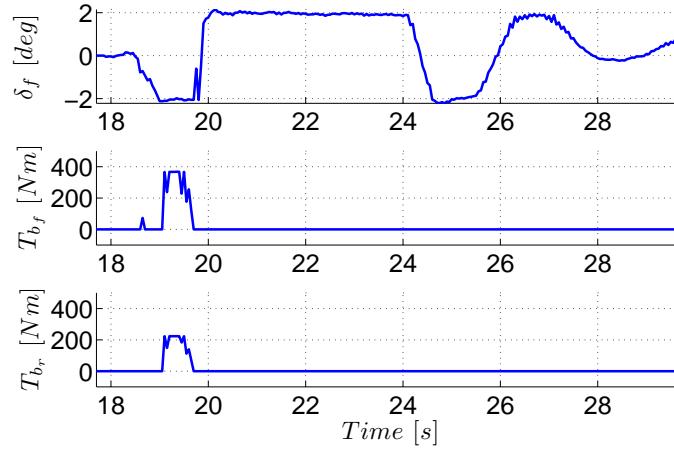


Figure 5.13: Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.11. The braking is applied by the controller at the beginning of the maneuver to reduce the speed so that the avoiding maneuver can be more smooth.

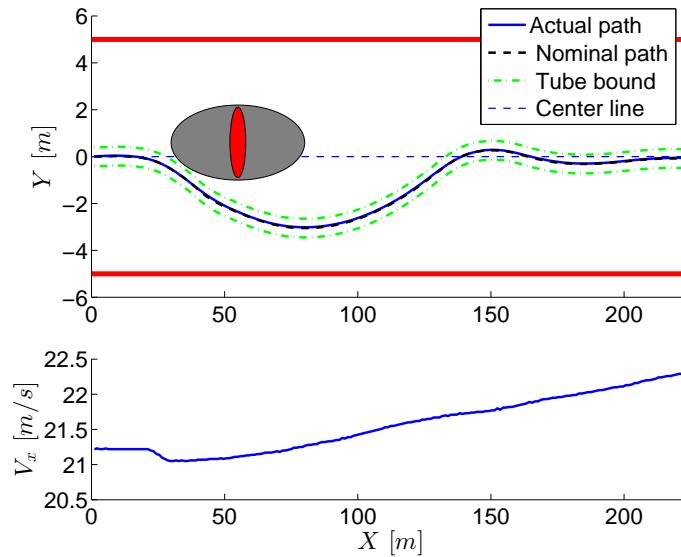


Figure 5.14: Experimental result: The vehicle enters the maneuver at 80Kph. The dashed black line and blue line are the nominal and actual vehicle trajectories respectively. The green dot-dashed lines indicate the robust bounds around the nominal trajectory. The actual vehicle path is seen to be very close to the nominal one and within the robust bounds. $\mu \approx 0.3$.

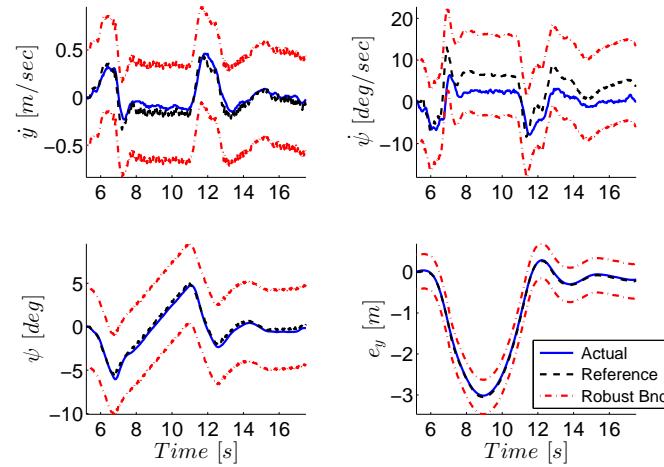


Figure 5.15: Experimental result: A plot of 4 states of the vehicle, $[y, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.14. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.

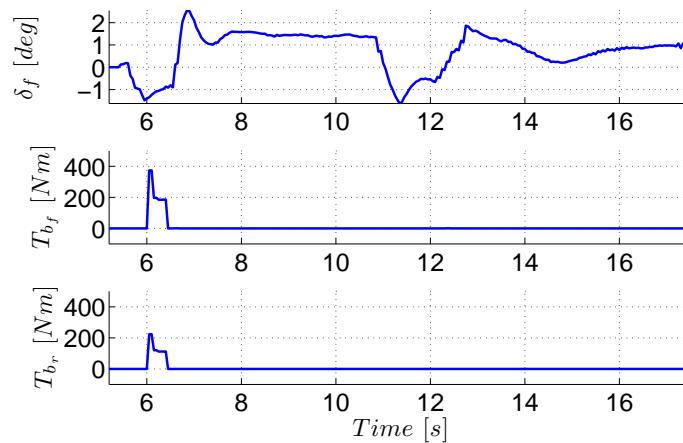


Figure 5.16: Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.14.

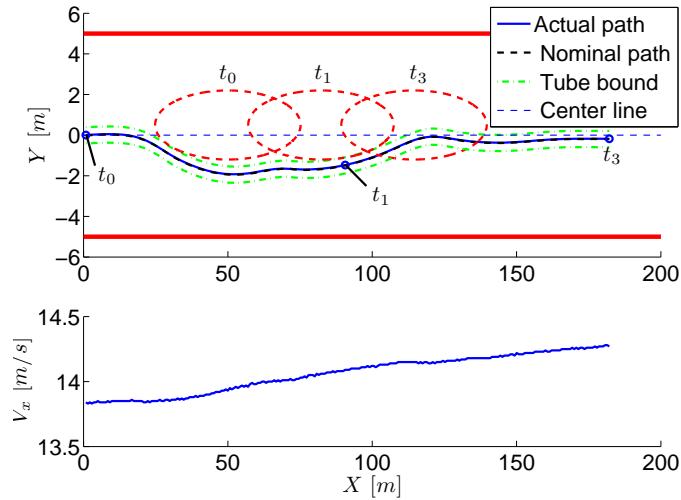


Figure 5.17: Experimental result: Avoiding a moving obstacle. Vehicle speed is 50 Kph, obstacle moves at 18 Kph. The obstacle position at time t_0 , t_1 and t_3 during the test are shown in dash red. The vehicle position at the corresponding times are marked with blue circles. $\mu \simeq 0.3$.

prediction horizon. Figure 5.18 and 5.19 show the states and inputs during that same trail. There was no braking applied by the controller in this trail.

Avoiding multiple obstacles

In this section the controller controls the vehicle to avoid two obstacles in line on the track. Figure 5.20 and 5.22 show the trajectory and input of the vehicle. The vehicle enters the maneuver at 50 kph, and it successfully avoids both obstacles and stays inside the road.

Robust performance against friction coefficient

In this section we test the vehicle on an ice track with a tire-road friction coefficient around 0.1. The controller is set up for a nominal μ of 0.3, as it is on a snow track. The vehicle is able to avoid the obstacle and track the center line at a speed of 35 kph. Figure 5.23 shows the trajectory of the vehicle during the maneuver and Figure 5.25 shows the inputs.

Computational time of the controller

One advantage of the proposed controller is its low computational burden compared to other robust approaches we have tried in our lab. Since the invariant set was computed off-line, the on-line computational burden of the robust controller is almost the same as the nominal

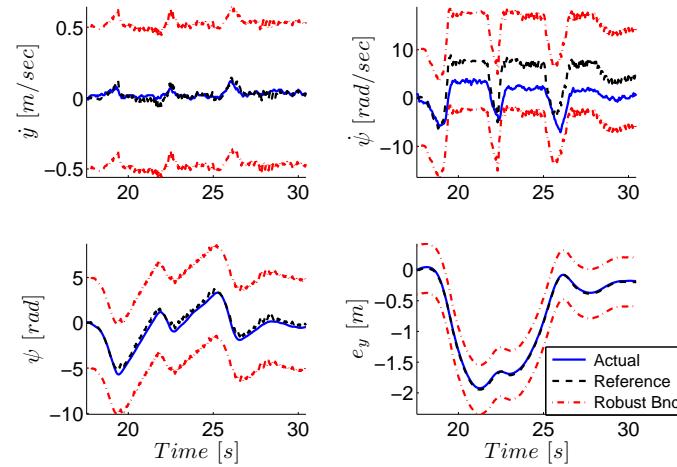


Figure 5.18: Experimental result: A plot of 4 states of the vehicle, $[y, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.19. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.

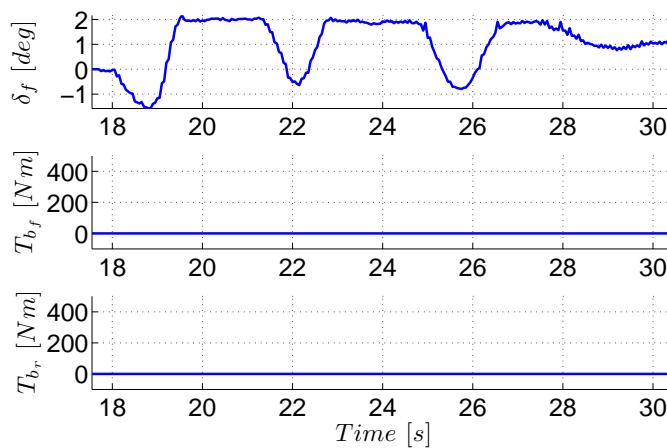


Figure 5.19: Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.19.

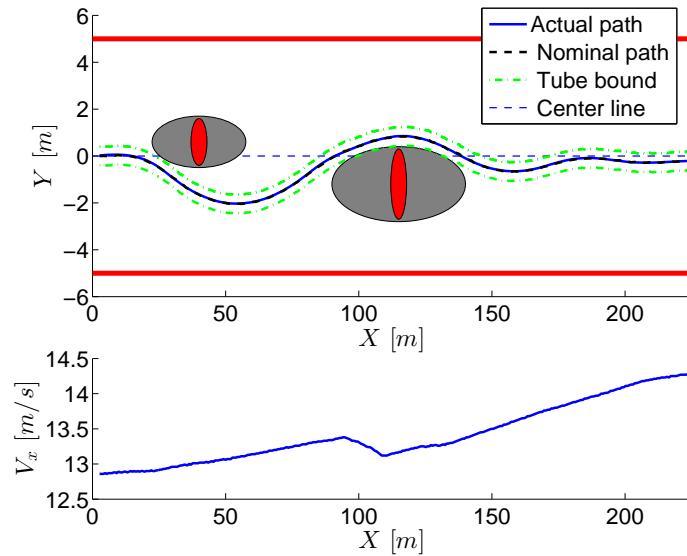


Figure 5.20: Experimental result: Vehicle trajectory. The vehicle enters the maneuver at 50Kph. $\mu \simeq 0.3$.

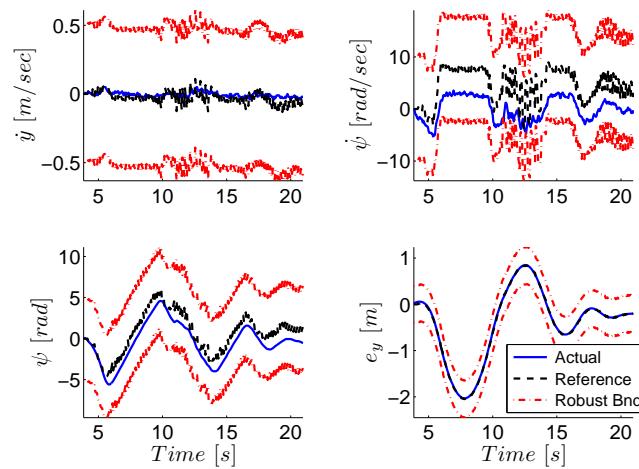


Figure 5.21: Experimental resultA plot of 4 states of the vehicle, $[y, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.20. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.

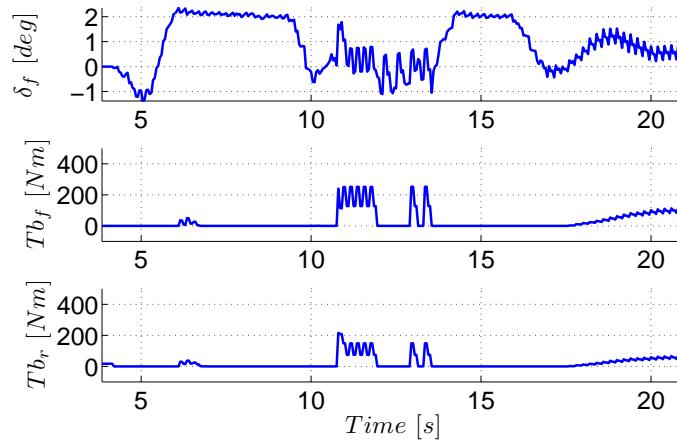


Figure 5.22: Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.20.

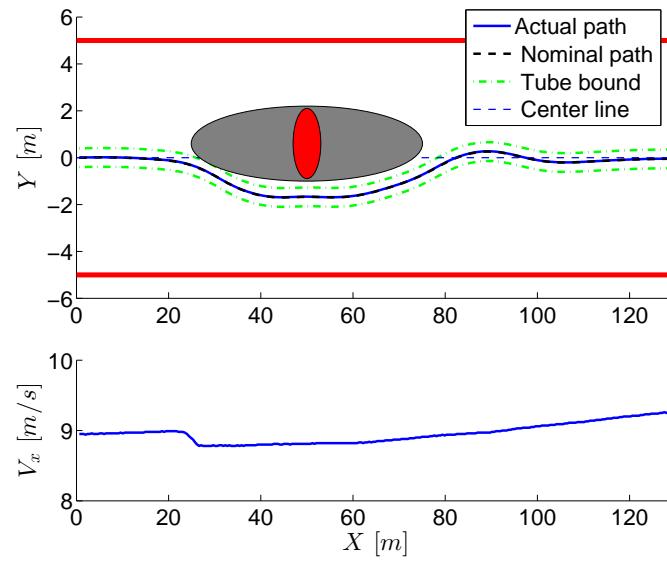


Figure 5.23: Experimental result: Vehicle trajectory on an ice track. The vehicle enters the maneuver at 35Kph. The actual μ on the track is 0.1, while the controller is set up for $\bar{\mu} = 0.3$ on snow track.

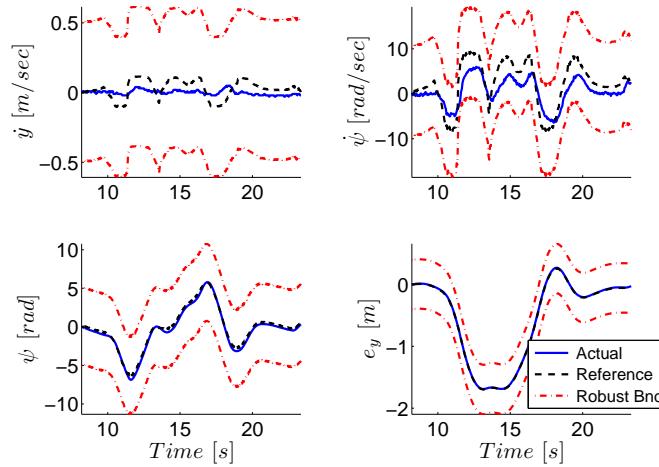


Figure 5.24: Experimental result: A plot of 4 states of the vehicle, $[y, \dot{\psi}, e_\psi, e_y]$, during the experiment of Figure 5.23. The nominal and actual states are shown in dashed black and solid blue lines respectively. The red dot-dashed lines indicate the robust bounds on each state.

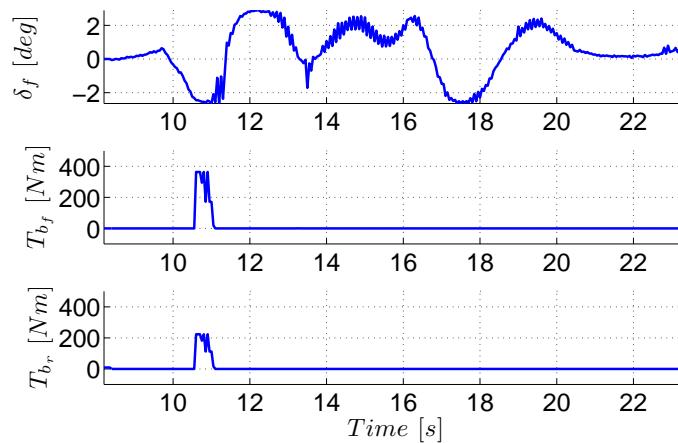


Figure 5.25: Experimental result: The steering and braking input to the vehicle during the experiment of Figure 5.23.

MPC. For the four tests reported previously, the average processing time was 59.6 ms and the maximum was 71.6 ms, both below the sampling time of 100ms.

5.9 Conclusions

This chapter presents a robust control framework for lane keeping and obstacle avoidance. The framework formulates the problem as a nonlinear model predictive control problem. A force-input nonlinear bicycle model is developed and used in the control design. A robust positively invariant set is computed for a given control law and a Robust NMPC is used with tightened input and state constraints to ensure constraint satisfaction in the presence of unknown disturbances. Multiple simulations with different scenarios have been run where the vehicle approaches multiple obstacles. The results show the effectiveness of the controller in the presence of uncertainty. The framework has been implemented on a test vehicle and the successful experimental results show the utility of this approach.

Bibliography

- [1] E. Bakker, L. Nyborg, and H. B. Pacejka. “Tyre modeling for use in vehicle dynamics studies”. In: *SAE paper # 870421* (1987).
- [2] Maximilian Balandat. “Constrained Robust Optimal Trajectory Tracking: Model Predictive Control Approaches”. MA thesis. Technische Universität Darmstadt, 2010.
- [3] R. Behringer et al. “Development of an autonomous vehicle for the DARPA Grand Challenge”. In: *IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon* (2004).
- [4] M Bertozzi, A Broggi, and A Fascioli. “Vision-based intelligent vehicles: State of the art and perspectives”. In: *Robot. Auton. Syst.* 32.1 (July 2000), pp. 1–16. ISSN: 09218890. DOI: 10.1016/S0921-8890(99)00125-6.
- [5] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*. Springer-Verlag, 2003. ISBN: 978-3-540-00257-4.
- [6] F. Borrelli et al. “An MPC/hybrid system approach to traction control”. In: *Control Systems Technology, IEEE Transactions on* 14, no. 3 (2006): 541–552 (2006).
- [7] F. Borrelli et al. “An MPC/Hybrid System Approach to Traction Control”. In: *IEEE Trans. Control Systems Technology* 14.3 (2006), pp. 541–552.
- [8] F. Borrelli et al. “Hybrid Decentralized Control of Large Scale Systems”. In: *Hybrid Systems: Computation and Control*. Lecture Notes in Computer Science. Springer Verlag, Mar. 2005.
- [9] F. Borrelli et al. “MPC-Based Approach to Active Steering for Autonomous Vehicle Systems”. In: *Int. J. Vehicle Autonomous Systems* 3.2/3/4 (2005), pp. 265–291.
- [10] F. Brugger, B. Manfred, and F. Andreas. “Anti-lock brake system”. In: *U.S. Patent 4,861,118* (1989).
- [11] E. Coelingh, A. Eidehall, and M. Bengtsson. “Collision Warning with Full Auto Brake and Pedestrian Detection C a practical example of Automatic Emergency Braking”. In: *The 13th International IEEE Annual conference on Intelligent Transportation Systems* (2010), pp. 155–160.
- [12] C. R. Cutler and B. L. Ramaker. “Dynamic matrix control-a computer control algorithm”. In: *Proceedings of the joint automatic control conference. Piscataway, NJ: American Automatic Control Council* 1 (1980), Wp5–B.

- [13] P Falcone. “Nonlinear Model Predictive Control for Autonomous Vehicles”. PhD thesis. Piazza Roma 21, 82100, Benevento, Italy, 2007.
- [14] P. Falcone et al. “A Model Predictive Control Approach for Combined Braking and Steering in Autonomous Vehicles”. In: *15th Mediterranean Conference on Control and Automation* (2007).
- [15] P. Falcone et al. “A Real-Time Model Predictive Control Approach for Autonomous Active Steering”. In: *Nonlinear Model Predictive Control for Fast Systems, Grenoble, France*. 2006.
- [16] P Falcone et al. “Integrated Braking and Steering Model Predictive Control Approach in Autonomous Vehicles”. In: *Fifth IFAC Symposium on Advances of Automotive Control* (2007).
- [17] P. Falcone et al. “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation”. In: *International Journal of Robust and Nonlinear Control* 18 (2007), pp. 862–875.
- [18] P. Falcone et al. “Linear Time Varying Model Predictive Control Approach to the Integrated Vehicle Dynamics Control Problem in Autonomous Systems”. In: *IEEE Conference on Decision and Control* 46 (2007).
- [19] P. Falcone et al. “Low Complexity MPC Schemes for Integrated Vehicle Dynamics Control Problems”. In: *9th International Symposium on Advanced Vehicle Control* (2008).
- [20] P Falcone et al. “Predictive Active Steering Control for Autonomous Vehicle Systems”. In: *IEEE Transactions on Control Systems Technology* 15.3 (2007), pp. 566–580.
- [21] P. Falcone et al. “Predictive Autonomous Vehicles: A Linear Time Varying Model Predictive Control Approach”. In: *46th Conference on Decision and Control* (2007).
- [22] H. J. Ferrau, H. G. Bock, and Moritz Diehl. “An Online Active Set Strategy for Fast parametric Quadratic Programming in MPC Applications”. In: *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, plenary talk* (2006).
- [23] Y. Gao et al. “Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads”. In: *Dynamic Systems and Control Conference* (2010).
- [24] Y. Gao et al. “Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles”. In: *11th International Symposium on Advanced Vechile Control*, (2012).
- [25] P. Gill et al. NPSOL – Nonlinear Programming Software. Stanford Business Software, Inc., Mountain View, CA, 1998.
- [26] A. Gray et al. “Predictive Control for Agile Semi-Autonomous Ground Vehicles using Motion Primitives”. In: *American Control Conference* (2012).
- [27] A. Gray et al. “Predictive Control for Agile Semi-Autonomous Ground Vehicles Using Motion Primitves”. In: *American Contr. Conf.* (2012).

- [28] Rami Y. Hindiyeh and J. Christian Gerdes. "Equilibrium Analysis of Drifting Vehicles for Control Design". In: *Dynamic Systems and Controls Conference* (2009).
- [29] *IEEE Spectrum Website*. URL: <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/googles-autonomous-car-takes-to-the-streets> (visited on 05/01/2014).
- [30] F. Kehrle et al. "Optimal control of Formula 1 race cars in a VDrift based virtual environment". In: *IFAC World Congress Milan*. 2011.
- [31] T. Keviczky and G. J. Balas. "Flight Test of a Receding Horizon Controller for Autonomous UAV Guidance". In: *Proc. American Contr. Conf.* 2005.
- [32] U Kiencke and L Nielsen. *Automotive Control Systems*. Springer, 2005.
- [33] S. G. Klauer et al. "The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data". In: *U.S. NHTSA Technical Report, DOT HS 810 594* (2006).
- [34] K. Kritayakirana and J. C. Gerdes. "Using the center of percussion to design a steering controller for an autonomous race car". In: *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility* (2012).
- [35] M Kvasnica, P Grieder, and M Baotic. *Multi-Parametric Toolbox (MPT)*. 2004. URL: <http://control.ee.ethz.ch/~mpt/>.
- [36] W. Langson et al. "Robust model predictive control using tubes". In: *Automatica* 40(1) (2004), pp. 125–133.
- [37] Raph Levien. "The euler spiral: a mathematical history". In: *Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report No. UCB/EECS-2008-111* (2008).
- [38] J. Levinson et al. "Towards Fully Autonomous Driving: Systems and Algorithms". In: *IEEE Intelligent Vehicles Symposium* (2011), pp. 163–168.
- [39] E. K. Liebemann et al. "Safety and Performance Enhancement: The Bosch Electronic Stability Control (ESP)". In: *SAE Paper 20004 (2004): 21-0060*. (2004).
- [40] L. Helgesson M. Christiansson M. Brannstrom J. Sjoberg. "A Real-time Implementation of an Intersection Collision Avoidance System". In: *The 18th World Congress of the International Federation of Automatic Control* (2011), pp. 9794–9798.
- [41] D. Margolis and J. Asgari. "Multipurpose Models of Vehicle Dynamics for Controller Design". In: *SAE Technical Paper 911927, doi:10.4271/911927*. (1991).
- [42] R. Matsumoto. "Vehicle traction control system for preventing vehicle turnover on curves and turns". In: *U.S. Patent 4,976,330* (1990).
- [43] D. Q. Mayne, M. M. Seron, and S. V. Rakovic. "Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances". In: *Automatica* 41(2) (2005), pp. 219–224.

- [44] D. Q. Mayne et al. “Robust output feedback model predictive control of constrained linear systems”. In: *Automatica* 42 (2006), pp. 1217–1222.
- [45] D. Q. Mayne et al. “Tube-based robust nonlinear model predictive control”. In: *International Journal of Robust and Nonlinear Control* 21 (2011), pp. 1341–1353.
- [46] D.Q. Mayne et al. “Constrained model predictive control: Stability and Optimality”. In: *Automatica* 36.6 (June 2000), pp. 789–814.
- [47] W. F. Milliken and D. L. Milliken. *Chassis Design: Principles and Analysis*. Society of Automotive Engineers, 2002.
- [48] M. Montemerlo et al. “Stanley, the robot that won the darpa grand challenge”. In: *Journal of Robotic Systems* 23, no. 9. Special Issue on the DARPA Grand Challenge, Part 2 (2006), pp. 661–692.
- [49] M. Morari and J.H. Lee. “Model predictive control: past, present and future”. In: *Computers & Chemical Engineering* 23.4–5 (1999), pp. 667–682.
- [50] NHTSA Data Resource Website. URL: <http://www-fars.nhtsa.dot.gov/Main/index.aspx> (visited on 05/01/2014).
- [51] H. B. Pacejka. *Tyre and Vehicle Dynamics*. Elsevier Science, 2005. ISBN: 9780080543338.
- [52] S. J. Qin and T. A. Badgwell. “A survey of industrial model predictive control technology”. In: *Control engineering practice* 11, no. 7 (2003), pp. 733–764.
- [53] Rajesh Rajamani. “Vehicle Dynamics and Control, 1st ed.” In: *Springer US* (2005).
- [54] S. V. Rakovic et al. “Invariant Approximations of the Minimal Robust Positively Invariant Set”. In: *IEEE Transactions on Automatic Control* 50 (2005), pp. 406–410.
- [55] S. V. Rakovic et al. “Robust Obstacle Avoidance for Constrained Linear Discrete Time Systems: A SetCTheoretic Approach”. In: *46th IEEE Conference on Decision and Control New Orleans, USA* (2007).
- [56] A. Scheuer and T. Fraichard. “Collision-free and continuous-curvature path planning for car-like robots”. In: *IEEE International Conference on Robotics and Automation* (1997).
- [57] D. H. Shin and S. Singh. “Path generation for robot vehicles using composite clothoid segments”. In: *Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-90-31* (1990).
- [58] J. C. Stutts and W. W. Hunter. “Driver Inattention, Driver Distraction and Traffic Crashes”. In: *Institute of Transportation Engineers*, 73(7):34C45 (2003).
- [59] Inc. The MathWorks. Model Predictive Control Toolbox. 2005.
- [60] E. H. Tseng et al. “The development of vehicle stability control at Ford”. In: *IEEE/ASME Transactions on Mechatronics* 4, no. 3 (1999), pp. 223–234.

- [61] D. Verscheure et al. “Time-Optimal Path Tracking for Robots: A Convex Optimization Approach”. In: *Automatic Control, IEEE Transactions on* 54.10 (2009), pp. 2318 – 2327. ISSN: 0018-9286. DOI: 10.1109/TAC.2009.2028959.
- [62] Y. Yoon et al. “Model-predictive active steering and obstacle avoidance for autonomous ground vehicles”. In: *Control Engineering Practice* 17.7 (2009), pp. 741–750.
- [63] S. Yu, H. Chen, and F. Allgower. “Tube MPC Scheme based on Robust Control Invariant Set with Application to Lipschitz Nonlinear Systems”. In: *50th IEEE Conference on Decision and Control and European Control Conference* (2011).
- [64] V. M. Zavala, C. D. Laird, and L. T. Biegler. “Fast Solvers and Rigorous Models: Can Both Be Accommodated in NMPC”. In: *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, plenary talk* (2006).
- [65] M. N. Zeilinger et al. “Real-time MPC C Stability through Robust MPC design”. In: *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference* (2009).