UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

An Integrated Framework for Planning and Control of Semi-Autonomous Vehicles

Permalink

https://escholarship.org/uc/item/1s2777sr

Author

Gray, Andrew Jacob

Publication Date

2013

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

An Integrated Framework for Planning and Control of Semi-Autonomous Vehicles

By

Andrew Jacob Gray

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

Engineering - Mechanical Engineering

Graduate Division

University of California, Berkeley

Committee in charge:

Professor J. Karl Hedrick, Co-Chair
Professor Francesco Borrelli, Co-Chair
Professor Pieter Abbeel

Spring 2013

Abstract

An Integrated Framework for Planning and
Control of Semi-Autonomous Vehicles
by
Andrew Jacob Gray
Doctor of Philosophy in Engineering - Mechanical Engineering
University of California, Berkeley
Professor J. Karl Hedrick, Co-Chair
Professor Francesco Borrelli, Co-Chair

This thesis presents the design of a novel active safety system preventing unintended roadway departures. The proposed framework unifies threat assessment, stability, and control of passenger vehicles into a single combined optimization problem. A nonlinear Model Predictive Control (NMPC) problem is formulated where the nonlinear vehicle dynamics, in closed- loop with a driver model, is used to optimize the steering and braking actions needed to keep the driver safe. A model of the drivers nominal behavior is estimated based on his observed behavior. The driver commands the vehicle while the safety system corrects the drivers steering and braking action in case theres a risk that the vehicle will unintentionally depart the road. The resulting predictive controller is always active and mode switching is not necessary. We show simulation results detailing the behavior of the proposed controller as well as experimental results obtained by implementing the proposed framework on embedded hardware in a passenger vehicle. The results demonstrate the capability of the proposed controller to detect and avoid roadway departures while avoiding unnecessary interventions.

List of Publications

[1] **A. Gray**, Y. Gao, T. Lin, J.K. Hedrick, E. Tseng, and F. Borrelli *Predictive Control for Agile Semi-Autonomous Ground Vehicles using Motion Primitives.* American Control Conference, Montreal, Canada, June, 2012

[2] Gao Y., **A. Gray**., J. Frasch, T. Lin, E. Tseng, J.K. Hedrick, and F. Borrelli *Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles.* International Symposium on Advanced Vehicle Control, Seoul, Korea, September, 2012.

[3] **A. Gray**., M. Ali, Y. Gao, J.K. Hedrick, and F. Borrelli *Integrated Threat Assessment and Control Design for Roadway Departure Avoidance.* Intelligent Transportation Systems, Anchorage, September, 2012.

[4] **A. Gray**., M. Ali, Y. Gao, J.K. Hedrick, and F. Borrelli *Semi-Autonomous Vehicle Control for Road Departure and Obstacle Avoidance.* IFAC Symposium on Control in Transportation Systems, Sofia, Bulgaria, September, 2012.

[5] J.V. Frasch, **A. Gray**, M. Zanon, H.J. Ferreau, S. Sager, F. Borrelli, and M. Diehl *An Auto-generated Nonlinear MPC Algorithm for Real-Time Obstacle Avoidance of Ground Vehicles.* European Controls Conference, Zurich, Switzerland, July, 2013.

[6] **A. Gray**., M. Ali, Y. Gao, J.K. Hedrick, and F. Borrelli *A Unified Framework for Semi-autonomous Control and Active Safety.* Transactions in Intelligent Transportation Systems, 2013.

[7] A. Hackbarth, **A. Gray** and E. Kreuzer *Multi-Agent Motion Control of Autonomous Vehicles in 3D Flow Fields.* Proceedings in Applied Mathematics and Mechanics, Hamburg, Germany, June, 2012.

[8] A. Hackbarth, **A. Gray**, E. Kreuzer and J.K. Hedrick *Collaborative Control of Multiple AUVs for Improving the Estimation of Flow Field Dependent Variables.* IEEE Underwater Vehicles, South Hampton, England, September, 2012.

[9] **A. Gray**., Y. Gao, J.K. Hedrick, and F. Borrelli *Robust Predictive Control for Semi-Autonomous Vehicles with an Uncertain Driver Model.* IEEE Intelligent Vehicles Symposium, Gold Coast, Australia, June, 2013.

[10] **A. Gray**., M. Ali, Y. Ga, J.K. Hedrick, and F. Borrelli *Multi-Objective Collision Avoidance.* Dynamic Systems and Controls Conference, Stanford, October, 2013.

[11] **A. Gray**., Y. Gao, J.K. Hedrick, and F. Borrelli *Stochastic Predictive Control for Semi-Autonomous Vehicles with an Uncertain Driver Model.* Intelligent Transportation Systems Conference, The Hague, Netherlands, 2013. Submitted.

[12] Y. Gao, **A. Gray**., J.K. Hedrick, and F. Borrelli *Robust Predictive Control of Semi-autonomous Ground Vehicle.* Transactions in Intelligent Transportation Systems, 2013. Submitted.

Acknowledgements

I would like to thank my research advisors J. Karl Hedrick and Francesco Borrelli as well as Pieter Abbeel for providing invaluable guidance and support while conducting the research presented in this thesis. Further I would like to acknowledge those I worked with in the Vehicle Dynamics and Control Lab. Namely, Brandon Basso, Jared Wood, Shih-Yuan Liu, Jared Garvey, Selina Pan and SangHyun Hong. I would also like to acknowledge those in the Model Predictive Control Lab; Yiqi Gao, Theresa Lin and Ashwin Carvalho. Further, this work received the invaluable support and influence from Mohammed Ali at Volvo Car. Thank you all.

Table of Contents

List of Publications **i**

Acknowledgments **iii**

1 Introduction **1**

1.1 Contributions 3

1.1.1 Hierarchical Model Predictive Control 3

1.1.2 A Unified Framework for Active Safety 3

1.1.3 Robust Model Predictive Control for Uncertain Driver Models 4

2 Vehicle Dynamics **5**

2.1 Nonlinear Four Wheel Model 8

2.2 Tire Models 13

2.2.1 Pacjeka Tire Model 13

2.2.2 Fiala Tire Model 19

2.2.3 Linear Tire Force Model 21

2.3 Bicycle Model 23

2.4 Spatial Model 25

2.5 Linear Model 28

2.6 Safety Constraints 31

3 Driver Models **33**

3.1 Nominal Driver Model 35

3.1.1 Parameter Estimation 35

3.2 Attentive Driver Model 40
3.3 Uncertain Driver Model 41
3.4 Stochastic Driver Model 42
3.5 Driver-in-the-Loop Vehicle Model 43

4 Driver Assistance Systems 44

5 Model Predictive Control 51
5.1 Nonlinear Model Predictive Control 53
5.2 Linear Time Varying Model Predictive Control 55
5.3 Robust Model Predictive Control 58
5.3.1 Set-based Robust MPC 58
5.3.2 Stochastic Robust MPC 65

6 Hierarchical Model Predictive Control 71
6.1 Motion Primitive Path Planner 75
6.1.1 High-Level Motion Primitive Framework 75
6.1.2 Motion Primitives for Collision Avoidance 78
6.1.3 Planning With Motion Primitives 84
6.1.4 Low-Level MPC Path Follower 85
6.1.5 Simulation And Experimental Results 86
6.2 Spatial Model Path Planner 94
6.2.1 High-Level Spatial Model Framework 96
6.2.2 Low-Level MPC Path Follower 97
6.2.3 Simulation and Experimental Results 98

7 Integrated Safety Framework 106
7.1 Integrated Active Safety 109
7.2 Simulation Results 111
7.2.1 Unintentional Roadway Departure 111
7.2.2 Collision Avoidance 125
7.3 Experimental Results 133
7.3.1 Test 1: Excessive Speed in a Curve 134
7.3.2 Test 2: Unintentional Drifting 137

7.4 Set-based Robust Active Safety 140
7.4.1 Simulation Results 143
7.5 Stochastic Robust Active Safety 149
7.5.1 Simulation Results 150

A Experimental Vehicle Setup 154

Bibliography 158

List of Figures 169

List of Tables 178

Chapter 1

Introduction

Advances in sensing technologies have enabled the introduction and commercialization of several automated driving features over the last two decades. Examples of such applications are threat assessment Warning Strategies [12], Adaptive Cruise Control (ACC) [91], Rear-end Collision Avoidance systems [22], as well as Lane Keeping systems [77]. In safety applications, autonomous interventions are activated automatically. Over-activation of automated safety interventions might be felt as intrusive by the driver, while on the other hand, a missed or delayed intervention might lead to a collision.

A typical active safety system architecture is modular [3], with separate threat assessment, decision making, and intervention modules. In particular, the threat assessment module deals with the task of determining whether interventions are necessary and plays an important role in the interaction with the driver. The threat assessment module repeatedly evaluates the driver's ability in maintaining safety in each situation and this information is used by the decision making module in order to decide whether and how to assist the driver. It is a challenge for an active safety system to properly assess when to intervene. In the literature, a large variety of threat assessment and decision making approaches can be found [22, 78, 62, 29]. In the simplest approaches, used in production vehicles, automated steering or braking interventions are issued when simple measures like the *time to collision* [22] or *time to line crossing* [78] pass certain thresholds.

More sophisticated approaches, on the other hand, include the computation of Bayesian collision probabilities [62] or sets of safe states from which the vehicle can safely evolve [29]. In advanced safety systems, such as roadway departure prevention, the intervention module has the goal to both determine a safe trajectory and coordinate the vehicle actuators. The literature on vehicle path planning and control is rich, see, e.g. [32, 72, 104, 48]. Because of its capability to systematically handle system nonlinearities and constraints, work in a wide operating region and close to the set of admissible states and inputs, Model Predictive Control (MPC) has been shown to be an attractive method for solving the path planning and control problem [32, 33]. Previous approaches to lane departure prevention using predictive control, as in [4], make the assumption that the vehicle is traveling at a constant velocity (and can therefore not consider braking), and does not use any information about the human driver.

In this thesis, we design a novel active safety system for prevention of unintended roadway departures with a human-in-the-loop. Rather than separately solving the threat assessment, decision making, and intervention problems, we reformulate them as a single combined optimization problem. In particular, a predictive optimal control problem is formulated which simultaneously uses predicted driver's behavior and determines the least intrusive intervention that will keep the vehicle in a region of the state space where the driver is deemed safe. The proposed controller is always active, which avoids the design of switching logic or the tuning of a sliding scale. In addition, since the proposed controller is designed to only apply the correcting control action that is necessary to avoid violation of the safety constraints, the intrusiveness of the safety application is kept minimal. Furthermore, the full nonlinear dynamics of the vehicle are considered in the optimization problem and the corrective action can augment both the driver's steering *and* braking.

The thesis is organized as follows: in Chapter 2 we detail the vehicle models that are used for control. Chapter 3 introduces simplified driver models that are estimated in real-time, as well as presents some extensions to model uncertainty in the driver's future behavior. Chapter 5 introduces our Model Predictive Control approach as well as details how we handle the uncertainty in the uncertain driver models. Chapter 6 presents simulation and experimental results from an implementation of Hierarchical MPC on an experimental platform and Chapter 7 unifies everything into an Integrated Safety Framework and presents simulative and experimental results.

1.1 Contributions

The following describes the list of contributions made by the author. The bibliographic references refer to the List of Publications at the beginning of this thesis.

1.1.1 Hierarchical Model Predictive Control

Papers [1]-[2] describe an approach that decomposes the planning and control problems into an upper and lower level. In [1] the upper level solves the path planning problem using motion primitives. The authors are the first to investigate the feasibility of planning using motion primitives for automotive applications by formulating the problem and implementing the planning architecture on an experimental test vehicle. The details are described in §6.1. In [2] the planning problem is made solvable in real-time by utilizing a coordinate transformation where the contribution made is the implementation of such a transformation within the model predictive control framework and is discussed in §6.2. The work in [6] further reduces the computation time of the MPC problem by using tailored algorithms.

1.1.2 A Unified Framework for Active Safety

The main contribution in this thesis is on the development and implementation of a novel active safety system framework for semi-autonomous vehicles, presented in Chapter 7. In [3] and [4] the framework was proposed to unify the threat assessment, planning, and control problem into a single combined optimization problem that incorporates closed-loop human behavior prediction. In [6] the framework was implemented on a test vehicle and the successful results show the utility of the proposed controller. In [10] we extend the approach to include moving obstacles.

1.1.3 Robust Model Predictive Control for Uncertain Driver Models

The remaining contributions extend the unified framework to include robust guarantees on safety by explicitly modeling the uncertainty on the driver's behavior. In [9] we use set-based methods to capture the spread on all possible future trajectories and constrain the controller to satisfy the safety constraints for the whole invariant set, as outlined in §5.3.1. In [12] we extend this approach to the nonlinear vehicle model by reforming the dynamics and approximating the robust invariant set and implementing the controller on an experimental test vehicle. In [11] we propose an uncertain driver model by modeling the uncertainty using a probability distribution function and probabilistically satisfy the safety constraints. The approach is presented in §5.3.2

Chapter 2

Vehicle Dynamics Models

This chapter introduces the various vehicle dynamics models useful for control design. The models presented capture the relevant dynamics deemed important for our application of threat assessment, planning, and control, but are, in general, oversimplified as further dynamics are neglected. Vehicle dynamics has been well studied and we draw extensively from the literature [91, 68]. Consider the vehicle in Figure 2.1,

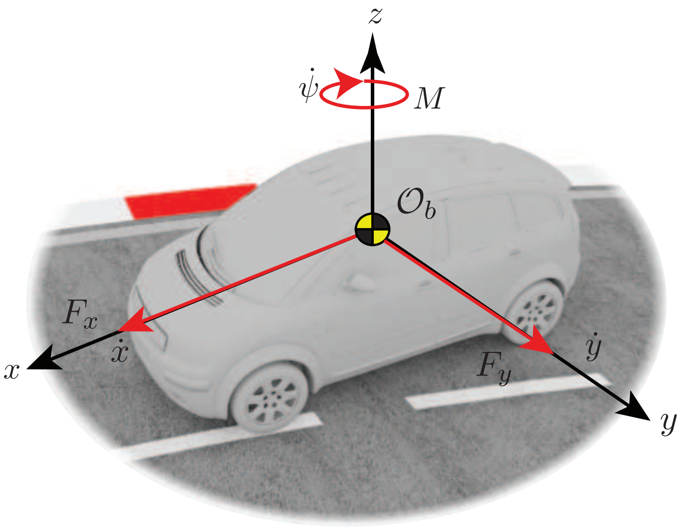


Figure 2.1: A vehicle sketch depicting the body-fixed coordinate frame, $\mathcal{O}_b$, as well as body-fixed longitudinal and lateral velocities, $\dot{x}$ and $\dot{y}$, forces F_x, F_y, and moment, M.

The translational body-fixed velocities are denoted as $\dot{x}$ and $\dot{y}$ for the longitudinal and lateral axes, respectively. The yaw angle is denoted as ψ and the yaw rate about the z-axis through the center of gravity is $\dot{\psi}$. The total longitudinal force acting on the vehicle is F_x, the total lateral force is F_y and the moment about the z-axis through the center of gravity is M. Newton's law is applied to the center of gravity (CoG) to obtain the general rigid-body dynamics. The following differential equations describe the longitudinal, lateral, and yaw motion,

$$m\ddot{x} = F_x, \tag{2.1a}$$

$$m\ddot{y} = F_y, \tag{2.1b}$$

$$I\ddot{\psi} = M, \tag{2.1c}$$

where m is the vehicle mass and I is the inertia about the z-axis. To model the planar motion of the vehicle a coordinate transformation is needed from the vehicle body-fixed frame $\mathcal{O}_b$ to the inertial frame $\mathcal{O}_I$. A simple rotation around the z-axis by the amount of the yaw angle ψ is used to calculate the velocities in the inertial frame,

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

$$\dot{Y} = \dot{y}\sin(\psi) - \dot{x}\cos(\psi), \tag{2.2b}$$

$$\dot{\Psi} = \dot{\psi}, \tag{2.2c}$$

where $\dot{X}$ and $\dot{Y}$ are the vehicle longitudinal and lateral velocities, respectively, in the inertial frame, and $\dot{\Psi}$ is the yaw rate.

The sections to follow will detail the calculations of the forces acting on the vehicle to arrive at mathematical models, of varying complexity, to describe the vehicle motion. The rest of the chapter is organized in the following way: in section 2.1 a Four Wheel Nonlinear Model is derived where the forces F_x, F_y, and M are computed as nonlinear functions of the vehicle state, steering, braking, and driving at the four wheels. In section 2.2 the Pacjeka and Fiala tire models are presented. A simplification from the Four Wheel Model to a Bicycle model is derived in section 2.3 where the reduced complexity is useful for real-time control. The Spatial Model presented in section 2.4 utilizes a coordinate transformation to convert space to the independent variable and is useful in modeling static obstacles as convex box constraints in the MPC chapter. Finally, a Linear Model is presented that will be utilized in the chapter on Robust MPC.

2.1 Nonlinear Four Wheel Model

In this section a useful model for control will be presented that models the vehicle as a rigid body and calculates the forces at each of the four wheels as nonlinear functions of the vehicle state and inputs. Consider the vehicle sketch in Figure 2.2.

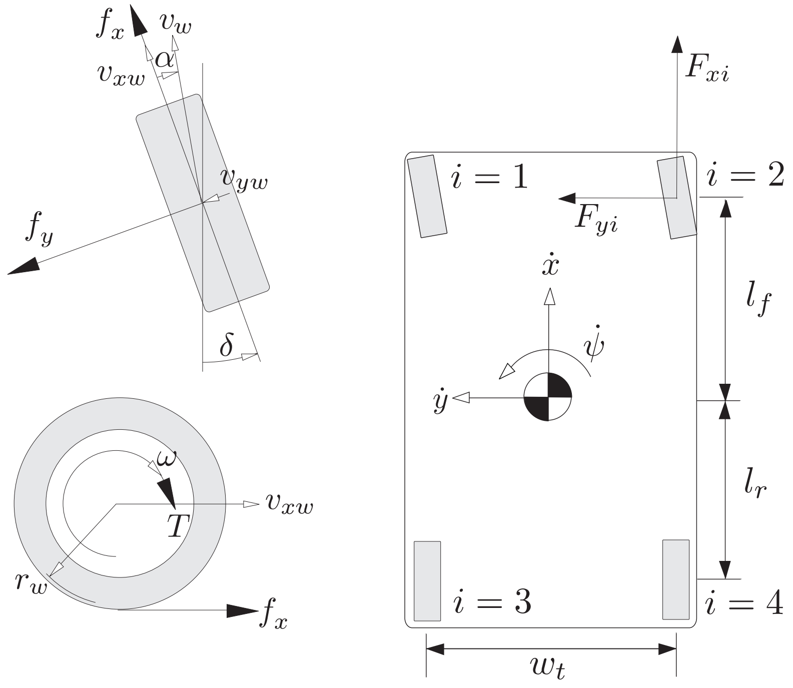


Figure 2.2: Modeling notation depicting the forces in the vehicle body-fixed frame, the forces in the tire-fixed frame, and the rotational and translational velocities. The relative coordinates e_y and e_ψ are illustrated on the sketch of the road as well as the driver model variable e_ψ^{lp} and the angle of the road tangent ψ_d.

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

$$m\ddot{x} = m\dot{y}\dot{\psi} + \sum_{i=1}^{4} F_{xi}, \tag{2.3a}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + \sum_{i=1}^{4} F_{yi}, \tag{2.3b}$$

$$I\ddot{\psi} = l_f(F_{y1} + F_{y2}) - l_r(F_{y3} + F_{y4}) + \tag{2.3c}$$

$$\frac{w_t}{2}(-F_{x1} + F_{x2} - F_{x3} + F_{x4}), \tag{2.3d}$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \tag{2.3e}$$

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

where m and I denote the vehicle mass and yaw inertia, respectively, l_f and l_r denote the distances from the vehicle center of gravity to the front and rear axles, respectively, and w_t denotes the track width. $\dot{x}$ 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 2.3 denote the vehicle orientation and lateral position, respectively, in a road aligned coordinate frame and ψ_{road} is the angle of the tangent to the road centerline in a fix coordinate frame.

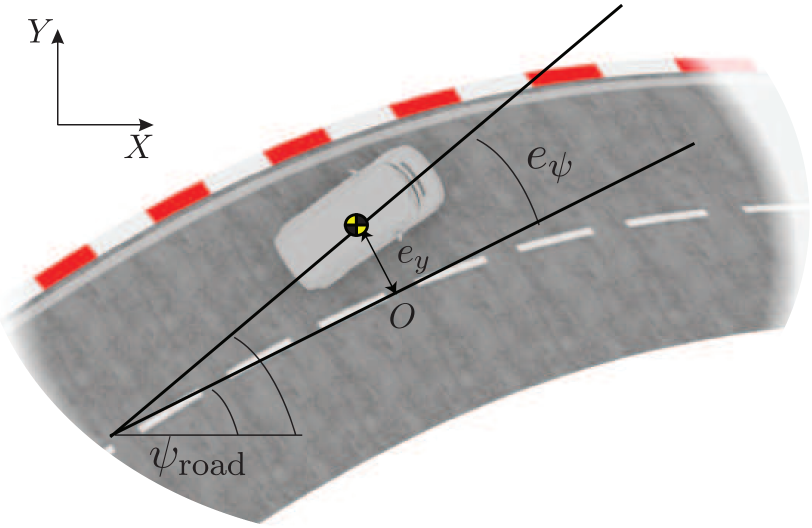


Figure 2.3: A road aligned coordinate frame illustrating the error states with respect to the road, e_y and e_ψ.

F_{yi} and F_{xi} are tire forces acting along the vehicle lateral and longitudinal axis, respectively, and f_{yi}, f_{xi} are forces acting along the tire lateral and longitudinal axis, respectively.

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

$$F_{xi} = f_{xi}\cos(\delta_i) - f_{yi}\sin(\delta_i), \tag{2.4a}$$

$$F_{yi} = f_{xi}\sin(\delta_i) + f_{yi}\cos(\delta_i), \qquad i \in \{1,2,3,4\}, \tag{2.4b}$$

where δ_i is the steering angle at wheel i. We introduce the following assumption on the steering angles.

Assumption 1 *Only the steering angles at the front wheels can be controlled and the steering angles at the right and left wheels of each axle are assumed to be the same, i.e., $\delta_1 = \delta_2 = \delta$ and $\delta_3 = \delta_4 = 0$. In addition, an actuator which corrects the driver commanded steering angle, such that $\delta = \delta_d + \delta_c$, is available, where δ_d is the driver commanded steering angle and δ_c is the correcting steering angle component.*

The longitudinal and lateral forces are, in general, complex functions of several parameters. In this thesis two such functions are detailed in the next section 2.2, the Magic Tire Formula, or the Pacjeka model [87], and the Fiala tire model [57]. A dependency could be written as

$$f_{xi} = f_l(\alpha_i,\, \sigma_i,\, \mu_i,\, F_{zi}), \tag{2.5a}$$

$$f_{yi} = f_c(\alpha_i,\, \sigma_i,\, \mu_i,\, F_{zi}), \qquad i \in \{1,2,3,4\}, \tag{2.5b}$$

where α_i is the slip angle at tire i, σ is the slip ratio, μ is the friction coefficient, and F_z is the normal force. These parameters will be defined next while the functions f_l and f_c are defined in section 2.2.

The slip angle α is defined as the angle between the velocity vector of the wheel and the orientation of the wheel itself, as shown in Figure 2.2. It is expressed as

$$\alpha_i = \tan^{-1}\left(\frac{v_{ywi}}{v_{xwi}}\right), \tag{2.6}$$

where v_{xw} is the longitudinal wheel velocity and v_{yw} is the lateral wheel velocity. The wheel velocities are computed as

$$v_y = \dot{y} + a\dot{\psi}, \qquad v_x = \dot{x} - c\dot{\psi}, \qquad i = 1, \tag{2.7a}$$

$$v_y = \dot{y} + a\dot{\psi}, \qquad v_x = \dot{x} + c\dot{\psi}, \qquad i = 2, \tag{2.7b}$$

$$v_y = \dot{y} - b\dot{\psi}, \qquad v_x = \dot{x} - c\dot{\psi}, \qquad i = 3, \tag{2.7c}$$

$$v_y = \dot{y} - b\dot{\psi}, \qquad v_x = \dot{x} + c\dot{\psi}, \qquad i = 4. \tag{2.7d}$$

The slip ratios in equation 2.5 are defined as

$$\sigma_i = \begin{cases} \dfrac{r_w\omega_i}{v_{ywi}} - 1, & \text{if } v_{ywi} > r_w\omega_i,\ v_{ywi} \neq 0, \quad \text{during braking} \\ 1 - \dfrac{v_{ywi}}{r_w\omega_i}, & \text{if } v_{ywi} < r_w\omega_i,\ \omega_i \neq 0, \qquad \text{during acceleration} \end{cases} \tag{2.8}$$

where r_w is the radius of the wheel and ω is again the angular velocity, as depicted in Figure 2.2, and the wheel velocities are computed in (2.7). Note that $\sigma \in [-1,\, 1]$. The angular velocities of the wheels, ω_i in equation (2.31), are obtained by integrating the set of differential equations

$$J_{wi}\dot{\omega}_i = -f_{xi}r_w - T_{bi} + T_{\text{eng,i}} - b\omega_i, \qquad i \in \{1,2,3,4\}. \tag{2.9}$$

J_{wi} is the inertia of wheel i, T_b is the braking torque generated at the wheel and T_{eng} is the accelerative torque generated at the wheel.

The main contribution of the normal force F_z in (2.5) is due to the weight of the vehicle and we make use of the following assumption,

Assumption 2 *The normal force F_z is constant and the distribution is governed by the parameters a and b describing the geometry of the car:*

$$F_{zi} = \frac{bmg}{2(a+b)}, \qquad i \in \{1,2\}, \tag{2.10a}$$

$$F_{zi} = \frac{amg}{2(a+b)}, \qquad i \in \{3,4\}. \tag{2.10b}$$

We note that equations (2.10) are an approximation of the actual normal force as longitudinal and lateral accelerations, as well as roll and pitch of the vehicle, can vary these forces. Equations (2.10) describe the normal forces in steady-state operation.

The Nonlinear Four Wheel Model is compactly written as

$$\xi(t) = f^{4w}(\xi(t), u(t)), \tag{2.11}$$

where $\xi = [\dot{x},\, \dot{y},\, \psi,\, \dot{\psi},\, X,\, Y,\, \omega_i]$ and $u = [\delta,\, T_i]$, $i \in \{1,2,3,4\}$ and the net torque input at each wheel is $T = T_{\text{eng}} - T_b$.

2.2 Tire Models

In this section two tire models, the Pacejka model, also known as The Magic Tire Formula, and the Fiala tire model, are introduced. The Pacejka tire model is semi-empirical and in general models the tire forces more accurately compared to the Fiala model, however, the Fiala model is less complex and is utilized during real-time implementation to speed up the sampling time. The Pacejka model is presented in subsection 2.2.1 and the Fiala model is presented in subsection 2.2.2.

2.2.1 Pacjeka Tire Model

The Pacjeka tire model, or as its called the Magic Tire Formula [6], is a semi-empirical formula that provides a method to calculate the longitudinal f_x and lateral f_y tire forces and aligning moment M. It provides a benefit over linear tire models, presented in section 2.2.3, in that it is accurate for a wide range of operating conditions, including large slip angles and slip ratios, and combined longitudinal and lateral force generation. The simplified Pacjeka model, presented in [91], where only lateral or longitudinal forces are being calculated, is expressed as

$$Y(X) = y(x) - S_v, \tag{2.12}$$

with

$$y(x) = D \sin\left[C \arctan\left(B\,x - E\left(B\,x - \arctan B\,x\right)\right)\right], \tag{2.13a}$$

$$x = X - S_h, \tag{2.13b}$$

where Y is the output variable, either lateral force f_y or longitudinal force f_x. X is the input variable, either slip angle α shown in Figure 2.2, or slip ratio σ calculated in equation (2.31), respectively. The model parameters B, C, D, E, S_v and S_h are summarized in Table 2.2.1.

Table 2.1: List of Pacejka tire model parameters.

Parameter	Name	Parameter	Name
B	stiffness factor	C	shape factor
D	peak value	E	curvature factor
S_h	horizontal shift	S_v	vertical shift

To illustrate the meaning of some of the factors the sketch in Figure 2.4 is produced. The coefficient D is shown to represent the peak value of the tire force characteristics, the product BCD corresponds to the slop at the origin, the value y_s is the asymptotic value of the output y at large values of x, the shape factor C controls the limits of the range of the sine function in equation (2.13) and therefore shapes the resulting curve,

$$C = \frac{2}{\pi} \sin^{-1} \left(\frac{y_s}{D} \right). \tag{2.14}$$

The offsets S_h and S_v account for ply-steer and conicity effects and possibly rolling resistance which can cause f_y and f_x curves to not pass through the origin [6, 91]. E, the curvature factor, can change the shape of the curve near the peak and controls the value of the sip x_m at which the peak of the curve occurs,

$$E = \frac{B\, x_m - \tan\left(\dfrac{\pi}{2\, C} \right)}{B\, x_m - \tan^{-1}\left(B\, x_m \right)}. \tag{2.15}$$

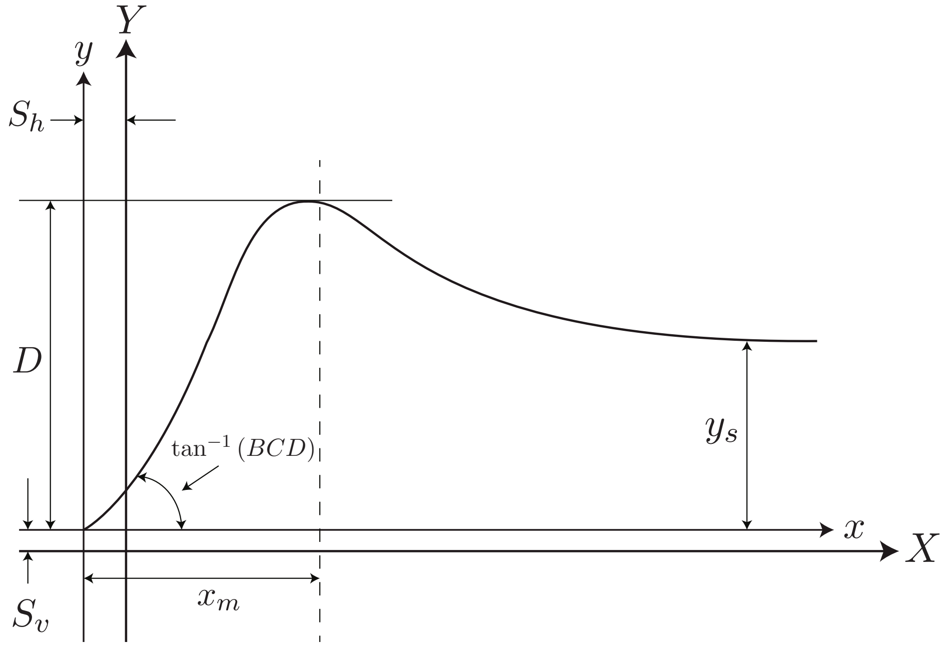


Figure 2.4: A sketch of the Pacjeka tire model function illustrating the parameters of the formula.

The geometry of the wheels, such as wheel camber, can introduce offset in the f_y vs. α curve. To accommodate the asymmetry, the curvature factor, E, is made dependent on the sign of the abscissa, x.

$$E = E_o + \Delta\, E\, \mathrm{sgn}(x) \tag{2.16}$$

This modification will also accommodate the difference in shape expected to occur in the f_x vs. σ characteristics between the driving and braking ranges [6].The value of the curve approaches an asymptotic value, y_s, at large slip values. This is expresses as

$$y_s = D \sin(\frac{\pi}{2}\, C) \tag{2.17}$$

The parameters discussed are functions of normal load and wheel camber angle. However, the parameters B, C, D, and E can be expressed as function of normal load F_z and friction coefficient μ. They are expressed as follows

$$D = a_1 F_z^2 + a_2 F_z \tag{2.18a}$$

$$BCD = a_3 \text{sin}(a_4 \text{arctan}(a_5 F_z)) \quad \text{(lateral)} \tag{2.18b}$$

$$BCD = \frac{a_3 F_z^2 + a_4 F_z}{e^{a_5 F_z}} \quad \text{(longitudinal)} \tag{2.18c}$$

$$E = a_6 F_z^2 + a_7 F_z + a_8 \tag{2.18d}$$

where a_1, a_2, ..., a_8 are constants that have to be determined for each tire. We make use of the following assumptions,

Assumption 3 *In equation (2.23) the vertical forces F_{z_i} are assumed constant and determined by the vehicle's steady state weight distribution when no lateral or longitudinal accelerations act at the vehicle center of gravity.*

Assumption 4 *The friction coefficient is assumed to be known and to be the same at all wheels, i.e., $\mu_i = \mu$, $\forall i$ and constant over a finite time horizon. At each time instant an estimate of μ is assumed available. See, e.g., [102, 94, 88, 97] for an overview on friction estimation techniques. The friction coefficient enters the system equations as a known parameter.*

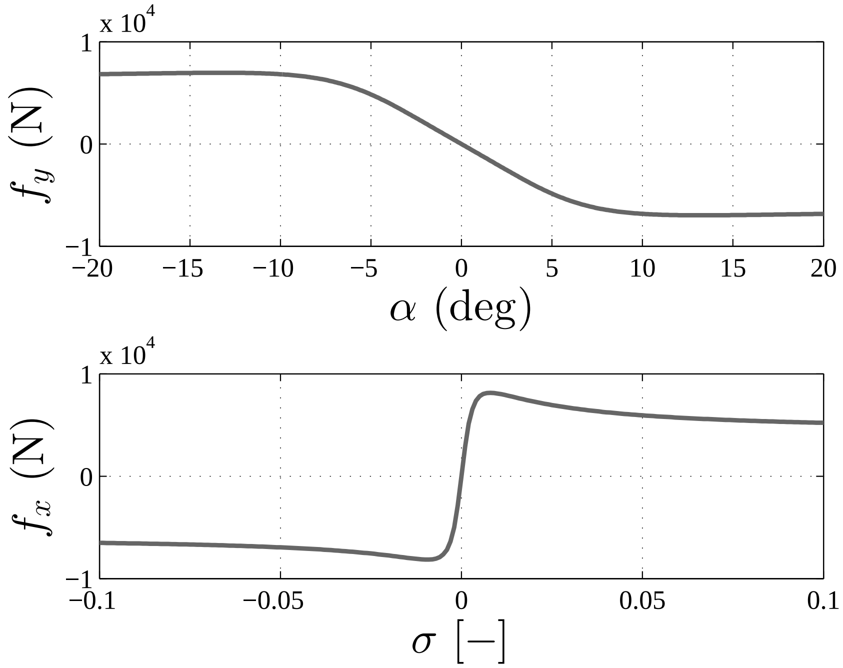


Figure 2.5: Pacejka tire force curve comparing lateral tire force f_y to slip angle α and longitudinal tire force f_x to slip ratio σ In these plots the friction coefficient $\mu = 1$.

Simplified Pacejka Tire Model

In this section a simplified version of the Pacejka Tire Model is presented. This model is useful to reduce the computation complexity for real-time applications. We assume the pedal braking distributes braking forces according to the following relation:

$$f_{x1} = f_{x2} = \sigma \tfrac{F_b}{2}, \quad f_{x3} = f_{x4} = (1 - \sigma)\tfrac{F_b}{2}, \tag{2.19}$$

where σ is a constant (vehicle dependant) distribution parameter and F_b is the total braking force. An actuator capable of augmenting the braking of the driver is assumed available.

The tire force components f_{xi}, f_{yi}, are generated at the tire-road contact patch. f_{yi} is computed using the Pacejka tire formula [87]. We let α_i denote

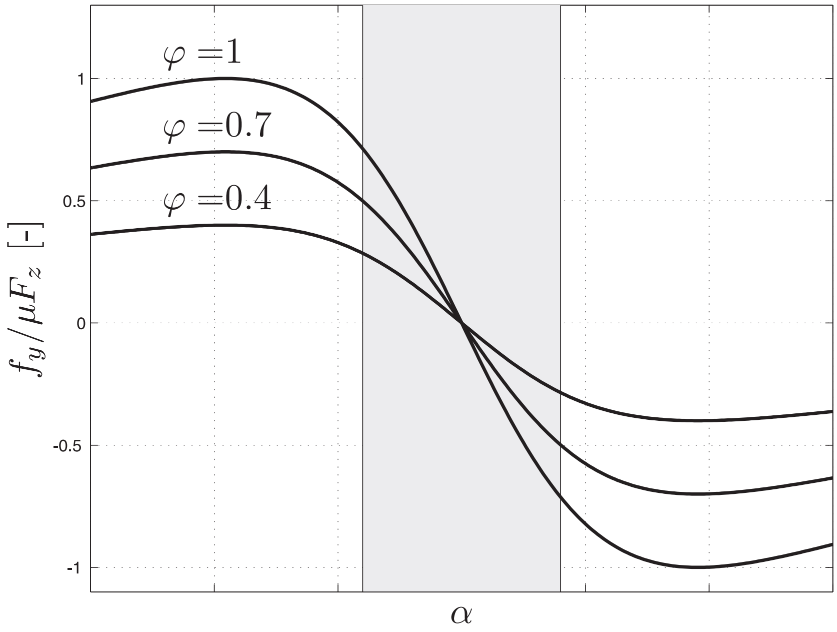


Figure 2.6: Sample plots of lateral tire force characteristics for different levels of braking, where $\varphi = \sqrt{(\mu_i F_{z_i})^2 - f_{x_i}^2}$.

the tire slip angle, μ_i denote the friction coefficient, F_{zi} denote the vertical load at each wheel and write the tire formula as:

$$f_{y_i} = \sqrt{(\mu_i F_{z_i})^2 - f_{x_i}^2}\,\sin\left(C_i \arctan\left(B_i \alpha_i\right)\right), \tag{2.20}$$

where C_i, B_i are tire parameters that are calibrated using experimental data. The tire slip angles α_i in (2.20) are approximated as:

$$\alpha_1 = \frac{v_y + l_f\dot{\psi}}{v_x - \frac{w_t}{2}\dot{\psi}} - \delta, \qquad \alpha_2 = \frac{v_y + l_f\dot{\psi}}{v_x + \frac{w_t}{2}\dot{\psi}} - \delta, \tag{2.21a}$$

$$\alpha_3 = \frac{v_y - l_r\dot{\psi}}{v_x - \frac{w_t}{2}\dot{\psi}}, \qquad \alpha_4 = \frac{v_y - l_r\dot{\psi}}{v_x + \frac{w_t}{2}\dot{\psi}}. \tag{2.21b}$$

where w_t denotes the vehicle track width. Specific values of the parameters B_i and C_i are reported in the results of Chapter 7.

2.2.2 Fiala Tire Model

The Fiala model, presented in [57], is a reduced complexity model that calculates the longitudinal tire force as a linear function of the required acceleration and the lateral tire force as a nonlinear function of the tire slip angle.

The longitudinal force in the tire frame, f_{xi} is calculated from the equation

$$f_{xi} = \beta_r \mu_i F_{z_i} \tag{2.22}$$

where $\beta_r \in [-1,\ 1]$ is referred to as the braking ratio. $\beta_r = -1$ corresponds to full braking and $\beta_r = 1$ corresponds to full throttle. In this thesis we consider an active safety application where $\beta_r \leq 0$.

Assumption 5 *An actuator capable of augmenting the driver's braking is assumed available and the total braking force is denoted by F_b.*

f_{yi} is computed by the equation,

$$f_{y_i} = \begin{cases} -C_{\alpha_i}\tan(\alpha_\mathrm{i}) + \dfrac{\mathrm{C}^2_{\alpha_\mathrm{i}}}{3\eta\mu\mathrm{F}_{\mathrm{z_i}}}|\tan(\alpha_\mathrm{i})|\tan(\alpha_\mathrm{i}) - \dfrac{\mathrm{C}^3_{\alpha_\mathrm{i}}}{27\eta^2\mu_\mathrm{i}^2\mathrm{F}^2_{\mathrm{z_i}}}\tan^3(\alpha_\mathrm{i}), \\ \qquad\qquad\qquad \text{if } |\alpha_i| < \alpha_{sl} \\ -\eta\mu_i F_{z_i}\mathrm{sgn}(\alpha_\mathrm{i}), \quad \text{if } |\alpha_\mathrm{i}| \geq \alpha_\mathrm{sl} \end{cases} \tag{2.23}$$

$\forall i$ where α denotes the tire slip angle, μ denotes the friction coefficient, and F_z denotes the vertical load at each wheel. C_α is the tire cornering stiffness and,

$$\eta = \sqrt{\mu^2 F_z^2 - f_x^2}/(\mu F_z), \tag{2.24}$$

which can be written as $\eta = \sqrt{1 - \beta_r^2}$. The lateral tire force characteristics are shown in Figure 2.7 where the region in dashed lines is where $|\alpha| \leq \alpha_{sl}$ and,

$$\alpha_{sl} = \tan^{-1}(\frac{3\eta\mu\mathrm{F_z}}{\mathrm{C}_\alpha}). \tag{2.25}$$

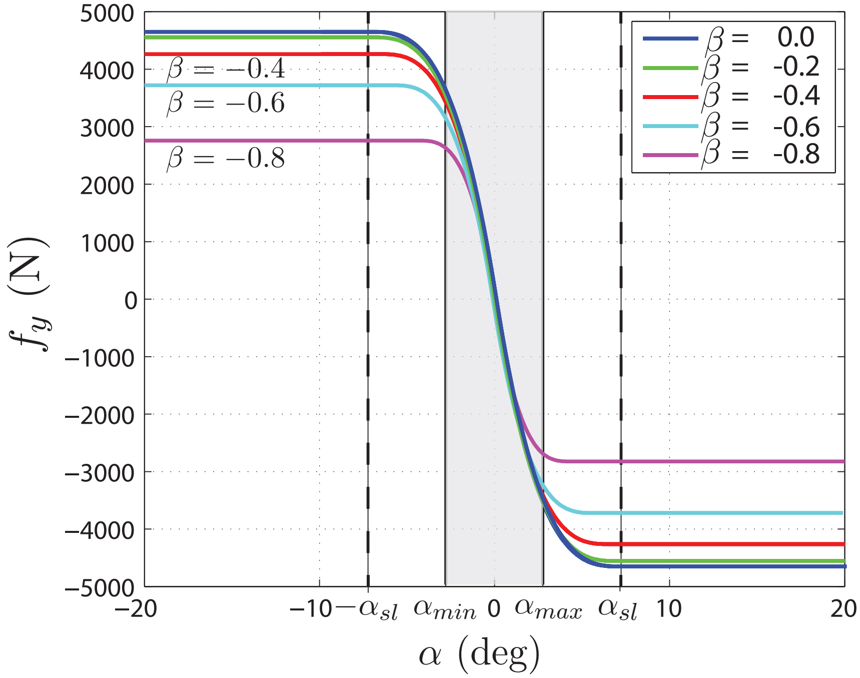


Figure 2.7: Lateral tire force characteristics compared to tire slip angle for different levels of braking.

The tire slip angles α_i in (2.23) are approximated as,

$$\alpha_1 = \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}} - \delta, \qquad \alpha_2 = \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}} - \delta, \tag{2.26a}$$

$$\alpha_3 = \frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}, \qquad \alpha_4 = \frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}. \tag{2.26b}$$

The same assumptions pertaining to the Pacejka tire model in section 2.2.1 are relevant here,

Assumption 6 *In equation (2.23) the vertical forces F_{z_i} are assumed constant and determined by the vehicle's steady state weight distribution when no lateral or longitudinal accelerations act at the vehicle's center of gravity.*

Assumption 7 *The friction coefficient is assumed to be known and to be the same at all wheels, i.e., $\mu_i = \mu$, $\forall i$ and constant over a finite time* *horizon. At each time instant an estimate of μ is assumed available. See, e.g., [102, 94, 88, 97] for an overview on friction estimation techniques. The friction coefficient enters the system equations as a known parameter.*

2.2.3 Linear Tire Force Model

For small slip angles and small slip ratio values, the linear tire force relationship between Y and X, defined in the Pacejka tire force model in section 2.2.1 can be approximated by

$$Y = (BCD)\,X \tag{2.27}$$

where BCD in equation (2.27) represents the cornering stiffness C_α or the longitudinal tire stiffness C_σ. Further, at small slip angles and slip ratio values the tire slip angles are approximated as they are in the Fiala tire model in section 2.2.2 equation (2.44). They are reproduced here for completeness,

$$\alpha_i = \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}} - \delta, \qquad i \in \{1,2\} \tag{2.28a}$$

$$\alpha_i = \frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}, \qquad i \in \{3,4\} \tag{2.28b}$$

Therefore, the front and rear lateral tire forces are approximated by,

$$f_{yi} = C_\alpha\left(\frac{\dot{y} + l_f\dot{\psi}}{\dot{x}} - \delta\right), \qquad i \in \{1,2\} \tag{2.29a}$$

$$f_{yi} = C_\alpha\left(\frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}\right), \qquad i \in \{3,4\} \tag{2.29b}$$

while the longitudinal tire forces are approximated by

$$f_{xi} = C_{\sigma i}\sigma_i \tag{2.30}$$

and the slip ratios σ_i are calculated by

$$\sigma_i = \begin{cases} \dfrac{r_w\omega_i}{v_{ywi}} - 1, & \text{if } v_{ywi} > r_w\omega_i,\ v_{ywi} \neq 0, \quad \text{during braking} \\ 1 - \dfrac{v_{ywi}}{r_w\omega_i}, & \text{if } v_{ywi} < r_w\omega_i,\ \omega_i \neq 0, \quad \text{during acceleration} \end{cases} \tag{2.31}$$

These relationships approximate the tire forces in the linear region of the tire characteristics functions of section 2.2.1 Figure 2.5 and section 2.2.2 Figure 2.7. They will, of course, diverge from the actual tire forces quite significantly at high slip values, but this problem may be mitigated by enforcing a tire slip constraint.

2.3 Bicycle Model

The bicycle model is derived from the Four Wheel Nonlinear Model presented in section 2.1. It is based upon a simplification where the two front tires are lumped together and the two rear tires are lumped together, as shown in the diagram of Figure 2.8. The differential equations describing the model are derived from the general rigid body dynamics from equation (2.1). They take the form,

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{xf} + 2F_{xr}, \tag{2.32a}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr}, \tag{2.32b}$$

$$I\ddot{\psi} = 2l_f F_{yf} - 2l_r F_{yr}. \tag{2.32c}$$

where m is the vehicle mass, I is the moment of inertia in the vertical axis, l_f and l_r are the distances from the center of gravity to the front and rear wheels, respectively. The state of the bicycle model is longitudinal velocity, $\dot{x}$, lateral velocity, $\dot{y}$, and yaw rate $\dot{\psi}$. The longitudinal and lateral tire forces are denoted as F_{xi} and F_{yi} where $i \in \{f, r\}$. We make the following remark,

Remark 1 *The tire forces F_{xi} and F_{yi} where $i \in \{f, r\}$ in equation* (2.32) *represent the forces generated by the contact of a single wheel with the ground. The factor of* 2 *recovers the forces generated by both wheels in the front or rear of the vehicle.*

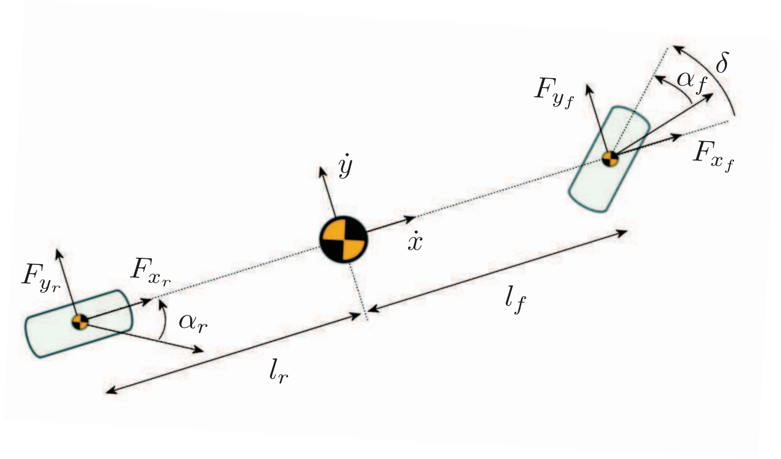


Figure 2.8: Modeling notation.

Note that the tire forces may be calculated by any of the methods presented in section 2.2 and are not reproduced here. We make the following assumption regarding the tire forces,

Assumption 8 *The friction coefficient μ_i, slip angles α_i, and slip ratios σ_i are assumed to be equal at the left and right wheels, i.e., no μ-split surfaces and the braking and acceleration is the same at the left and right sides.*

The Bicycle Model is written in the following compact form,

$$\dot{\xi}(t) = f^{2w}(\xi(t), u(t)), \tag{2.33}$$

where the state vector $\xi = [\dot{x},\, \dot{y},\, \psi,\, \dot{\psi},\, Y,\, X]$ and the input $u = \delta_f$. We make the following assumption,

Assumption 9 *The slip ratios σ_i and coefficient of friction μ are assumed to be known and enter the system as known parameters.*

2.4 Spatial Vehicle Model

A particular vehicle model that will be important for the obstacle avoidance model predictive controller of chapter 7 will be the Spatial Vehicle Model. This model utilizes a coordinate transformation to derive the dynamics of the vehicle about an arbitrary path, for our purposes we define this to be the center line of the lane. The independent variable becomes distance along the path s instead of time t. This transformation allows the position of the vehicle to be known explicitly at each sampling instant of an optimization routine, yet still retains the freedom of the solver to vary the velocity of the vehicle, which is critical in safety applications.

Figure 2.9 shows the curvilinear coordinate system used in the spatial model as well as the states of the model. 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.

The following kinematic equations can be derived from Figure 2.9:

$$v_s = (\rho - e_y)\,\dot{\psi}_s, \tag{2.34a}$$

$$v_s = \dot{x}\,cos(e_\psi) - \dot{y}\,sin(e_\psi), \tag{2.34b}$$

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

$$\dot{s} = \rho\,\dot{\psi}_s \; = \; \left(\frac{\rho}{\rho - e_y}\right)(\dot{x}\,cos(e_\psi) - \dot{y}\,sin(e_\psi)) \tag{2.35}$$

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,

$$\frac{d\xi^s}{ds} = \frac{d\xi^s}{dt}\frac{dt}{ds}, \tag{2.36}$$

we can calculate the derivative of ξ^s with respect to s as follows $((\cdot)'$ represents

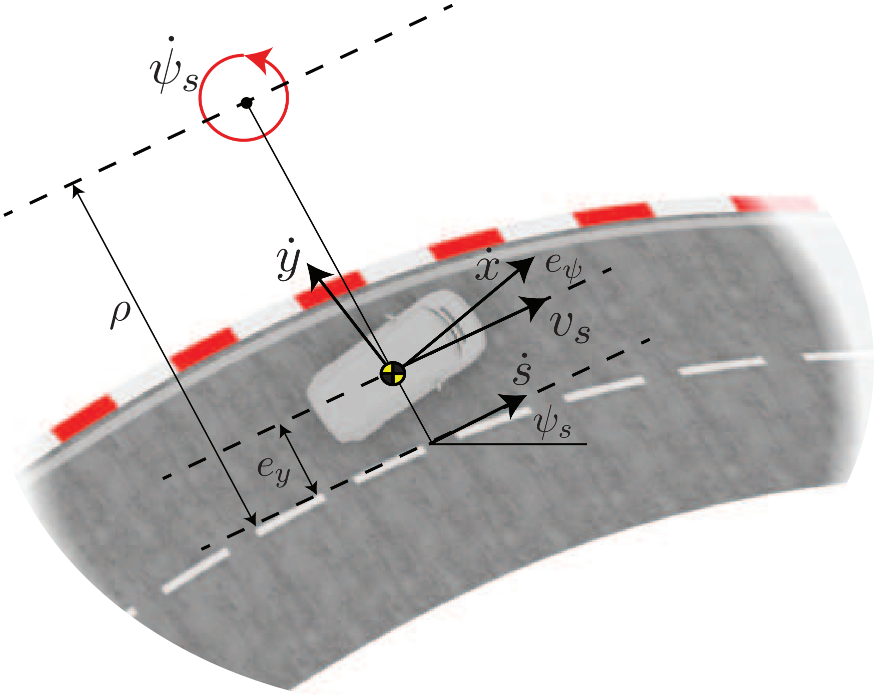


Figure 2.9: The curvilinear coordinate system. The dynamics are derived about a curve defining the center-line of a track. The coordinate s defines the arc-length along the track. The relative spatial coordinates e_y and e_ψ are shown.

the derivative with respect to s):

$$\dot{y}' = \ddot{y}/\dot{s}; \qquad \dot{x}' = \ddot{x}/\dot{s}; \qquad \dot{\psi}' = \ddot{\psi}/\dot{s}; \tag{2.37a}$$

$$e'_\psi = (\psi - \psi_s)' = \dot{\psi}/\dot{s} - \psi'_s; \tag{2.37b}$$

$$e'_y = \dot{e}_y/\dot{s} = (\dot{x}\, sin(e_\psi) + \dot{y}\, cos(e_\psi))/\dot{s} \tag{2.37c}$$

where $\ddot{y}$, $\ddot{x}$ and $\ddot{\psi}$ are computed from the bicycle model in section 2.3 and ξ^s denotes the vehicle state for the spatial model. We make the following assumption

Assumption 10 *A sensor exists that extracts the road information, ψ'_s, at* *each time instant as well as over a future given horizon. The signal enters the model as a known parameter.*

Therefore, the dynamics of the spatial vehicle model are written as,

$$\dot{x}' = \left(\dot{y}\dot{\psi} + \frac{1}{m}(2F_{xf} + 2F_{xr})\right)/\dot{s}, \tag{2.38a}$$

$$\dot{y}' = \left(-\dot{x}\dot{\psi} + \frac{1}{m}(2F_{yf} + 2F_{yr})\right)/\dot{s}, \tag{2.38b}$$

$$\dot{\psi}' = \frac{1}{I}\left(2l_f F_{yf} - 2l_r F_{yr}\right)/\dot{s}, \tag{2.38c}$$

$$e'_\psi = (\psi - \psi_s)' = \dot{\psi}/\dot{s} - \psi'_s, \tag{2.38d}$$

$$e'_y = \dot{e}_y/\dot{s} = (\dot{x}\, sin(e_\psi) + \dot{y}\, cos(e_\psi))/\dot{s} \tag{2.38e}$$

where $\xi^s = [\dot{x},\, \dot{y},\, \psi,\, \dot{\psi},\, e_\psi,\, e_y]$, m is the vehicle mass, I is the moment of inertia, and the forces F_{ji}, $j \in \{x, y\}$, $i \in \{f, r\}$ are calculated using the Fiala tire model in section 2.2.2.

The spatial vehicle dynamics are compactly written as,

$$\xi^{s'}(s) = f^s(\xi^s(s), u^s(s)). \tag{2.39}$$

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 ($u^s = [\delta_f, \beta_r]$), where β_r is calculated from the Fiala tire model in section 2.2.2. We make the following remark,

Remark 2 *The time as function of s, $t(s)$, can be retrieved by integrating $t' = 1/\dot{s}$ if needed. That is,*

$$t(s_f) - t(s_0) = \int_{s_0}^{s_f} \left(\frac{1}{\left(\frac{\rho}{\rho - e_y}\right)\, (\dot{x}\, cos(e_\psi) - \dot{y}\, sin(e_\psi))}\right) ds \tag{2.40}$$

2.5 Linear Vehicle Error Model

If the longitudinal velocity is kept constant and the slip values small, as in nominal driving conditions, a linear vehicle model can be employed. Such a model is very useful for robust control methods that are presented later. This section introduces the required simplifications and assumptions and presents the Linear Vehicle Model. The error dynamics of a vehicle are linear with respect to the lateral motion within the lane by assuming a constant velocity, V_x, and constraining the slip angles, α_i, to operate in the linear region of the tire forces [91]. The differential equations describing the motion are compactly written as,

$$\dot{x}(t) = \mathbf{A}\, x(t) + \mathbf{B}\, u(t) + \mathbf{E}\dot{\psi}_{\text{road}}(t) \tag{2.41}$$

where the state is $x(t) = [e_y,\, \dot{e}_y,\, e_\psi,\, \dot{e}_\psi]^T \in \mathbb{R}^{4\times1}$ at time t and $x_0 = x(0)$. The state of the system is in a road aligned coordinate frame and is shown in Figure 2.10. The control input $u = \delta$ is the steering angle command and the system matrices are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & -\frac{2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x} & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix}, \tag{2.42a}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} - V_x \\ 0 \\ -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix}, \tag{2.42b}$$

where $\mathbf{A} \in \mathbb{R}^{4\times4}$, $\mathbf{B} \in \mathbb{R}^{4\times1}$, and $\mathbf{E} \in \mathbb{R}^{4\times1}$. m and I_z denote the vehicle mass and yaw inertia and l_f and l_r denote the distances from the vehicle center of gravity to the front and rear axles, depicted in Figure 2.8. e_ψ and e_y denote the vehicle orientation and lateral position, respectively, in a road aligned coordinate frame.

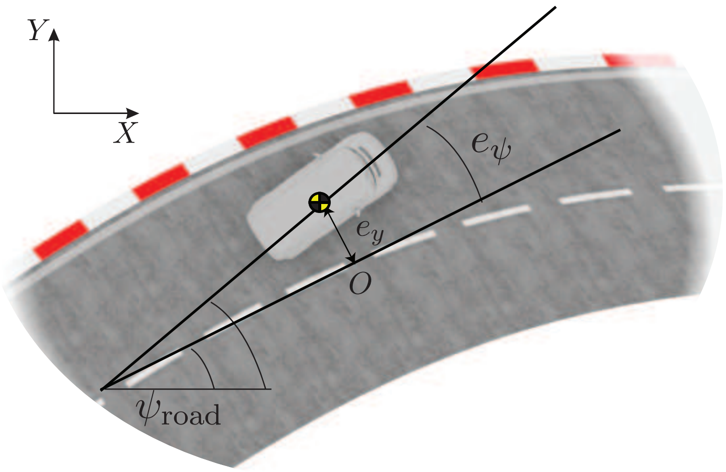


Figure 2.10: A road aligned coordinate frame illustrating the error states with respect to the road, e_y and e_ψ.

The tire cornering stiffness is denoted $C_{\alpha f}$ and $C_{\alpha r}$ for the front and rear tires, respectively. The lateral tire force components in the vehicle body frame are modeled as,

$$F_{yi} = -C_{\alpha i}\alpha_i, \qquad i \in \{f, r\} \tag{2.43}$$

where α_i is the slip angle at wheel i. We assume only the steering angles of the front wheels can be controlled, i.e., $\delta_f = \delta$ and $\delta_r = 0$. In addition, an actuator which corrects the driver commanded steering angle, such that $\delta = \delta_d + \delta_c$, is available, where δ_d is the driver commanded steering angle and δ_c is the correcting steering angle component. The tire slip angles α_i in (2.43) are approximated as,

$$\alpha_f = \frac{v_y + l_f\dot{\psi}}{v_x} - \delta, \quad \alpha_r = \frac{v_y - l_r\dot{\psi}}{v_x} \tag{2.44}$$

We make use of the following assumptions,

Assumption 11 *The friction coefficient is assumed to be known and to be the same at all wheels, i.e., $\mu_i = \mu$, $\forall i$ and constant over a finite time horizon. At each time instant an estimate of μ is assumed available.*

Assumption 12 *The signal $\dot{\psi}_{\text{road}}$ is assumed to be known and every time instant an estimate of $\dot{\psi}_{\text{road}}$ is available over a finite time horizon. See [10] for an overview of sensing technologies that can be used to obtain this signal.*

The linear vehicle model is written in the following compact form,

$$\dot{\xi}(t) = f^{\text{lin}}(\xi(t), u(t), p(t)), \tag{2.45}$$

where the state $\xi = [e_y,\, \dot{e}_y,\, e_\psi,\, \dot{e}_\psi]^T$, the input $u = \delta$, and the parameter $p = \dot{\psi}_{\text{road}}$.

2.6 Safety Constraints

We recall that the overall aim of the safety system proposed in this thesis is to keep the vehicle in the lane and avoid collision with obstacles while maintaining a stable vehicle motion. In this section, we express the requirements that the vehicle stays in the lane while operating in a stable operating region as constraints on the vehicle state and input variables.

Let $e_{y_i},\ i \in \{1,2,3,4\}$ be the distances of the four vehicle corners from the lane centerline. e_{y_i} is written as

$$\begin{aligned} e_{y_1} &= e_y + \frac{c}{2}\cos(e_\psi) + a\sin(e_\psi) \\ e_{y_2} &= e_y - \frac{c}{2}\cos(e_\psi) + a\sin(e_\psi) \\ e_{y_3} &= e_y + \frac{c}{2}\cos(e_\psi) - b\sin(e_\psi) \\ e_{y_4} &= e_y - \frac{c}{2}\cos(e_\psi) - b\sin(e_\psi) \end{aligned} \tag{2.46}$$

where c is the vehicle width, a and b are the distances of the center of gravity from the front and rear vehicle bumpers, respectively. Note these lengths differ from w_t, l_f, and l_r, which measure the chassis for the vehicle dynamics. a, b, and c measure the body of the vehicle to geometrically impose the safety constraints. The requirement that the vehicle stays in the lane is then expressed as,

$$-e_{y_{\max}} \leq e_{y_i} \leq e_{y_{\max}},\ \forall i. \tag{2.47}$$

In addition to staying in the lane, we require that the vehicle operates in a region of the state space where the vehicle is easily maneuverable by a normally skilled driver. Consider, for example, the tire force characteristics shown in Figure 2.7. In the shaded region, the nonlinearity in the lateral tire force characteristics is less evident. In this region the vehicle behavior is predictable by most drivers and Electronic Stability Control (ESC) systems are inactive [50, 96]. The requirement that the vehicle operates in stable operating conditions is thus ensured by limiting the tire slip angles α_i,

$$\alpha_{i_{\min}} \leq \alpha_i \leq \alpha_{i_{\max}},\ \forall i. \tag{2.48}$$

The constraints (2.47)-(2.48) are compactly written as,

$$h(x(t), u(t), p(t), t) \leq \mathbf{0}, \tag{2.49}$$

where $\mathbf{0}$ is the zero vector with appropriate dimension. We make the following remark regarding feasibility.

Remark 3 *The constraints on the state may be made soft by introducing a slack variable ε and enforcing the modified constraints*

$$h(x(t), u(t), p(t), t) \leq \mathbf{1}\varepsilon, \tag{2.50a}$$

$$\varepsilon \geq 0. \tag{2.50b}$$

Note that any constraints on the input $u(t)$ remain hard as these constraints reflect physical limitations of the actuators.

Chapter 3

Driver Behavior Models

We utilize a model of the driver's steering behavior. In general, an accurate description of the driver's behavior requires complex models accounting for a large amount of exogenous signals [19, 98]. We are interested in very simple model structures, enabling the design of a low complexity model-based threat assessment and control design algorithm. In this thesis the driver's steering behavior is described by a model, where the vehicle state and the road geometry information are exogenous signals, the steering angle is the model output and the steering model parameters are estimated based on the observed behavior of the driver. The modeling and estimation of the nominal driver behavior considered in this thesis was presented first in [29]. In this chapter we will detail the nominal model and extend the model to include uncertainty in the driver's behavior.

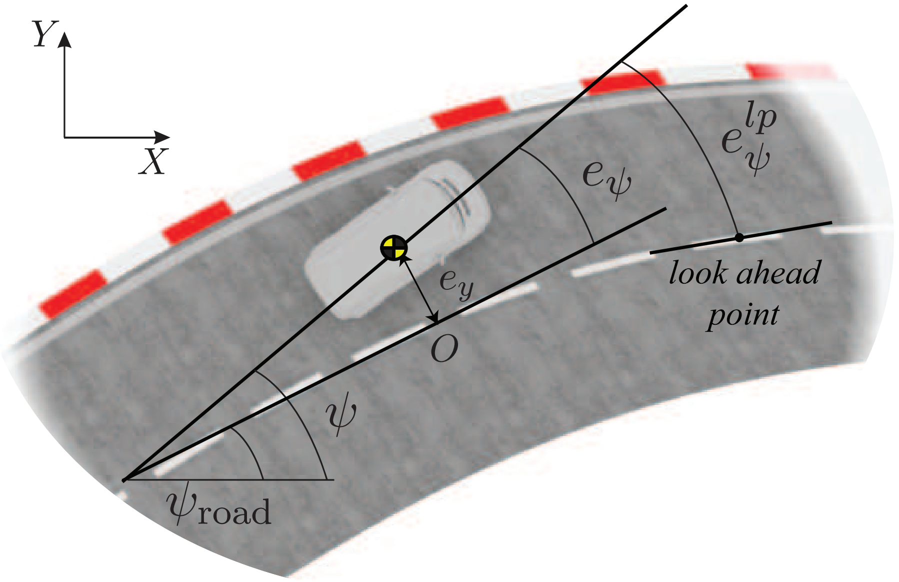


Figure 3.1: A road aligned coordinate frame illustrating the error states with respect to the road, e_y and e_ψ, as well as the driver model coordinates, the look-ahead point, lp, the error angle of the road at the look-ahead point e_ψ^{lp}, as well as the inertial frame angles of the vehicle ψ and road ψ_{road}.

3.1 Nominal Driver Model

Define the orientation error e_ψ^{lp}, w.r.t. a look-ahead point as in Figure 3.1,

$$e_\psi^{lp} = \psi - \psi_d^{lp} = e_\psi + \Delta\psi_{\text{road}}, \tag{3.1}$$

where ψ_{road}^{lp} is the desired orientation at time $t + t_{lp}$, with t the current time, $\Delta\psi_{\text{road}} = \psi_{\text{road}} - \psi_{\text{road}}^{lp}$ and t_{lp} the preview time that can be mapped into the preview distance d_{lp} under the assumption of constant speed $\dot{x}$. We compute an estimate of the driver commanded steering angle $\hat{\delta}_d$ as,

$$\hat{\delta}_d = K_y e_y + K_\psi e_\psi^{lp}, \tag{3.2a}$$

$$\hat{\delta}_d = K_y e_y + K_\psi e_\psi + K_\psi \Delta\psi_{\text{road}}, \tag{3.2b}$$

with K_y and K_ψ as gains that are, in general, time varying and are updated online. Clearly, $\Delta\psi_{\text{road}}$ in (3.1) depends on the preview time t_{lp} that, in our modeling framework, is considered as a parameter of the driver model. We also remark that the steering model (3.17) is velocity dependant since $\Delta\psi_{\text{road}}$ also depends on the vehicle speed $\dot{x}$.

Estimation results of the driver model parameters in (3.1)-(3.17), obtained using a nonlinear recursive least squares algorithm, are presented in [29] for both normal and aggressive driving styles. The estimation results for nominal driving are reported in Figure 3.2 and for aggressive driving in Figure 3.3.

The feedback equation for the driver model is compactly written as,

$$\hat{\delta}_d = \mathbf{F}x(t) + \mathbf{G}\Delta\psi_{\text{road}} \tag{3.3}$$

where $\mathbf{F}$ and $\mathbf{G}$ are defined by the particular vehicle model employed.

3.1.1 Recursive Least Squares Estimation

In this section the Recursive Least Squares Parameter Estimation algorithm is presented by which the parameters of the driver model of Equation (3.21) are estimated. This is run online and updated in real-time and will converge to different driver behavior after sufficient excitation of the estimation algorithm. A linear recursive estimator is written as,

$$y_k = H_k \mathbf{x} + v_k, \tag{3.4a}$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_{k-1} + K_k(y_k - H_k \hat{\mathbf{x}}_{k-1}). \tag{3.4b}$$

where H_k is a row vector and $K_k \in \mathbb{R}^{n \times k}$ is the estimator gain matrix. The term $y_k - H_k \hat{\mathbf{x}}_{k-1}$ is the innovation. The current estimation error is denoted as,

$$\epsilon_k = \mathbf{x} - \hat{\mathbf{x}}_k. \tag{3.5}$$

The mean of this error is computed as follows,

$$\begin{aligned} E(\epsilon_k) &= E(\mathbf{x} - \hat{\mathbf{x}}_k), && (3.6a) \\ &= (I - K_k H_k) E(\epsilon_{k-1}) - K_k E(v_k), && (3.6b) \end{aligned}$$

obtained by substitution of (3.4) in (3.5) where $I \in \mathbb{R}^{n \times n}$ is the identity matrix. To determine the optimal gain matrix K_k we minimize the aggregated variance of the estimation errors at time k,

$$\begin{aligned} J_k &= E(\|\mathbf{x} - \hat{\mathbf{x}}_k\|^2), && (3.7a) \\ &= E(\epsilon_k^T \epsilon_k), && (3.7b) \\ &= E(tr(\epsilon_k \epsilon_k^T)), && (3.7c) \\ &= tr(P_k) && (3.7d) \end{aligned}$$

where tr is the trace operator[1] and $P_k = E(tr(\epsilon_k \epsilon_k^T))$ is the estimation error covariance. P_k is found by substitution of Equation (3.6),

$$\begin{aligned} P_k &= E\left(((I - K_k H_k) E(\epsilon_k) - K_k v_k)((I - K_k H_k) E(\epsilon_k) - K_k v_k)^T\right), && (3.8a) \\ &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T. && (3.8b) \end{aligned}$$

[1] The trace of a matrix is the sum of its diagonal elements.

where $R_k = E(v_k v_k^T)$ is the covariance of v_k. Note the estimation error ϵ_{k-1} at time $k-1$ is independent of the measurement noise v_k at time k, i.e. $E(v_k \epsilon_{k-1}^T) = E(v_k)E(\epsilon_{k-1}) = 0$. Equation (3.8) is the recurrence for the covariance of the least squares estimation error. Note that P_k is a covariance matrix and is positive definite, $P_k \succ 0$. Finding the value of the optimal gain matrix K_k that minimizes the cost function in Equation (3.7) is found by taking the partial derivative of J_k with respect to K_k and setting it equal to 0. We make the following remarks,

Note 1 *The derivative of a function f with respect to a matrix $A = (a_{ij})$ is a matrix $\frac{\partial f}{\partial A} = \left(\frac{\partial f}{\partial a_{ij}}\right)$.*

Note 2 *The partial derivative $\frac{\partial}{\partial A} tr(ABA^T) = 2AB$ when B is symmetric.*

Therefore,

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_k H_k)P_{k-1}(-H_k^T) + 2K_k R_k \tag{3.9}$$

and setting Equation (3.9) equal to zero and solving for K_k yields the optimal gain matrix,

$$K_k = P_{k-1}H_k^T(H_k P_{k-1} H_k^T + R_k)^{-1}. \tag{3.10}$$

By setting $S_k = H_k P_{k-1} H_k^T + R_k$ we write,

$$K_k = P_{k-1} H_k^T S_k^{-1}. \tag{3.11}$$

A simple substitution of Equation (3.11) into (3.8) yields the recursion,

$$P_k = (I - K_k H_k) P_{k-1}. \tag{3.12}$$

Note 3 *The above P_k, R_k, and S_k are symmetric.*

For our problem of estimating the parameters for the nominal driver model, the estimator variables take the following form,

$$y_k = \delta_d, \qquad \text{actual driver steering angle,} \tag{3.13a}$$

$$H = [e_y \ e_\psi^{lp}], \qquad \text{state matrix,} \tag{3.13b}$$

$$\mathbf{x} = [K_y \ K_\psi]^T, \qquad \text{parameters to be estimated,} \tag{3.13c}$$

Then, the parameters are updated as follows,

$$\begin{bmatrix} K_y \\ K_\psi \end{bmatrix}_k = \begin{bmatrix} K_y \\ K_\psi \end{bmatrix}_{k-1} + K_k \left(\delta_{d,k} - [e_y \ e_\psi^{lp}]_k \begin{bmatrix} K_y \\ K_\psi \end{bmatrix}_{k-1} \right), \tag{3.14}$$

where K_k and P_k are updated by Equations (3.11) and (3.8), respectively. The initial values of the estimated parameters and covariance matrix may be set using prior knowledge, or if none is available one may set $P = \infty I \in \mathbb{R}^{n \times n}$. For our application the initial values are set as follows,

$$P = \begin{bmatrix} 10 & -0.0005 \\ -0.0005 & 25.7 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix}. \tag{3.15}$$

and the results from the estimation algorithm are reported in Figure 3.2 for nominal driving and Figure 3.3 for aggressive driving.

Note that the look ahead point is also estimated in this framework, which gives rise to the nonlinearity from the bilinear term in Equation (3.17), i.e. $K_\psi e_\psi^{lp}$. The estimation routine described in this section is for linear least squares. We handle this nonlinearity by discretizing the look ahead point and estimating the optimal parameters for each discrete distance and then choosing the set of parameters that minimize the cost.

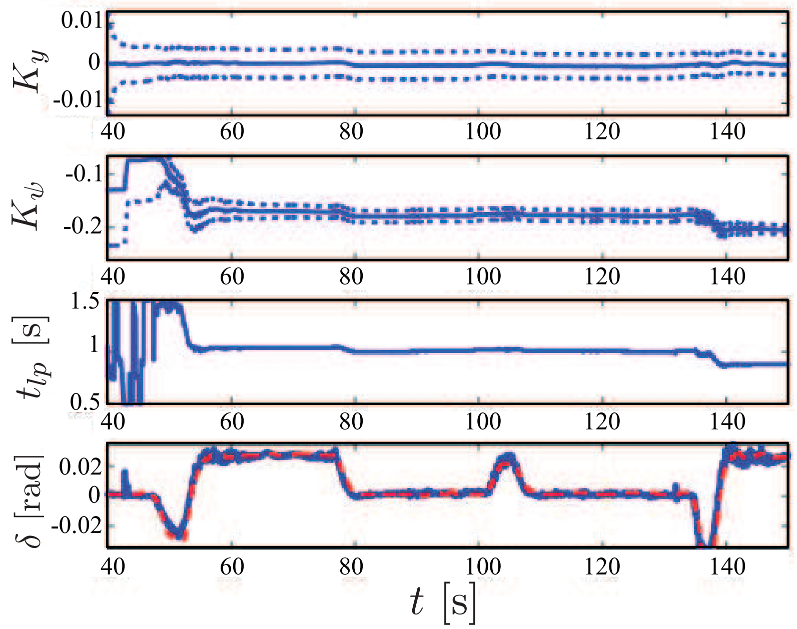


Figure 3.2: Estimation results from [29] of the nonlinear estimation of the feedback parameters for the driver model of (3.17) for nominal driving.

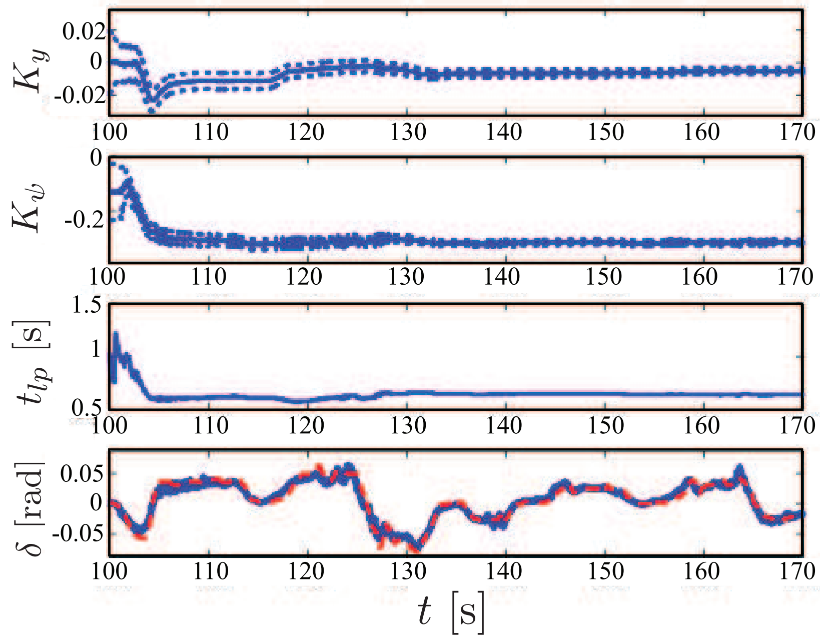


Figure 3.3: Estimation results from [29] of the nonlinear estimation of the feedback parameters for the driver model of (3.17) for aggressive driving.

3.2 Attentive Driver Model

In this section the nominal driver model of the previous section §3.1 is adapted to incorporate obstacles if the driver is deemed attentive and aware of an upcoming roadside obstacle.

Remark 4 *A driver monitoring system capable of determining driver distraction, such as the one suggested in [89], is assumed available.*

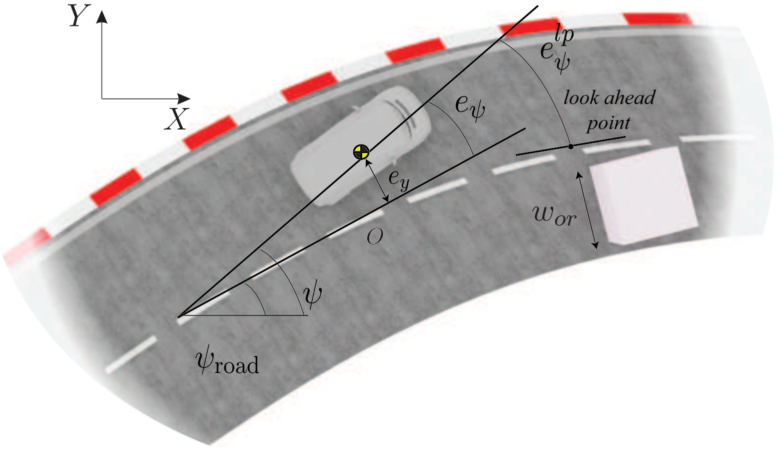


Figure 3.4: The road-aligned coordinate frame adapted to accommodate roadside obstacles.

Consider the sketch in Figure 3.4. A roadside obstacle has been added but is otherwise the same as Figure 3.1. Denote by w_{or}, w_{ol}, the width of an obstacle located at the right and left lane borders, respectively, which are zero in case no obstacle is present. We introduce the position error,

$$e_y^{lp} = e_y - \frac{1}{2}w_{or} + \frac{1}{2}w_{ol}, \tag{3.16}$$

which is the distance of the vehicle's center of gravity from the center of the free portion of the lane. Then, the driver model is adapted from Equation 3.17 to incorporate the variable e_y^{lp} and is written as,

$$\hat{\delta}_d = K_y e_y^{lp} + K_\psi e_\psi^{lp}. \tag{3.17}$$

3.3 Uncertain Driver Model

While the nominal driver model of section 3.1 represents the expected nominal driver behavior, this equation fails to capture the uncertainty in the driver's behavior. We assume the actual value of δ_d is assumed to lie in an interval centered at $\hat{\delta}_d$. Then,

$$\mathcal{W}(x) = \left\{ \delta_d : \|\delta_d - \hat{\delta}_d\| \leq \epsilon > 0,\ \|\delta_d\| \leq \delta_{d,\max} \right\}, \tag{3.18}$$

where ϵ is a parameter that must be chosen. The constraint $\delta_d \in \mathcal{W}(x)$ can also be expressed in terms of a polytopic contraint in $\mathbb{R}$, independent of x, by assuming the worst-case approximation. That is,

$$\delta_d \in \mathcal{W}(x) \Rightarrow \delta_d \in \{\hat{\delta}_d \oplus \mathcal{W}\} \subseteq \mathbb{R} \tag{3.19}$$

Note 4 *The Monkowski sum of two polytopes $\mathcal{P}$ and $\mathcal{Q}$ is a polytope*

$$\mathcal{P} \oplus \mathcal{Q} := \{x + q \in \mathbb{R}^n \ : \ x \in \mathcal{P},\ q \in \mathcal{Q}\}. \tag{3.20}$$

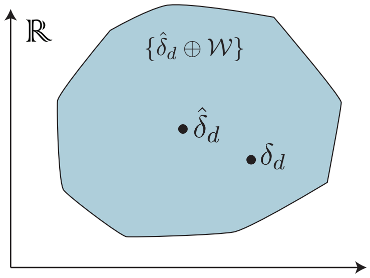


Figure 3.5: A drawing of an uncertain interval set $\hat{\delta}_d \oplus \mathcal{W}$ centered on $\hat{\delta}_d$ where the actual value δ_d lies somewhere in the set.

The feedback equation for the driver model is compactly written as,

$$\hat{\delta}_d = \mathbf{F}x(t) + \mathbf{G}\Delta\psi_{\text{road}} + w(t), \tag{3.21}$$

where $w(t) \in \mathcal{W}$, and $\mathbf{F}$ and $\mathbf{G}$ are defined by the particular vehicle model employed.

3.4 Stochastic Driver Model

Another useful method to incorporate uncertainty in the driver prediction is to capture the uncertainty as a probability distribution, measured from real data. In the literature accurate driver models often predict a distribution over future driver inputs [86, 93, 45, 5]. By explicitly considering this distribution, we can convert the uncertainty in the behavior of the driver to probabilistic constraint satisfaction in the model predictive controller.

We use the value of $\hat{\delta}_d$ obtained in (3.17) as a linear state-dependent estimate of the driver's steering input. The actual value of δ_d is assumed to lie in a normal distribution centered at $\hat{\delta}_d$. Then,

$$\delta_d \sim \mathcal{N}(\hat{\delta}_d, \Sigma), \tag{3.22}$$

where Σ is the covariance and we will denote the stochastic driver input as $w(t)$ and the mean-value $\hat{\delta}_d(t)$ as $\bar{w}(t)$. The feedback equation for the driver model is compactly written as,

$$u(t) = F\xi(t) + G\Delta\psi_{\text{road}} + w(t) + v(t) \tag{3.23}$$

where $w(t) \sim \mathcal{N}(\bar{w}(t), \Sigma)$, $F = [K_y,\, 0,\, K_\psi,\, 0] \in \mathbb{R}^{1\times 4}$ and $G = [K_\psi] \in \mathbb{R}$. Clearly, $u(t) = \hat{\delta}_d(t) + w(t) + v(t)$ where $v(t)$ has been introduced as an exogenous input signal and will be determined by a robust control law.

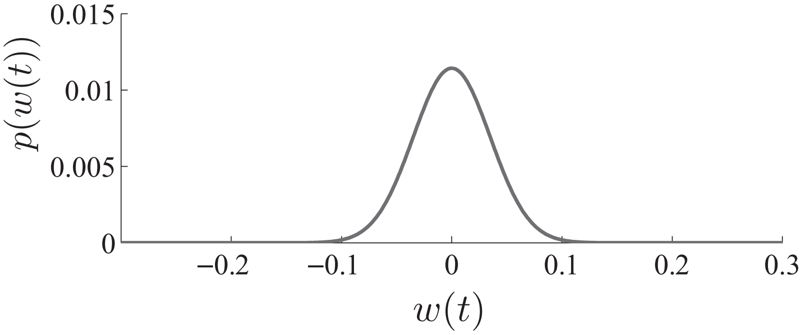


Figure 3.6: A normal distribution over the uncertain driver input w at time t given $\hat{\delta}_d = 0$ and $\Sigma = 0.035$ where $w(t) \sim \mathcal{N}(\hat{\delta}_d, \Sigma)$.

3.5 Driver-in-the-Loop Vehicle Model

Consider the Linear Vehicle Model of §2.5 and the Uncertain Driver Model of §3.3 and the Stochastic Driver Model of §3.4. Both sections §3.3 and §3.4 presented methods to model the uncertainty of the driver's inputs, set-based and probabilistically, respectively. In this section we will call the uncertainty $w(t)$ and this will refer to either approach.

We write the vehicle dynamics of Equation (2.41) in closed-loop with the uncertain driver model as

$$\dot{x}(t) = \mathbf{A}_{dm}\, x(t) + \mathbf{B}\, v(t) + \mathbf{E}_{dm} p(t) + \mathbf{B} w(t) \tag{3.24}$$

where $\mathbf{A}_{dm} = (\mathbf{A} + \mathbf{BF}) \in \mathbb{R}^{4\times4}$ is the closed-loop system matrix, $\mathbf{E}_{dm} \in \mathbb{R}^{4\times2}$ is the augmented parameter matrix where $\mathbf{E}_{dm} = [\mathbf{E} \;\; \mathbf{BG}]$, $p(t) = [\dot{\psi}_{\text{road}} \;\; \Delta\psi_{\text{road}}]^T \in \mathbb{R}^{2\times1}$, and $w(t)$ is the additive disturbance vector. By propagating the state according Equation (3.24) a prediction that incorporates both the vehicle dynamics and the driver's behavior is obtained. The signal $v(t)$ has been introduced as an exogenous input. The closed-loop driver controlled vehicle model presented here is shown in block diagram form in Figure 3.7.

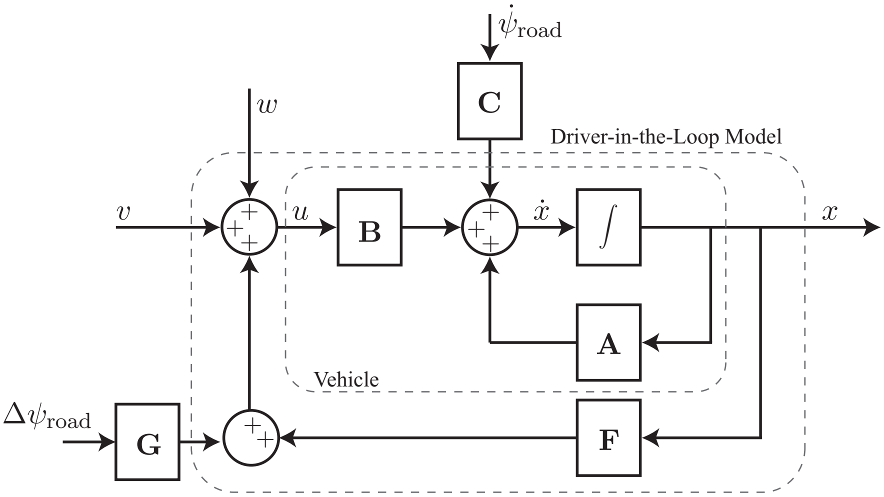


Figure 3.7: Block diagram illustrating the structure of the simulated system. The actual plant is not shown.

Chapter 4

Driver Assistance Systems

In this chapter a brief overview of existing Advanced Driver Assistance Systems (ADAS) will be introduced. ADAS are designed to essentially augment the sensing capabilities of the driver and provide more information than a human would otherwise be able to obtain. In addition, ADAS can process and interpret data at a much higher frequency than a human driver, which is especially important at high speeds. Recently, many automobile manufacturers have started implementing such systems. Examples include, but are not limited to, systems such as in-vehicle navigation systems with GPS (global positioning system) and TMC (traffic message channel) to provide up-to-date traffic information, ACC (adaptive cruise control), lane departure warning system, lane change assistance, collision avoidance system, intelligent speed adaptation, night vision, adaptive light control, pedestrian protection system, automatic parking, traffic sign recognition, blind spot detection, driver drowsiness detection, vehicular communication systems, hill descent control, etc. Although these systems are very different they share a common goal: to increase traffic safety. ADAS can typically be classified into two categories, Reaction Systems and Warning Systems [82], where reaction systems apply a control to keep the driver safe and warning systems simply warn the driver if the system predicts a threat to the driver's safety. An example of a reaction system is a CAS (collision avoidance system) and an example of a warning system is a LDWS (lane departure warning system). The effectiveness of such systems rely heavily on the sensors which enable them. Typically these sensors involve video sensors, radar, lidar, and infrared sensors. See Figure 4.1 for an example of a sensor enabled ADAS.

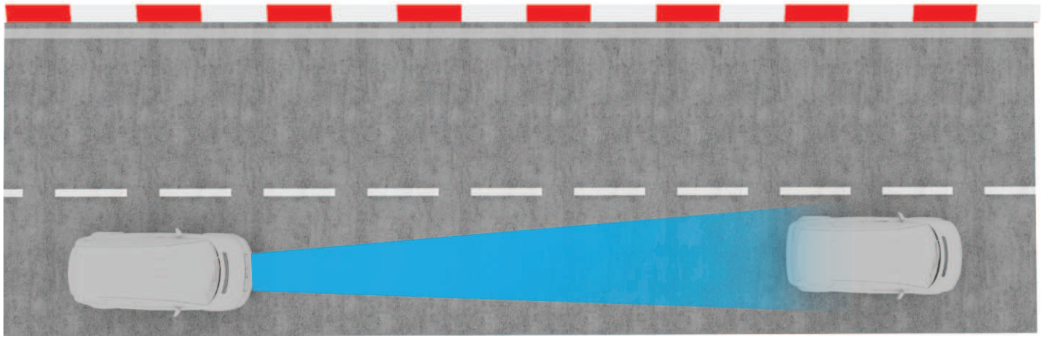


Figure 4.1: This figures shows a sketch of a front mounted sensor, e.g. a radar, providing range and range rate to the preceding vehicle. These measurements can be used to implement an ACC (adaptive cruise control) system that will maintain a constant distance or time gap to the preceding vehicle.

Two important safety systems are Electronic Stability Control (ESC) systems and Lane Keeping Systems (LKS), both of which will be considered in this thesis. ESC, or sometimes referred to as dynamic stability control, is a technology that improves the safety of a vehicle by detecting and reducing the loss of traction. When the vehicle loses traction and steering control, for example during a cornering maneuver, the ESC will brake the vehicle's wheels independently in order to recover stability and steer the vehicle to where the driver intends to go. Figure 4.2 and 4.3 show examples of an understeered and oversteered vehicle, respectively. An ESC system would be able to detect and control the vehicle to avoid such situations. It has been estimated that one-third of fatal accidents could have been prevented by such technology [85, 40].

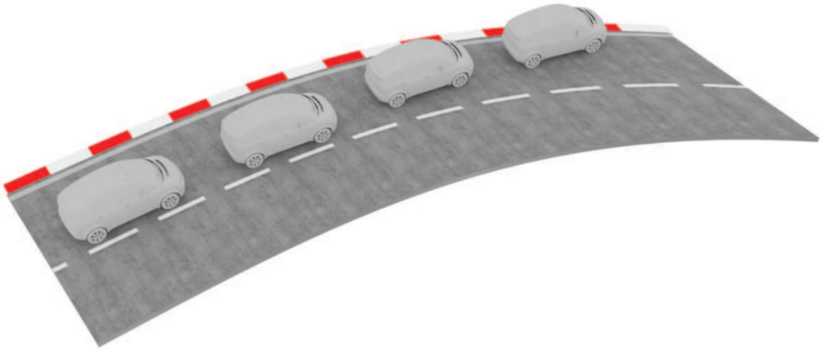


Figure 4.2: A depiction of a car during a maneuver that is understeered: the car does not turn enough and leaves the road.

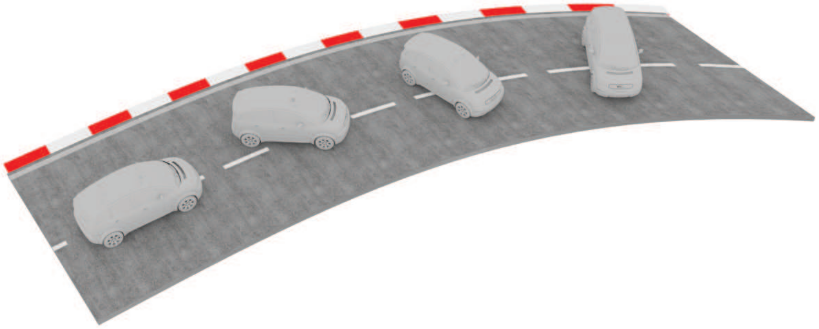


Figure 4.3: A depiction of a car during a maneuver that is oversteered: the car turns more sharply than intended and could go into a spin.

Figure 4.4 shows a plot where the slip angles of the front and rear tires have been constrained to lie within the linear operating region of the tire forces, as presented in §2.2, and thus avoid under- or oversteering. The framework to achieve this will be presented in Chapter 7.

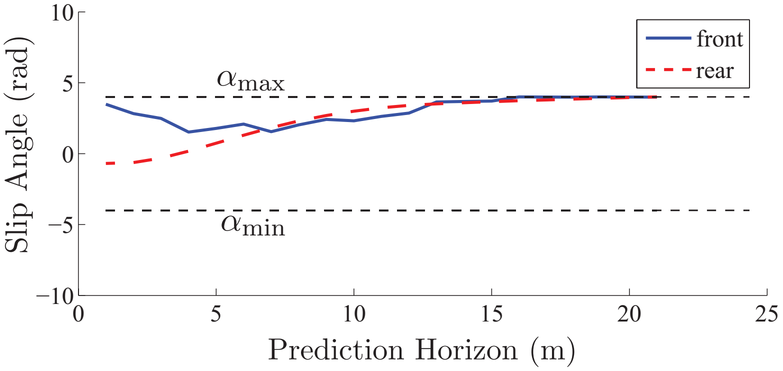


Figure 4.4: An electronic stability control system can constrain the tire slip angles to lie within the linear operating region of the tire forces and thus be easily maneuverable by a driver.

A Lane Keeping System is designed to detect when the vehicle is about to depart the roadway, and if no action is taken by the driver, automatically take steps to ensure the vehicle stays in its lane.

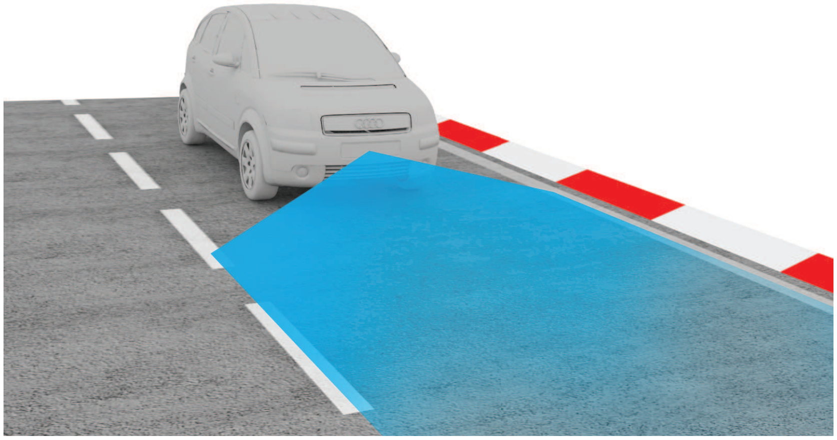


Figure 4.5: A sketch of a vehicle with a lane detection system running to support safety systems such as Lane Keeping Systems.

The ability to implement systems such as ESC and LKS is made easier by the parallel development of sensors, such as those developed by Mobileye [1], in order to detect the lanes using an affordable optical camera system.

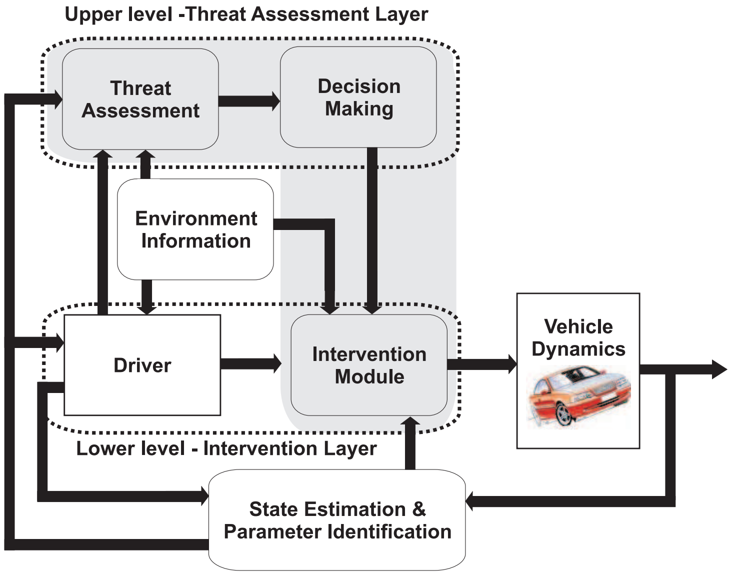

Upper level -Threat Assessment Layer
Threat Assessment
Decision Making
Environment Information
Driver
Intervention Module
Vehicle Dynamics
Lower level - Intervention Layer
State Estimation & Parameter Identification

Figure 4.6: Safety system architecture

A vehicle safety specific DAS is devoted to the task of deploying automated interventions only in safety critical situations. Typically this type of safety system is modular. Figure 4.6 shows an example of how such an architecture can appear.

The environment information and state estimation and parameter identification blocks provide various signals, or estimated signals, that the safety system needs in order to operate. This can include vehicle states such as velocity and yaw rate, or external information such as road curvature, or friction coefficient estimation. In general not every needed signal is available and estimation techniques must be employed in order to recover the desired information. A wide range of estimation problems are treated in [23, 41, 18, 61, 24, 25].

The threat assessment module deals with the task of determining whether interventions are necessary and plays an important role in the interaction with the driver. The threat assessment module repeatedly evaluates the driver's ability in maintaining safety in each situation and this information is used by the decision making module in order to decide whether and how to assist the driver. It is a challenge for an active safety system to properly assess when to intervene. In the literature, a large variety of threat assessment and decision making approaches can be found [22, 78, 62, 29]. In the simplest approaches, used in production vehicles, automated steering or braking interventions are issued when simple measures like the *time to collision* [22] or *time to line crossing* [78] pass certain thresholds.

The overall aim of the framework presented in this thesis is to aid in the design of an Advanced Driver Assistance System in which the system is more informed and can make safer, more coordinated, and smarter use of the roadway. A sketch of the semi-autonomous ADAS is shown in Figure 4.7.

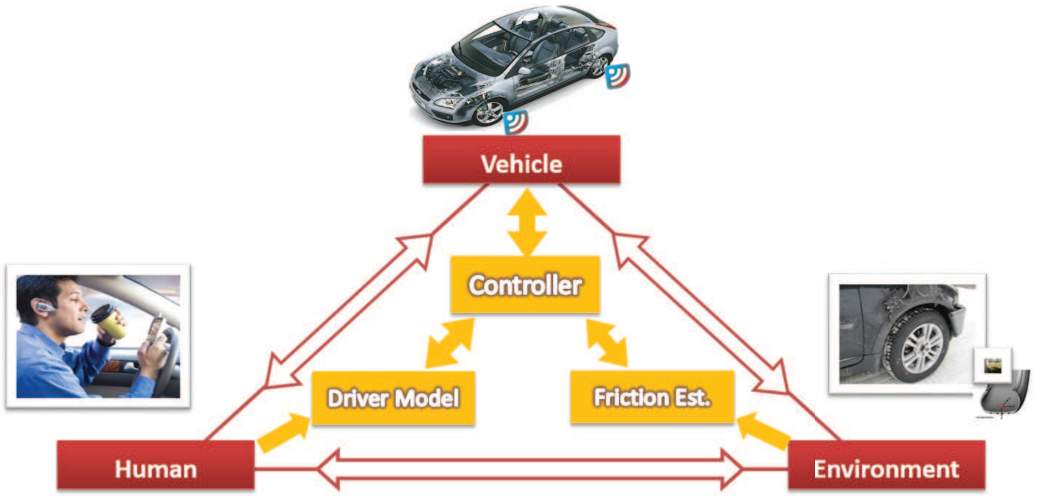


Figure 4.7: A sketch of the structure of the ADAS designed in this thesis: the overall goal is to be more informed in order to make safer, smarter, and more coordinated decisions about the safety of the driver.

The focus of this thesis is on the control design where the controller is informed by the Human and Environment modules of Figure 4.7. Simple driver models have been presented in Chapter 3. In general, an accurate description of the driver's behavior requires complex models accounting for a large amount of exogenous signals [19, 98, 76]. The Environment module can provide information such as location of obstacles, pedestrians and road boundaries, as well as provide parameter identification such as the real-time estimation of the friction coefficient. See [59, 58, 26, 27] for recent work on online friction estimation. We abandon the modular architecture in favor of a unified approach where the planning, threat assessment, and control of the vehicle is combined into a single optimization problem. The formulation of this optimization problem is presented in Chapter 5. Various planning methods are explored in Chapter 6 and the unified safety framework is presented in Chapter 7.

Chapter 5

Model Predictive Control

In this chapter we present the Model Predictive Control methodology. Because of its capability to systematically handle system nonlinearities and constraints, work in a wide operating region and close to the set of admissible states and inputs, Model Predictive Control (MPC) has been shown to be an attractive method for solving the path planning and control problems we are considering in this thesis [35, 32]. MPC is a control technique where the current input is found by solving a Constrained Finite Time Optimal Control Problem (CFTOC). At each sampling instant, starting at the current state of the system, an open-loop optimal control problem is solved over a finite horizon. The input is applied to the plant only during the following sampling interval $[t,\ t+1]$. At the next time step, $t+1$, a new optimal control problem based on new measurements of the updated state is solved over the shifted horizon. MPC has been widely used in industrial process and control [90] because of its ability to cope with hard constraints on controls and states where high efficiency is achieved at operating points at the boundary of the sets of admissable states and controls. The reader is referred to the following papers for an overview of the basics of MPC theory [74, 80, 81, 83].

In this chapter we introduce the MPC problem formulation. This is used in the remaining chapters to formulate the threat assessment, path planning, and control problems. Section 5.2 introduces the linear MPC theory and some stability results. Further, in section 5.3, this theory will be used to formulate a Robust MPC problem where constraint satisfaction can be guaranteed even in the presence of uncertainties, such as the uncertain driver model introduced in section 3.3. Section 5.1 introduces the general formulation for the Nonlinear MPC problem.

5.1 Nonlinear Model Predictive Control

Consider the various dynamic systems presented in Chapter 2. Without loss of generality we discretize the dynamics with a fixed sampling time T_s to obtain,

$$\xi(t+1) = f^d(\xi(t), u(t)). \tag{5.1}$$

where $f^d(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, with $f^d(\cdot,\cdot) \in \mathcal{C}^1$, defines the state update function and $\xi \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the control input. Note the superscript $(\cdot)^d$ emphasizes the *discrete* dynamics, as opposed to the *continuous* dynamics presented in Chapter 2. We drop the superscript for simplicity for the remainder of this section. The origin of the state space is an equilibrium point , i.e. $f(0,0) = 0$. The system (5.1) is subject to the following state and input constraints,

$$\xi(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U}, \quad \forall t \geq 0 \tag{5.2}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ are polyhedra [13]. Assume that a full measurement or estimate of the state $\xi(t)$ is available at the current time t. Then, the finite time optimal control problem $\mathbb{P}_{NL}$ to be solved at each time step,

$$\begin{aligned}
\mathbb{P}_{NL} : \qquad & \min_{U_t, \Xi_t} \; J_N(\Xi(t), U(t)) && \text{(5.3a)}\\
& s.t. \; \xi_{k+1,t} = f(\xi_{k,t}, u_{k,t}), \quad k = t, ..., N-1, && \text{(5.3b)}\\
& \xi_{k,t} \in \mathcal{X}, \quad k = t+1, ...t+N-1, && \text{(5.3c)}\\
& u_{k,t} \in \mathcal{U}, \quad k = t, ..., t+N-1, && \text{(5.3d)}\\
& \xi_{t,t} = \xi(t), && \text{(5.3e)}\\
& \xi_{t+N,t} \in \mathcal{X}_f, && \text{(5.3f)}
\end{aligned}$$

is solved at time t, where $\xi_{k,t}$ denotes the predicted state at time $t+k$ obtained by applying the control sequence $U_t = \{u_{t,t}, \ldots, u_{t+k,t}\}$ to the system (5.1) with $\xi_{t,t} = \xi(t)$. $N \in$ [1]$\mathbb{Z}^+$ denotes the prediction horizon. $\Xi_t = \{\xi_{t,t}, \ldots, \xi_{t+k,t}\}$ is the state trajectory obtained by applying the control

[1] $\mathbb{Z}^+$ is the set of positive integers

sequence U_t to the system (5.1) starting from the initial state $\xi(t)$. Equation (5.3f) is the terminal constraint and $\mathcal{X}_f$ is a polytope. The cost function $J_N(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^{Nm} \to \mathbb{R}^+$ is generally defined as,

$$J_N(\Xi(t), U(t)) = \sum_{k=t}^{N-1} l(\xi(k), u(k)) + P(\xi(t+N)), \tag{5.4}$$

where $l(\cdot,\cdot) \in \mathcal{C}^1$ and $l(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+$ is the stage cost and $P(\cdot) : \mathbb{R}^n \to \mathbb{R}^+$ is the terminal cost.

Let $\Phi(t;\xi,U_t)$ denote the solution of system (5.1)-(5.2) at time t controlled by U_t when $\xi(0) = \xi$. Denote by $U_t^* = \{u_{t,t}^*, ..., u_{t+N-1,t}^*\}$ the optimal solution of problem $\mathbb{P}_N L$ of equations (5.3) at time t and $J_N^*(\cdot,\cdot)$ the optimal value funcion. Then, the optimal state trajectory is $\Phi^*(t;\xi,U_t^*) = \{\xi_{t,t}^*, ..., \xi_{t+N-1,t}^*\}$, which we write as Ξ_t^* for simplicity. The first sample of U_t^* is applied to the system (5.1),

$$u(t) = u_{t,t}^*(\xi(t)). \tag{5.5}$$

At the next sampling time the optimization problem $\mathbb{P}_{NL}$ is solved over the shifted horizon based on the new state $\xi(t+1)$.

The problem $\mathbb{P}_{NL}$ is a nonlinear and, in general, non-convex optimization problem with $2N$ optimization variables, nN nonlinear equality constraints (constraints (5.3b)), and a number of linear constraints (constraints (5.3c)-(5.3d)) depending on the polytopes $\mathcal{X}$ and $\mathcal{U}$ [28]. The control law (5.3)-(5.5) is referred to as Non-Linear Model Predictive Control (NLMPC). Let $\kappa_t(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ denote the NLMPC law that associates the optimal input $u_{t,t}^*$ with the current state $\xi(t)$, $\kappa_t(\xi(t)) = u_{t,t}^*$. Then, the closed-loop system obtained by solving $\mathbb{P}_{NL}$ at time t with the NLMPC law (5.3)-(5.5)

$$\xi(k+1) = f(\xi(k), \kappa_t(\xi(k))) \overset{\triangle}{=} f_{cl}(\xi(k), k), \quad k \geq 0. \tag{5.6}$$

Remark 5 *Note the difference in notation used to distinguish between the input $u(t+k)^*$ applied to the plant at time $t+k$, and the optimizer $u_{t+k,t}^*$ of the problem $\mathbb{P}_{NL}$ at time $t+k$ obtained by solving* (5.3) *at time t with $\xi_{t,t} = \xi(t)$.*

5.2 Linear Time Varying Model Predictive Control

In this section the concept of Linear Time Varying Model Predictive Control (LTVMPC) is introduced. This is a suboptimal MPC algorithm that has a lower complexity than the general Nonlinear MPC law of problem $\mathbb{P}_N L$ in Equations (5.3)-(5.3). At each time step the model in (5.1) is approximated by a linear time varying model. In this section we will follow closely the notation of [38]. Further references regarding LTVMPC can be found in [21, 65, 67, 69, 100]. We define the system

$$\hat{\xi}_0(k+1) = f(\hat{\xi}_0(k), u(k)), \tag{5.7a}$$

$$u(k) = u_0, \tag{5.7b}$$

$$\hat{\xi}_0(0) = \xi_0 \tag{5.7c}$$

where $\xi_0 \in \mathcal{X}$ and $u_0 \in \mathcal{U}$. Denote by $\hat{\xi}_0(k)$, $\forall k \geq 0$ the state trajectory obtained by applying the input sequence $u(k) = u_0$, $\forall k \geq 0$ to the system (5.7) with $\hat{\xi}_0(0) = \xi_0$. The system (5.1) can be approximated by the following LTV system,

$$\delta\xi(k+1) = A_{k,0}\delta\xi(k) + B_{k,0}\delta u(k), \tag{5.8}$$

where $A_{k,0} \in \mathbb{R}^{n\times n}$ and $B_{k,0} \in \mathbb{R}^{n\times m}$ are defined as,

$$A_{k,0} = \frac{\partial f}{\partial \xi}|_{\hat{\xi}_0(k),u_0}, \qquad B_{k,0} = \frac{\partial f}{\partial u}|_{\hat{\xi}_0(k),u_0}, \tag{5.9a}$$

$$\delta\xi(k) = \xi(k) - \hat{\xi}_0(k), \qquad \delta u(k) = u(k) - u_0, \tag{5.9b}$$

The system (5.8) can be rewritten as,

$$\xi(k+1) = A_{k,0}\xi(k) + B_{k,0}u(k) + d_{k,0}(k), \tag{5.10}$$

where $d_{k,0}(k) = \hat{\xi}_0(k+1) - A_{k,0}\xi(k) - B_{k,0}u(k)$ for $k \geq 0$.

The LTV system of Equation (5.8) describes the deviation of the nonlinear system (5.1) from the state trajectory $\hat{\xi}_0(k)$ when a constant sequence u_0 is applied. In other words, system (5.8) and (5.10) are identical first order approximations of the system (5.1) around the nominal state trajectory $\hat{\xi}_0(k),\ k \geq 0$.

Consider the LTV Model in Equation (5.10) and the cost function in (5.4). Assume that a full measurement or estimate of the state $\xi(t)$ is available at the current time t. Then, the finite time optimal control problem $\mathbb{P}_{LTV}$ to be solved at each time step,

$$\begin{aligned}
\mathbb{P}_{LTV}: \qquad & \min_{U_t, \Xi_t} \ J_N(\Xi(t), U(t)) && (5.11a)\\
& s.t. \ \xi_{k+1,t} = A_{k,t}\xi_{k,t} + B_{k,t}u_{k,t} + d_{k,t}, && (5.11b)\\
& \qquad\qquad k = t, ..., N-1, && (5.11c)\\
& \xi_{k,t} \in \mathcal{X}, \quad k = t+1, ...t+N-1, && (5.11d)\\
& u_{k,t} \in \mathcal{U}, \quad k = t, ..., t+N-1, && (5.11e)\\
& \xi_{t,t} = \xi(t), && (5.11f)\\
& \xi_{t+N,t} \in \mathcal{X}_f, && (5.11g)
\end{aligned}$$

is solved at time t, where $\xi_{k,t}$ denotes the predicted state at time $t+k$ obtained by applying the control sequence $U_t = \{u_{t,t}, \ldots, u_{t+k,t}\}$ to the system (5.11b) with $\xi_{t,t} = \xi(t)$. $N \in \mathbb{Z}^+$ denotes the prediction horizon. $\Xi_t = \{\xi_{t,t}, \ldots, \xi_{t+k,t}\}$ is the state trajectory obtained by applying the control sequence U_t to the system (5.11b) starting from the initial state $\xi_{t,t} = \xi(t)$. Equation (5.11g) is the terminal constraint and $\mathcal{X}_f$ is a polytope. The matrices $A_{k,t}$, $B_{k,t}$, and the vector $d_{k,t}$ are defined in (5.9) and (5.10). Note the fixed index 0 has been replaced by t. The solution to problem $\mathbb{P}_{LTV}$ is $U_t^* = \{u^*_{t,t}, \ldots, u^*_{t+k,t}\}$ and the first sample is applied to the system,

$$u(\xi(t)) = u^*_{t,t}. \qquad (5.12)$$

Remark 6 *The cost funcion* (5.11a) *is convex piecewise linear or quadratic, the constraints* (5.11b)-(5.11g) *are linear, therefore the optimization problem* (5.11) *is convex. It can be solved with efficient Linear Programming (LP)* *or Quadratic Programming (QP) solvers, if the functions $l(\xi, u)$ and $P(\xi)$ in* (5.4) *are linear or quadratic, respectively.*

Note that although the computational complexity of LTVMPC in $\mathbb{P}_{LTV}$ is greatly reduced compared the general nonlinear formulation of problem $\mathbb{P}_{NL}$, the formulation in Equations (5.11) requires N linearizations of the model (5.1), which can be expensive for higher order models with long prediction horizons.

Assumption 13 *To reduce the complexity of the LTVMPC law in* (5.11)-(5.12) *we assume $A_{k,t} = A_t$ and $B_{k,t} = B_t$ for $k = t, ...t + N - 1$. i.e. the model is held constant over the prediction horizon.*

Stability is not guaranteed in Model Predictive Control and the reader is referred to the survey paper [81] for a thorough discussion on stability and optimality and the thesis [38] for a stability analysis of the LTVMPC.

5.3 Robust Model Predictive Control

The formulation for Nonlinear Model Predictive control and Linear Time Varying Model Predictive Control was presented in §5.1 and §5.2, respectively. The formulations in problems $\mathbb{P}_{NL}$ and $\mathbb{P}_{LTV}$ have been studied extensively and have strong theoretical results regarding stability and feasibility. However, they do not address the scenario when the predicted system evolution differs from the actual system behavior. Robustness against uncertainties is an important discussion for practical applications as uncertainty can arise from modeling errors, unknown or neglected system dynamics, or exogenous disturbances. In addition, explicitly modeling an uncertainty can provide robustness guarantees against known or bounded disturbances if a robust approach is adopted.

Although employing a Model Predictive Control approach introduces feedback into the system, and therefore an inherent degree of robustness over an open-loop system, an analysis of the robustness for a given MPC control law is difficult [9]. The first appearance of Robust Model Predictive Control was proposed in [20] where a worst-case objective function was considered, i.e. the worst realization of the uncertainty. This approach is commonly known in the literature as Min-Max Model Predictive Control. In this method closed-loop predictions are obtained that contain the spread of trajectories resulting from the influence of uncertainty. This, however, leads to controllers of comparably high computational complexity. In this thesis two methods are proposed to incorporate robustness in the MPC problem to handle uncertain driver models. Section 5.3.1 handles the uncertainty using set-based methods and section 5.3.2 converts the uncertainty in the driver model to probabilistic constraints in the optimization problem.

5.3.1 Set-based Robust MPC

A method, proposed in [73] and adopted in this thesis, is the Tube-Based Robust MPC. This form of Robust MPC builds upon set invariance theory and its ideas can be traced back to [101, 11, 51]. The basic idea is to essentially tighten the constraints of the original nominal system while bounding the error between the nominal and the uncertain system state by a robust positively invariant set. This will allow the uncertain system to evolve in a set centered on the nominal system state trajectory while the original constraints are satisfied at all times, even in the presence of uncertainty. In this section we first introduce set invariance theory in §5.3.1 and provide all the required assumptions and definitions. Then, in §5.3.1 the Robust MPC control law is proposed that is later studied in the simulations of §7.4.1.

Background on Set Invariance Theory

In this section several definitions are provided that will be important in developing the Robust MPC later in this thesis. We follow the notation used in [64].

Denote by f_a the constrained, discrete time linear autonomous system perturbed by a bounded, additive disturbance. The system dynamics are

$$x(t+1) = f_a(x(t), w(t)) = \mathbf{A}x(t) + w(t) \tag{5.13}$$

where $x(t)$ and $w(t)$ denote the state and the disturbance vectors. System (5.13) is subject to the constraints

$$x(t) \in \mathcal{X} \subseteq \mathbb{R}^n, \quad w(t) \in \mathcal{W} \subseteq \mathbb{R}^d, \tag{5.14}$$

where $\mathcal{X}$ and $\mathcal{W}$ are polyhedra that contain the origin in their interiors. For the autonomous system (5.13)-(5.14),

Definition 1 *(Reachable set for autonomous systems) we define the one-step robust reachable set for initial states x contained in the set $\mathcal{S}$ as*

$$\begin{aligned} &\text{Reach}_{f_a}(\mathcal{S}, \mathcal{W}) \triangleq \{x \in \mathbb{R}^n | \\ &\quad \exists\, x(0) \in \mathcal{S},\ \exists\, w \in \mathcal{W} : x = f_a(x(0), w)\}. \end{aligned} \tag{5.15}$$

For the nominal system, i.e., with $w(t) = 0$, $\forall t$, the one-step reachable set is *defined as*

$$\mathrm{Reach}_{f_a}(\mathcal{S}) \overset{\triangle}{=} \{x \in \mathbb{R}^n | \exists\ x(0) \in \mathcal{S} : x = f_a(x(0))\}. \tag{5.16}$$

Similarly, for systems with inputs

$$x(t+1) = f(x(t), u(t), w(t)) = \mathbf{A}x(t) + \mathbf{B}u(t) + w(t), \tag{5.17}$$

subject to the constraints

$$x(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U} \subseteq \mathbb{R}^m, \quad w(t) \in \mathcal{W}, \tag{5.18}$$

the one-step robust reachable set is defined as:

Definition 2 *(Reachable set for systems with external inputs) the one-step robust reachable set for initial states x contained in the set $\mathcal{S}$ is*

$$\begin{gathered}\mathrm{Reach}_{f}(\mathcal{S}, \mathcal{W}) \overset{\triangle}{=} \{x \in \mathbb{R}^n | \\ \exists\ x(0) \in \mathcal{S},\ \exists\ u \in \mathcal{U},\ \exists\ w \in \mathcal{W} : x = f_a(x(0), u, w)\}.\end{gathered} \tag{5.19}$$

Therefore, all states contained in $\mathcal{S}$ are mapped into the reach 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.* Robust invariant sets are computed for the autonomous system (5.13)-(5.14). We define the robust invariant set as follows:

Definition 3 *(Robust Positive Invariant Set) A set $\mathcal{Z} \subseteq \mathcal{X}$ is said to be a robust invariant set for the autonomous system* (5.13) *subject to the constraints in* (5.14)*, if*

$$x(0) \in \mathcal{Z} \ \Rightarrow\ x(t) \in \mathcal{Z}, \quad \forall w(t) \in \mathcal{W}, \quad \forall t \in \mathbb{N}^+ \tag{5.20}$$

Definition 4 *(Maximal Robust Positive Invariant Set) The set $\mathcal{Z}_\infty \subseteq \mathcal{X}$ is the maximal robust invariant set for the autonomous system* (5.13) *subject to the constraints in* (5.14)*, if $\mathcal{Z}_\infty$ is a robust invariant set and $\mathcal{Z}_\infty$ contains all positive invariant sets contained in $\mathcal{X}$ that contain the origin.*

Two important operations for the discussion to follow are the Pontryagin difference and the Monkowski sum. The Pontryagin difference of two polytopes $\mathcal{P}$ and $\mathcal{Q}$ is a polytope

$$\mathcal{P} \ominus \mathcal{Q} := \{x \in \mathbb{R}^n \ : \ x + q \in \mathcal{P}, \ \forall q \in \mathcal{Q}\}, \tag{5.21}$$

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

$$\mathcal{P} \oplus \mathcal{Q} := \{x + q \in \mathbb{R}^n \ : \ x \in \mathcal{P}, \ q \in \mathcal{Q}\}. \tag{5.22}$$

Robust Control Law

In this section we introduce the framework used to develop the Robust Active Safety controller in §7.5. We follow a notation similar to [8]. The control problem is divided into two components: (1) a feedforward control input computed for the nominal system and (2) a linear state feedback controller that acts on the error between the actual state and the predicted nominal state. We denote the control sequence and the disturbance sequence as $\mathbf{u} = \{u_0, u_1, ..., u_{N-1}\}$ and $\mathbf{w} = \{w_0, w_1, ..., w_{N-1}\}$ for system (5.17)-(5.18) for $t = 0...N-1$. Let $\Phi(t; x, \mathbf{u}, \mathbf{w})$ denote the solution of (5.17) at time t controlled by $\mathbf{u}$ when $x(0) = x$. Furthermore, let $\bar{\Phi}(t, x, \bar{\mathbf{u}})$ denote the solution of the *nominal* system

$$\bar{x}(t+1) = \mathbf{A}\bar{x}(t) + \mathbf{B}\bar{u}(t) \tag{5.23}$$

at time t controlled by the nominal control sequence $\bar{\mathbf{u}} = \{\bar{u}_0, \bar{u}_1, ..., \bar{u}_{N-1}\}$ when $x(0) = x$. Denote the predicted nominal state trajectory by $\bar{\mathbf{x}} = \{\bar{x}_0, \bar{x}_1, ..., \bar{x}_{N-1}\}$. We write the controller as

$$v(t) = \bar{u}(t) + K(x(t) - \bar{x}(t)) \tag{5.24}$$

where $\bar{u}(t)$ is the feedforward component for the nominal system and $K(x(t) - \bar{x}(t))$ is the linear state feedback component acting on the error between the actual state and the predicted nominal state. We make use of the following assumption,

Assumption 14 *(Stabilizing Disturbance Rejection Controller) The linear state feedback gain $K \in \mathbb{R}^{m \times n}$ in* (5.24) *is chosen such that $A_K = A + BK$ is Hurwitz. Note this can always be satisfied if the pair (A, B) is controllable.*

Using the above definitions we can formulate the following Proposition. The details can be found in [79].

Proposition 1 *Suppose that assumption 14 is satisfied and that $\mathcal{Z}$ is a robust positively invariant set (definition 3) for the perturbed system* (5.17)-(5.18) *with control law* (5.24). *If $x \in \{\bar{\mathbf{x}}\} \oplus \mathcal{Z}$, then $x(t+1) \in \{\bar{\mathbf{x}}(t+1)\} \oplus \mathcal{Z}$ for all admissible disturbance sequences $w(t) \in \mathcal{W}$.*

This proposition states that if the control law (5.24) is used it will keep the states $x(t) = \Phi(t; x, \mathbf{u}, \mathbf{w})$ of the uncertain system (5.17) within the robust positive invariant set $\mathcal{Z}$ centered on the predicted state trajectory $\bar{\Phi}(t, x, \bar{\mathbf{u}})$ of the nominal system (5.23) for all admissible disturbance sequences $\mathbf{w}$.

$$x(0) \in \{\bar{x}_0\} \oplus \mathcal{Z} \;\Rightarrow\; x(t) \in \{\bar{x}_t\} \oplus \mathcal{Z} \;\; \forall w(t) \in \mathcal{W}, \;\forall t \geq 0, \tag{5.25}$$

where $x(0)$ and $\bar{x}_0$ are the initial states of (5.17) and (5.23). Proposition 1 suggests that if the optimal control problem for the *nominal* system (5.23) is solved for the tightened constraints

$$\bar{\mathcal{X}} = \mathcal{X} \ominus \mathcal{Z}, \quad \bar{\mathcal{U}} = \mathcal{U} \ominus K\mathcal{Z}, \tag{5.26}$$

then the use of the control law (5.24) will ensure persistent contraint satisfaction for the controlled *uncertain* system (5.17) [8].

Robust Invariant Set Computation

The robust analysis is done off-line and the notion of robust invariant sets is important for the discussion to follow.

Stabilizing Controller K Recall control law (5.24). The choice of the stabilizing state feedback gain matrix K will determine the size of the robust invariant set. See [8] for a thorough discussion on this topic. In this paper we choose K as the optimal infinite horizon LQR solution K^{∞}_{LQR}. Then,

$$v(t) = \bar{u}(t) + K^{\infty}_{\text{LQR}}(x(t) - \bar{x}(t)) \tag{5.27}$$

Defining this stabilizing controller allows us to compute the robust positive invariant set $\mathcal{Z}$ needed to calculate the tightened constraints $\bar{\mathcal{X}}$ and $\bar{\mathcal{U}}$ defined in (5.26). Note that $\mathcal{Z}$ is dependent upon the choice of K.

Robust Positive Invariant Set $\mathcal{Z}$ The robust invariant set $\mathcal{Z}$ is used to determine the tightened constraints for the nominal system. The following algorithm provides a procedure for computing $\mathcal{Z}$ for system (3.24)-(5.27). An initial set of states $\mathcal{X}_0 \subset \mathcal{Z}$ is chosen and the bounded disurbance $\mathcal{W}$ is defined. Algorithm 1 will calculate the reachable set $\mathcal{Z}$ (definition 3) if it converges in a finite number of steps. For the problem to be well-posed we make the assumption that the tightened constraints $\bar{\mathcal{X}}$ and $\bar{\mathcal{U}}$ exist and contain the origin. For this assumption to hold it is required that $\mathcal{W}$ is sufficiently small. Clearly there is a design trade-off between disturbance rejection properties (large K) and the size of the tightened constraints.

Algorithm 1 Computation of $\mathcal{Z}$

```
Require: $f_a$, $\mathcal{X}_0$, $\mathcal{W}$
Ensure: $\mathcal{Z}$
1: Let $\Omega_0 = \mathcal{X}_0$
2: Let $\Omega_{k+1} = \text{Reach}_{f_a}(\Omega_k, \mathcal{W}) \cup \Omega_k$
3: if $\Omega_{k+1} = \Omega_k$ then
4:     $\mathcal{Z} \leftarrow \Omega_{k+1}$
5: else
6:     GOTO 2.
7: end if
```

Set-based MPC Problem Formulation

Once the robust positively invariant set has been calculated and the contstraints have been sufficiently tightened, the optimization problem to be

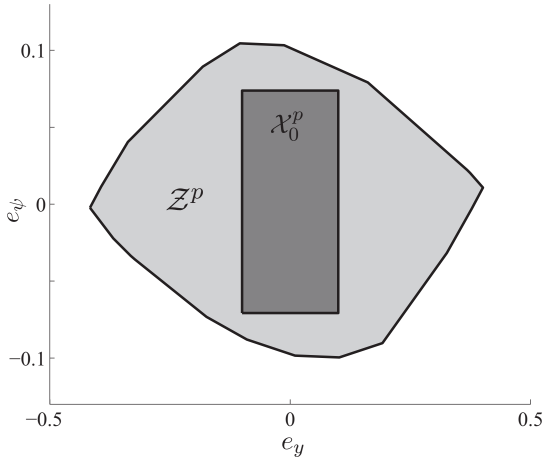


Figure 5.1: The initial set of states $\mathcal{X}_0^p$ and the final robust positive invariant set $\mathcal{Z}^p$ projected onto the e_y-e_ψ plane where $\mathcal{X}_0^p = \text{proj}_{e_y - e_\psi}(\mathcal{X}_0)$ and $\mathcal{Z}^p = \text{proj}_{e_y - e_\psi}(\mathcal{Z})$.

solved at each time step is no more computationally demanding that a standard MPC problem for the nominal system. The formulation is,

$$\begin{aligned}
\mathbb{P}_{ROB}: \quad & \min_{\bar{U},\varepsilon} \; J_N(\bar{U}(t), \Delta\bar{U}(t)) && (5.28a)\\
& s.t. \; \xi_{k+1,t} = f^{dm}(\xi_{k,t}, u_{k,t}), \quad k = t, ..., N-1, && (5.28b)\\
& \xi_{k,t} \in \bar{\mathcal{X}}, \quad k = t+1, ...t+N-1, && (5.28c)\\
& u_{k,t} \in \bar{\mathcal{U}}, \quad k = t, ..., t+N-1, && (5.28d)\\
& \xi_{t,t} = \xi(t), && (5.28e)\\
& \xi_{N,t} \in \bar{\mathcal{X}}_f, && (5.28f)
\end{aligned}$$

where the model $f^{dm}(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ in Equation (5.28b) is model (5.23) with control law (5.27), i.e. the *nominal* system dynamics. The constraints in Equations (5.28c)-(5.28f) are the tightened constraint sets,

$$\bar{\mathcal{X}} = \mathcal{X} \ominus \mathcal{Z}, \quad \bar{\mathcal{U}} = \mathcal{U} \ominus K\mathcal{Z}. \tag{5.29}$$

Then, the corrective steering action calculated by problem $\mathbb{P}_{ROB}$ for the nominal system (5.23) and tightened constraints (5.29) will ensure persistent constraint satisfaction for the controlled uncertain system.

5.3.2 Stochastic Robust MPC

In this section we incorporate the stochastic driver model of §3.4 and provide probabilistic robust guarantees of constraint satisfaction in the presence of the driver's uncertain behavior. In the literature accurate driver models often predict a distribution over future driver inputs [86, 93, 45, 5]. By explicitly considering this distribution, we convert the uncertainty in the behavior of the driver to probabilistic constraint satisfaction in the model predictive controller. The uncertainty in the driver model is handled at the design stage by the computation of an upper bound on the disturbance propagation. By tightening the constraints by this amount we can ensure constraint satisfaction of the original system, within a given probability, for the stochastic system.

Stochastic MPC Problem Formulation

A Model Predictive Controller is implemented to constrain e_y within the lane using minimal control effort. By formulating the problem this way, the driver assumes full control of the vehicle until a lane departure is predicted, at which point the controller adds the minimal corrective action to keep the driver safe. Since we are only concerned with keeping the states of the vehicle within the lane constraint and do not perform tracking, only the control input is penalized in the cost function J. Because of the stochastic nature of the driver disturbance, J is taken as the expected value, $\mathbb{E}\{\cdot\}$, of control inputs. We discretize the dynamics of Equation (2.41) with a fixed sampling time T_s to obtain,

$$\xi(k+1) = A^d_{dm}\xi(k) + B^d v(k) + E^d_{dm}p(k) + D^d w(k), \tag{5.30}$$

where the $(\cdot)^d$ denotes the discrete-time dynamics. The optimization problem is formulated as follows,

$$
\begin{aligned}
\mathbb{P}_{PROB}: \qquad \min_{\mathbf{v_t}} \quad & J = \mathbb{E}\{\sum_{k=t}^{t+N-1} ||u_{k,t}||_2^2\} && \text{(5.31a)}\\
\text{s.t. } & \xi_{k+1,t} = A^d_{dm}\xi_{k,t} + B^d v_{k,t} + E^d_{dm} p_{k,t} + D^d w_{k,t} && \text{(5.31b)}\\
& |v_{k,t}| \leq v_{\max} && \text{(5.31c)}\\
& w_{k,t} \sim \mathcal{N}(\bar{w}_{k,t}, \Sigma) && \text{(5.31d)}\\
& \Pr\{g^T \xi_{k+1,t} \leq h\} \geq p && \text{(5.31e)}\\
& \xi_{t,t} = \xi(t) && \text{(5.31f)}
\end{aligned}
$$

where t denotes the current time instant and $\xi_{k,t}$ denotes the predicted state at time $t+k$ obtained by applying the control sequence $\mathbf{v_t} = \{v_{t,t},\ v_{t+1,t}, ...,\ v_{t+N-1,t}\}$ to the system (5.30) with $\xi_{t,t} = \xi(t)$. N denotes the prediction horizon. Equation (5.31b) is the discretized dynamics in Equation (5.30). Equation (5.31c) is the constraint on the controller, which corresponds to the physical limit of the active front steering actuator. Equation (5.31d) assigns a normal distribution to the driver disturbance (3.22). Equation (5.31e) is the probabilistic constraints imposed on e_y, and equation (5.31f) is the state feedback at the start of the prediction horizon. Since the lateral position e_y is dependent on the driver's steering action, which is stochastic, we can only find a solution to the problem that satisfies the constraint with probability p. We can specify the probabilistic constraint (5.31e) as,

$$
g^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix},\ h = \begin{bmatrix} e_{y,\max} \\ e_{y,\min} \end{bmatrix}
$$

Closed-Loop Paradigm In order to handle the stochastic nature of (5.31), the problem is reformulated and a new optimization variable, $c_{k,t}$, representing the perturbation of the control input about a linear feedback law, is added. Using the closed-loop paradigm in [71], (5.31b) can be decomposed into,

$$\xi_{k,t} = z_{k,t} + e_{k,t} \tag{5.32a}$$

$$v_{k,t} = K\xi_{k,t} + c_{k,t} \tag{5.32b}$$

$$z_{k+1,t} = \Phi z_{k,t} + B^d c_{k,t} \tag{5.32c}$$

$$e_{k+1,t} = \Phi e_{k,t} + D^d w_{k,t} \tag{5.32d}$$

where $\Phi = A^d_{dm} + B^d K$ is the closed-loop system matrix with state-feedback controller K. We make the following assumption.

Assumption 15 $\Phi = A^d_{dm} + B^d K$ *is Hurwitz. Note this assumption can always be satisfied if* (A^d_{dm}, B^d) *is controllable.*

Remark 7 *We choose the state-feedback stabilizing controller K to be the optimal infinite horizon LQR gain matrix.*

Remark 8 *In the discussion here and to follow we set the road parameters* $p(t) = 0,\ \forall t$ *for visual clarity.*

By this decomposition the states $\xi_{k,t}$, which are random variables, are now decomposed into a deterministic component $z_{k,t}$ and a stochastic component $e_{k,t}$, where z and e are driven by c and w, respectively.

Probabilistic Constraints Given the probability distribution on the predicted driver inputs (3.22), we can formulate the requirements that the vehicle stays within the lane and in a stable operating region as probabilistic constraints (5.31e). We will convert the probabilistic constraints to linear constraints by using information on the disturbances.

Lane Departure Constraints The constraints on lateral lane position e_y is expressed by (5.31e). We will omit the second subscript t in the later discussion since t does not change in each optimization. We will also drop the superscript $(\cdot)^d$ for visual clarity. From (5.32), and with $e_t = 0$, we have,

$$z_{t+i} = \Phi^i z_t + H_i \mathbf{c}_t \tag{5.33a}$$

$$e_{t+i} = \Phi^{i-1} D w_t + \Phi^{i-2} D w_{t+1} + ... + D w_{t+i-1} \tag{5.33b}$$

$$i = 1, 2, ..., N-1$$

where $H_i = [\Phi^{i-1}B \quad \Phi^{i-2}B \quad ... \quad B \quad 0 \quad 0 ... 0]$, and $\mathbf{c}_t^T = [c_t^T \;\; c_{t+1}^T \;\; ... \;\; c_{t+N-1}^T]$. From (5.32a) we have $g^T\xi_{t+i} = g^T z_{t+i} + g^T e_{t+i}$, thus the constraint $\Pr\{g^T\xi_{t+i} \leq h\} \geq p$ is equal to,

$$\Pr\{g^T H_i \mathbf{c}_t + g^T \Phi^i z_t \leq h - g^T e_{t+i}\} \geq p \tag{5.34}$$

which in turn equals to,

$$g^T H_i \mathbf{c}_t + g^T \Phi^i z_t \leq h - \gamma_i \tag{5.35a}$$

$$\Pr\{g^T(\Phi^{i-1} D w_t + \Phi^{i-2} D w_{t+1} + ... + D w_{t+i-1}) \leq \gamma_i\} = p \tag{5.35b}$$

$$i = 1, 2, ..., N$$

Bounds on γ_i are implied by the following result [70].

$$\gamma_i \leq \sqrt{\frac{p}{1-p} \times g^T P_i g} \tag{5.36}$$

where $P_1 = D\mathbb{E}(w_t w_t^T)D^T$ and $P_{i+1} = \Phi P_i \Phi^T + D\mathbb{E}(w_{t+i} w_{t+i}^T)D^T, i = 1, 2....$ Result of (5.36) holds for arbitrary distributions and is a direct product of Chebyshev's one-sided inequality. Here, since we assumed the disturbance has a Gaussian distribution, we can estimate the value of γ_i more accurately.

Note 5 *The bound in* (5.36) *is overly-conservative. By assuming the disturbance has a Gaussian distribution, the values of γ_i can be estimated more accurately by discretizing the distributions of w_{t+i} and then performing discrete convolutions to get the estimated distribution of the sum $g^T(\Phi^{i-1} D w_t + \Phi^{i-2} D w_{t+1} + ... + D w_{t+i-1})$. Since γ_i does not depend on the states ξ, it can* *be calculated off-line (if the disturbance w_i can be known off-line).*

Once γ_i is calculated, the constraints (5.31e) are ready to be converted to linear constraints:

$$g^T \begin{bmatrix} B & 0 & 0 & ... & 0 & 0 \\ \Phi B & B & 0 & ... & 0 & 0 \\ \Phi^2 B & \Phi B & B & ... & 0 & 0 \\ : & : & : & ... & : & : \\ : & : & : & ... & : & : \\ \Phi^{N-2}B & \Phi^{N-3}B & \Phi^{N-4}B & ... & B & 0 \\ \Phi^{N-1}B & \Phi^{N-2}B & \Phi^{N-3}B & ... & \Phi B & B \end{bmatrix} \mathbf{c_t^T} \tag{5.37}$$

$$+g^T \begin{bmatrix} \Phi \\ \Phi^2 \\ \Phi^3 \\ : \\ : \\ \Phi^{N-1} \\ \Phi^N \end{bmatrix} z_k \leq \begin{bmatrix} h-\gamma_1 \\ h-\gamma_2 \\ h-\gamma_3 \\ : \\ : \\ h-\gamma_{N-1} \\ h-\gamma_N \end{bmatrix},$$

Formulating an Optimization Problem with the Batch Method To solve the optimization problem posed in (5.31), the batch method [14] is used to eliminate constraint (5.31b) before solving the problem. By recursively substituting the state update, (5.32) can be re-written in terms of the state feedback $\xi(t)$, the optimization vector $\mathbf{c_t^T} = [c_t^T \; c_{t+1}^T \; ... \; c_{t+N-1}^T]$, and the disturbance vector $\mathbf{w_t^T} = [w_t^T \; w_{t+1}^T \; ... \; w_{t+N-1}^T]$ over the prediction horizon. Taking equation (5.33),

$$z_{t+i,t} = \Phi^{i-1}B_d c_{t,t} + ... + B_d c_{t+i-1,t} + \Phi^i z_{t,t} \tag{5.38a}$$

$$e_{t+i,t} = \Phi^{i-1}D_d w_{t,t} + ... + D_d w_{t+i-1,t} \tag{5.38b}$$

the i-th predicted state, can be expressed as,

$$\xi_{t+i} = \Phi^i \xi(t) + H_i B \mathbf{c_t} + H_i D \mathbf{w_t} \tag{5.39}$$

where $H_i = [\Phi^{i-1} \,...\, \Phi^0 \; 0 \,...\, 0]$. By combining (5.39) and (5.32b), the control input vector $\mathbf{v_t^T} = [v_t^T \; v_{t+1}^T \,...\, v_{t+N-1}^T]$ can be expressed as,

$$\mathbf{v_t^T} = K\mathbf{H}_\xi \xi(t) + (K\mathbf{H_c} + I)\mathbf{c_t} + K\mathbf{H_w}\mathbf{w_t} \tag{5.40}$$

$$\mathbf{H}_\xi = \begin{bmatrix} I \\ \Phi \\ \vdots \\ \Phi^{N-1} \end{bmatrix}$$

$$\mathbf{H_c} = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ B & 0 & \cdots & \cdots & 0 \\ \Phi B & B & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \Phi^{N-2}B & \Phi^{N-3}B & \cdots & B & 0 \end{bmatrix}$$

$$\mathbf{H_w} = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ D & 0 & \cdots & \cdots & 0 \\ \Phi D & D & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \Phi^{N-2}D & \Phi^{N-3}D & \cdots & D & 0 \end{bmatrix}$$

Substituting (5.40) into cost function (5.31a) and removing terms which do not involve $\mathbf{c_t}$, a new cost function $\tilde{J}$ is defined,

$$\begin{aligned} \tilde{J} &= \mathbb{E}\{\mathbf{c_t^T P c_t} + 2\xi(t)^T\mathbf{Q c_t} + 2\mathbf{w_t^T R c_t}\} \qquad (5.41) \\ \mathbf{P} &= (K\mathbf{H_c} + I)^T(K\mathbf{H_c} + I) \\ \mathbf{Q} &= (K\mathbf{H}_\xi)^T(K\mathbf{H_c} + I) \\ \mathbf{R} &= (K\mathbf{H_w})^T(K\mathbf{H_c} + I) \end{aligned}$$

Now the optimization problem defined in (5.31) is completely specified as a Quadratic Program, which can be easily handled by solvers such as CVX.

Chapter 6

Hierarchical Model Predictive Control

The capability of handling constraints in a systematic way makes MPC a very attractive control technique, especially for applications where the process is required to work in wide operating regions and close to the boundary of the set of admissible states and inputs. Parallel advances in theory and computing systems have enlarged the range of applications where real-time MPC can be applied [15, 17, 66, 39, 103]. Yet, for a wide class of *fast* applications, the computational burden of Nonlinear MPC (NMPC), presented in §5.1, is still a barrier. As an example, in [16] 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 [34, 32] a Linear Time Varying MPC approach is presented to tackle the same problem. Experimental results [32, 37, 31, 35] demonstrated the capability of the controller to stabilize the vehicle at higher speeds, up to 72kph, in a double lane change maneuver on a slippery (snow covered) surface.

In all the aforementioned literature, obstacle avoidance is not explicitly considered in the control design. The work presented in this thesis continues to investigate the application of MPC techniques in the situation where a given trajectory fails to avoid one or more pop-up obstacles. The given trajectory represents the initial driver intent and the MPC has to safely avoid the obstacle while trying to track the desired trajectory. Previous work by the authors [46] have accounted for pop-up obstacles by decomposing the problem into the two-level NMPC problem depicted in Figure 6.1. The high-level "Path Planner" uses a simplified point-mass vehicle model to generate an obstacle avoiding trajectory by using a NMPC controller (§5.1). The trajectory is fed to the low-level "Path Follower" designed by using a NMPC based on a higher fidelity vehicle model (§2.1) [33]. In [46] the proposed hierarchical framework has been implemented on an autonomous ground vehicle driving at high speeds on an icy road.

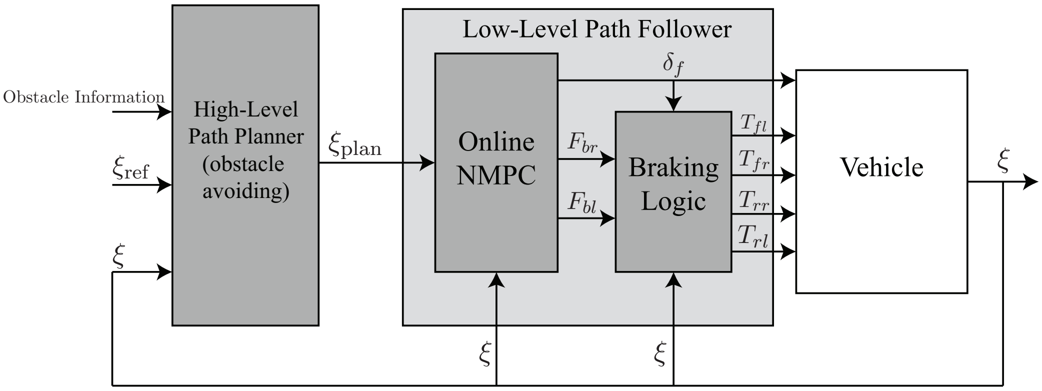


Figure 6.1: Architecture of the two-level MPC. The *motion primitive* or *spatial model* is used for high-level path planning. The four-wheel vehicle model (§2.1) is used for low-level path tracking.

Although the decomposition in Figure 6.1 allows for real-time implementation, the trajectories generated by the path planner are not always feasible when using a low-fidelity vehicle model, such as the point mass, in [46]. The lower level tracking performance deteriorates and obstacle collisions can be observed in conditions where the obstacle could have been avoided.

Two different solutions to this problem will be presented. In Section 6.1, in order to overcome this issue and still maintain real-time feasibility, we *study the use of a motion primitive path planner*. Path planners based on motion primitives for dynamical systems have been first introduced in [42]. There, the author considered two types of primitives: trims and maneuvers. Trims are steady-state equilibrium trajectories and maneuvers are pre-specified trajectories connecting trims, designed offline, and stored in a library. The method shifts the complexity of nonlinear dynamical optimization to the sequencing of "useful" trims and maneuvers in order to generate a feasible path. If the number of motion primitives is small, we expect the combinatorial nature of the resulting problem to be easily handled by state-of-the-art mixed-integer solvers. Moreover, if the environment is highly structured (as in urban driving), one can expect to generate feasible trajectories for a variety of scenarios with only a few motion primitives.

The motion primitive approach has been used to successfully plan time-optimal paths for small robotic helicopters [42, 43] and other simple mechanical systems [84]. However, to the best of the authors' knowledge, this method has not yet been implemented on autonomous ground vehicles. The section is structured in two parts. In the first part we describe in detail the proposed hierarchical control architecture and the approach to generate a trajectory by using motion primitives. We will provide details on specific lane-change and drifting motion primitives which are very useful for designing a semi-autonomous vehicle with adjustable degree of autonomy. In the second part, simulative and experimental results with an autonomous and semi-autonomous ground vehicle traveling at high speed on an icy surface are presented [54].

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 an approach on current electronic control units if the motion library is large. In section 6.2, in order to plan a feasible path in real-time, we study the use of *spatial predictive control at the high level.* We follow the approach presented in [99, 63] 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 [47].

6.1 Motion Primitive Path Planner

In this section we introduce the motion primitive framework. We use the ideas presented in [42] and adapt the formalism to fit our application. Consider the Nonlinear Four Wheel Vehicle Model presented in §2.1. In this section a sliding model controller is used to control the braking torques T_i at each wheel. Model (2.2)-(2.3) in closed-loop with the slip ratio controller is compactly written as

$$\dot{\xi}(t) = f_{\sigma}^{4w}(\xi(t),\, u(t)), \tag{6.1}$$

where the input $u(t) = [\delta_f,\, \sigma_r]$ and δ_f is the front steering angle and σ_r is the desired rear slip ratio. The state $\xi = [\dot{x},\, \dot{y},\, \dot{\psi},\, X,\, Y,\, \psi] \in \mathbb{R}^6$ is grouped into two vectors $\xi = [\xi_g,\, \xi_b]$ where the vector $\xi_b = [\dot{x},\, \dot{y},\, \dot{\psi}]$ collects longitudinal and lateral velocities and yaw rate in the body-fixed frame $\mathcal{O}_b$ and $\xi_g = [X,\, Y,\, \psi]$ collects longitudinal and lateral coordinates and yaw angle in the inertial frame $\mathcal{O}_I$. The tire forces in Equation (6.1) are calculated using the Pacejka tire model of §2.2.1.

6.1.1 High-Level Motion Primitive Framework

Two types of motion primitives are defined: trims and maneuvers. A *trim* is a trajectory for system (6.1) with constant body-fixed frame states $\bar{\xi}_b^q = [\bar{\dot{x}}, \bar{\dot{y}}, \bar{\dot{\psi}}]$ and constant inputs $\bar{u}^q = [\bar{\delta}_f, \bar{\sigma}_r]$, i.e., $(\bar{\xi}_b^q, \bar{u}^q)$ is an equilibrium of the dynamical system (6.1). By changing initial conditions of the vehicle states in the global frame $\xi_g(t_0)$, a set of trim trajectories is generated. We denote this set of trim trajectories as Ξ_q where q is used to highlight the dependence of the trim trajectories to the single equilibrium point $(\bar{\xi}_b^q, \bar{u}^q)$. In summary, a trim $\xi^q(t)$ (or simply “a trim q”) is the solution to:

$$\dot{\xi^q}(t) = f_{\sigma}^{4w}([\xi_g(t)', \xi_b(t)']', \bar{u}^q),\ \forall t \geq t_0 \tag{6.2a}$$

$$\xi_b(t) = \bar{\xi}_b^q,\ \forall t \geq t_0 \tag{6.2b}$$

$$\xi_g(t_0) = \xi_{g,0} \tag{6.2c}$$

where the initial condition belongs to a compact set $\xi_{g,0} \in \mathcal{X} \times \mathcal{Y} \times \Psi$. A trim trajectory $q \in \Xi_q$ requires the specification of the initial conditions $\xi_g(t_0) = [X_0, Y_0, \psi_0]$ and does not have any constraint on its duration. The set of all trims is $\bigcup_{q \in Q} \Xi_q$. Table 6.1 shows a subset of the trims used later in the examples in Section 6.1.5.

The time spent in trim q will be denoted as τ_q and will be called the *coasting time*. Since the body-fixed velocity and inputs are constant in a trim motion, the Nonlinear Four Wheel Model (6.2) is easily simplified to predict the state of the vehicle at time $t_0 + \tau_q$ when a trim q is initiated at time t_0:

$$\xi_g(t^+) = \xi_g(t_0) + \mathbf{R}(\psi)\, \bar{\xi}_{\mathbf{b}}^{\mathbf{q}} \tau_{\mathbf{q}} \tag{6.3a}$$

$$\xi_b(t^+) = \bar{\xi}_b^q \tag{6.3b}$$

$$q(t^+) = q(t_0) \tag{6.3c}$$

$$t^+ = t_0 + \tau_q \tag{6.3d}$$

where $\mathbf{R}(\psi)$ is the rotation matrix for coordinate transformation:

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

The table of trims used in the example considered here are presented in Table 6.1.

Table 6.1: Table of Trims

trim	description	$\xi_b = [\dot{x}(m/s), \dot{y}(m/s), \dot{\psi}(r/s)]$	$[\delta_f(rad), \sigma_r]$
q_1	straight	$[11, 0, 0]$	$[0, 0.0015]$
q_2	left turn	$[11, 0.02, 0.27]$	$[0.09, 0.0015]$
q_3	right turn	$[11, -0.02, -0.27]$	$[-0.09, 0.0015]$
q_4	drift left	$[8, -1.09, 0.50]$	$[0, 0.1]$

A *maneuver* is a finite time trajectory which is used to transition between two trims. A maneuver p is defined by its starting and ending trim, q_{from} and q_{to}, respectively, a fixed duration τ_p, a fixed displacement in the global coordinates $\Delta\xi_g^p = [\Delta X, \Delta Y, \Delta\psi]$ and the corresponding input profile $\tilde{u}^p : [t_0,\ t_0 + \tau_p] \rightarrow U$ where U is the set of feasible inputs.

A maneuver is executed by applying to the vehicle the input $\tilde{u}^p(t)$ for t in the interval $t_0 \leq t \leq t_0 + \tau_p$. If the vehicle is in trim q_{from} at time t_0, then by executing the maneuver p, the end state at time $t_0 + \tau_p$ is

$$\xi_g(t^+) = \xi_g(t_0) + \Delta\xi_g^p \tag{6.5a}$$

$$\xi_b(t^+) = \bar{\xi}_b^{q_{\text{to}}} \tag{6.5b}$$

$$q(t^+) = q_{\text{to}} \tag{6.5c}$$

$$t^+ = t_0 + \tau_p \tag{6.5d}$$

Transitions between trims are only permitted through the execution of a maneuver. We will use the set P to collect the index p of all stored maneuvers. Table 6.2 shows the maneuvers designed to transition between the trims in Table 6.1. A maneuver $p \in P$ is defined by defining control sequence $\tilde{u}^p$ and is a solution to,

$$\dot{\xi^p}(t) = f_\sigma^{4w}([\xi_g(t)', \xi_b(t)']', \tilde{u}^p),\ \forall t \in [t_0\ \tau_p] \tag{6.6a}$$

$$\xi_b(t_0) = \bar{\xi}_b^{q_{\text{from}}} \tag{6.6b}$$

$$\xi_g(t_0) = \xi_{g,0} \tag{6.6c}$$

Table 6.2: Table of Maneuvers

maneuver	description	q_{from}	q_{to}	$\Delta\xi_g = [\Delta X(m), \Delta Y(m), \Delta\psi(rad)]$	$\tau_p(sec.)$
p_1	straight to left turn	1	2	$[16.58, 1.61, 0.28]$	1.5
p_2	left turn to straight	2	1	$[5.53, 0.08, 0.02]$	0.5
p_3	straight to right turn	1	3	$[16.58, -1.61, -0.28]$	1.5
p_4	right turn to straight	3	1	$[5.53, -0.08, -0.02]$	0.5
p_5	straight to drift	1	4	$[17.19, 3.59, 0.77]$	2.0
p_6	drift to straight	4	1	$[17.19, 3.59, 0.77]$	2.0
p_7	right lane-change 1	1	1	$[39.67, -6.16, 0]$	3.7
p_8	right lane-change 2	1	1	$[55.14, -6.09, 0]$	5.2
p_9	left lane-change	1	1	$[87.93, 6.00, 0]$	8.1
p_{10}	parameterized lane-change	1	1	$(a, L) \leftrightarrow \Delta\xi_g$	$\tau_{(a,L)}$

The resulting system can be reformulated as a hybrid system with the steady-state trims as the discrete states and a maneuver used as a control input to transition between two discrete states. Figure 6.2 shows the hybrid system obtained by using the trims and maneuvers described in Tables 6.1 and 6.2. With this model at hand, the motion planning problem is then formulated as a hybrid optimal control problem. In particular, finding an optimal path is reduced to finding a sequence of coasting times in trims and corresponding maneuvers that optimize a given cost function. More details can be found in [44] and [43].

6.1.2 Motion Primitives for Collision Avoidance

The trims and maneuvers described in Tables 6.1 and 6.2 have been derived by using their definitions and forward simulations of the vehicle model (6.1). The intent was to generate motion primitives useful for collision avoidance of a ground vehicle. Clearly the library of trims and maneuvers can be easily expanded.

The objective of this section is to provide details on two specific maneuvers in Table 6.2. First, we discuss maneuver p_{10} which is a parameterized lane change obtained by using clothoids. Later in this thesis we will show that this type of maneuver is very useful for obtaining a semi-autonomous

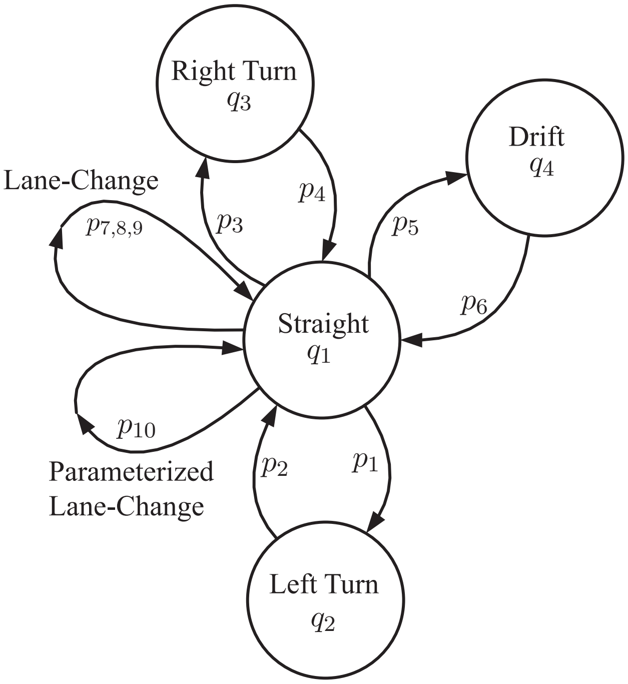


Figure 6.2: An example hybrid system with steady-state trims as discrete states and maneuvers execute transitions.

vehicle with an adjustable degree of autonomy. Second, we discuss the drift-straight-drift maneuvers p_5 and p_6. Drifting maneuvers are used to improve the vehicle agility. Moreover, at high speed on slippery surfaces it may be the only feasible maneuver to avoid an obstacle.

Parameterized Clothoids

Because of the structured environment of on-road driving, a common motion primitive to avoid an oncoming collision is a lane-change [55]. A lane change motion primitive is the sequence of a straight trim, a lane-change maneuver, and a straight trim. Maneuvers p_7, p_8, p_9 in Table 6.1 are classical lane change maneuvers.

Maneuver p_{10} is a parameterized lane change obtained by sequencing four clothoids. A clothoid is a curve with a curvature that changes linearly with respect to the curve length, see Figure 6.3. It is widely used in railway and high way engineering for curvature transition. Also, its use in road design and robot path planning has a long history, relevant references include [95, 92, 75]. 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 [95] as:

$$\begin{aligned} X &= \frac{1}{a}\int_0^{aL} \cos(\mathrm{s}^2)\mathrm{ds} \\ Y &= \frac{1}{a}\int_0^{aL} \sin(\mathrm{s}^2)\mathrm{ds} \end{aligned} \tag{6.7}$$

where X and Y are the global coordinates and L is the maximum curve length. In numerical computation schemes, the integrals in Equation (6.7) are approximated by their Taylor expansion:

$$\begin{aligned} \int_0^{aL} \cos(\mathrm{s}^2)\mathrm{ds} &= aL - \frac{(aL)^5}{5 \times 2!} + \frac{(aL)^9}{9 \times 4!} - \frac{(aL)^{13}}{13 \times 6!}... \\ \int_0^{aL} \sin(\mathrm{s}^2)\mathrm{ds} &= \frac{(aL)^3}{3} - \frac{(aL)^7}{7 \times 3!} + \frac{(aL)^{11}}{11 \times 5!} - \frac{(aL)^{15}}{15 \times 7!}... \end{aligned} \tag{6.8}$$

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 6.3.

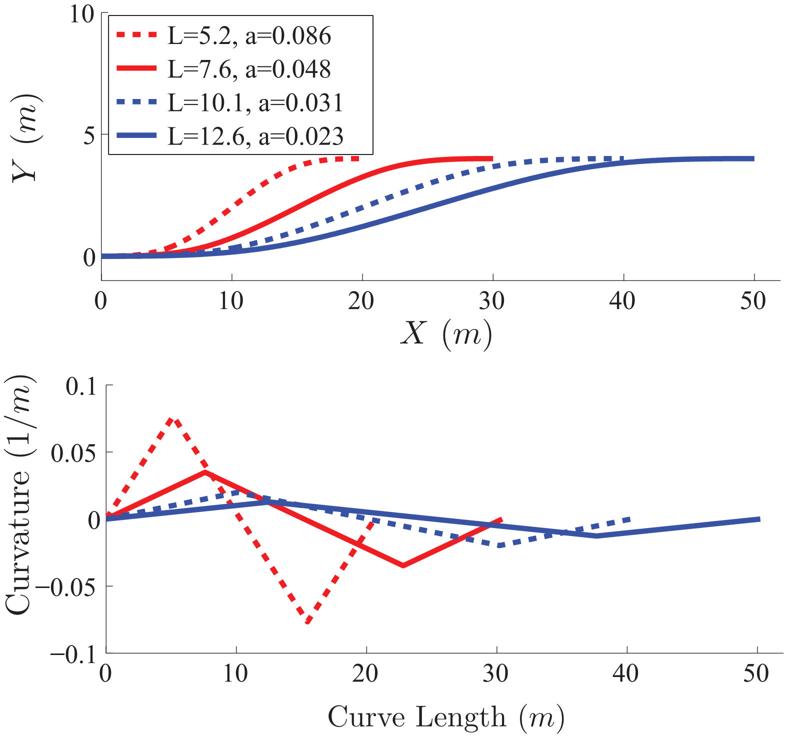


Figure 6.3: Lane change paths with different aggressiveness. Each is generated by connecting four pieces of clothoids. The upper figure shows the shapes of the paths in global frame. The lower figure shows the piecewise affine relation between curvature and curve length for each path.

In our approach, in order to avoid storing lane change maneuvers for different a and L, we do not pre-compute the maneuver p_{10}. Instead, we find conditions on a and L guaranteeing that the maneuver is feasible and let the path planner find the optimal and feasible a and L. The maximum curvature of the clothoid defined by Equation (6.7) 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 and therefore that the clothoid can be followed without tracking error.

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 Equation 6.1, 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,

$$2a^2L \;\leq\; \frac{0.46\mu g}{\dot{x}^2} \tag{6.9}$$

where μ is the friction coefficient and $\dot{x}$ is the body-fixed longitudinal velocity. Specifying a and L determines $\Delta\xi_g^{p_{10}}$ and the time to track the clothoid $\tau_{a,L}$ depends on the low-level MPC.

Drift Trim and Maneuver

Maneuvers p_5 and p_6 are designed to bring the vehicle into a steady-state drift. During a drifting trim the tires are operating outside their linear region, see, e.g., Figure 2.7, and the constant side slip angle of the vehicle is high. This is commonly achieved by saturating the rear tire forces.

Bringing the system to a steady-state drift is a maneuver often difficult to achieve since the drifting trim is an unstable equilibrium point of the system (6.1). We compute the equilibrium points as in [56] by solving a set of nonlinear equations, $\dot{\xi} = f_\sigma^{4w}(\xi_e, u_e) = 0$ for constant ξ_e and u_e. Once a drift trim has been computed, we compute the drifting maneuvers off-line by solving a sequence of nonlinear optimization problems to find a trajectory in the body-frame that constrains the desired steady-state point as a boundary condition. The cost in the nonlinear optimization problem is

$$J(\xi_b, u) = \min_U \sum_{k=1}^{N} \|\xi_{b,k} - \xi_{b,e,k}\|_Q^2 + \|u_k - u_{e,k}\|_R^2 \tag{6.10}$$

where $\xi_b = [\dot{x}, \dot{y}, \dot{\psi}]$ is the body-fixed state, $\xi_{b,k}$ is calculated from the discretized model in (6.1) with a fixed sampling time T_s, and Q and R are appropriate weighting matrices. The solution to this problem yields an input vector $U = \{\delta_f, \sigma_r\}_k,\ k = 1...N$ where δ_f is the front steering angle, σ_r is the desired slip ratio of the rear tires and N is the planning horizon. Computing offline drifting maneuvers allows for the path planner to quickly plan a feasible path which can be difficult to find in real-time with online nonlinear optimization.

In Figure 6.4 the vehicle approaches an obstacle at $\dot{x} = 11$ m/s on an icy surface with $\mu = 0.55$. In these conditions, the drifting maneuver may be the only safe maneuver to navigate the corner.

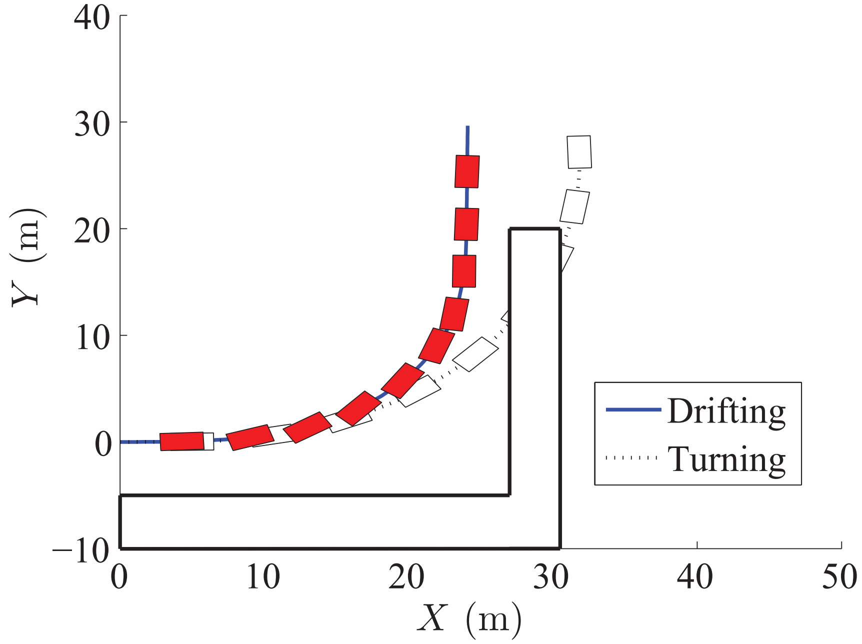


Figure 6.4: A drifting maneuver compared to an aggressive turn. The turn is the sharpest allowable maneuver within the constraint of the slip angles of the tires. At this initial vehicle state ($\xi_g = [0, 0, 0],\ \dot{x} = 11$m/s, q_1) the drifting maneuver is the only safe maneuver to avoid the obstacle, where the coefficient of friction $\mu = 0.55$.

6.1.3 Planning With Motion Primitives

Casting the trajectory planning problem in the motion primitive framework allows us to compute an optimal feasible path by solving a hybrid optimal control problem. It is our desire to track a reference trajectory while avoiding collisions with any obstacles. We use a cost function that is a weighted sum of the tracking error, distance to any obstacles, and the *aggressiveness* of the maneuver measured as a norm of the maximum tire slip angle.

Let the cost of a trim q be defined as:

$$C_q = K_r \int_{t_0}^{T_q} (\eta(t) - \eta_{ref}(t))^2 dt + K_o \frac{\dot{X}_{T_q}}{d(T_q) + \varepsilon} \tag{6.11}$$

where $\eta = \left[\dot{x}, \psi, \dot{\psi}, Y\right]$ and η_{ref} is the reference trajectory. K_r and K_o are weighting scalars. $T_q = t_0 + \tau_q$ where trim q is initiated at time t_0 and τ_q is the time spent in trim q. $\dot{X}_{T_q}$ is the vehicle speed in the inertial X-direction at time T_q, $d(T_q)$ is the distance to the obstacle at time T_q (see [46] for the explicit calculation of this term), and $\varepsilon \geq 0$ is a small constant. The cost of initiating a maneuver p is defined as:

$$C_p = K_r^m \int_{t_0}^{T_p} (\eta(t) - \eta_{ref}(t))^2 dt + K_p^m \|\alpha_{\max}\| \tag{6.12}$$

where K_r^m and K_p^m are weighting scalars. $T_p = t_0 + \tau_p$ where t_0 is the start time of the maneuver p and τ_p is the duration of maneuver p. $\alpha_{\max}$ is the maximum slip angle of the front tires during the maneuver. Higher weights K_p^m on $\alpha_{\max}$ favor paths that are less aggressive.

The problem of finding a sequence of maneuvers and trims minimizing the cost $C_p + C_q$ in (6.11) and (6.12) can be formulated as a dynamic program:

$$J(\xi_g, q) = \min_{\tau_q, p, (a,L)} \left[C_q + C_p + J(\xi_g^+, q^+)\right] \tag{6.13}$$

where ξ_g^+ and q^+ are the next vehicle states and trim obtained by taking maneuver p, as defined in Equations (6.3) and (6.5), respectively. In (6.13) $\xi_g = [X, Y, \psi]$, $(\xi_g(0), q(0)) = (\xi_{g,0}, q_0)$ and $J(\xi_{g,f}, q_f) = 0$. τ_q is the time spent in trim q and p is the maneuver executed. For the parameterized clothoid the minimization is also taken over parameters a and L. Equation (6.13) is a form of the Bellman equation. We solve (6.13) offline by discretizing the value function over a grid in the (X, Y, ψ, q) space and performing value iteration until the algorithm converges. The solution is an optimal value function from all hybrid states (ξ_g, q) to end state and trim $(\xi_{g,f}, q_f)$.

Operational Envelope

As stated in the previous section, the cost function of the optimization problem in equation (6.13) captures the aggressiveness of the curve as defined by maximum slip angle during a maneuver. This slip angle quantifies how close the vehicle is to the dynamic envelope defined by the linear region of Figure 2.7. This region is considered the *safe* driving region. The solution to the planning problem is a path $L = \{\tau_q,\ p\}_k$, where $k = 1, 2 \ldots N$ and N is the length of the planning horizon. The sequence of τ_q and p define the optimal time in trim and the optimal switching sequence. Then, the maximum slip angle of a given path is defined as

$$\alpha_{\max} = \max_{\alpha}\{L\}, \tag{6.14}$$

The maneuvers are designed to be within the linear region of the tire model, in Figure 2.7 where $|\alpha_{\max}| < |\alpha_{\text{linear}}|$.

6.1.4 Low-Level MPC Path Follower

The trajectory computed by the high-level motion-primitive path planner is sent to the low level path follower. A Model Predictive Controller (MPC) is used to accurately track the path and thus mitigate the effect of model mismatch and exogenous disturbances. The optimization problem to be solved at each time instant is formulated in Problem $\mathbb{P}_N L$ of Equations (5.3). The general cost function defined in Equation (3.7) is specified as,

$$
\begin{aligned}
&J_N(\Xi_t, U_t, \Delta U_t) = \\
&\sum_{k=t}^{t+N-1} \|\eta_{k,t} - \eta_{\text{plan}_{\text{k,t}}}\|_Q^2 + \|u_{k,t}\|_R^2 + \|\Delta u_{k,t}\|_S^2
\end{aligned} \tag{6.15}
$$

where the tracking reference is the output of the motion primitive planner and is the planned path:

$$
\eta_{\text{plan}_{\text{k,t}}} = [\dot{x}_{\text{plan}_{\text{k,t}}}, \psi_{\text{plan}_{\text{k,t}}}, \dot{\psi}_{\text{plan}_{\text{k,t}}}, Y_{\text{plan}_{\text{k,t}}}]' \tag{6.16}
$$

At each time step t, the performance index $J_N(\Xi_t, U_t, \Delta U_t)$ is optimized under the constraints specified in Equation (5.3) starting from the state $\xi_{t,t} = \xi(t)$ to obtain an optimal control sequence, $U_t^* = [u_{t,t}^{*}{}', ..., u_{t+N-1,t}^{*}{}']'$ and optimal state sequence $\Xi_t^* = [\xi_{t,t}^{*}{}', ..., \xi_{t+N-1,t}^{*}{}']'$.

Note 6 *A difference from problem $\mathbb{P}_{NL}$ in Equation (5.3) and the problem presented here is the addition of the optimization variables ΔU_t which has been added to constrain the change rate of the control inputs U_t. The constraints of Equation (5.3) should be modified to include,*

$$
u_{k,t} \in \Delta\mathcal{U},\ \ k = t, ..., t+N-1, \tag{6.17}
$$

6.1.5 Simulation And Experimental Results

Extensive simulations and experiments have been performed to test the proposed hierarchical control architecture composed of a high-level motion primitive planner and a low-level MPC path follower. This section has two objectives: first we will show in simulation the capability of the proposed scheme in generating aggressive feasible motion plans which include drifting in order to avoid lane departures. Then, we will show the benefit of using parameterized clothoids in semi-autonomus drive by using simulations and experiments conducted on a vehicle while driving high-speed on ice.

Agile Driving - Simulations

The simulations presented in this section show the utility of the motion primitive planner and its capability of generating aggressive feasible trajectories. The simulated scenario shows the vehicle approaching a right-angle at a velocity of $\dot{x} = 11$ m/s in trim q_1 on an icy surface with $\mu = 0.55$. Figure 6.5 shows the results of the motion primitive planner and the low-level tracking. In the simulation the target set is $X_f \in [32\ 37]$ and $Y_f \in [38\ 42]$. Using the nonlinear solver NPSOL [49] a feasible solution could not be found. Without considering the drifting primitives the lane departure is unavoidable with steady-state turning.

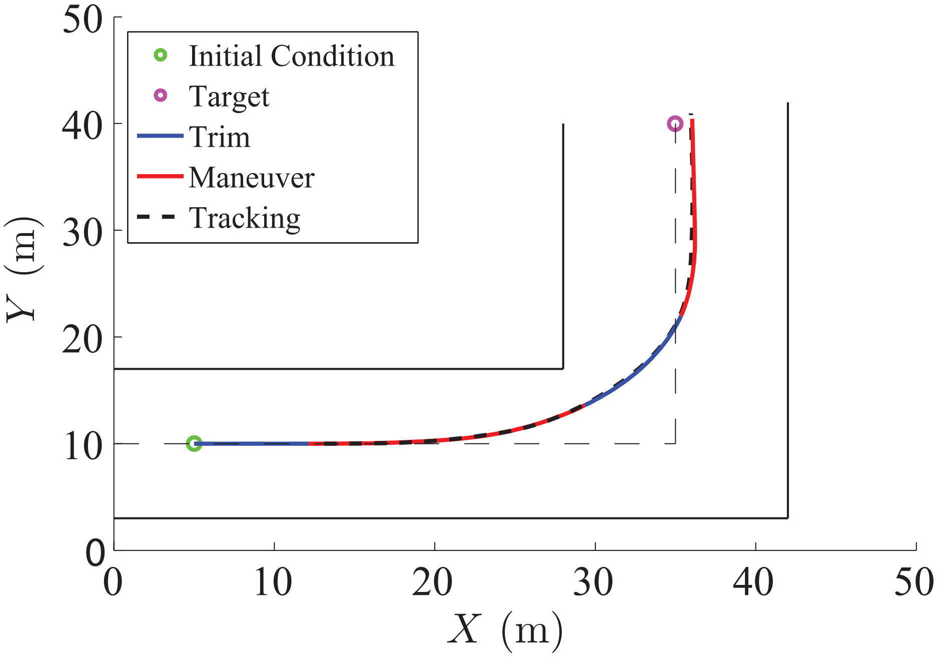


Figure 6.5: A successfull navigation of the corner planned with the drifting primitives. The vehicle starts in trim q_1 (straight travel, $\dot{x} = 11$m/s), takes maneuver p_5 to a drifting trim q_4, then takes maneuver p_6 back to q_1. The aggressive path is able to be tracked with no tracking error where the coefficient of friction $\mu = 0.55$.

Semi-Autonomous Drive - Simulations

The simulations presented in this section are based on single and double lane-change maneuvers to avoid an upcoming obstacle. The vehicle is initially tracking a reference trajectory down the center of the lane. The high-level path planner is generating trajectories at each time step based on the current vehicle position. The autonomous system does not take over from the driver until the aggressiveness of the planned path, defined by its maximum slip angle $\alpha_{\max}$, is greater than a specified threshold.

We use Matlab to simulate the closed-loop system. The MPC optimization problem has been implemented as a C-coded s-Function. The commercial NPSOL software package [49] is used for solving the nonlinear programming problem in Equation (5.3). The first element of the optimized control sequence is passed to an external block which uses the Four Wheel Model (§2.1) and Pacejka tire model (§2.2.1) to simulate the dynamics of the vehicle, and feeds the current state of the vehicle back to the controller.

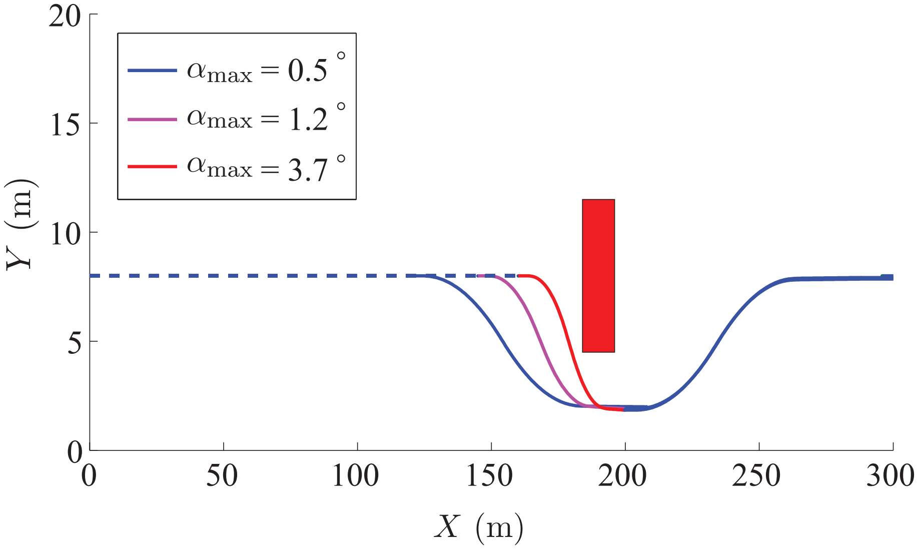


Figure 6.6: Various lane change maneuvers are compared. As the vehicle approaches the obstacle the planned paths become more aggressive (high slip angles).

Figure 6.6 shows three different lane change maneuvers. They are generated from the high-level path planner starting in a steady-state trim q_1 and $X_0 = 120,\ 140,$ and 160m. As the vehicle approachs the obstacle, the planned paths become more aggressive, that is, the maximum slip angle of the front tires approaches the stability limit for the vehicle. The maneuver with $\alpha_{\max} = 3.7°$ is the most aggressive maneuver the vehicle can handle at the given speed. The simulations presented in Figure 6.7 use a binary driver model. The driver is either attentive, as in the upper plot of Figure 6.7, 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 6.7, and the low-level control begins tracking the planned path with a lower degree of curvature. Because the planned path is generated using the higher fidelity Four Wheel Model (§2.1), both simulations show accurate tracking by the low-level MPC controller.

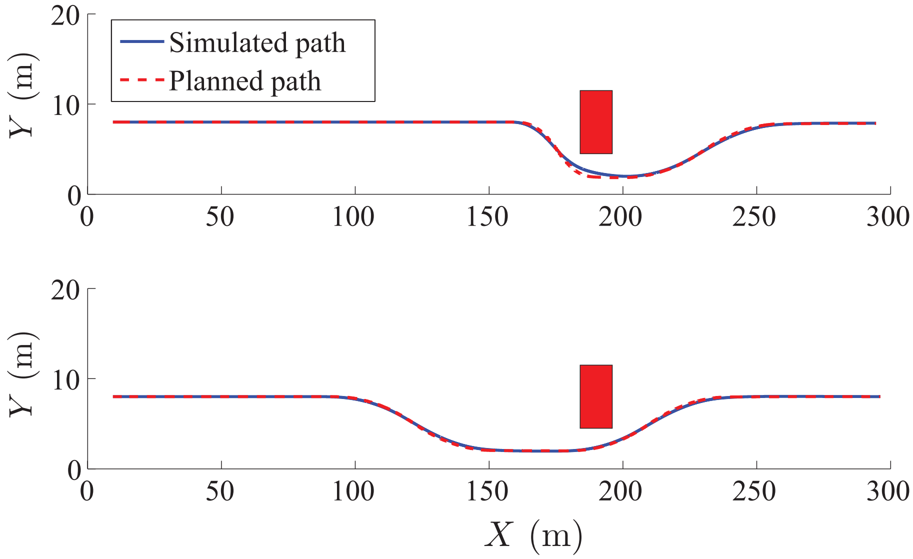


Figure 6.7: In the upper plot an attentive driver is assumed. The low-level control takes over when the planned path becomes aggressive. In the lower plot the low-level control takes over for a distracted driver and the result is a smoother and safer path.

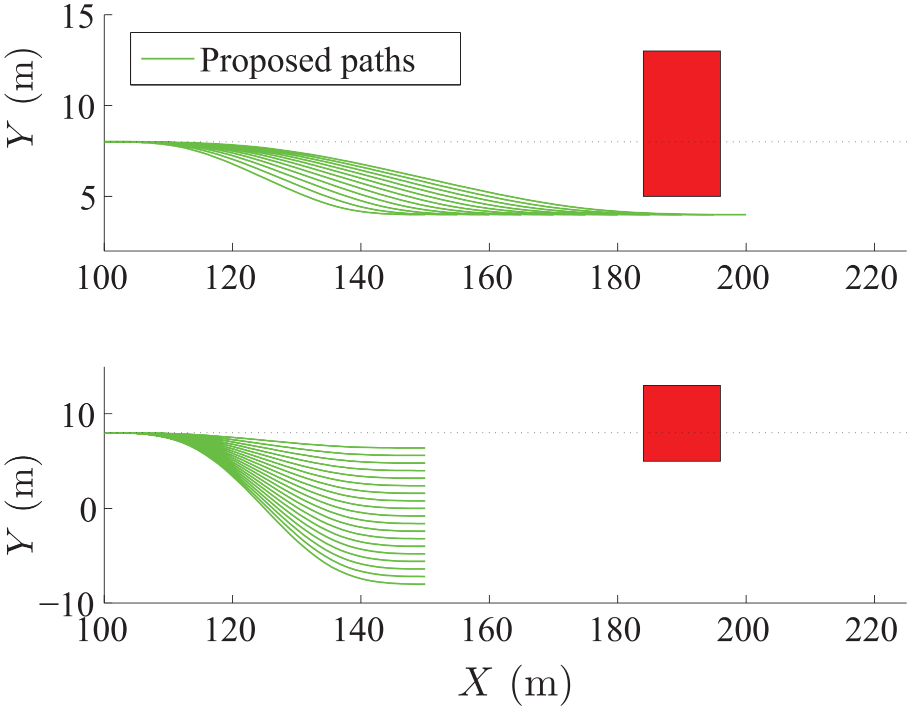


Figure 6.8: Various candidate paths considered during the online optimization of the parameterized maneuver.

Figure 6.8 shows some of the candidate paths considered during the online optimization of the parameterized clothoid maneuver, where the full set are all possible combinations of the paths in the upper and lower plots of Figure 6.8. The candidate path chosen is the one that minimizes the cost function in Equation (6.13) and satisfies the constraint in (6.9). Section 6.1.5 below details an experiment conducted on ice for the online optimization of this maneuver.

Semi-Autonomous Drive - Experiments

The control framework presented has been tested through simulation and the online optimization of paramterized maneuvers has been tested through experiments conducted on an icy surface. 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 intertia 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 differential GPS receiver, inertial measurement unit (IMU), and a DSP. It is equipped with a single antenna to receive GPS information. The IMU inclused three accelerometers and three angular rate sensors. The DSP receives both the measurements from the IMP and GPS, utilizes a Kalman filter for sensor fusion, and calculates the position, orientation, and other states of the vehicle such as lateral and longitudinal velocities.

The car is equipped with an Active Front Steering (AFS) and Differential Braking system which utilizes an electric drive motor to change the relation bewteen the hand steering wheel and the road wheel angles. This is done independently from the hand wheel position, thus the front road wheel angle is obstained 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. 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 pressed, the inertial frame 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. The inertial frame becomes the desired path coordinate system. Once the initialization procedure is concluded, the vehicle executes the experiment. Note that noise may affect the yaw angle measurement due to the single sensor setup. Compared to a dual antenna setup, a single antenna system has to elarn 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 presenece of a small offset in the orientation measurement while traveling at low speed or standing still.

The proposed control framework was tested with the high-level motion primitive path planner using parameterized clothoids to construct the path. The test was run on an icy and slippery test track. In the first test shown in Figure 6.9 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 6.10 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.

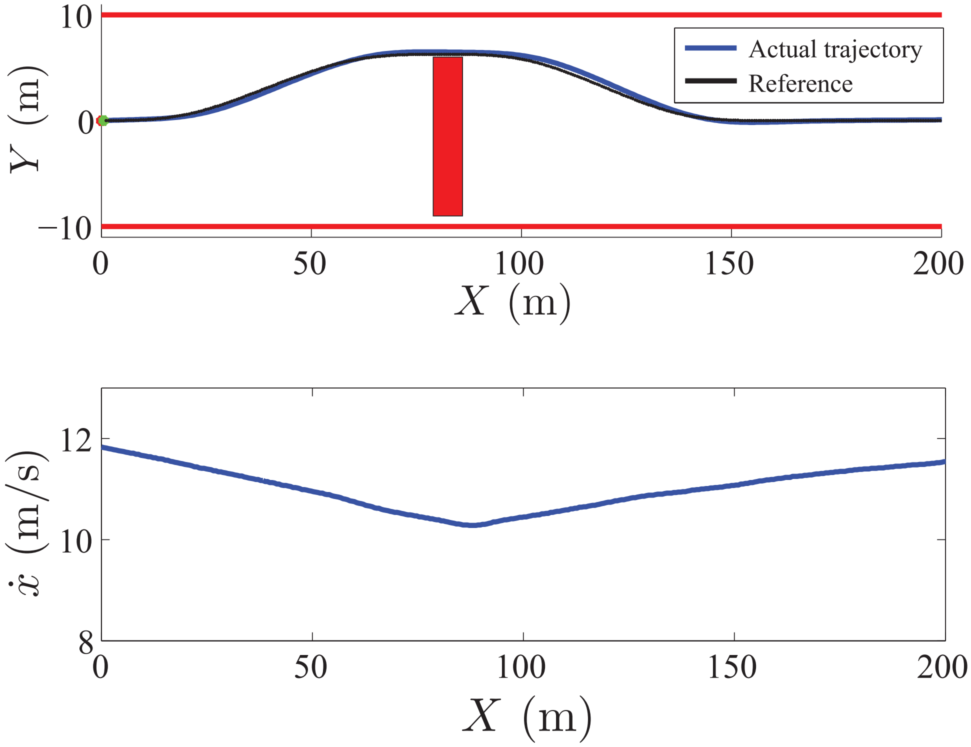


Figure 6.9: Vehicle successfully avoids the obstacle using maneuvers based on clothoids

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 6.10 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. In Fig 6.10 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.

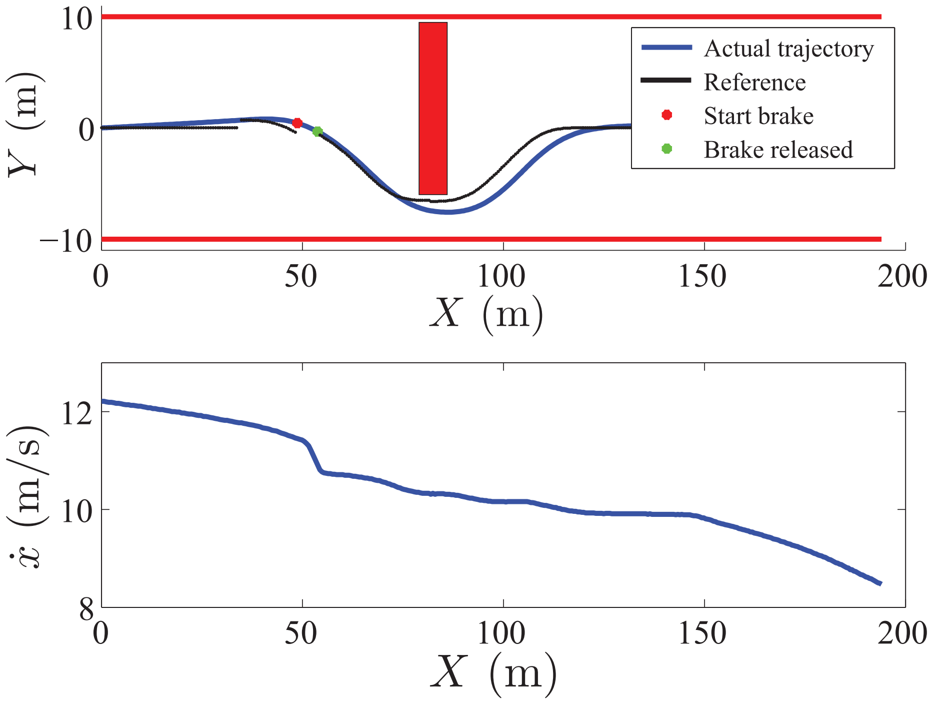


Figure 6.10: Actual path of the vehicle deviates from the planned path due to model mismatch and caused infeasibility of tracking. Braking was invoked to enlarge the feasible region in that situation.

6.2 Spatial Model Path Planner

In this section we introduce a novel spatial MPC framework. Consider the Nonlinear Spatial Vehicle Model presented in §2.4. This model is similar to the Bicycle Model of §2.3 but has undergone a coordinate transformation to make space, more specifically the arc length, s, along an arbitrary curve, the independent variable, and time, t, has become a dependent variable. The utility in this transformation becomes apparent when formulating the state constraints $\mathcal{X}$ for the optimization problem $\mathbb{P}_N L$ in §5.1. If the optimization routine retains the freedom to vary the vehicle's velocity, by considering braking and throttling, then the time-dependent vehicle models will not have pre-specified positions at the sampling points along the receding horizon. Consider the scenario in Figure 6.11 where a vehicle approaches a static object. For a time-based vehicle model the position of each sampled point is not known and the problem must be transformed into a complex mixed-integer optimization program. However, after a spatial transformation the sampled points are explicitly known in space and the constraints on the vehicle's position in the lane, e_y, becomes convex, as in Figure 6.12.

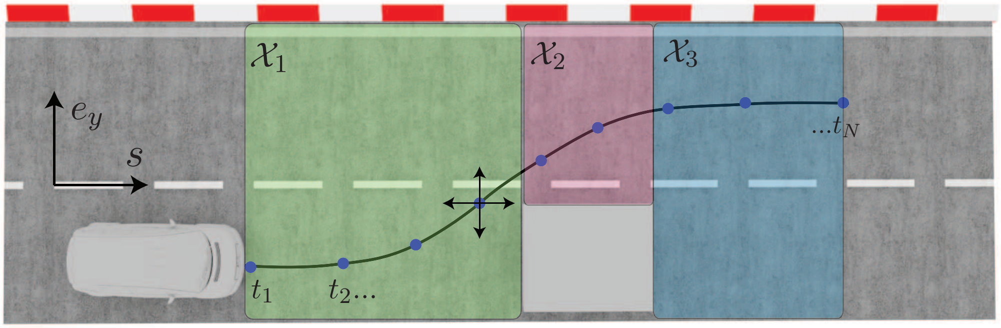


Figure 6.11: For the NMPC problem with steering and braking the obstacle avoidance problem may be posed as a complex mixed-integer program, shown here with 3 discrete constraint sets $\mathcal{X}_i$, $i \in \{1, 2, 3\}$, where the sampled points in time, t_i, $i \in \{1...N\}$, are not fixed in the spatial-direction s.

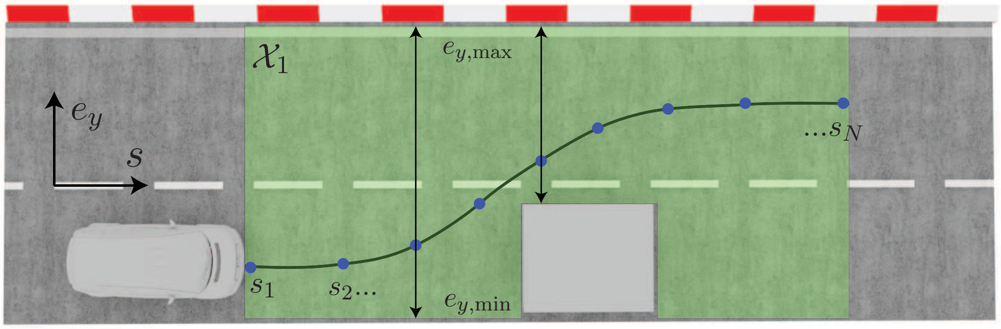


Figure 6.12: The spatial transformation reduces the complexity of the NMPC problem with steering and braking by a translation of the obstacle constraints to convex constraints on the state vector. The constraint set Σ_1 is then defined as interval constraints $\mathcal{X}_1 := \{e_y \,|\, e_{y,\min} \leq e_y \leq e_{y,\max}\}$ and the sampled points, $s_i,\ i \in \{1...N\}$, are fixed in the spatial-direction s.

Remark 9 *In this section only static objects are considered and the dynamics of moving objects are ignored. The obstacles considered here are roadside obstructions and varying lane boundaries.*

As in the previous section, the obstacle avoiding control problem considered here is decomposed into two parts: a high-level path planner and a low-level path follower. The high-level plans an obstacle-free path using the Nonlinear Spatial Vehicle Model, introduced in §2.4. The planned path (in spatial coordinates) is then fed to the low-level MPC path follower, which uses the Nonlinear Four Wheel Model, in §2.1, 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, as formulated in Chapter 5. This chapter is organized as follows: first, the high-level spatial model planner is detailed is §6.2.1, followed by the presentation of the low-level path tracker in §6.2.2. Simulation results using real-time control on dSpace embedded hardware is shown and the proposed framework is implemented on a test vehicle and the experimental results are presented in §6.2.3

6.2.1 High-Level Spatial Model Framework

The high-level MPC uses the spatial bicycle vehicle model from §2.4. We discretize the dynamics in (2.39) with a fixed sampling *distance*, ds, for the vehicle dynamics constraint in problem $\mathbb{P}_N L$ Equation (5.3b), which we write as,

$$\xi(s+1) = f^s(\xi(s), u(s)), \tag{6.18}$$

We define the general cost function of Equation (5.4) to consider the deviation of the tracking states $\eta^{hl}_{k,s} = [\dot{x}_{k,s}\ , \dot{\psi}_{k,s},\, e_{\psi_{k,s}}, e_{y_{k,s}}]^T$ with respect to the reference $\eta^{hl}_{ref_{k,s}} = [\dot{x}_{k,s}\ , \dot{\psi}_{k,s},\, e_{\psi_{k,s}},\, e_{y_{k,s}}]^T$, as well as the input and input rate. That is,

$$J_N(\Xi_s, U_s, \Delta U_s) = \sum_{k=s}^{s+N-1} \|\eta^{hl}_{k,s} - \eta^{hl}_{ref_{k,s}}\|^2_{Q_{hl}} + \|u_{k,s}\|^2_{R_{hl}} + \|\Delta u_{k,s}\|^2_{S_{hl}}, \tag{6.19}$$

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. Further, the constraints of Equations (5.3c)-(5.3d),

$$\xi_{k,t} \in \mathcal{X}, \quad u_{k,t} \in \mathcal{U}, \quad k = t+1, ...t+N-1, \tag{6.20}$$

are easily written as interval constraints,

$$\mathcal{X} := \{e_y \,|\, e_{y,\min} \leq e_y \leq e_{y,\max}\}, \quad \mathcal{U} := \{u \,|\, u_{\min} \leq u \leq u_{\max}\}, \tag{6.21}$$

Note 7 *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 (5.3c). At each prediction step, the vehicle position along the lane center is known to be $(s+k)\,ds$. 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 thesis, a simple heuristic based on the vehicle position and the size of each feasible region is used to determine on which side the vehicle should pass. 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 §2.4 Equation (2.40). 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.

6.2.2 Low-Level MPC Path Follower

The low-level MPC uses the Nonlinear Four Wheel Vehicle Model from section 2.1. At each time step t, the system dynamics are linearized around the equilibrium trajectory $[\xi_{k,t}, u_{k,t}]$, with $u_{k,t} = u_{t,t}\ \forall k = t, ..., t + H_{p,ll}$ 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 [30].

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.

$$J_N(\Xi_t, U_t, \Delta U_t) = \sum_{k=t}^{s+N-1} \|\eta_{k,t}^{ll} - \eta_{ref_{k,t}}^{ll}\|_{Q_{ll}}^2 + \|u_{k,t}\|_{R_{ll}}^2 + \|\Delta u_{k,t}\|_{S_{ll}}^2, \quad (6.22)$$

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_{rl} and right braking force F_{br}. The braking logic in [33] is used to distribute the corresponding torques at the four wheels.

6.2.3 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 [49] is used to solve the high level optimization. At the low-level, the nonlinear program is solved by using a sequential quadratic programming approach [36] and the quadratic program is solved using the QP solver routine available in [2], 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 [7], 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 implemented on a test vehicle with the following tuning parameters. The high-level of the hierarchical Spatial MPC framework solves the Nonlinear MPC problem (5.3) with model (2.39) and cost (6.19) and with the following parameters.

Table 6.3: Tuning parameters for high-level Spatial MPC

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

The low-level of the hierarchical Spatial MPC framework solves a Linear Time-Varying MPC with model (2.3) and cost (6.22) and with the following parameters

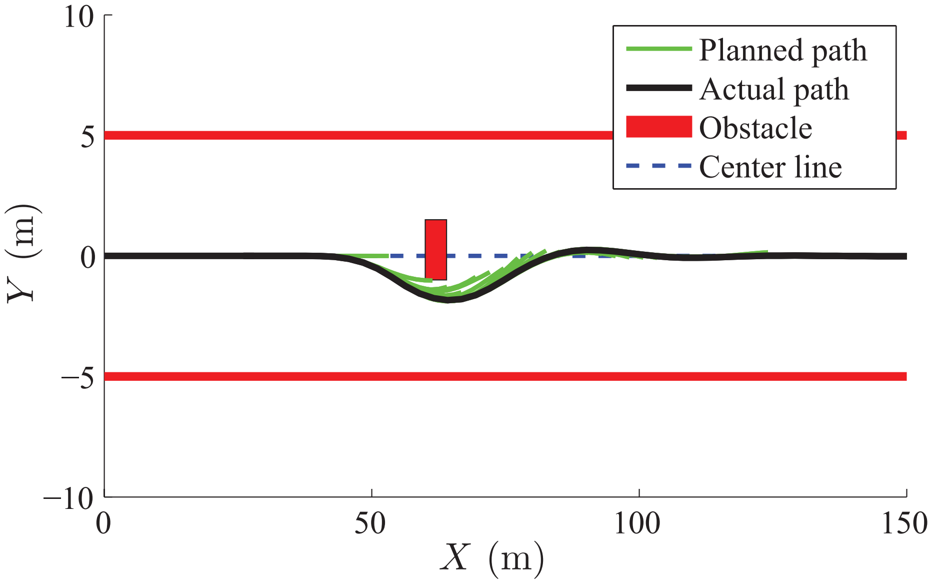


Figure 6.13: Simulated result. The vehicle entered the maneuver at 50 kph. The green lines are planned paths from the high-level which are updated every 200 ms. The black line is the actual trajectory the vehicle traveled.

Table 6.4: Tuning parameters for low-level LTV MPC

$T_{s,ll} = 0.05\text{s}$	$H_{p,ll} = 5$
$H_{u,ll} = 3$	$iH_{u,ll} = 1$
$\delta_f \in [-10^\circ, 10^\circ]$	$\Delta\delta_f \in [-17^\circ, 17^\circ] \times T_{s,ll}$
$F_{b\bullet} \in [-1500, 0]$	$\Delta F_{b\bullet} \in [-1000, 1000] \times T_{s,ll}$
$Q = \text{diag}(10, 20, 10, 50)$	$R = \text{diag}(1, 0.5, 0.5)$
$S = \text{diag}(1, 0.1, 0.1)$	

The simulation and experimental results are summarized in Figures 6.13 to 6.16. 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

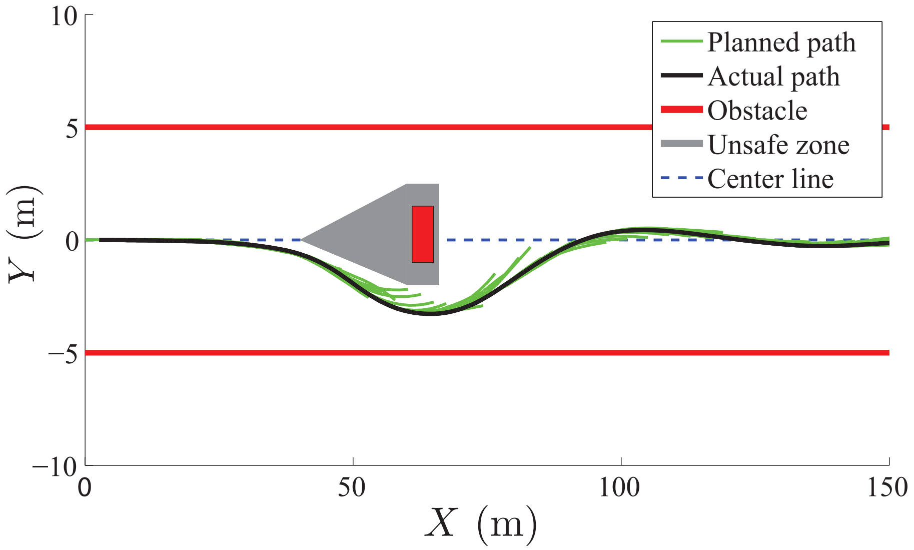


Figure 6.14: Experimental result. The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the obstacle and continued to track the lane center. The green lines are planned paths from the high-level which were updated every 200 ms. The black line is the actual trajectory the vehicle traveled.

Table 6.5: Low Level Lateral Position (Y) Tracking Error Comparison. Simulation data corresponds to figure 6.17 and experimental data corresponds to figure 6.14.

	HL Point Mass ([cm])	HL Spatial Bicycle ([cm])
Mean	7.37	1.63
Max	49,42	11.18

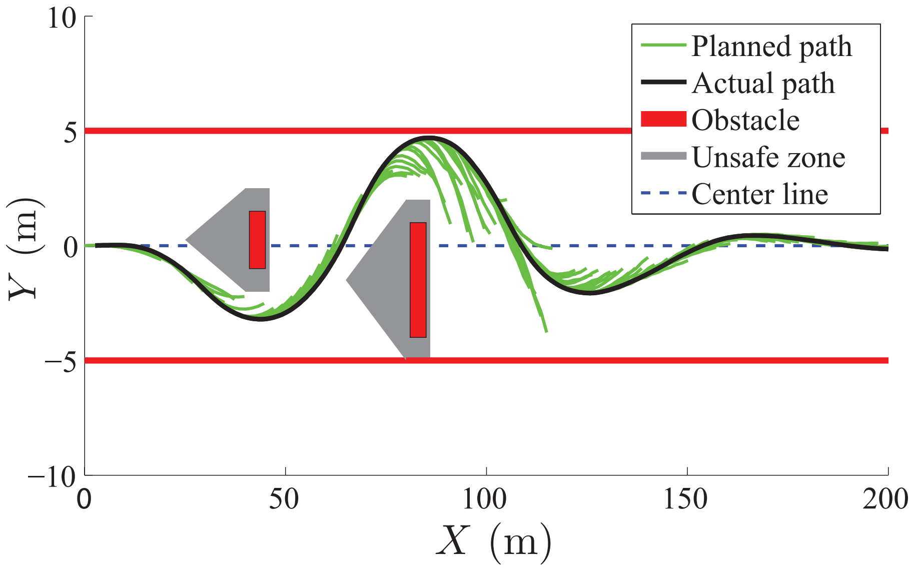


Figure 6.15: Experimental result. The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the two obstacles and continued to track the lane center.

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 6.13 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 Figures 6.14-6.16. Figure 6.13 and 6.14 are the simulation and experimental results of the vehicle avoiding one obstacle on a slippery road. The two tests show consistent performance. Figure 6.15 and 6.16 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 6.16 where 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 6.17 compares the controller's performance with a previously proposed controller in [46], 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

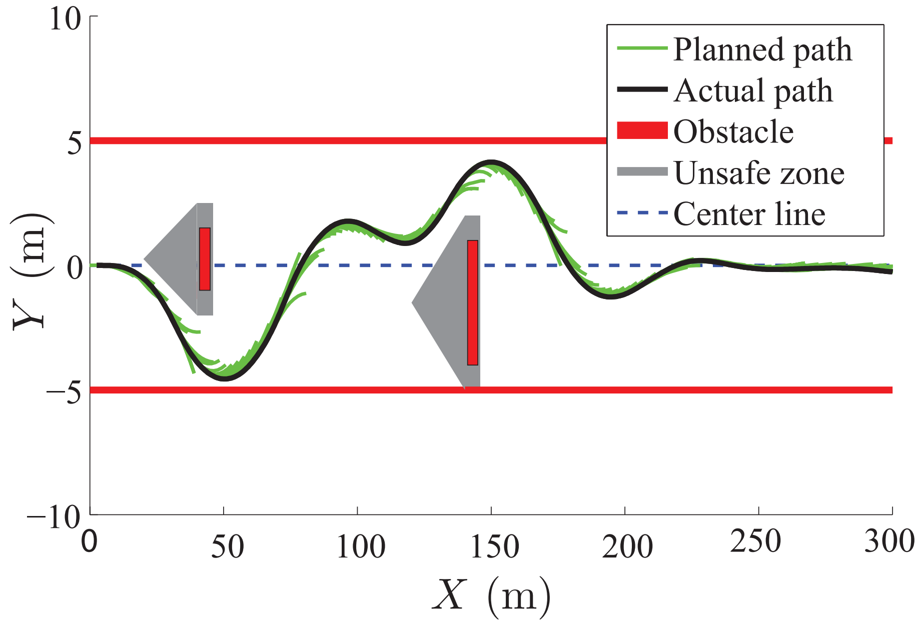
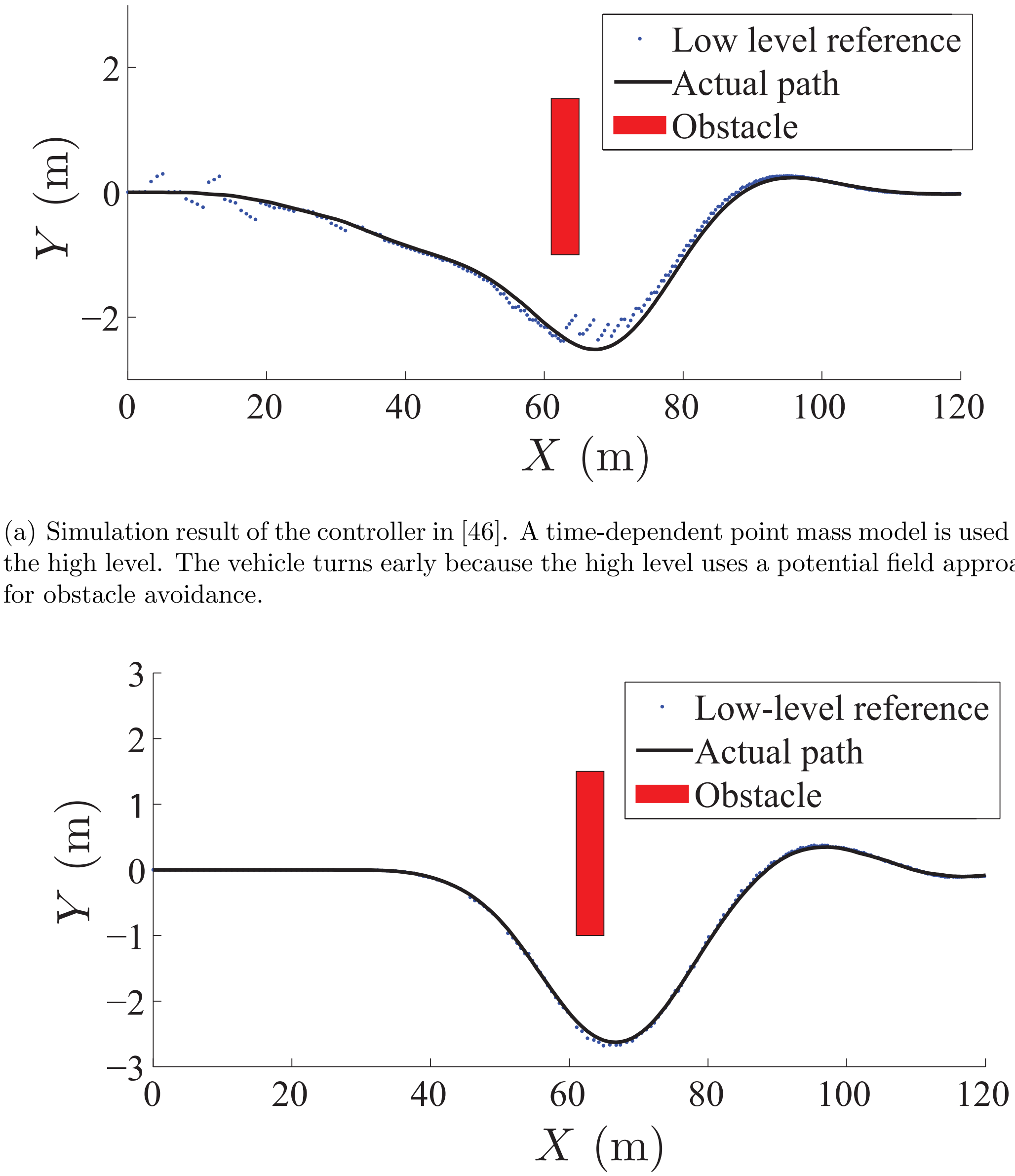


Figure 6.16: Experimental result. The vehicle entered the maneuver at 50 kph. Friction coefficient of the ground was approximately 0.3. The vehicle avoided the first obstacle and continued to track the lane center until the second obstacle came into sight. It then turned left to avoid the second obstacle.



(a) Simulation result of the controller in [46]. 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 6.17: Simulation result comparison with the controller proposed in [46]. 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.

at each sampling time. We observe that the use of a nonlinear bicycle model at the high level greatly improves the tracking performance. The use of a spatial model at the high-level makes the construction of obstacle constraints straight forward as described in §6.2. On the other hand, the hard constraint formulation for obstacles is not trivial for general obstacles in the MPC framework of Chapter 5 with time-dependent models. Potential field approaches are a commonly used solution to this problem. Of course, it is possible to find indicating functions whose level sets can be used to form the obstacle constraints. This is usually not straightforward, especially if one wants the indicating function to have nice properties, such as differentiability, in order to speed up the optimization problem.

Chapter 7

Integrated Safety Framework

In this chapter we design a novel active safety system for prevention of unintended roadway departures and collision avoidance with a human-in-the-loop. A typical active safety system architecture is modular [3], with separate threat assessment, decision making, and intervention modules, see Figure 7. In particular, the threat assessment module deals with the task of determining whether interventions are necessary and plays an important role in the interaction with the driver. The threat assessment module repeatedly evaluates the driver's ability in maintaining safety in each situation and this information is used by the decision making module in order to decide whether and how to assist the driver. It is a challenge for an active safety system to properly assess when to intervene. In the literature, a large variety of threat assessment and decision making approaches can be found [22, 78, 62, 29]. In the simplest approaches, used in production vehicles, automated steering or braking interventions are issued when simple measures like the *time to collision* [22] or *time to line crossing* [78] pass certain thresholds.

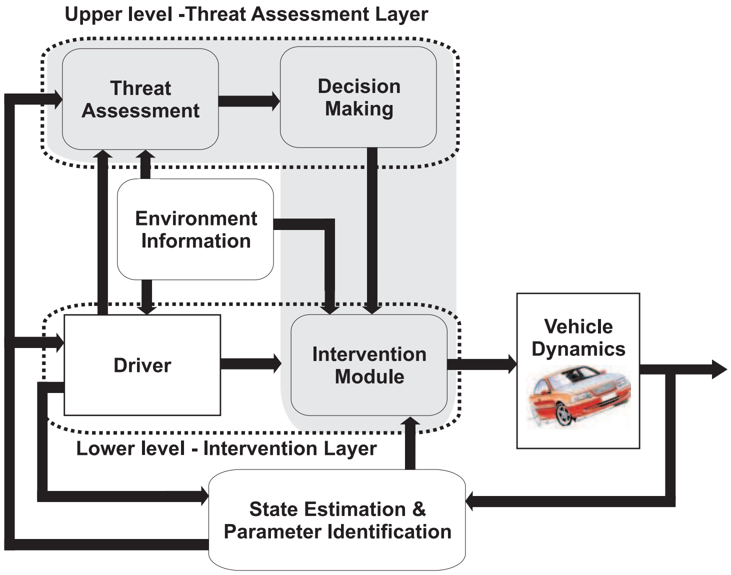

Upper level -Threat Assessment Layer
Threat Assessment
Decision Making
Environment Information
Driver
Intervention Module
Vehicle Dynamics
Lower level - Intervention Layer
State Estimation & Parameter Identification

Figure 7.1: A block diagram of the typical structure of a modular active safety system.

More sophisticated approaches, on the other hand, include the computation of Bayesian collision probabilities [62] or sets of safe states from which the vehicle can safely evolve [29]. In advanced safety systems, such as roadway departure prevention, the intervention module has the goal to both determine a safe trajectory and coordinate the vehicle actuators. The literature on vehicle path planning and control is extensive, see, e.g. [32, 72, 104, 48]. Because of its capability to systematically handle system nonlinearities and constraints, work in a wide operating region and close to the set of admissible states and inputs, Model Predictive Control (MPC) has been shown to be an attractive method for solving the path planning and control problem [32, 33]. Previous approaches to lane departure prevention using predictive control, as in [4], make the assumption that the vehicle is traveling at a constant velocity (and can therefore not consider braking), and does not use any information about the human driver.

Rather than separately solving the threat assessment, decision making, and intervention problems, we reformulate them as a single combined optimization problem. In particular, a predictive optimal control problem is formulated which simultaneously uses predicted driver's behavior and determines the least intrusive intervention that will keep the vehicle in a region of the state space where the driver is deemed safe. The proposed controller is always active, which avoids the design of switching logic or the tuning of a sliding scale. In addition, since the proposed controller is designed to only apply the correcting control action that is necessary to avoid violation of the safety constraints, the intrusiveness of the safety application is kept minimal. Furthermore, the full nonlinear dynamics of the vehicle are considered in the optimization problem and the corrective action can augment both the driver's steering *and* braking. Preliminary findings were reported by the authors in [53, 52] with simulation results. In this chapter we detail the proposed framework and show its effectiveness through experimental results implemented on a passenger vehicle.

The chapter is organized as follows: in Section 7.1 we present the Integrated Active Safety Framework for collision avoidance and unintentional lane departure. In Section 7.2 we present simulation results both from live interaction with a driving simulator as well as from human driver data collected offline. Then, in Section 7.3 we present experimental results collected from implementing the proposed active safety system on a passenger vehicle at a testing facility at Volvo in Gothenburg, Sweden. Finally, Section 7.5 proposes an approach to incorporate uncertainty into the framework and still guarantee safety. Simulation results approaching single and multiple obstacles are presented.

7.1 Integrated Active Safety

In this section we introduce the Active Safety Framework where we formulate a single combined optimization problem, drawing from the vehicle models of Chapter 2, the driver models of Chapter 3, and the Model Predictive Control methodology of Chapter 5. Consider the discretized Nonlinear Spatial Vehicle Model presented in §2.4,

$$\xi(s+1) = f^s(\xi(s), u(s)). \tag{7.1}$$

To explicitly consider the driver's intention over the prediction horizon of the MPC problem $\mathbb{P}_{NL}$ we define the input $u = [\delta,\, \beta_r]$ where $\delta = \hat{\delta}_d + \delta_c$ and $\hat{\delta}_d$ is the nominal driver's input from Equation (3.17),

$$\hat{\delta}_d = K_y e_y + K_\psi e_\psi + K_\psi \Delta\psi_{\text{road}}, \tag{7.2}$$

and δ_c is the corrective steering action decided by the MPC control law, along with corrective braking action β_r. Then, the optimization problem, following the general formulation of problem $\mathbb{P}_{NL}$, is stated as,

$$\begin{aligned}
\mathbb{P}_{DM}: \qquad & \min_{U_c,\varepsilon} \; J_N(U_c, \Delta U_c) && \text{(7.3a)}\\
& s.t. \; \xi_{k+1,s} = f^{dm}(\xi_{k,s}, u_{k,s}), \quad s = t, ..., N-1, && \text{(7.3b)}\\
& \xi_{k,s} \in \mathcal{X}, \quad k = s+1, ...s+N-1, && \text{(7.3c)}\\
& u_{k,s} \in \mathcal{U}, \quad k = s, ..., s+N-1, && \text{(7.3d)}\\
& \xi_{s,s} = \xi(t), && \text{(7.3e)}\\
& \xi_{N,s} \in \mathcal{X}_f, && \text{(7.3f)}
\end{aligned}$$

where the model $f^{dm}(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ in Equation (7.3b) is constructed by placing the nominal driver model of Equation (7.2) in closed-loop with the vehicle dynamics of Equation (7.1). That is,

$$\xi(s+1) = f^{dm}(\xi(s), [\hat{\delta}_d(s) + \delta_c(s),\, \beta_r(s)]), \tag{7.4}$$

where $\delta_c(s)$ and $\beta_r(s)$ are a solution to problem $\mathbb{P}_{DM}$ at distance s. By formulating the problem in this way the predicted driver's inputs are explicitly considered and the optimization problem tries only to minimize the corrective action needed to satisfy the safety constraints in Equation (5.3c). By doing this, we can then write the cost function to be minimized as,

$$J(U_c, \Delta U_c) = \sum_{s=k}^{s+N-1} \|u_c(s)\|_Q^2 + \|\Delta u_c(s)\|_R^2 + \rho\varepsilon, \tag{7.5}$$

where $U_c = \{u_{c,s}, u_{c,s+1}, ..., u_{c,s+N}\}$ is the vector of optimal corrective actions, $\Delta U_c = \{\Delta u_{c,s}, \Delta u_{c,s+1}, ..., \Delta u_{c,s+N}\}$ is the vector of optimal change rates of corrective actions, and the slack variable ε has been added to soften the constraints, as detailed in §2.6. Q and R are weights of appropriate dimension.

Note 8 *We note that no penalty on deviation from a tracking reference* η_{ref} *is imposed in the cost function* (7.14)*, as was the case in Chapter 6. The objective here is to ensure that the safety constraints* (7.3c) *are not violated while utilizing minimal control action. If the driver steering model* (7.2) *is alone capable of steering the vehicle without violating the safety constraints* (7.3c) *no control action will be applied and the optimal cost will thus be zero.*

Remark 10 *In addition to the soft constraints we have imposed hard constraints. Equation* (5.3d) *reflects limitations set by the actuators.*

7.2 Simulation Results

In this section we present simulations showing the behavior of the proposed framework. The next section will present experimental results of the proposed controller implemented on a passenger vehicle. We present results from simulations using human driver data stored offline. We solve problem $\mathbb{P}_{DM}$ using Matlab and Tomlab/NPSOL to solve the optimization problem. Then, we implement the controller on a dSPACE embedded control unit and run simulations using real-drivers interacting with CarSim vehicle simulation software. The proposed controller is run in real-time. Section 7.2.1 focuses on unintended roadway departure scenarios while Section 7.2.2 focuses on collision avoidance of roadside obstacles. Both subsections are organized internally by the type of scenario considered.

7.2.1 Unintentional Roadway Departure

This subsection focuses on ensuring the safety of the driver during an unintentional lane departure. We will focus on two scenarios that are a common cause of traffic accidents, as noted in Chapter 4, and are both caused by driver distraction: drifting out of the lane and approaching a curve too quickly. The next subsection 7.2.2 will focus on collision avoidance.

Simulation Results with Offline Human Driver Data

In this section we validate the behavior of the proposed active safety system by analyzing the results from Matlab simulations. We consider scenarios three scenarios where the driver,

1. safely negotiates a curve,
2. approaches a curve too quickly and is not prepared to safely navigate the turn,
3. unintentionally veers off a straight road into oncoming traffic.

Human drivers interacting with a driving simulator were used to collect the data for the results presented. The estimation algorithm presented in

§3.1.1 and [29] is implemented to estimate parameters of the driver model (3.17). The vehicle and design parameters in Tables 7.1 and 7.2 were used to implement the predictive controller of Equations (5.3).

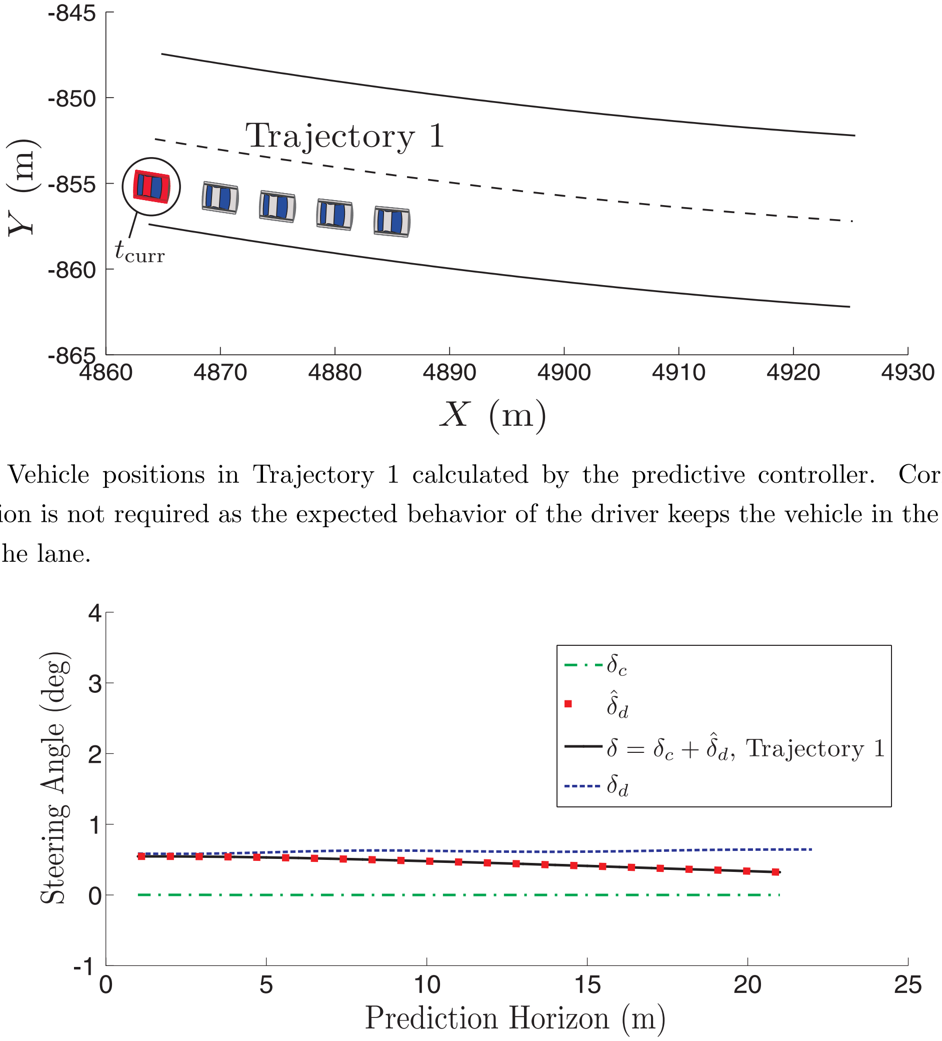
Table 7.1: Vehicle parameters

$m =$ 2050 kg	$\sigma =$ 0.7	$w_t =$ 1.63 m	$B_1, B_2 =$ -10.5	$C_1, C_2 =$ 0.5
$J_z =$ 3344 kgm^2	$l_f =$ 1.43 m	$l_r =$ 1.47 m	$B_3, B_4 =$ -12.7	$C_3, C_4 =$ 0.5
μ	a	b	c	
1.0	2.12 m	2.66 m	1.77 m	

Table 7.2: Design parameters

$u_{\max} =$ $[0.7 \text{ rad}, 0 \; N]^T$	$N =$ 21	$\alpha_{\max} =$ 4°
$u_{\min}=$ $[-0.7 \text{ rad}, -\mu m g \; N]^T$	$ds =$ 1 m	$\alpha_{\min} =$ -4°
$\Delta u_{\max} =$ $[1.4 \text{ rad}, \mu m g \; N]^T$	$\rho =$ 10^4	$e_{y_{\max}} =$ 2.5 m
$\Delta u_{\min} =$ $[-1.4 \text{ rad}, -\mu m g \; N]^T$	$R =$ $diag(1, 10)$	$e_{y_{\min}} =$ -2.5 m

The driver estimation algorithm adapts and updates the parameters of the driver model as new data becomes available. Since the estimation is conducted in nominal driving conditions, the resulting driver model is expected to be representative of the nominal behavior of the driver. The implications of this is discussed next, as the behavior of the suggested predictive controller is analyzed for the three considered scenarios.



(a) Vehicle positions in Trajectory 1 calculated by the predictive controller. Corrective action is not required as the expected behavior of the driver keeps the vehicle in the center of the lane.

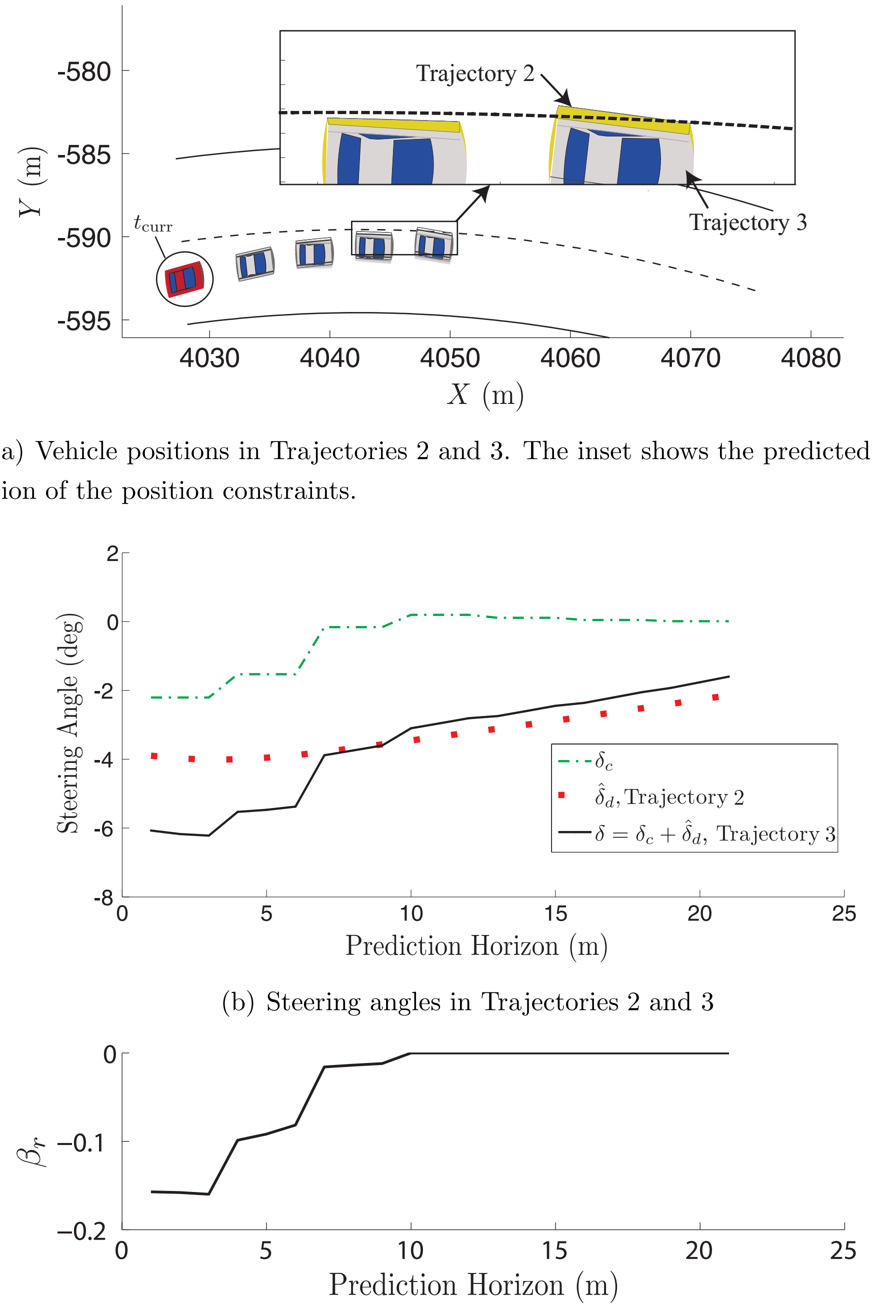
(b) Steering angles: δ_c, the corrective action from the controller, $\hat{\delta}_d$, the driver model prediction, δ, the closed-loop steering used in the predictive control problem, and δ_d, the actual steering from the human driver.

Figure 7.2: Simulation results. In these plots we show nominal driving behavior where the driver is not in need of assistance. These results show the behavior of the controller when the driver is capable of maintaining safety. The minimally invasive nature is illustrated by the controller calculating zero assistance in a safe situation.

Nominal Behavior Consider Figure 7.2(a) which depicts a driving situation where the driver is attentive and is safely steering the vehicle down the center of the lane, within the road bounds. The circled vehicle indicates the current vehicle position, marked t_{curr}, and the others illustrate the future vehicle positions, predicted by the predictive controller. We refer to this trajectory as Trajectory 1. In this situation, the estimated driver model is capable of keeping the vehicle in the lane which indicates that the nominal behavior of the driver is safe. The action that minimizes the cost function (7.12a) is thus zero corrective steering and braking, hence the driver remains in control of the vehicle.

Figure 7.2(b) shows a comparison of the predicted steering trajectory, Trajectory 1, and the actual steering trajectory of the driver, who was allowed to remain in full control of the vehicle. We note that the corrective steering action δ_c is zero in Trajectory 1, hence the closed-loop trajectory is predicted by the driver model only, i.e., $\delta = \hat{\delta}_d$. We also note that the steering angle $\hat{\delta}_d$ in Trajectory 1 corresponds well with the driver's actual steering angle δ_d. In this situation the adopted predictive controller could correctly predict the nominal behavior of the driver and thus avoided intervening.

Excessive Speed in a Curve Next we consider a scenario where the driver is approaching a curve too quickly to safely navigate the turn. An intervention from the active safety system is required to keep the driver safely within the constraints of the lane.



(a) Vehicle positions in Trajectories 2 and 3. The inset shows the predicted violation of the position constraints.

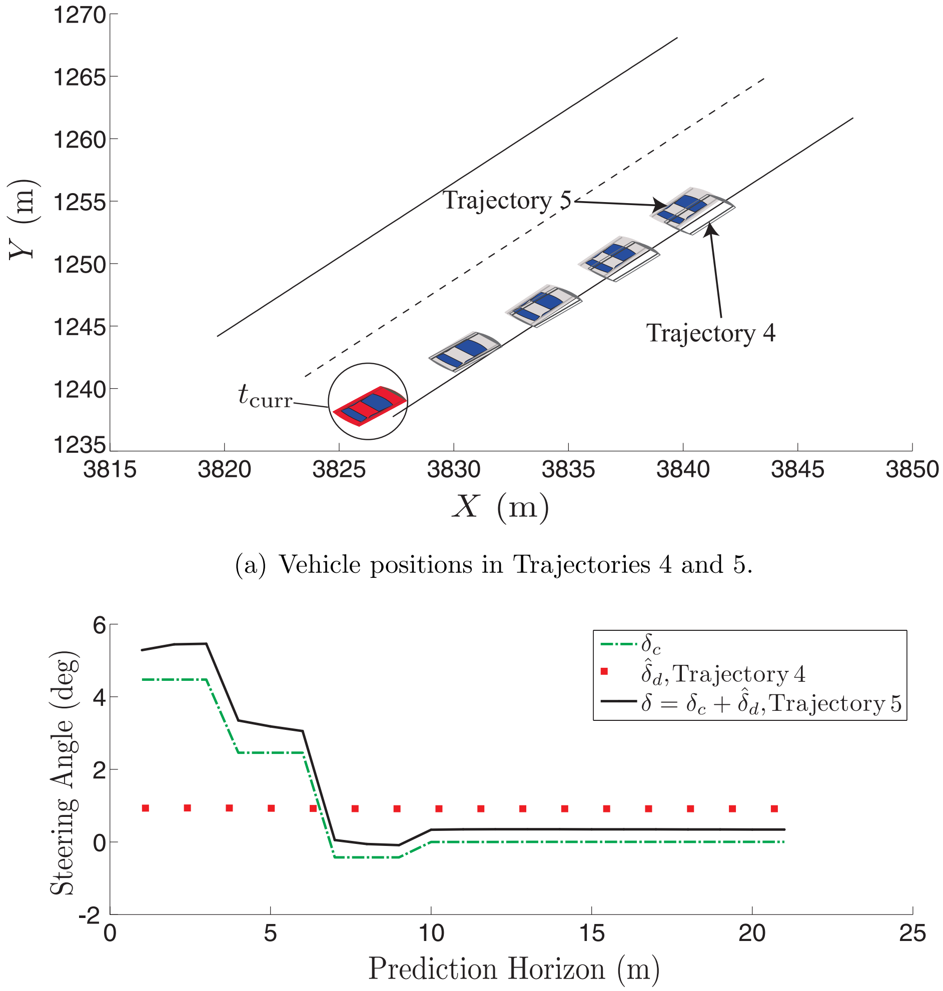
(b) Steering angles in Trajectories 2 and 3

(c) β_r in Trajectory 3

Figure 7.3: Simulation results. The plots capture a snapshot of the prediction that detects a lane departure by excessive speed in a curve. The controller augments the driver's steering and braking to bring the vehicle back within the lane.

In Figure 7.3(a) two trajectories are shown. The circled vehicle shows the vehicle's current position. The vehicles shown in outline illustrate the future trajectory of the vehicle controlled by the driver model only (Trajectory 2). Trajectory 2 indicates that the driver's nominal behavior leads to a violation of the position constraints (2.47). Consequently, the predictive controller corrects the driver's control action to avoid the constraint violation. The vehicles in Figure 7.3(a) show the trajectory predicted by the predictive controller (Trajectory 3). Compared to Trajectory 2, the vehicle motion has been slightly corrected such that the vehicle remains in the lane.

Figure 7.3(b) shows the steering angles δ, $\hat{\delta}_d$ and δ_c in Trajectories 2 and 3 and Figure 7.3(b) shows the braking signal β_r in Trajectory 3. Figure 7.3(b) shows that, as indicated by $\hat{\delta}_d$ in Trajectory 2, the driver is expected to steer and attempt to follow the path prescribed by the road. However, we note that the magnitude of $\hat{\delta}_d$ in Trajectory 2 is too small, hence in order to maintain the vehicle within the road boundaries, the driver would have to deviate from the nominal behavior described by the estimated driver model. Figures 7.3(b) and 7.3(c) show how the predictive controller simultaneously corrects the driver's steering and slightly brakes the vehicle. In particular we note that the steering magnitude $|\delta|$ in Trajectory 3 is initially significantly higher than $|\hat{\delta}_d|$ in Trajectory 2. We also note that the control signals δ_c and β_r vanish smoothly as the vehicle path has been recovered and the driver model is again capable of keeping the vehicle in the lane.



(a) Vehicle positions in Trajectories 4 and 5.

(b) Steering angles in Trajectories 4 and 5.

Figure 7.4: Simulation results. These plots capture a snapshot of the prediction where the driver will depart the lane, as shown in (a). The controller augments the driver's steering, (b), to satisfy the safety constraints and keep the vehicle within the lane.

Unintentional Drifting Consider Figure 7.4 where the driver is distracted and is veering off the shoulder of the lane. In Trajectory 4 the vehicle is controlled by the driver model and is illustrated with the vehicles in outline. In Trajectory 5 the vehicle is instead controlled by the predictive controller. At the points of the drawn vehicles in Figure 7.4, the nominal driver behavior is no longer sufficient to keep the vehicle in the lane, as indicated by Trajectory 4.

Since the lane departure in this situation is related to distraction rather than excessive speed, the predictive controller does not brake. Instead, we note that the predictive controller corrects the driver's steering to steer the vehicle back in the lane. We also note that in Trajectory 5 the signal δ_c smoothly vanishes to zero as the vehicle's path is recovered and the driver model is again capable of keeping the vehicle in the lane.

We remark that the adopted driver model does not capture the driver's distraction. Consequently the predictive controller does not explicitly account for this. In the considered scenarios, the predictive controller simply identified that even though it is still possible to keep the vehicle in the lane, the driver would have to deviate from the nominal behavior described by the driver model. Even though, the performance of the considered approach could potentially be enhanced by incorporating a driver monitoring system, we observed that in these two scenarios, the proposed approach is beneficial without depending on such a system.

Real-time Simulation Results with Human Driver

In this section the proposed framework is tested through real-time simulation using human drivers. The controller is run on a dSPACE MicroAutobox. A high-fidelity vehicle model is simulated using state of the art vehicle simulation software CarSim. The scenario is tested live with a real driver interacting with the system by manipulating steering, braking, and throttle inputs. Table 7.3 lists the parameters used for the real-time simulation. We make the following remark for the real-time implementation:

Remark 11 *Each control input, δ_c and β_r (the solution from the optimization problem) is held constant for three sampling times, $H_i = 3$. This reduces the complexity and the number of optimization variables becomes $\frac{N}{H_i} \cdot m_r + 1 = 9$ where m_r is the number of inputs and the addition of* 1 *comes from the slack variable, ϵ.*

Table 7.3: Real-Time Design parameters

$u_{\max} = [0.2\ \text{rad}, 0.5\ [-]]^T$	$N = 12$	$\alpha_{\max} = 4°$
$u_{\min} = [-0.2\ \text{rad}, -0.5\ [-]]^T$	$T_s = 200\ \text{ms}$	$\alpha_{\min} = -4°$
$\Delta u_{\max} = [0.4\ \text{rad}, 1\ [-]]^T$	$Q = diag(1,1)$	$e_{y_{\max}} = 0.7\ \text{m}$
$\Delta u_{\min} = [-0.4\ \text{rad}, -1\ [-]]^T$	$R = diag(1,1)$	$e_{y_{\min}} = -0.7\ \text{m}$

Three scenarios are detailed, two are shown in Figure 7.6 inset A and B and the third is shown in Figure 7.8.

Distracted Driver Approaching a Curve In Figure 7.6 inset A, the driver approaches a curve and simulates distraction by removing his hands from the steering wheel. The vehicle enters the curve at 90 kph. The controller correctly predicts the vehicle will exit the lane and adds corrective steering to safely negotiate the curve. After exiting the curve the driver resumes control.

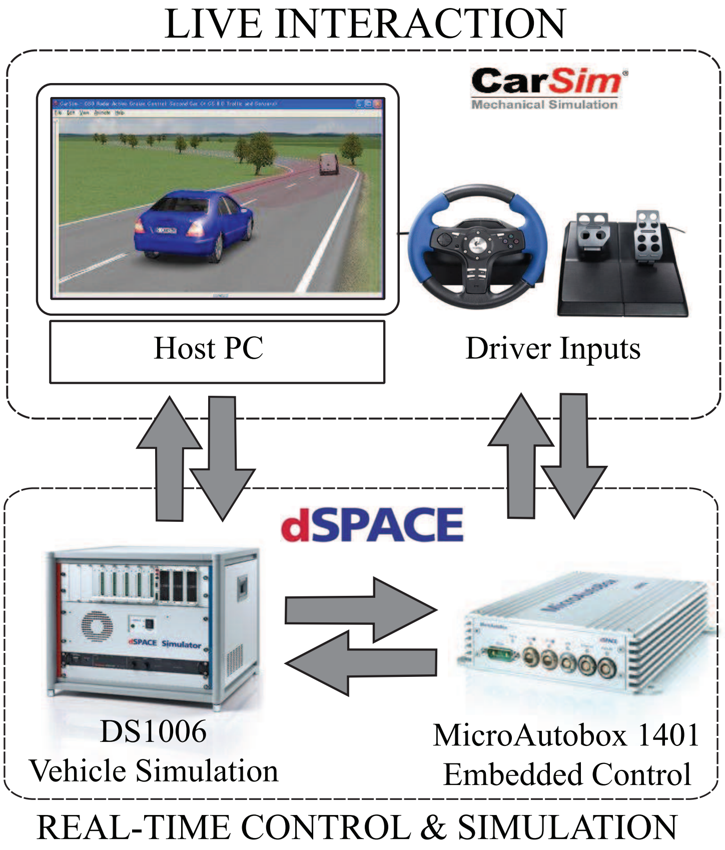


Figure 7.5: Setup

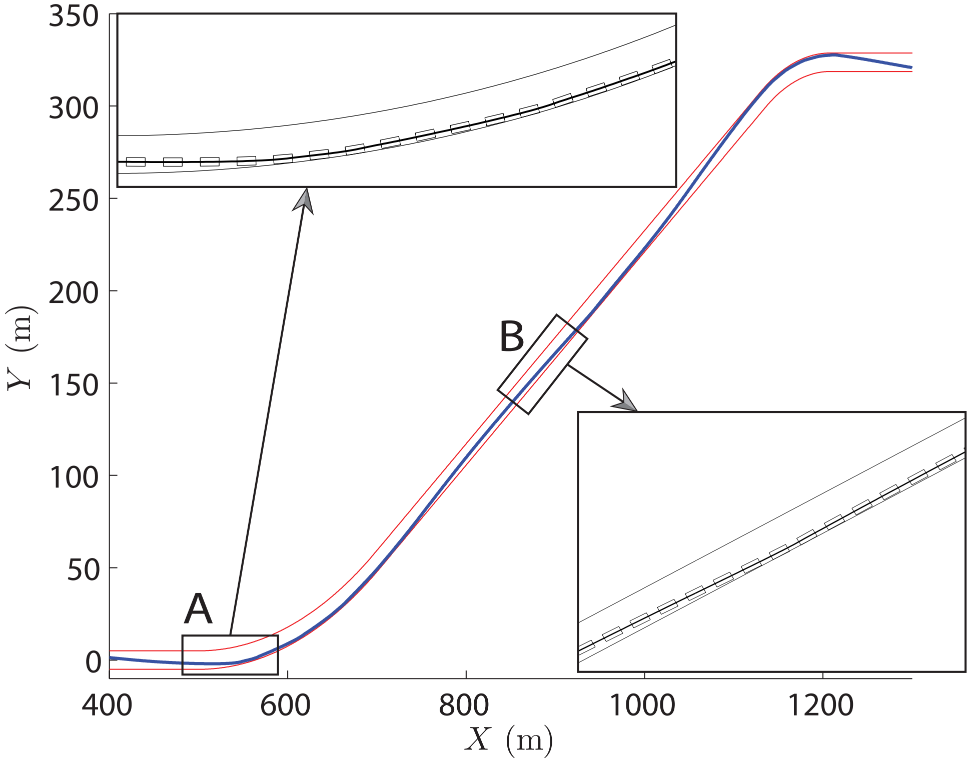


Figure 7.6: The complete trajectory of the vehicle within the lane with two insets, A and B, detailing two interventions. The first occurs when the distracted driver approaches a curve and the second occurs when the driver lets the vehicle stray off the road. In both situations the controller corrected the steering to keep the vehicle safely on the road.

In Figure 7.6 inset B, the driver again simulates distraction while driving on a straight section of the road to let the car stray off the road. The controller predicts a roadway departure and adds corrective steering to keep the vehicle within the lane. In both situations, A and B, braking was not needed to keep the driver and vehicle safe.

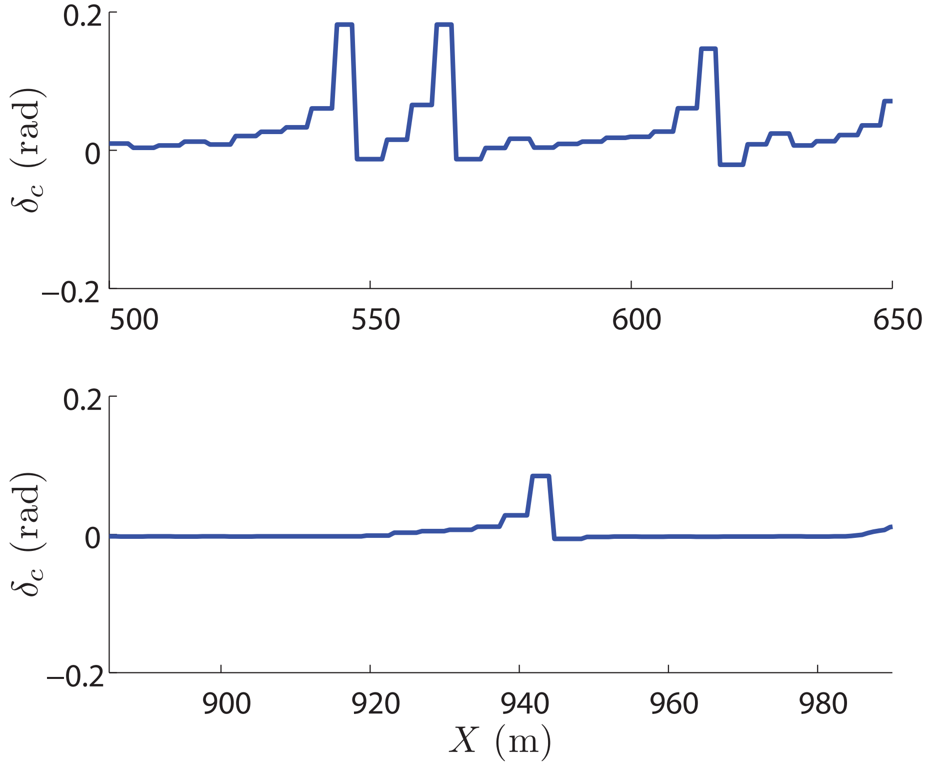


Figure 7.7: The upper plot shows the corrective steering inputs for the real-time simulation shown in Figure 7.6 inset A and the lower plot depicts the corrective steering for Figure 7.6 inset B.

Figure 7.7 plots the corrective steering action needed for the two interventions detailed in Figure 7.6 inset A and B. In scenario A the driver did not manipulate the steering wheel and it can be seen that multiple corrective actions were needed to keep the vehicle within the lane and navigate around the curve. Whereas in scenario B only a brief intervention was needed to correct the vehicle from straying out of the lane.

Distracted Driver Drifting from Lane. A scenario is shown in Figure 7.8 where the driver unintentionally drifts over to the left side of the road. In this scenario the driver is distracted and is heading for a collision by roadway departure. The active safety system adds corrective steering to keep the vehicle on the road. As shown in Figure 7.10 only corrective steering was needed and only a minimal amount of braking. The controller briefly augments the steering angle to straighten the vehicles trajectory and the safety constraints are not violated. A plot of 4 vehicle states of interest, $[e_y, e_\psi, \dot{x}, \dot{\psi}]$ are shown in Figure 7.9. The lateral error offset, e_y, clearly indicates the vehicle is corrected to stay within the upper road bound of 5 m.

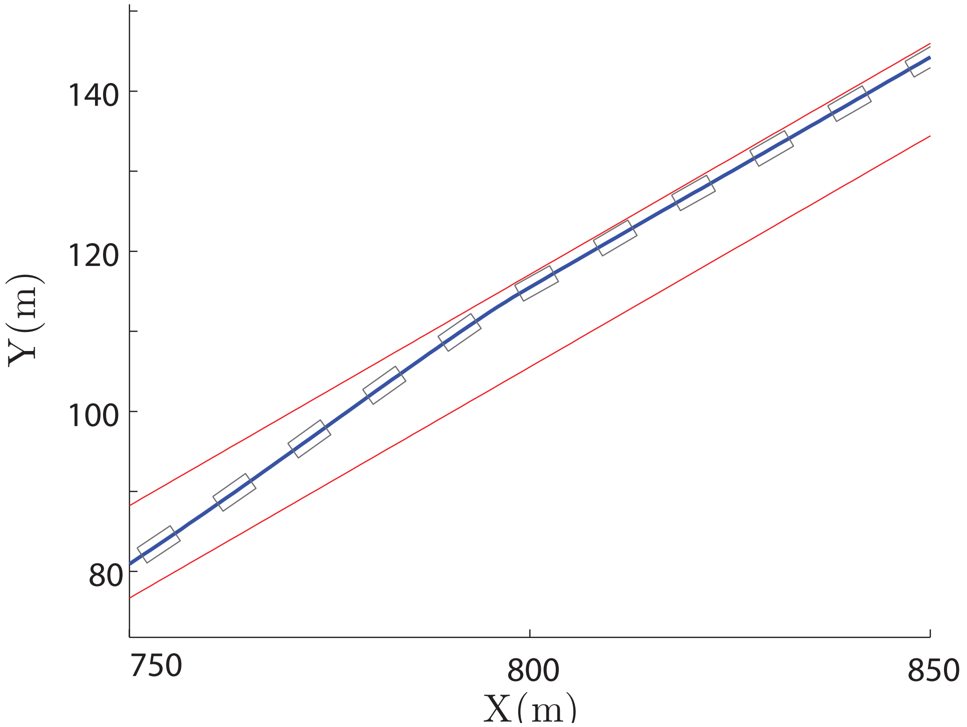


Figure 7.8: The trajectory of the vehicle during a roadway departure intervention. The driver is distracted and the vehicle drifts to the left of the road. An intervention occurs and the vehicle safely remains within the safety constraints imposed by the road bounds. The associated states and inputs can be seen in Figures 7.9 and 7.10, respectively.

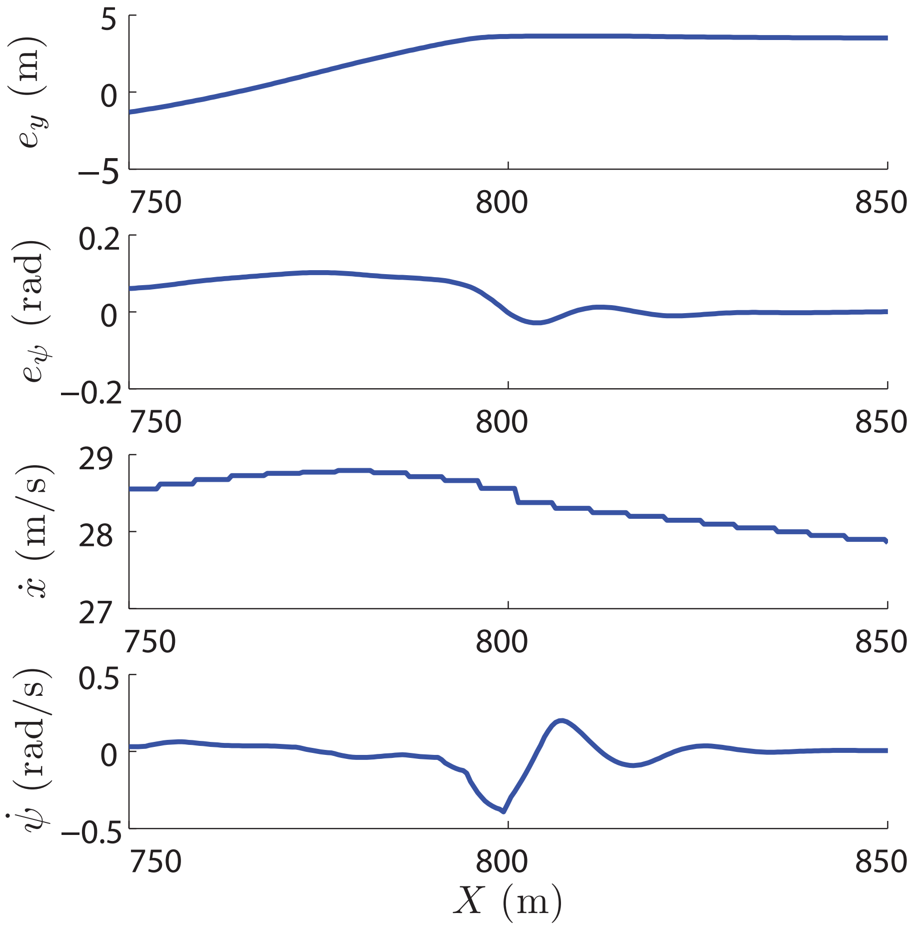


Figure 7.9: A plot of 4 vehicle states, $[e_y,\, e_\psi,\, \dot{x},\, \dot{\psi}]$, during the intervention depicted in Figure 7.8. The controller intervenes just before the vehicle reaches 800 m in the X-direction. The lateral offset, e_y, is constrained by the upper limit of the roadway, in this case 5 m.

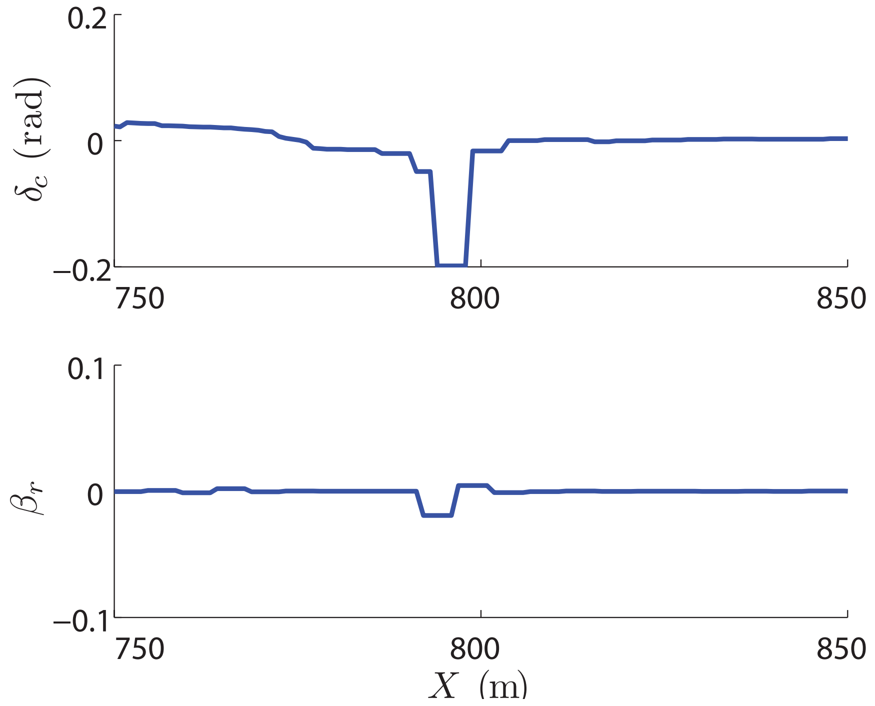


Figure 7.10: A plot of the inputs added by the controller during the intervention depicted in Figure 7.8. The controller briefly adds corrective steering to keep the vehicle on the road. In this scenario very little braking action was required.

7.2.2 Collision Avoidance

In this section we validate the behavior of the proposed active safety system. In Section 7.2.2 we first show a situation where the driver is attentive and capable of avoiding an obstacle . In this situation, the safety system correctly detects that the driver is capable of performing the driving task and does not intervene. Next, in Section 7.2.2, we demonstrate the ability of the adopted approach to detect critical situations and adequately assist the driver in avoiding accidents. We also demonstrate how the performance of the safety system can potentially be influenced if the system is complemented with a driver monitoring system that is capable of assessing whether the driver is distracted. Finally, in Section 7.2.2, we consider a scenario with multiple obstacles and demonstrate the ability of the controller to avoid collisions in such challenging situations.

For the results presented next, the estimation algorithm used in [29] is implemented to estimate parameters of the driver model (3.17) and the vehicle and design parameters in Tables 7.6 and **??** are used to implement the predictive controller (5.3).

The driver estimation algorithm adapts and updates the parameters of the driver model as new data becomes available. Since the estimation is conducted in nominal driving conditions, the resulting driver model is expected to be representative of the nominal behavior of the driver. The implications of this are discussed next, as the behavior of the suggested predictive controller is analyzed for the considered scenarios.

Attentive Driver

We first show a situation where the driver is attentive. Consider Figure 7.11 which shows a situation where an attentive driver is negotiating a curve and encounters an obstacle in the path.

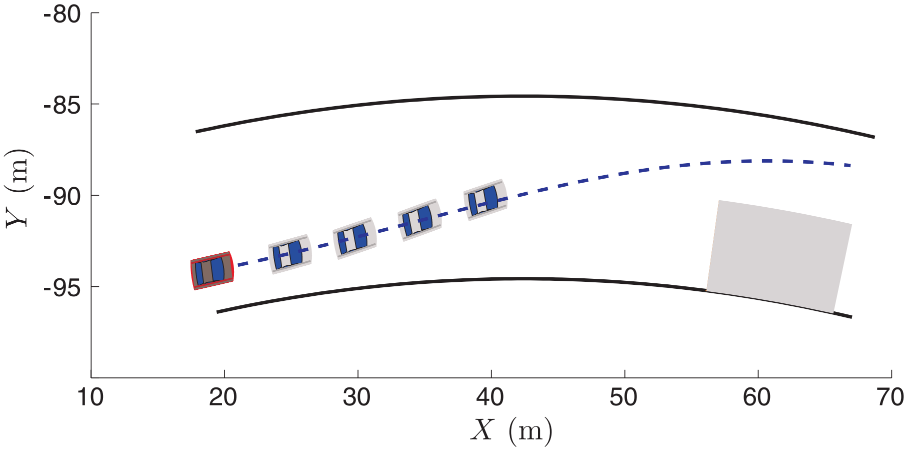


Figure 7.11: A situation where an attentive driver encounters an obstacle in the path. The driver is capable of avoiding the obstacle and for this reason, the MPC controller doesn't intervene.

The dashed line shows the path traversed by the driver. Clearly, the driver has no problems avoiding the obstacle. The shaded vehicles illustrate the trajectory that is predicted by the MPC controller when the vehicle is in the position shown with a darker color. In this situation the driver behavior, modeled by (3.17), is capable of avoiding the obstacle without assistance from the MPC controller. The MPC controller correctly predicts that the driver can maintain a safe trajectory and the decision to not intervene in this situation is correct.

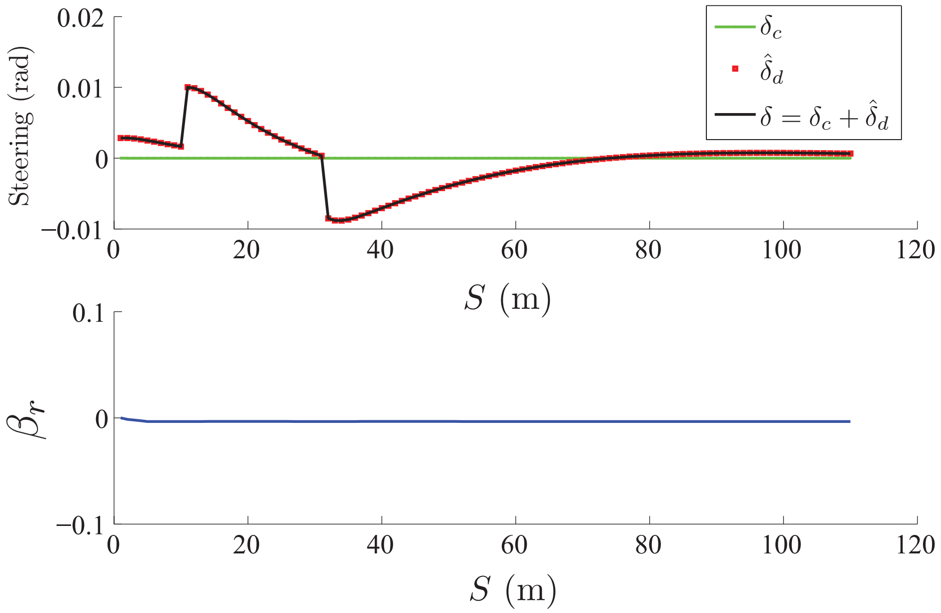


Figure 7.12: The inputs from the optimization, $[\delta_c, \beta_r]$, the driver model, $\hat{\delta}_d$, and the closed-loop steering, δ.

The final vehicle trajectory, along with the box geometry of the vehicle, is shown in Figure 7.11 and the inputs in Figure 7.12. The discrete jumps in inputs from the driver model, $\hat{\delta}_d$, are from the moving of the preview point to account for the obstacle. For an active safety system it is important to not intervene when it is not necessary. This simulation shows the behavior when the driver is able to maintain safety himself. The optimal solution from the controller is zero and an intervening action does not occur.

Distracted Driver

In this section we consider a scenario where the driver is distracted. Consider Figure 7.13

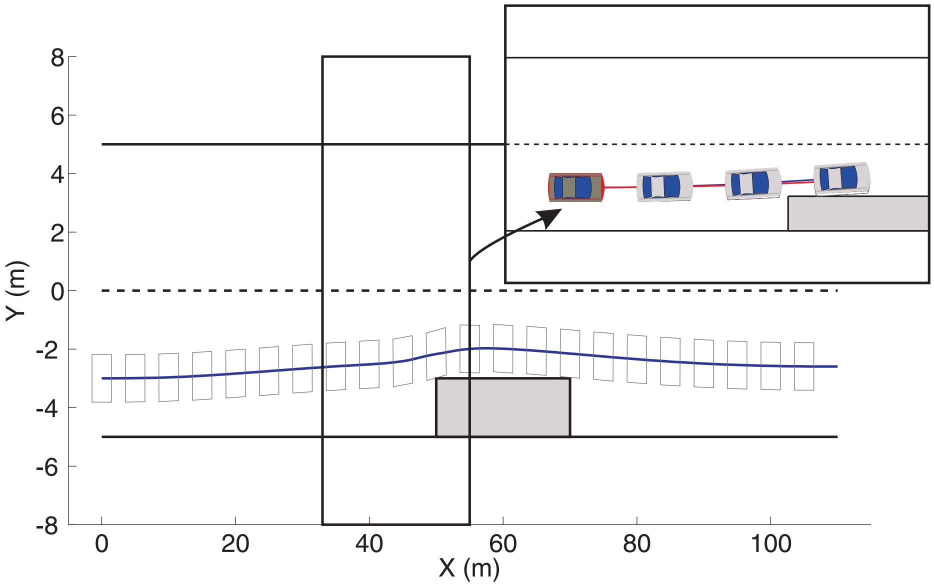


Figure 7.13: The closed-loop vehicle trajectory where the driver is assisted by the MPC controller. The controller detects that the nominal driver behavior is no longer sufficient to maintain a safe vehicle trajectory and intervenes in order to avoid the road side obstacle.

where the vehicle is approaching an obstacle and the driver is distracted. In this situation the driver doesn't account for the obstacle and instead just drives as if the object wasn't present. The inset in Figure 7.13 shows a comparison between the predicted trajectory obtained with the driver model (3.17) and the trajectory obtained when the driver is assisted by the MPC controller. These predictions are initiated at the position where the MPC controller first starts assisting the driver, i.e., when the two predicted trajectories no longer overlap. We note that, at this position, even though the driver is assumed attentive and will try to avoid the obstacle, the vehicle is already in a state where the driver would have to deviate from the nominal behavior, described by the model (3.17), in order to avoid the obstacle.

The MPC controller, therefore, assists the driver with as much control action as necessary to avoid the obstacle while minimizing the cost function (7.12a). Figure 7.14 shows the steering angle of the distracted driver and the controller's corrective steering angle and braking force.

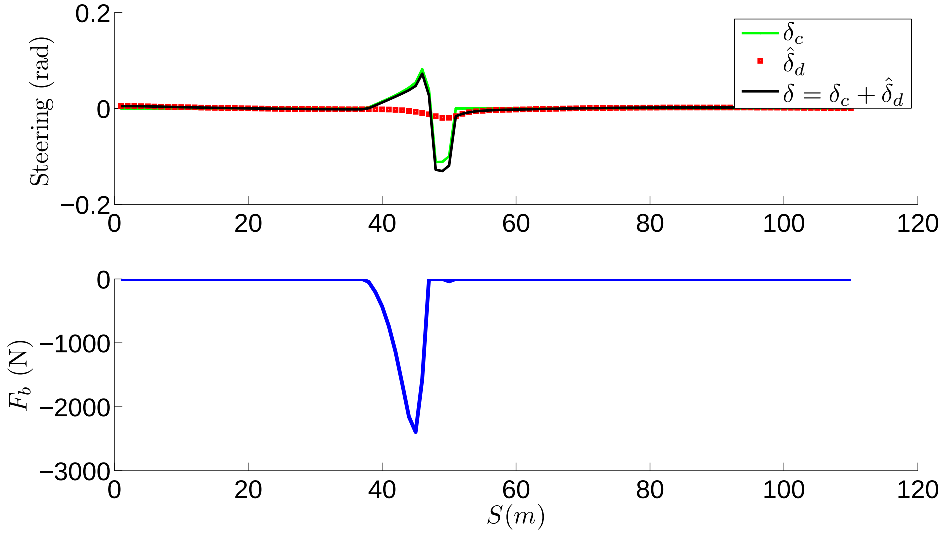


Figure 7.14: The inputs required from the safety system, the controller slightly steers and brakes the vehicle to avoid the roadside obstacle and smoothly gives back control to the driver once the obstacle has been avoided.

We note that the controller slightly steers and brakes the vehicle to avoid the roadside obstacle and smoothly gives back control to the driver once the obstacle has been avoided and the vehicle is again in a state where the driver is expected to be capable of avoiding violation of the safety constraints.

Figure 7.13 demonstrates the ability of the adopted MPC approach to intervene and avoid roadside objects without utilizing any information about driver's distraction. Consider the scenario shown in Figure 7.15 where the driver model in the MPC controller has been modified to account for a distracted driver.

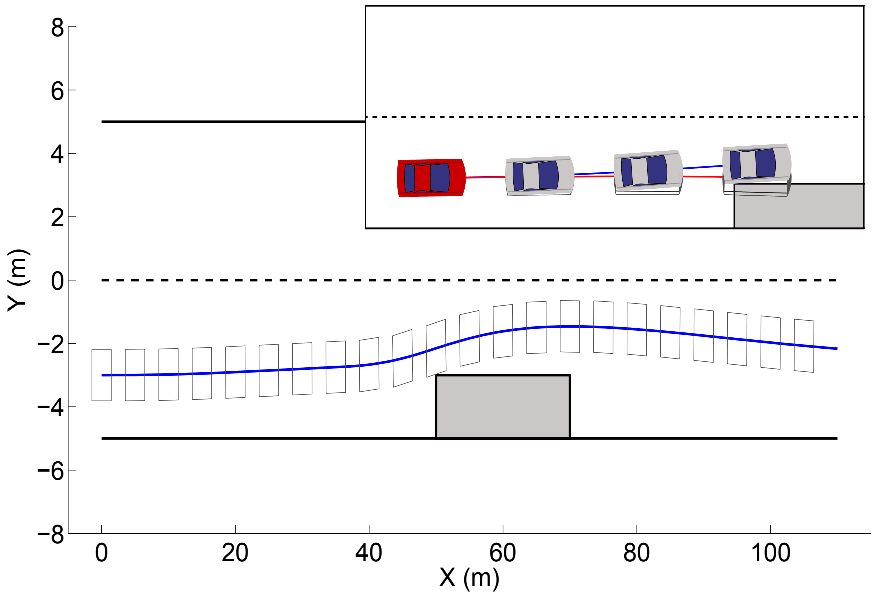


Figure 7.15: The closed-loop trajectory of the vehicle. The inset shows a snapshot of the time instant an intervention was required. In this case a driver monitoring system has detected that the driver is distracted and consequently, we note that the intervention is activated earlier.

Information about driver distraction might be obtained through a driver monitoring system like e.g. the monitoring system suggested in [89]. Figure 7.15 shows that the MPC controller is capable of avoiding the obstacle in this case and the inset shows how the distracted driver is expected to hit the obstacle. Nonetheless, we note in Figure 7.16 that the intervention is activated earlier in this case than in Figure 7.14 and that consequently the control signals are kept smaller.

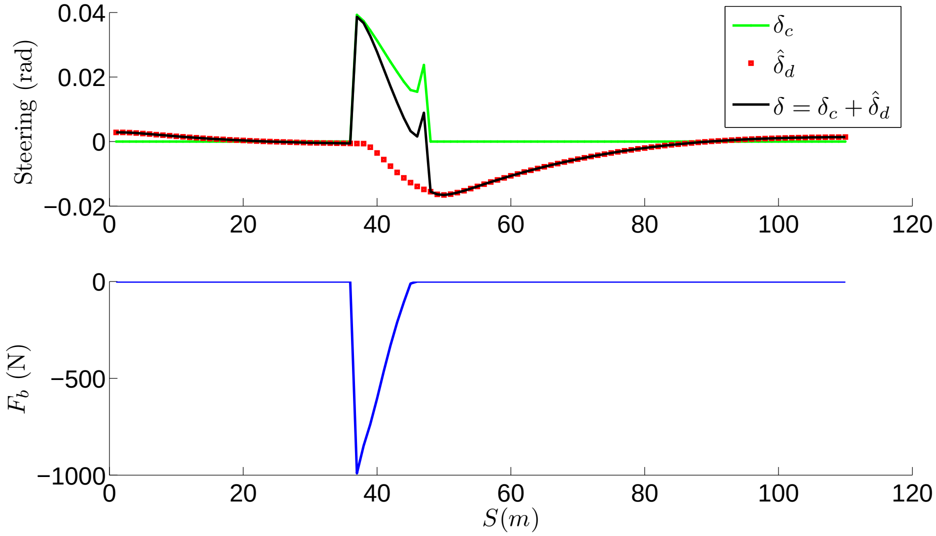


Figure 7.16: The input signals in the scenario shown in Figure 7.15. Since the intervention is issued early, the control action is smaller when compared to the input signals shown in Figure 7.14 .

Multiple Obstacles

Next we show a scenario that is more challenging. In Figures 7.17-7.18, a distracted driver is approaching multiple obstacles. The inset in Figure 7.17 shows a snapshot of the moment the second obstacle is encountered. A driver monitoring system has detected that the driver is distracted and the driver model in the MPC algorithm has been modified accordingly. Again we note that the controller intervenes to satisfy the constraints in order to minimize the control action, as prescribed by the cost function (7.12a). The results show the ability of the active safety system to successfully navigate around multiple obstacles while minimizing the interference to the driver.

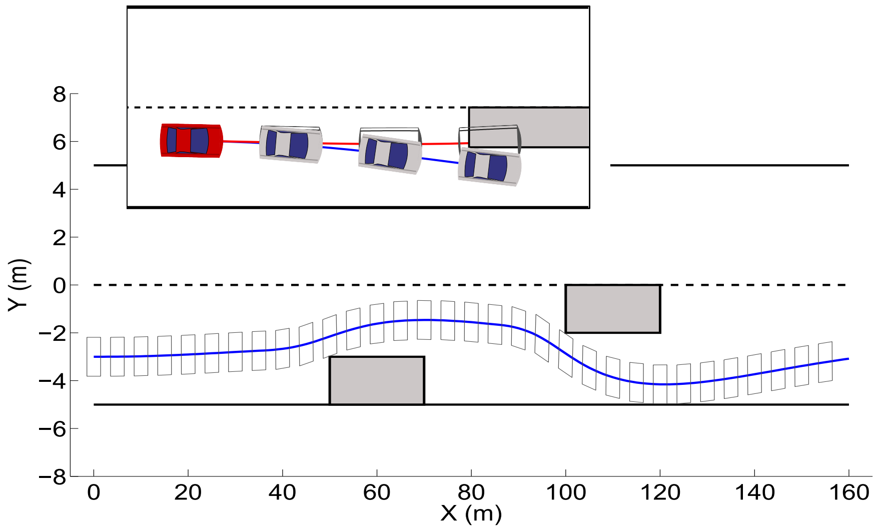


Figure 7.17: The closed-loop trajectory of the vehicle in a situation with multiple obstacles. The inset shows a snapshot of the moment the second obstacle was encountered and intervention was required.

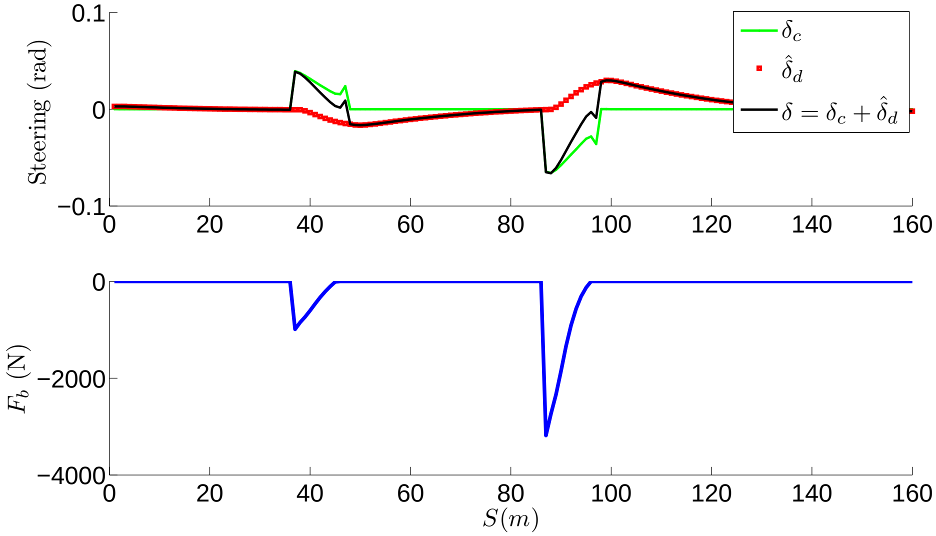


Figure 7.18: The inputs showing the corrective action, $[\delta_c, F_b]$, of the safety system required to keep the driver safe while navigating through multiple obstacles.

7.3 Experimental Results

In this section we present the results from real-world tests performed on an experimental test bed, a prototype Volvo S60 passenger vehicle. The experiments were performed at Volvo's headquarters in Gothenburg, Sweden. The controller was written in C-coded s-Functions and compiled to a dSPACE embedded control platform using Matlab's Realtime Workshop. The ECU is equipped with a 1 GHz DS1005 PPC Board. All the road information, such as lane boundaries, curvature, and road angles, as well as the vehicle state within the lane, is calculated using vision processing from an optical camera and communicated through the system using a CAN-interface. The parameters for the vehicle model, used by the model predictive controller, are stated in Table 7.6. The tuning parameters for the controller were adjusted for the real-time implementation and are reported in Table 7.4. To speed-up the computation time on the embedded hardware we make the following remark:

Remark 12 *The geometric constraint on e_y from Equation* (2.47) *is simplified to a tightened constraint on the center of gravity of the vehicle. A box constraint is set where $e_y \in [-0.7, 0.7]$.*

Table 7.4: Real-Time Design parameters

$u_{\max} =$ $[0.2\ \text{rad}, 0.5\ [-]]^T$	$H_p = H_c =$ 12	$\alpha_{\max} =$ 4°
$u_{\min}=$ $[-0.2\ \text{rad}, -0.5\ [-]]^T$	$T_s =$ $200\ \text{ms}$	$\alpha_{\min} =$ -4°
$\Delta u_{\max} =$ $[0.4\ \text{rad}, 1\ [-]]^T$	$Q =$ $diag(1,1)$	$e_{y_{\max}} =$ $0.7\ \text{m}$
$\Delta u_{\min} =$ $[-0.4\ \text{rad}, -1\ [-]]^T$	$R =$ $diag(1,1)$	$e_{y_{\min}} =$ $-0.7\ \text{m}$

7.3.1 Test 1: Excessive Speed in a Curve

The intent of this experiment was to capture the controller's performance for a scenario in which the driver is distracted and approaching a curve at an excessive speed. The driver brought the car to an initial velocity of 50 kph and maintained steady driving in the center of the lane. The driver then let go of the steering wheel and approached a curve. The resulting performance follows.

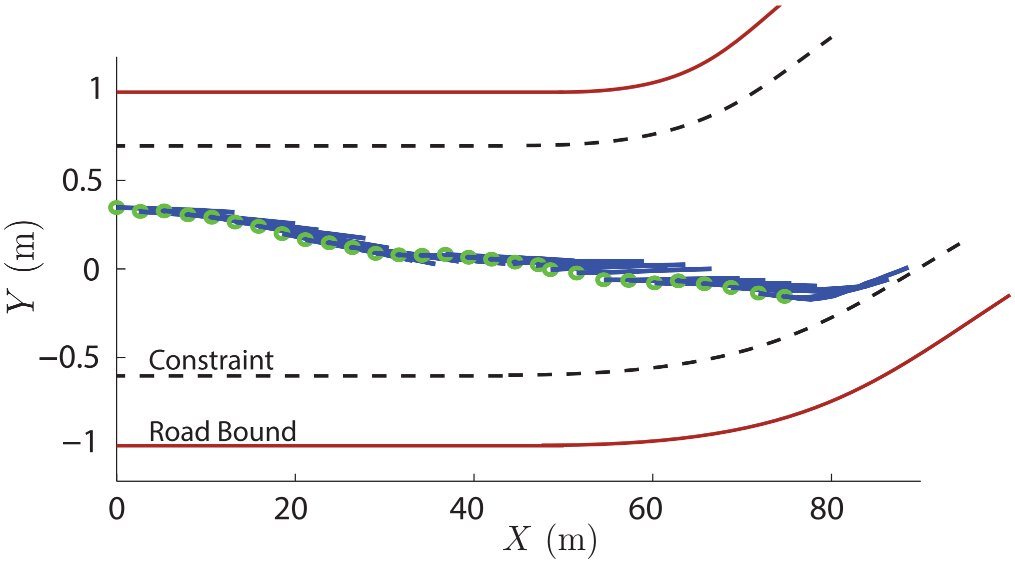


Figure 7.19: The paths predicted by the controller at each time instant. The initial condition is the circle marker, the dashed lines are the constraints imposed by the lane on the center of the vehicle, and the solid lines are the actual road bounds. This plot shows the time history leading up to the first intervention only at approximately $t = 6$ seconds.

Figure 7.19 shows the paths predicted by the controller at each time instant. As the vehicle approaches the curve it begins to stray out of the lane. As the lane constraints become active the controller adds corrective steering to keep the predicted trajectory within the lane, as can be seen by the curved paths at the lower lane constraint.

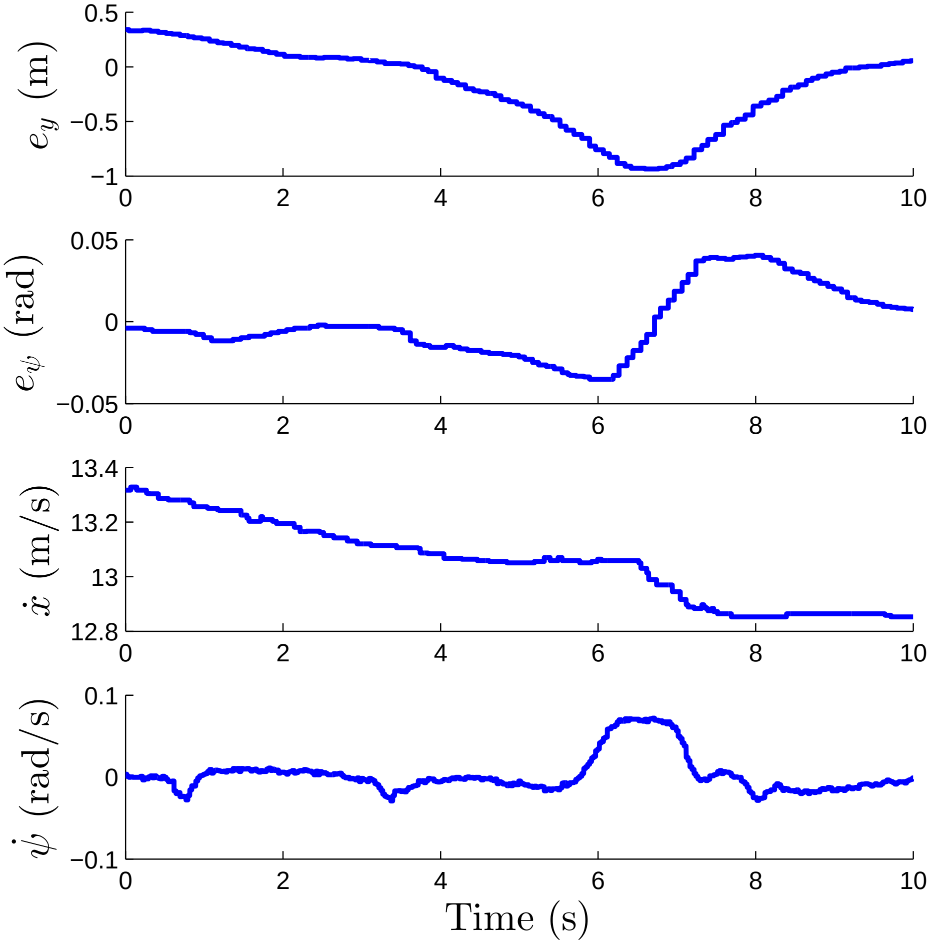


Figure 7.20: A plot of 4 states of the vehicle, $[e_y, e_\psi, \dot{x}, \dot{\psi}]$, during the experiment in section 7.3.1. This plot details the moment the controller intervenes at approximately 6 seconds, as shown in Figure 7.19. The velocity is slightly reduced and the lateral offset is corrected to keep the vehicle inside the lane. The associated inputs are shown in Figure 7.21.

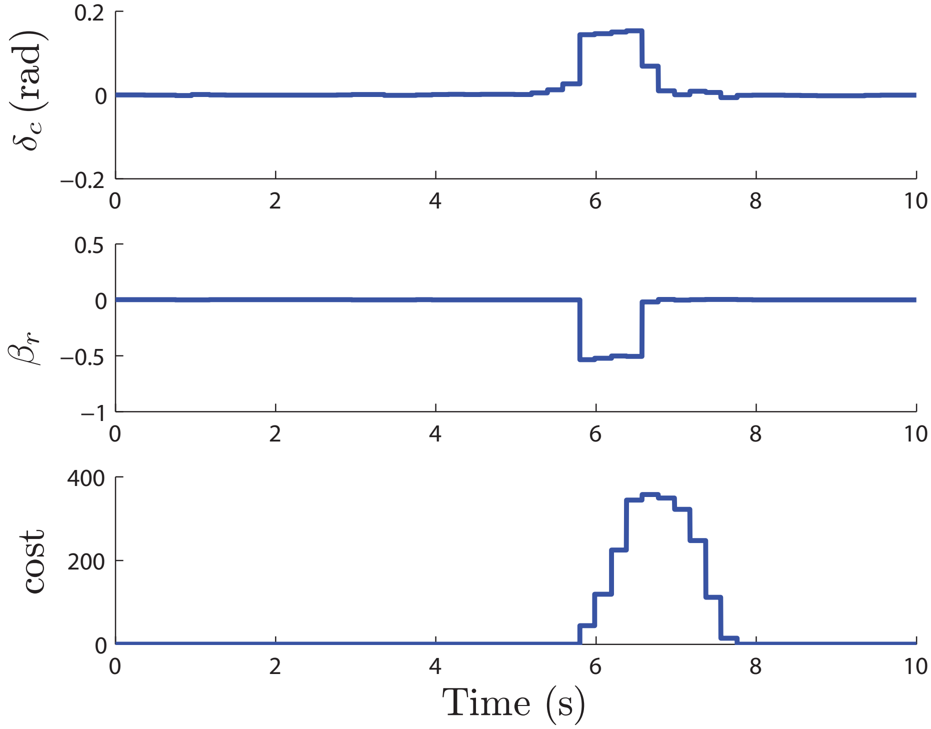


Figure 7.21: The corrective steering and braking action induced by the controller during the intervention depicted in Figures 7.19 and 7.20 during the experiment in section 7.3.1. The cost is also shown where the spike at approximately 6 seconds coincides with the soft constraint, ϵ, taking a small value. The inputs are shown correcting the constraint violation.

The associated states and inputs corresponding to Test 1 are shown in Figures 7.20 and 7.21, respectively. In Figure 7.20 a time-history of the states of the vehicle is shown. The lateral offset, e_y, is seen to approach the lower bound of the road as the controller intervenes. A jump in yaw angle, e_ψ, and yaw rate, $\dot{\psi}$, is seen, as well as a decrease in velocity, as the controller acts to keep the vehicle within the lane. The inputs calculated by the controller, both steering, δ_c, and braking, β_r, are shown in Figure 7.21. As the road boundary is approached the controller adds corrective steering and braking. The moment the vehicle is no longer threatened to depart the road the corrective action returns to zero. Videos of the experiments can be found online in [60].

7.3.2 Test 2: Unintentional Drifting

In this experiment we emulate a distracted driver by leaving the steering wheel unattended. The vehicle starts to stray off the road to the left into oncoming traffic. As the road boundary is approached the controller adds corrective steering and braking to keep the vehicle within the lane. The states are shown in Figure 7.22 and the inputs are shown in Figure 7.23. The intervention during this experiment is more subtle than the experiment in section 7.3.1 because the vehicle approached the upper bound at a shallower angle.

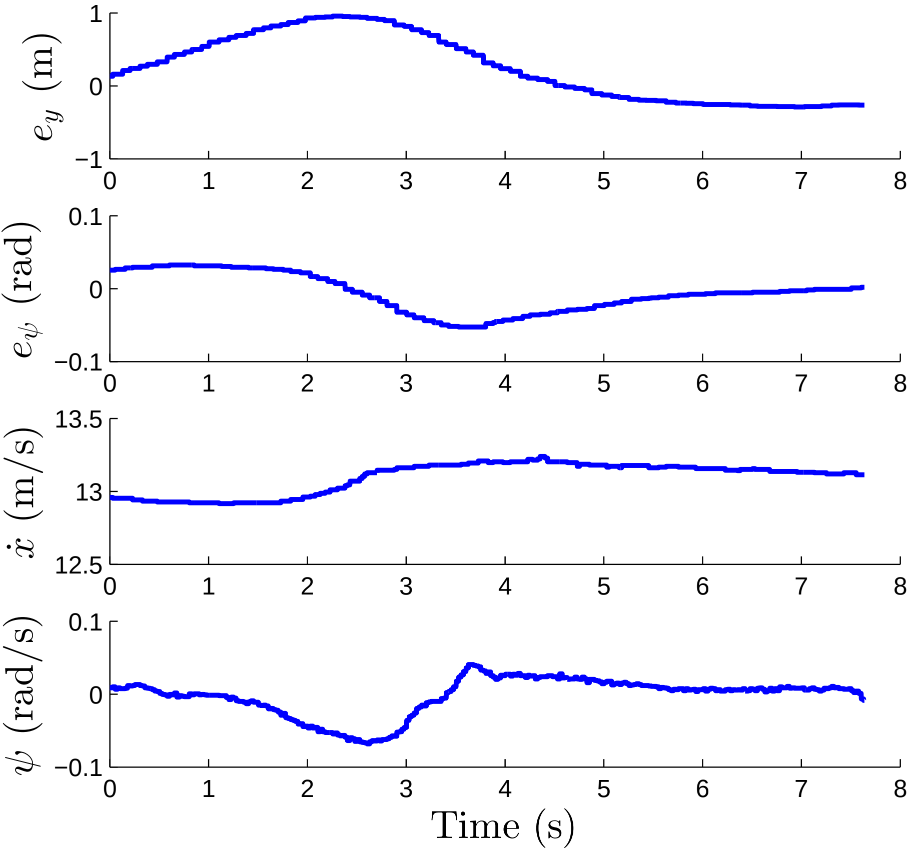


Figure 7.22: A plot of 4 states of the vehicle, $[e_y,\ e_\psi,\ \dot{x},\ \dot{\psi}]$, during the experiment in section 7.3.2. The vehicle approaches the upper bound on e_y and the controller corrects the steering and braking to remain inside the lane. The associated inputs are shown in Figure 7.23.

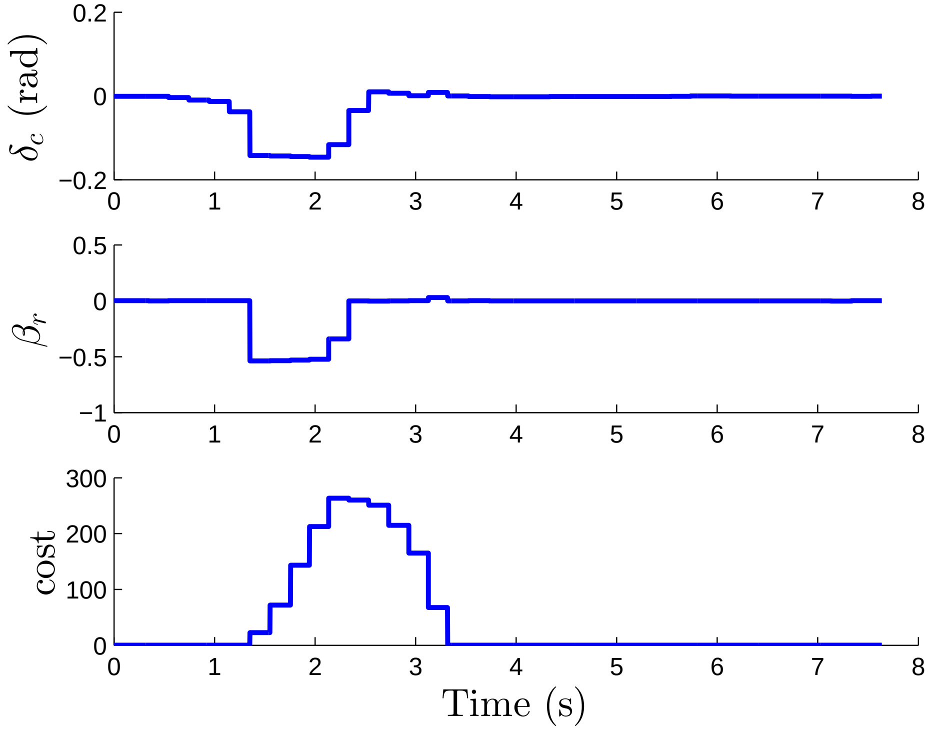


Figure 7.23: The corrective steering and braking action induced by the controller during the intervention depicted in Figure 7.22 during the experiment in section 7.3.2.

It is important to note here that the soft-constraint is violated (e_y exceeds 0.7 m). This is attributed to the time-delay in the system. The controller itself has a sample time of 200 ms, as noted in Table 7.4. The vision system also has a time-delay that is estimated to also be approximately 200 ms. This additive delay limits our vehicle speed to be around 50 kph for effective interventions.

7.4 Set-based Robust Active Safety

Consider the uncertain driver model presented in §3.3. It is our aim to provide robust guarantees of constraint satisfaction even in the presence of the driver's uncertain behavior. The uncertainty in the driver model is handled at the design stage by the computation of a robust invariant set that captures the spread of the vehicle's future trajectories given the uncertainty in the driver model. By tightening the constraints of the original nominal system we solve the optimization problem to yield the optimal corrective action needed to augment the driver's steering to ensure satisfaction of the safety constraints in the presence of the uncertain driver behavior. This approach is often called Tube-based MPC in the literature [73] and was presented in §5.3.1. The proposed controller is always active, which avoids the design of switching logic or the tuning of a sliding scale. In addition, since the proposed controller is designed to only apply the correcting control action that is necessary to avoid violation of the safety constraints, the intrusiveness of the safety application is kept minimal.

In this section we introduce a Robust Model Predictive Controller (RMPC) for use with the modeled uncertainty on the future driver inputs. The Integrated Framework of §7.1 is slightly modified to include a robust control law that can explicitly account for this uncertainty. The objective of the Robust Model Predictive Controller is to determine a corrective steering action to keep the driver safe in the presence of uncertain driver input. Consider the Linear Vehicle Model of §2.5, compactly written as,

$$\dot{\xi}(t) = \mathbf{A}\,\xi(t) + \mathbf{B}\,u(t) + \mathbf{E}\dot{\psi}_{\text{road}}(t), \tag{7.6}$$

and the Uncertain Driver Model of §3.3, written as,

$$\hat{\delta}_d(t) = \mathbf{F}\xi(t) + \mathbf{G}\Delta\psi_{\text{road}}(t) + w(t). \tag{7.7}$$

The closed-loop Driver-in-the-Loop Uncertain Model of §3.5 is obtained by setting,

$$u(t) = \hat{\delta}_d(t) + v(t), \tag{7.8}$$

where $v(t)$ is chosen as the Robust MPC law and the closed-loop system is compactly written as,

$$\dot{\xi}(t) = \mathbf{A}_{dm}\,\xi(t) + \mathbf{B}\,v(t) + \mathbf{E}_{dm}p(t) + \mathbf{B}w(t). \tag{7.9}$$

Denote the control sequence and the disturbance sequence as $\mathbf{v} = \{v_0, v_1, ..., v_{N-1}\}$ and $\mathbf{w} = \{w_0, w_1, ..., w_{N-1}\}$ for $t = 0...N-1$. Let $\Phi(t; \xi_0, \mathbf{v}, \mathbf{w})$ denote the solution of (7.9) at time t controlled by $\mathbf{v}$ when $\xi(0) = \xi_0$. Furthermore, let $\bar{\Phi}(t, \xi_0, \bar{\mathbf{u}})$ denote the solution of the *nominal* system

$$\dot{\bar{\xi}}(t) = \mathbf{A}_{dm}\,\bar{\xi}(t) + \mathbf{B}\,\bar{u}(t) + \mathbf{E}_{dm}p(t) \tag{7.10}$$

at time t controlled by the nominal control sequence $\bar{\mathbf{u}} = \{\bar{u}_0, \bar{u}_1, ..., \bar{u}_{N-1}\}$ when $\xi(0) = \xi_0$. Denote the predicted nominal state trajectory by $\bar{\xi} = \{\bar{\xi}_0, \bar{\xi}_1, ..., \bar{\xi}_{N-1}\}$.

By following Algorithm 1 in §5.3 a Robust Positive Invariant Set (definition 3) is calculated using the robust control law of Equation 5.24,

$$v(t) = \bar{u}(t) + K(\xi(t) - \bar{\xi}(t)), \tag{7.11}$$

where $\bar{u}(t)$ is the feed-forward term for the nominal system and $K(\xi(t) - \bar{\xi}(t))$ is the feed-back term acting on the error between the nominal state and the actual state where K is chosen according to the discussion in §5.3. This proposition states that if the control law (7.11) is used it will keep the states $\xi(t) = \Phi(t; \xi_0, \mathbf{v}, \mathbf{w})$ of the uncertain system (7.9) within the robust positive invariant set $\mathcal{Z}$ (definition 3) centered on the predicted state trajectory $\bar{\Phi}(t, \xi_0, \bar{\mathbf{u}})$ of the nominal system (7.10) for all admissible disturbance sequences $\mathbf{w}$. Now, we formulate the Robust Model Predictive Control Problem following the general formulation of problem $\mathbb{P}_{NL}$ in §5.1:

$$
\begin{aligned}
\mathbb{P}_{ROB}: \qquad & \min_{\bar{U},\varepsilon} \ J_N(\bar{U}(t), \Delta\bar{U}(t)) && \text{(7.12a)}\\
& s.t. \ \xi_{k+1,t} = f^{dm}(\xi_{k,t}, u_{k,t}), \quad k = t, ..., N-1, && \text{(7.12b)}\\
& \xi_{k,t} \in \bar{\mathcal{X}}, \quad k = t+1, ...t+N-1, && \text{(7.12c)}\\
& u_{k,t} \in \bar{\mathcal{U}}, \quad k = t, ..., t+N-1, && \text{(7.12d)}\\
& \xi_{t,t} = \xi(t), && \text{(7.12e)}\\
& \xi_{N,t} \in \bar{\mathcal{X}}_f, && \text{(7.12f)}
\end{aligned}
$$

where the model $f^{dm}(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ in Equation (7.12b) is model (7.10) with control law (7.11), i.e. the *nominal* system dynamics. The constraints in Equations (7.12c)-(7.12f) are the tightened constraint sets,

$$\bar{\mathcal{X}} = \mathcal{X} \ominus \mathcal{Z}, \quad \bar{\mathcal{U}} = \mathcal{U} \ominus K\mathcal{Z}. \tag{7.13}$$

Then, the corrective steering action calculated by problem $\mathbb{P}_{ROB}$ for the nominal system (7.10) and tightened constraints (7.13) will ensure persistent constraint satisfaction for the controlled uncertain system. The cost function to be minimized is,

$$J(\bar{U}, \Delta\bar{U}) = \sum_{t=k}^{t+N-1} \|\bar{u}(t)\|_Q^2 + \|\Delta\bar{u}(t)\|_R^2 + \rho\varepsilon, \tag{7.14}$$

where $\bar{U} = \{\bar{u}_t, \bar{u}_{t+1}, ..., \bar{u}_{t+N}\}$ is the vector of optimal corrective actions, $\Delta\bar{U} = \{\Delta\bar{u}_t, \Delta\bar{u}_{t+1}, ..., \Delta\bar{u}_{t+N}\}$ is the vector of optimal change rate of corrective actions, and the slack variable ε has been added to soften the state constraints, as detailed in §2.6. Q and R are weights of appropriate dimension.

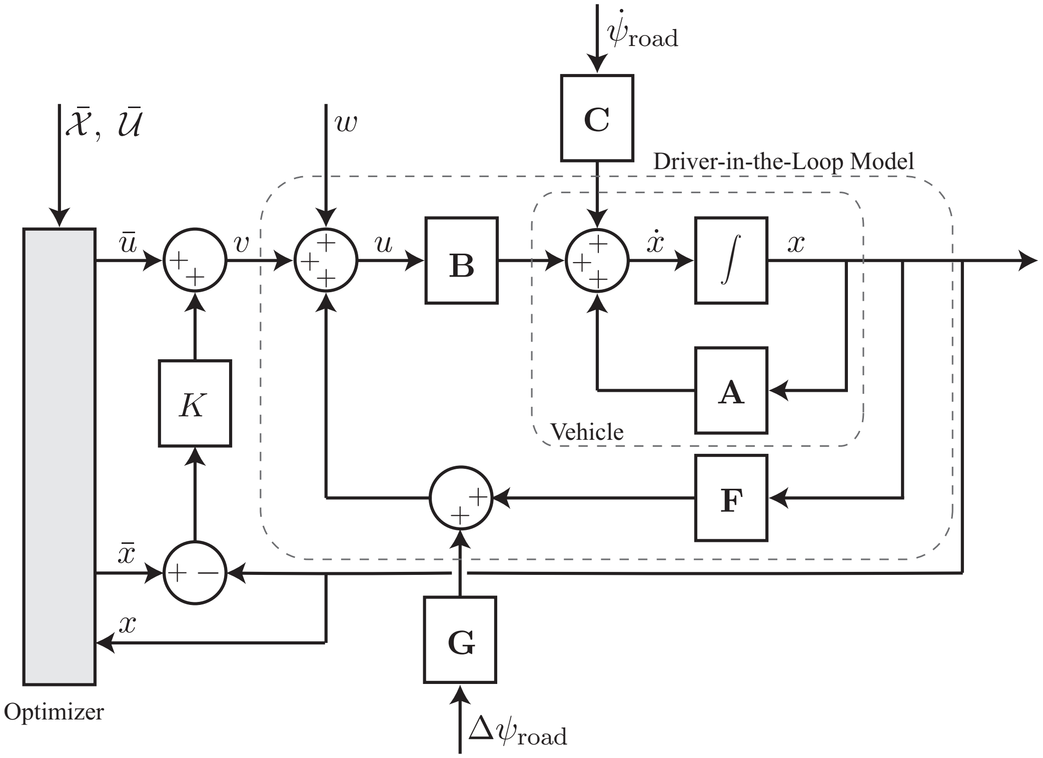


Figure 7.24: Block diagram illustrating the structure of the simulated system. The actual plant is not shown.

7.4.1 Simulation Results

In this section the results from simulations are presented. The model predictive control problem is solved using Tomlab/NPSOL at each time step. The off-line analyses to determine the robust invariant sets and solve for the tightened constraints was done by running Algorithm 1 in Matlab using MPT Toolbox. Table 7.6 lists the parameters used in the simulations. Two scenarios are considered, where the

1. driver approaches an obstacle on the right of the roadway,

2. driver navigates between two obstacles obstructing the lane.

Table 7.5: Vehicle parameters

$m =$ 2050 kg	$C_{\alpha i}$ $80,000$	$Q =$ 1	$\alpha_{\max} =$ $4°$	$e_{y_{\max}} =$ 0 m	$\bar{u}_{\max} =$ 0.16
$I_z =$ 3344 kgm^2	$N =$ 15	$R =$ 1 m	$\alpha_{\min} =$ $-4°$	$e_{y_{\min}} =$ -5 m	$\bar{u}_{\min}$ -0.16
$\mu =$ 1.0	$T_s =$ 50 ms	$\rho =$ 10^4	$\bar{e}_{y_{\max}} =$ -0.365 m	$\bar{e}_{y_{\min}} =$ -4.635 m	$\mathcal{W} =$ $[-0.1, 0.1]$

Single Obstacle

Figure 7.32 captures a snapshot of the moment the model predictive controller must add corrective steering action. Two trajectories are shown. Trajectory 2 is the one predicted by the nominal driver model only, in Equation (3.17), and is depicted by the vehicles in outline. This is the expected steering input predicted for the driver. Clearly, the vehicle is predicted to collide with the obstacle, denoted by the constraint $\mathcal{X}$. Trajectory 1 is the corrected trajectory and ensures satisfaction for the tightened constraints $\bar{\mathcal{X}}$ for any disturbance in the predicted driver model, $w \in \mathcal{W}$. The tightened constraints are shown in dashes and the drawn vehicles show the predicted trajectory with the corrective action. Figure 7.26 plots the inputs for Trajectory 1 and 2. $\bar{u}$ is the added corrective action from the solution to the nominal optimization problem, $\hat{\delta}_d$ is the nominal driver model from Trajectory 2, and u is the final augmented steering from Trajectory 1.

Multiple Obstacles

In this section we simulate the proposed controller during a scenario where the vehicle approaches two obstacles. Similar to the scenario detailed in section 7.4.1, the vehicle encounters an obstacle in the road and the controller must intervene to keep the driver safe. Immediately after the first obstacle the vehicle encounters a second, an intervention is again needed to ensure the safety constraints are satisfied.

In Figure 7.27 the nominal trajectory, $\bar{x}$, as well as the actual disturbed

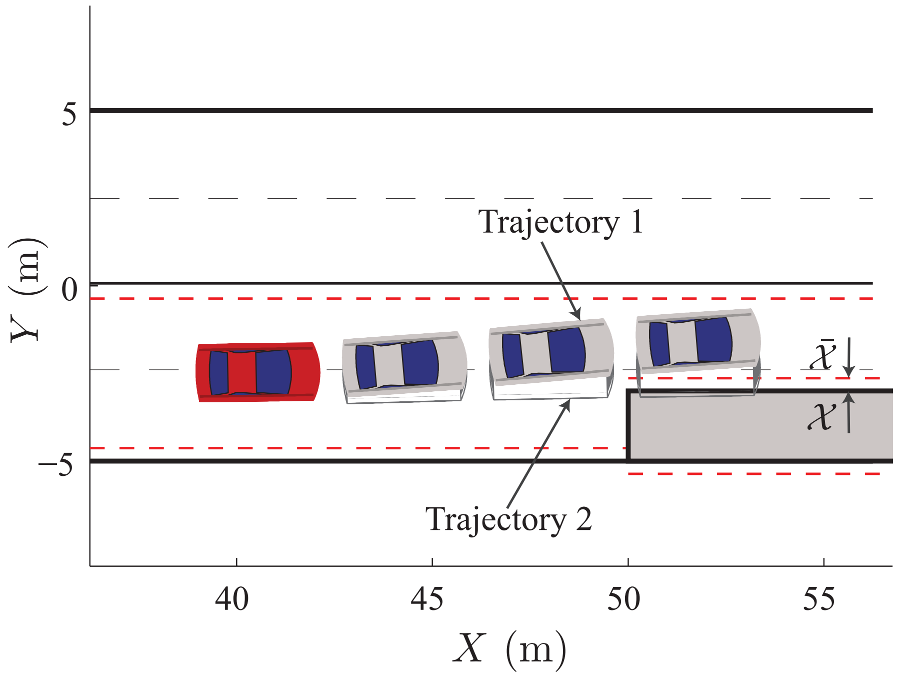


Figure 7.25: Trajectory 1 is assisted by the controller to keep the driver safe. Trajectory 2 is the expected driver input and collides with the obstacle.

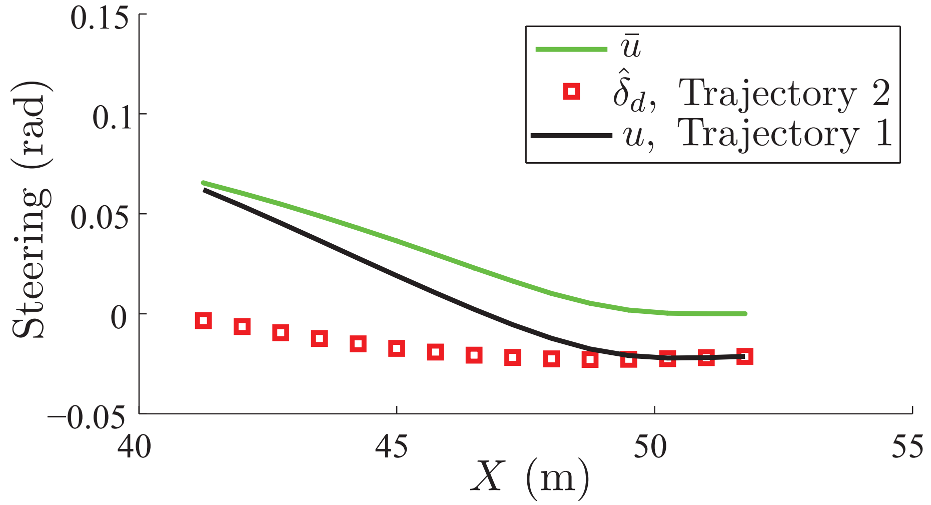


Figure 7.26: The inputs in Trajectory 1 and 2 as well as the corrective action $\bar{u}$ determined by the controller.

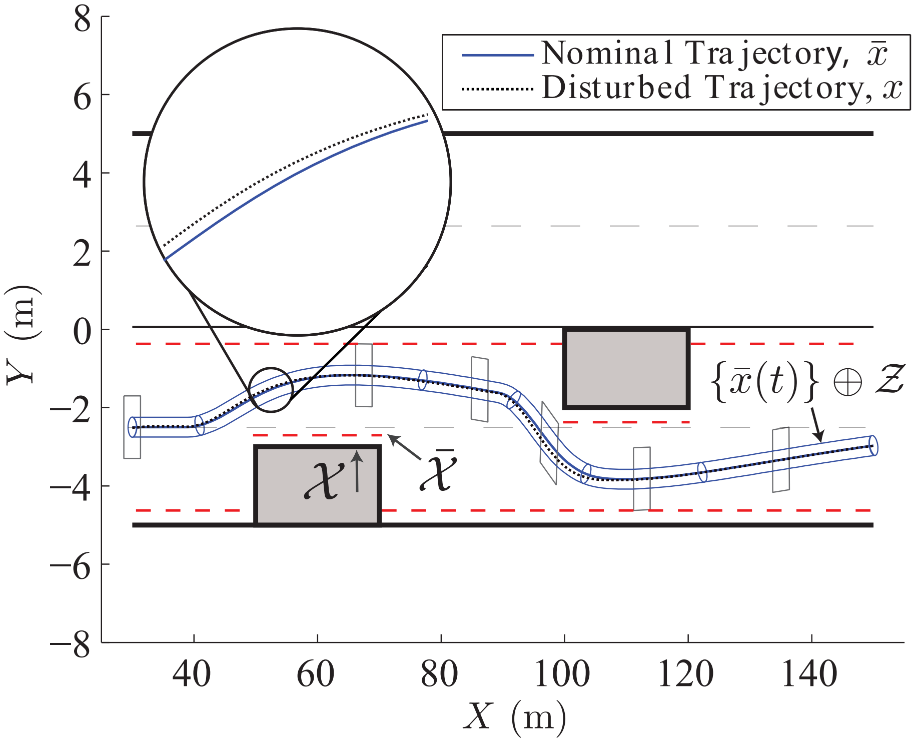


Figure 7.27: A plot of the nominal trajectory, $\bar{x}$, the disturbed trajectory, x, and a projection of the robust invariant set along the nominal trajectory.

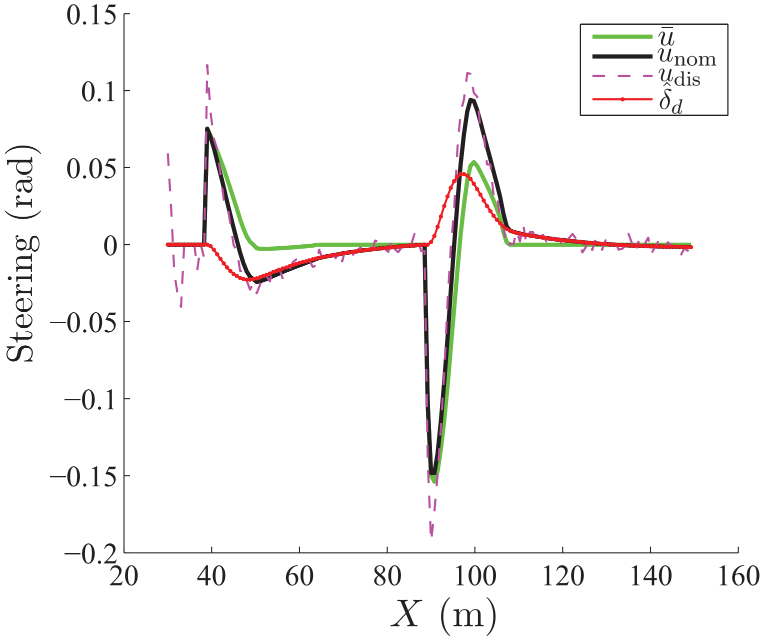


Figure 7.28: The inputs showing the corrective action from the controller, $\bar{u}$, the input from the driver model augmented with the controller action, u_{nom}, the input with additive disturbance, u_{dis}, and the steering from the driver model, $\hat{\delta}_d$.

trajectory, x, are shown. Boxes are plotted along the trajectory to show the geometry of the vehicle at various points in time. In addition, a sketch $\mathcal{Z}^{e_y} = \text{proj}_{e_y}(\mathcal{Z})$ is plotted to illustrate the size of the robust invariant set in the e_y dimension. The tightened constraints $\bar{\mathcal{X}}$ are shown in dashed lines. In Figure 7.28 the calculated inputs are shown for the scenario presented in Figure 7.27. $\hat{\delta}_d$ is the nominal steering angle determined by the driver model, $\bar{u}$ is the corrective action calculated by the nominal MPC problem, and u_{nom} is the final augmented input for the nominal trajectory, $\bar{x}$. Further, the disturbed input, u_{dis}, is plotted. This trajectory is detailed in the magnification in Figure 7.28. It is clear to see the disturbed trajectory is contained within the robust invariant set around the nominal state trajectory.

7.5 Stochastic Robust Active Safety

Consider the stochastic driver model presented in §3.4. It is our aim to provide robust guarantees of constraint satisfaction even in the presence of the driver's uncertain behavior. The uncertainty in the driver model is handled at the design stage by the computation of an upper bound on the propagation of the uncertain disturbance. By tightening the constraints of the original nominal system we solve the optimization problem to yield the optimal corrective action needed to augment the driver's steering to ensure satisfaction of the safety constraints, to a given probability, in the presence of the uncertain driver behavior. The proposed controller is always active, which avoids the design of switching logic or the tuning of a sliding scale. In addition, since the proposed controller is designed to only apply the correcting control action that is necessary to avoid violation of the safety constraints, the intrusiveness of the safety application is kept minimal.

In this section we introduce a Stochastic Model Predictive Controller (SMPC) for use with the modeled uncertainty on the future driver inputs. The Integrated Framework of §7.1 is slightly modified to include a robust control law that can explicitly account for this uncertainty. The objective of the Stochastic Model Predictive Controller is to determine a corrective steering action to keep the driver safe in the presence of uncertain driver input. The problem formulation was presented in §5.3.2 in Problem $\mathbb{P}_{PROB}$. We can specify the probabilistic constraint in Equation (5.31e) as

$$g^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \; h = \begin{bmatrix} e_{y,\max} \\ e_{y,\min} \end{bmatrix}$$

where $\Pr\{g^T \xi_{k+1} \leq h\} \geq p$, $\xi = [e_y, \, \dot{e}_y, \, e_\psi, \, \dot{e}_\psi]$ and we set $p = 0.99$. Following the results presented in §5.3.2 the probabilistic constraints are converted to linear constraints using information on the disturbance and the problem is able to be solved as a standard QP. Simulation results detailing the behavior of the stochastic MPC are presented next.

7.5.1 Simulation Results

In this section we present the simulation results. The parameters to implement the controller are listed in Table 7.6.

Table 7.6: Simulation Parameters

Parameter	Value	Units	Parameter	Value	Units
m	2050	kg	N	15	-
I_z	3344	kg.m^2	K_y	-0.005	-
e_y	[2.5, -2.5]	m	K_ψ	-0.2	-
C_α	80,000	-	μ	1	-
l_f	1.43	m	T_s	50	ms
l_r	1.47	m	Σ	0.1	rad
v	[0.2,-0.2]	rad	t_{lp}	2	sec

We detail two aspects of the proposed framework. In section 7.5.1 a *snapshot* of the prediction is captured. This illustrates the tightened constraints over the prediction horizon as well as the effect the corrective action has on the vehicle. Section 7.5.1 depicts the *actual* trajectories taken by the vehicle while the controller adds corrective action to avoid obstacles. Ten trajectories are simulated with the disturbance sampled from the normal distribution $\mathcal{N}(\bar{w}(t), \Sigma)$ in Equation (3.22), as well as the worst-case disturbances with $\bar{w} \pm 3\Sigma$ (99.7%). These simulations show the statistical behavior of the proposed framework in the presence of uncertainties.

Snapshot of Prediction

In this section a snapshot of the prediction is detailed as the vehicle approaches a roadside obstacle. Figure 7.29 illustrates this scenario. The current vehicle position is marked by ξ_{curr} and the predicted vehicles extend over the horizon and is marked by ξ_{pred}. Trajectory 2 shows the prediction of the vehicle position controlled by the nominal driver model only (3.17) and is drawn in boxed-outline. Trajectory 1 shows the corrected trajectory calculated by the model predictive controller. The corrective action has been added to satisfy the tightened safety constraints (5.35). The constraints are

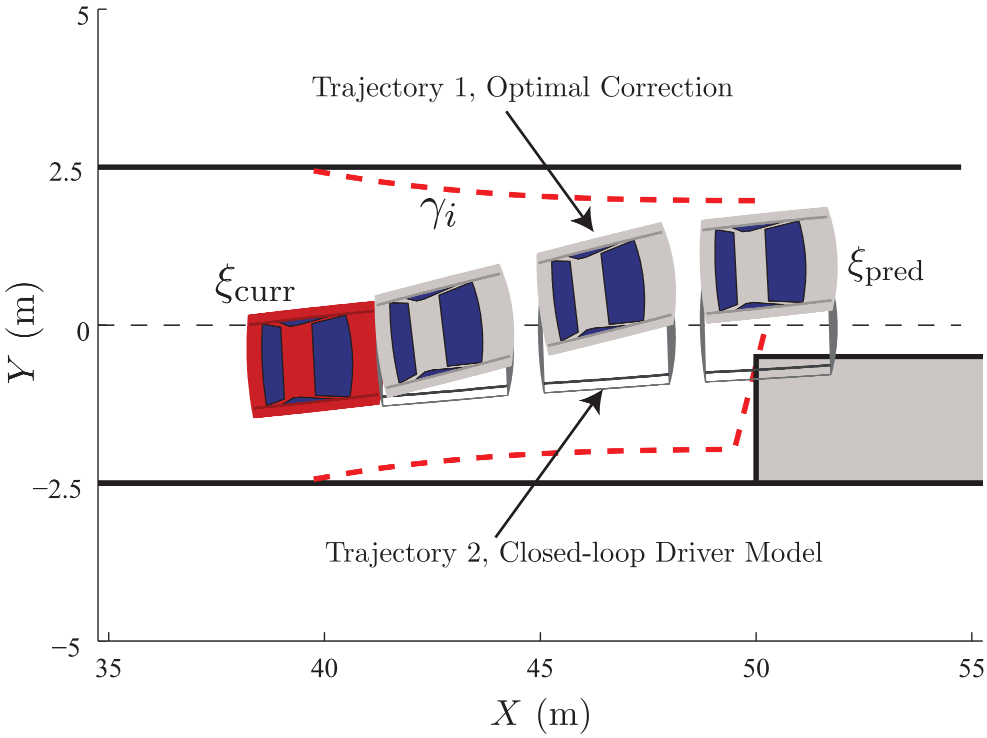


Figure 7.29: A snapshot of the prediction where the vehicle encounters an obstacle. The tightened constraints are labeled by γ_i. The vehicles in boxed-outline, Trajectory 2, is the trajectory with the driver model only and Trajectory 1 depicts the corrected trajectory to satisfy the safety constraints.

tightened by amount γ_i, $i = 1...N$ over the horizon and is drawn as the dashed lines in Figure 7.29.

Figure 7.30 plots the inputs for the scenario captured in Figure 7.29. The input $v = K\xi + c$ is the variable to be minimized and c is the optimization variable able to add the corrective action to satisfy the safety constraints. The driver model is $\hat{\delta}_d$ and it is clear the driver alone is not capable of avoiding the obstacle, as noted by Trajectory 2.

Figure 7.31 plots the tightened constraints over the prediction horizon, obtained by applying Equation (5.36) for $i = 1...N$.

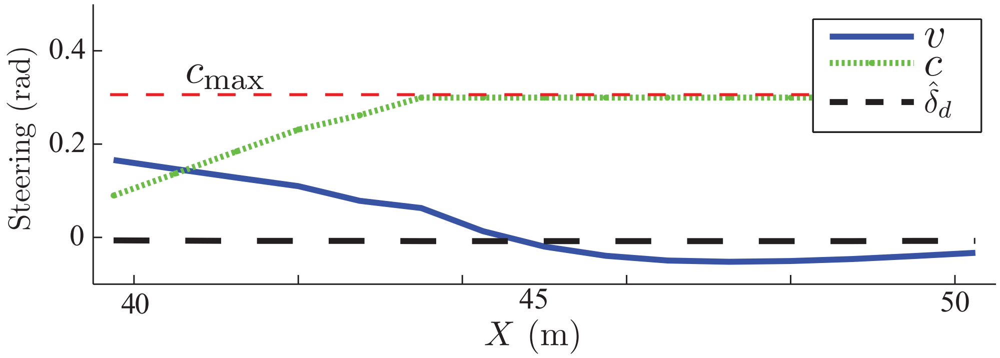


Figure 7.30: The inputs for the snapshot depicted in Figure 7.29. $v = K\xi + c$ is the minimization variable, c is the optimization variable, and $\hat{\delta}_d$ is the nominal driver input.

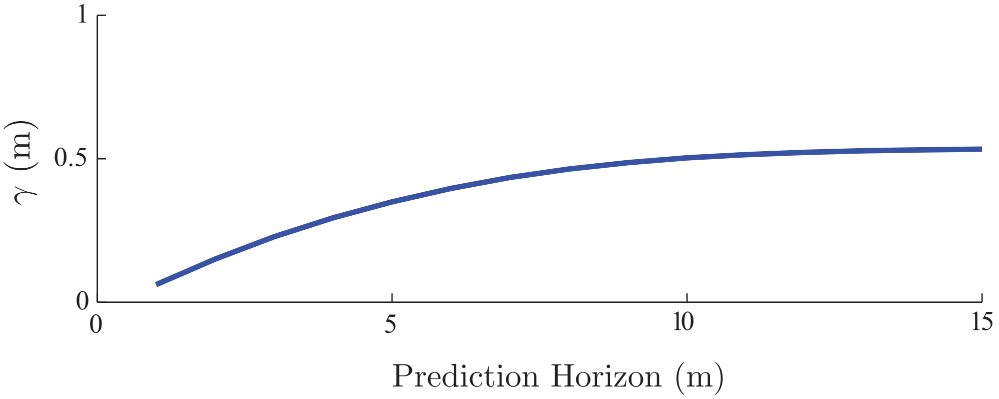


Figure 7.31: The upper bound γ on the disturbance propagation over the prediction horizon.

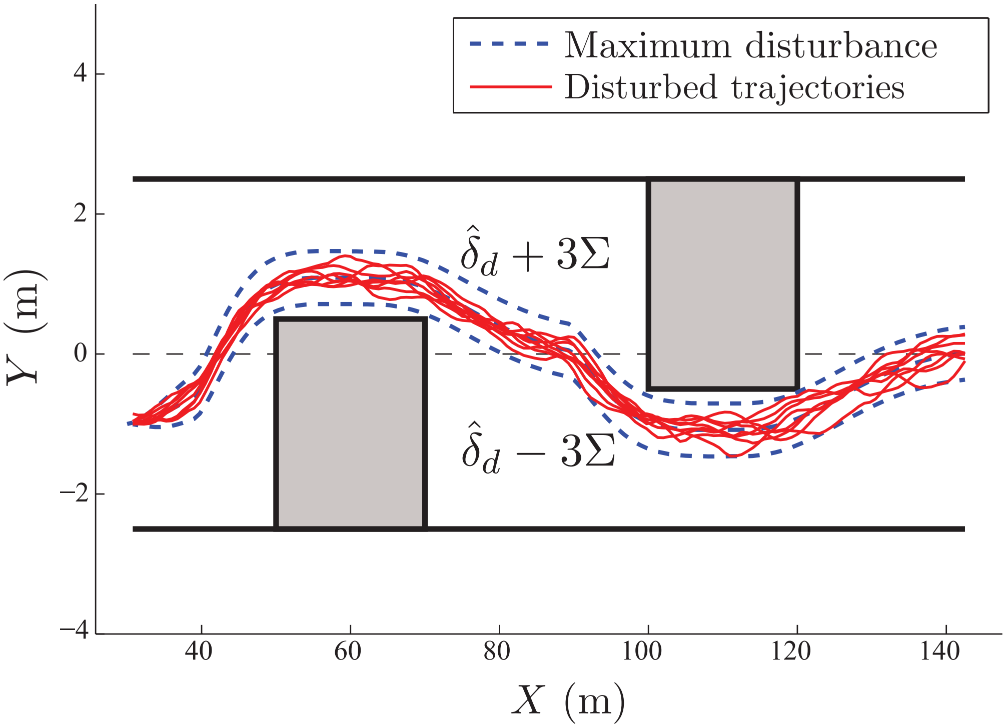


Figure 7.32: Plot of randomly simulated trajectories depicting the statistical behavior of the controller. The worst-case trajectories are marked in dashed line.

Sampled Trajectories

In this section we illustrate the statistical behavior of the proposed controller by plotting ten randomly disturbed trajectories, drawn from the normal distribution $\mathcal{N}(\bar{w}(t), \Sigma)$, as well as the worst-case $\bar{w} \pm 3\Sigma$ trajectories. Figure 7.32 shows the tightened constraints create a probabilistic satisfaction for the original constraints and the vehicle successfully avoids the obstacles.

Note 9 *In this scenario the obstacles have been widened compared to Figure 7.29 and the constraint has been placed on the vehicle center-of-gravity to more clearly depict the trajectories.*

Appendix A

Experimental Vehicle Setup

Here we introduce the experimental vehicle setup. This thesis contains experimental results from various experimenting platforms, including those from Ford, Volvo, and Hyundai, and therefore only a general description will be given. Figure A.1 shows a diagram of the experimental testing vehicle.

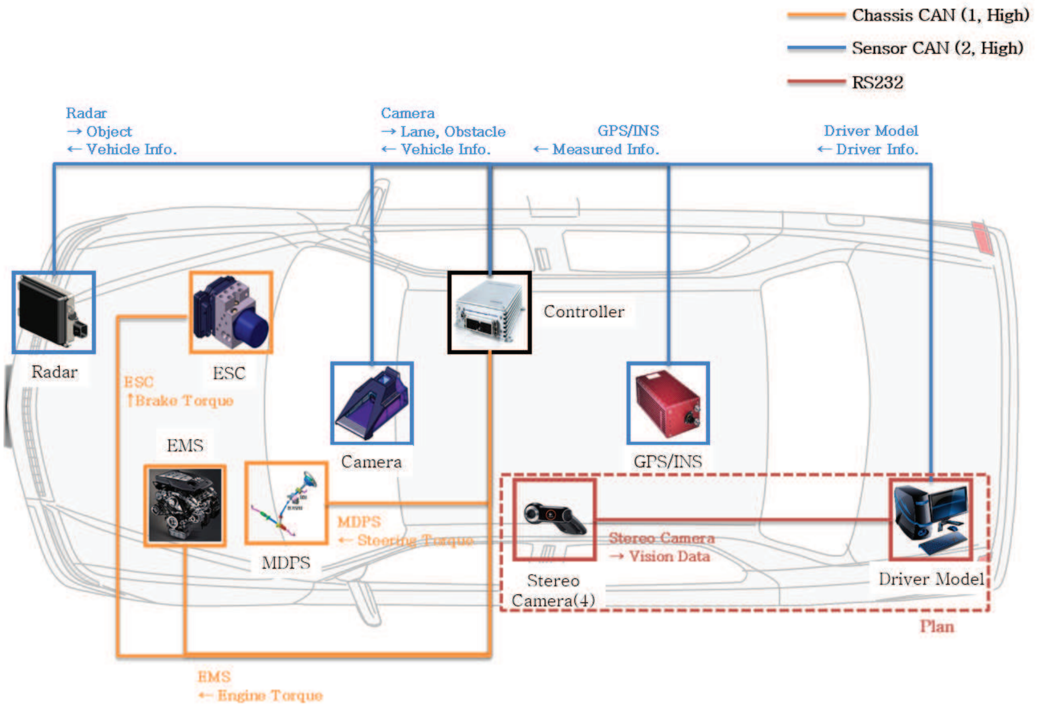


Figure A.1: A diagram of the experimental setup.

We used a dSPACE Rapid Prototyping Control Unit to run real-time control. Various sensors were utilized depending on the control task at hand. These sensors include lane recognition from Mobileye [1], GPS and IMU from Oxford Technical Solutions, and radar.

The MPC controllers have been tested on a passenger car, with a mass of approximately 2050 Kg and an inertia of approximately 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 GPS receiver, Inertial Measurement Unit (IMU), and a 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 GPS, 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.

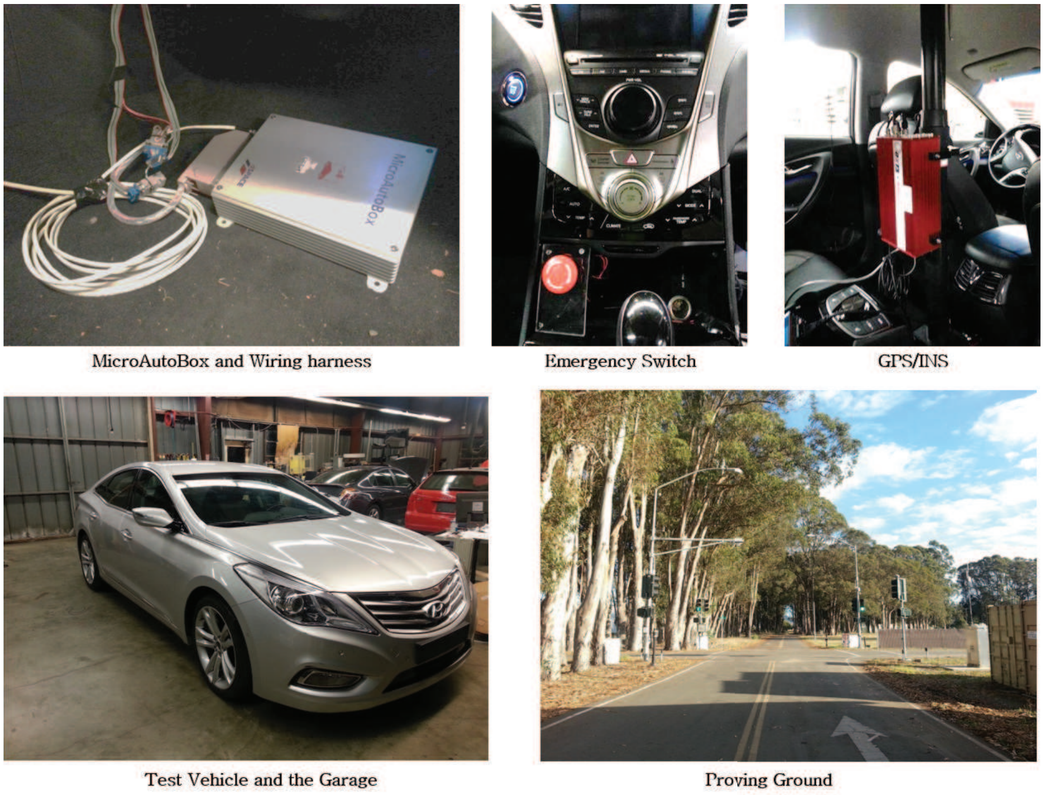
The car was equipped with an Active Front Steering (AFS) and Differential Braking system which utilizes an electric drive motor to change the relation between the hand steering wheel and road wheel angles. This is done independently from the hand wheel position, thus the front road wheel angle is obtained 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 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 not 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. All the data about the actuator cannot be disclosed. The sensor, the dSPACE Autobox and the actuators communicate through a CAN bus. Measurement noises/bias: Compared to a dual antenna setup, in which the vehicle orientation can be set and maintained (even during vehicle standing still), a single antenna system have to learn the vehicle orientation and/or coordinate during vehicle motion. Therefore, a small orientation error may occur in the single antenna setup. That is, while RT3002 knows if the vehicle, (in particular, the antenna), is traveling north, for example, it has to learn if the vehicle orientation is also pointing toward the north during this travel. In the experimental results, we will see that an orientation bias/error may lead to some additional transient dynamics as well as a steady state error in vehicle orientation. The results also show that the overall performance in lane change behavior is robust to this orientation bias.

The orientation bias discussed above can be eliminated if a dual antenna system is used.

Before starting the experiment, an initialization procedure was required to set up the OTS sensor. In this warm-up phase the vehicle has to be driven and dynamic maneuvers are required in order to estimate the bias in the accelerometers and in the angular rate sensors. Since we used a single antenna sensor, the best accuracy in the yaw angle measurement is achieved after a warm up phase. In a dual antenna sensor, the difference between the position of the two antennas is used to compute directly the yaw angle and the measurement is not affected by errors due to an imperfect initialization. After the OTS sensor has been initialized, an inertial co-ordinate frame has to be defined. This is done when the experiment starts.

The test is initiated by the driver with a button. When the button is pushed, the inertial frame 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. The origin of the inertial frame is fixed at the current vehicle position, while the X, Y axis are oriented as the current vehicle longitudinal and lateral axis respectively. The position in the inertial frame is then derived from the GPS coordinates by a coordinate transformation. The path reference is given in the defined inertial frame.

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.



MicroAutoBox and Wiring harness

Emergency Switch

GPS/INS

Test Vehicle and the Garage

Proving Ground

Figure A.2: Pictures of the Hyundai setup.

Bibliography

[1] Mobileye: http://www.mobileye.com/en-us/.

[2] *Model predictive control toolbox.* The MathWorks, Inc., Natick, MA, 2005.

[3] Mohammad Ali, Paolo Falcone, and Jonas Sjöberg. Threat Assessment Design under Driver Parameter Uncertainty. *In Proc. IEEE Conference on Decision and Control*, 2012.

[4] S.J. Anderson, S.C. Peters, T.E. Pilutti, and K. Iagnemma. An optimal-control-based framework for trajectory planning , threat assessment , and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *Int. J. Vehicle Autonomous Systems*, 8:190–216, 2010.

[5] Pongtep Angkititrakul, Ryuta Terashima, and Toshihiro Wakita. On the use of stochastic driver behavior model in lane departure warning. *Transactions on Intelligent Transportation Systems*, 12(1), mar 2011.

[6] E Bakker, L Nyborg, and H Pacejka. Tyre Modeling for Use in Vehicle Dynamics Studies. *SAE*, 1989.

[7] E. Bakker, L. Nyborg, and H. B. Pacejka. Tyre modeling for use in vehicle dynamics studies. *SAE paper # 870421*, 1987.

[8] Maximilian Balandat. Constrained robust optimal trajectory tracking: Model predictive control approaches. Master's thesis, Technische Universitat Darmstadt, 2010.

[9] A. Bemporad and M. Morari. Robust model predictive control: A survey. *Lecture Nots in Control and Information Sciences*, 245(1999), 1999.

[10] M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robot. Auton. Syst.*, 32(1):1–16, July 2000.

[11] D. Bertsekas and I. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7(2), 1971.

[12] Francesco Biral, Mauro Da Lio, and Roberto Lot. Vehicle and driver modeling and threat assessment for driving support functions. *Atti XX Congresso dell'Associazione Italiana di Meccanica Teorica e Applicata*, 2011.

[13] F. Borrelli. Constrained optimal control of linear and hybrid systems. *Lecture Notes in Control and Information Sciences*, 290, 2003.

[14] F. Borrelli. *Introduction to Model Predictive Control. Lectures 9-10: Optimal Control.* 2009.

[15] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat. An MPC/hybrid system approach to traction control. *IEEE Trans. Control Systems Technology*, 14(3):541–552, May 2006.

[16] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat. MPC-based approach to active steering for autonomous vehicle systems. *Int. J. Vehicle Autonomous Systems*, 3(2/3/4):265–291, 2005.

[17] F. Borrelli, T. Keviczky, G. J. Balas, G. Stewart, K. Fregene, and D. Godbole. Hybrid decentralized control of large scale systems. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer Verlag, March 2005.

[18] M. Brannstrom, E. Coelingh, and J. Sjoberg. model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):658–669.

[19] C Cacciabue. *Modelling Driver Behaviour in Automotive Environments*. Springer, 2007.

[20] P.J. Campo and M. Morari. Robust model predictive control. pages 1021–1026, 1987.

[21] L. Chisci, P. Falugi, and G. Zappa. Gain-scheduling mpc of nonlinear systems. *International Journal of Robust and Nonlinear Control*, 13:295–308, 2003.

[22] M Distner, M Bengtsson, T Broberg, and L Jakobsson. City Safety - A System Addressing Rear-End Collisions at Low Speeds. *In Proc. 21st International Technical Conference on the Enhanced Safety of Vehicles*, 2009.

[23] A. Eidehall, J. Pohl, and F. Gustafsson. joint road geometry estimation and vehicle tracking. *Control Engineering Practice*, 15(12):1484–1494.

[24] A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark. toward autonomous collision avooidance by steering. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):84–94.

[25] N. Enache, S. Mammar, M. Netto, and B. Lusetti. Driver steering assistance for lane-departure avoidance based on hybrid automoata and composite lyapunov function. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):28–39.

[26] G. Erdogan, S. Hong, F. Borrelli, and K. Hedrick. Tire sensors for the measurement of slip angle and friction coefficient and their use in traction control systems. *SAE International Journal of Passenger Cars-Mechanical Systems*, june 2011.

[27] G. Erdogan, S. Hong, F. Borrelli, and K. Hedrick. Tire sensors for the measurement of slip angle and friction coefficient and their use in traction control systems. *SAE 2011 World Congress*, april 2011.

[28] P. Falcone. *Nonlinear Model Predictive Control for Autonomous Vehicles*. PhD thesis, Piazza Roma 21, 82100, Benevento, Italy, 2007.

[29] P Falcone, M Ali, and J Sjöberg. Predictive Threat Assessment via Reachability Analysis and Set Invariance Theory. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1352–1361, 2011.

[30] P. Falcone, F. Borrelli, J. Asgari, and E. Tseng. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International Journal of Robust and Nonlinear Control*, 18:862–875, 2007.

[31] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. A model predictive control approach for combined braking and steering in autonomous vehicles. *Submitted to the 15th Mediterranean Conference on Control and Automation (*available at http://www.grace.ing.unisannio.it/home/pfalcone*)*, 2007.

[32] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. on Control System Technology*, 15(3), 2007.

[33] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. Low complexity mpc schemes for integrated vehicle dynamics control problems. *9th International Symposium on Advanced Vehicle Control*, 2008.

[34] P. Falcone, F. Borrelli, J. Asgari, H. Eric Tseng, and D. Hrovat. A real-time model predictive control approach for autonomous active steering. In *Nonlinear Model Predictive Control for Fast Systems, Grenoble, France*, 2006.

[35] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat. Integrated braking and steering model predictive control approach in autonomous vehicles. *Fift IFAC Symposium on Advances of Automotive Control*, 2007.

[36] P. Falcone, E. Tufo, F. Borrelli, J. Asgari, and E. Tseng. Linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. *IEEE Conference on Decision and Control*, 46, 2007.

[37] P. Falcone, F. Borrelli M. Tufo, H. E. Tseng, and J. Asgari. Predictive autonomous vehicles: A linear time varying model predictive control approach. *46th Conference on Decision and Control*, 2007.

[38] Paolo Falcone. *Nonlinear model predictive control for autonomous vehicles*. Ph.d thesis, University of Sannio, Benevento, 2007.

[39] H. J. Ferrau, H. G. Bock, and Moritz Diehl. An online active set strategy for fast parametric quadtratic programming in mpc applications. *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, plenary talk*, 2006.

[40] Insurance Institute for Highway Safety. Electronic stability control could prevent nearly one-third of all fatal crashes and reduce rollover risk by as much as 80; effect is found on single- and multiple-vehicle crashes, 2006.

[41] U. Franke, H. Loose, and C. Knoppel. lane recognition on country roads. *IEEE Intelligent Vehicle Symposium*, june.

[42] E. Frazzoli. *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD thesis, Massachusetts Institute of Technology, 2001.

[43] E. Frazzoli. Maneuver-based motion planning for nonlinear systems with symmetries. volume 21, pages 1077–1091, Dec. 2005.

[44] E. Frazzoli, M.A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 25:116–129, 2002.

[45] Magomed Gabibulayev and Bahram Ravani. A stochastic form of a human driver steering dynamics model. *Transactions of the ASME*, 129:322–336, may 2007.

[46] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. *Dynamic Systems and Control Conference, 2010*, 2010.

[47] Yiqi Gao, Andrew Gray, Janick Frasch, Theresa Lin, Eric Tseng, J. Karl Hedrick, and Francesco Borrelli. Spatial predictive control for agile semi-autonomous ground vehicles. Seoul, Korea, 2012.

[48] J C Gerdes, E J Rossetter, and U Saur. Combining Lanekeeping and Vehicle Following with Hazard Maps. *Vehicle System Dynamics*, 36(4-5):391–411, 2001.

[49] P. Gill, W. Murray, M. Saunders, and M. Wright. NPSOL – Nonlinear Programming Software. Stanford Business Software, Inc., Mountain View, CA, 1998.

[50] T D Gillespie. *Fundamentals of Vehicle Dynamics.* Society of Automotive Engineers, Inc, 1992.

[51] J. Glover and F. Schweppe. Control of linear dynamic systems with set constrained disturbances. *IEEE Transactions on Automatic Control*, 16(5), 1971.

[52] Andrew Gray, M. Ali, Y. Gao, J.K. Hedrick, and Francesco Borrelli. Semi-autonomous vehicle control for road departure and obstacle avoidance. *IFAC Control of Transportation Systems*, 2012.

[53] Andrew Gray, Mohammed Ali, Yiqi Gao, J. Karl Hedrick, and Francesco Borrelli. Integrated threat assessment and control design for roadway departure avoidance. *Intelligent Transportation Systems Conference*, 2012.

[54] Andrew Gray, Yiqi Gao, Theresa Lin, J. Karl Hedrick, and Francesco Borrelli. Predictive control for agile semi-autonomous ground vehicles using motion primitives. Montreal, Quebec, Canada, 2012.

[55] R. Hanowski, R. Olson, J. Hickman, and T. Dingus. The 100-car naturalistic driving study: A descriptive analysis of light vehicle-heavy vehicle interactions from the light vehicle drivers perspective. Technical Report FMCSA-RRR-06-004, Virginia Tech Transportation Insti-

tute, 3500 Transportation Research Plaza (0536) Blacksburg, Virginia 24061, March 2006.

[56] R. Y. Hindiyeh and J. C. Gerdes. Equilibrium analysis of drifting vehicles for control design. *Dynamic Systems and Control Conference*, Oct. 2009.

[57] Rami Y. Hindiyeh and J. Christian Gerdes. Equilibrium analysis of drifting vehicles for control design. *Dynamic Systems and Controls Conference*, 2009.

[58] S. Hong, G. Erdogan, J.K. Hedrick, and F. Borrelli. Tyre-road friction coefficient estimation based on tyre sensors and lateral tyre deflection: Modeling, simulations and experiments. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, january 2013.

[59] S. Hong and J.K. Hedrick. Tire-road friction coefficient estimation with vehicle steering. *IEEE Intelligent Vehicles Symposium*, june 2013.

[60] http://www.mpc.berkeley.edu/research/advanced-vehicle-dynamic control.

[61] J. Jansson and F. Gustafsson. a framework and automotive application of collision avoidance decision making. *Automatica*, 44(9):658–669.

[62] J Jansson and F Gustafsson. A framework and automotive application of collision avoidance decision making. *Automatica*, 44(9):2347–2351, 2008.

[63] F. Kehrle, J.V. Frasch, C. Kirches, and S. Sager. Optimal control of formula 1 race cars in a vdrift based virtual environment. In *IFAC World Congress Milan*, 2011.

[64] E C Kerrigan. Robust Constraint Satisfaction: Invariant Sets and Predictive Control, 2000.

[65] T. Keviczky and G. J. Balas. Flight test of a receding horizon controller for autonomous uav guidance. Portland, Oregon, 2005.

[66] T. Keviczky and G. J. Balas. Flight test of a receding horizon controller for autonomous uav guidance. 2005.

[67] T. Keviczky and G. J. Balas. Software-enabled receding horizon control for autonomous uav guidance. *AIAA Journal of Guidance, Control, and Dynamics*, 29(3):680–694, May-June 2006.

[68] U. Kiencke and L. Nielson. Automotive control systems. *Springer*, 2000.

[69] M. V. Kothare, B. Mettler, M. Morari, P. Bendotti, and C. M. Falinower. Level control in the steam generator of a nuclear power plant. *IEEE Transactions on Control System Technology*, 8(1):55–69, jan 2000.

[70] B. Kouvaritakis, M. Cannon, and S.V. Rakovic. Explicit use of probabilistic distributions in linear predictive control. *Automatica*, 46:1719–1724, 2010.

[71] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8), aug 2000.

[72] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazolli, and J.P. How. Real-time Motion Planning with Applications to Autonomous Urban Driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, 2009.

[73] W. Langson, I. Chryssochoos, S. V. Rakovic, and D. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[74] J.H. Lee and B. Cooley. Recent advances in model predictive control. *Chemical Process Control - V*, 93(316):201, 1997.

[75] Levien and Raph. The euler spiral: a mathematical history. Technical Report UCB/EECS-2008-111, EECS Department, University of California, Berkeley, sep 2008.

[76] Theresa Lin, Eric Tseng, and Francesco Borrelli. Modeling driver behavior during complex maneuvers. *American Controls Conference*, 2013.

[77] Jing-Fu Liu, Jui-Hung Wu, and Yi-Feng Su. Development of an interactive lane keeping control system for vehicle. *Vehicle Power and Propulsion Conference*, 2007.

[78] S Mammar, S Glaser, and M Netto. Time to Line Crossing for Lane Departure Avoidance: A Theoretical Study and an Experimental Setting. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):226–241, 2006.

[79] D. Q. Mayne, M. M. Seron, and S. V. Rakovic. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[80] D.Q. Mayne. Nonlinear model predictive control. *Chemical Process Control - V*, 93(316):217, 1997.

[81] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789, 2000.

[82] Dejan Mitrovi and ore Vuleti. A survey of advanced driver assistance systems (adas). Technical report, University of Belgrade, Belgrade, Serbia, 2010.

[83] M. Morari and J.H. Lee. Model predictive control: past, present and future. *Computers and chemical engineering*, 23(4), 1999.

[84] C.B. Neas and M. Farhood. A hybrid architecture for maneuver-based motion planning and control of agile vehicles. *IFAC, Internationl Federation of Automatic Control*, 18, 2011.

[85] U.S. Department of Transportation. preliminary results analyzing the effectiveness of electronic stability control (esc) systems. Technical report, 2004.

[86] Nuria Oliver and Alex P. Pentland. Graphical models for driver behavior recognition in a smartcar. *Intelligent Vehicle Symposium*, 2000.

[87] H B Pacejka. *Tyre and Vehicle Dynamics.* Elsevier Science, 2005.

[88] W R Pasterkamp and H B Pacejka. The Tyre as a Sensor to Estimate Friction. *Vehicle System Dynamics*, 27(5):409–422, June 1997.

[89] J Pohl, W Birk, and L Westervall. A driver-distraction-based lane-keeping assistance system. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 221(4):541–552, January 2007.

[90] S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. *Chemical Process Control - V*, 93(316):232, 1997.

[91] Rajesh Rajamani. Vehicle dynamics and control, 1st ed. *Springer US*, 2005.

[92] A. Scheuer and Th. Fraichard. Collision-free and continuous-curvat ure path planning for car-like robots. *IEEE Intemational Conference on Robotics and Automation*, April 1997.

[93] Shogo Sekizawa, Shinkichi Inagaki, Tatsuya Suzuki, Soichiro Hayakawa, Nuio Tsuchida, Taishi Tsuda, and Hiroaki Fujinami. Modeling and recognition of driver behavior based on stochastic switched arx model. *Transactions on Intelligent Transportation Systems*, 8(4), dec 2007.

[94] Taehyun Shim and Donald Margolis. Model-Based Road Friction Estimation. *Vehicle System Dynamics*, 41(4):249–276, April 2004.

[95] Dong H. Shin and Sanjiv Singh. Path generation for robot vehicles using composite clothoid segments. Technical Report CMU-RI-TR-90-31, Robotics Institute, Carnegie Mellon University, dec 1990.

[96] H E Tseng, B Ashrafi, D Madau, T A Brown, and D Recker. The development of vehicle stability control at Ford. *IEEE/ASME Transactions on Mechatronics*, 4(3):223–234, September 1999.

[97] H Tsunashima, M Murakami, and J. Miyataa. Vehicle and road state estimation using interacting multiple model approach. *Vehicle System Dynamics*, 44(sup1):750–758, January 2006.

[98] R. Vasudevan, V. Shia, Y. Gao, R. Cervera-Navarro, R. Bajcsy, and F. Borrelli. Safe semi-autonomous control with enhanced driver modeling. *American Controls Conference*, 2012.

[99] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl. Time-optimal path tracking for robots: A convex optimization approach. *Automatic Control, IEEE Transactions on*, 54(10):2318 –2327, oct. 2009.

[100] Z. Wan and M. V. Kothare. Efficient scheduled stabilizing model predictive control for constrained nonlinear systems. *International Journal of Robust and Nonlinear Control*, 13:331–346, March-April 2003.

[101] H. Witsenhausen. A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control*, 13(1):5–21, 1968.

[102] S Yamazaki, O Furukawa, and T Suzuki. Study on Real Time Estimation of Tire to Road Friction. *Vehicle System Dynamics*, 27(sup001):225–233, January 1997.

[103] V. M. Zavala, C. D. Laird, and L. T. Biegler. Fast solvers and rigorous models: Can both be accomodated in nmpc? *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, plenary talk*, 2006.

[104] C Zhao, W Xiang, and P Richardson. Vehicle Lateral Control and Yaw Stability Control through Differential Braking. *In Proc. IEEE Industrial Electronics Symposium*, 2006.

List of Figures

2.1 A vehicle sketch depicting the body-fixed coordinate frame, $\mathcal{O}_b$, as well as body-fixed longitudinal and lateral velocities, $\dot{x}$ and $\dot{y}$, forces F_x, F_y, and moment, M. 5

2.2 Modeling notation depicting the forces in the vehicle body-fixed frame, the forces in the tire-fixed frame, 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 driver model variable e_ψ^{lp} and the angle of the road tangent ψ_d. 8

2.3 A road aligned coordinate frame illustrating the error states with respect to the road, e_y and e_ψ. 10

2.4 A sketch of the Pacjeka tire model function illustrating the parameters of the formula. 15

2.5 Pacejka tire force curve comparing lateral tire force f_y to slip angle α and longitudinal tire force f_x to slip ratio σ In these plots the friction coefficient $\mu = 1$. 17

2.6 Sample plots of lateral tire force characteristics for different levels of braking, where $\varphi = \sqrt{(\mu_i F_{z_i})^2 - f_{x_i}^2}$. 18

2.7 Lateral tire force characteristics compared to tire slip angle for different levels of braking. 20

2.8 Modeling notation. 24

2.9 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. 26

2.10 A road aligned coordinate frame illustrating the error states with respect to the road, e_y and e_ψ. 29

3.1 A road aligned coordinate frame illustrating the error states with respect to the road, e_y and e_ψ, as well as the driver model coordinates, the look-ahead point, lp, the error angle of the road at the look-ahead point e_ψ^{lp}, as well as the inertial frame angles of the vehicle ψ and road ψ_{road}. 34

3.2 Estimation results from [29] of the nonlinear estimation of the feedback parameters for the driver model of (3.17) for nominal driving. 39

3.3 Estimation results from [29] of the nonlinear estimation of the feedback parameters for the driver model of (3.17) for aggressive driving. 39

3.4 The road-aligned coordinate frame adapted to accommodate roadside obstacles. 40

3.5 A drawing of an uncertain interval set $\hat{\delta}_d \oplus \mathcal{W}$ centered on $\hat{\delta}_d$ where the actual value δ_d lies somewhere in the set. 41

3.6 A normal distribution over the uncertain driver input w at time t given $\hat{\delta}_d = 0$ and $\Sigma = 0.035$ where $w(t) \sim \mathcal{N}(\hat{\delta}_d, \Sigma)$. . . 42

3.7 Block diagram illustrating the structure of the simulated system. The actual plant is not shown. 43

4.1 This figures shows a sketch of a front mounted sensor, e.g. a radar, providing range and range rate to the preceding vehicle. These measurements can be used to implement an ACC (adaptive cruise control) system that will maintain a constant distance or time gap to the preceding vehicle. 45

4.2 A depiction of a car during a maneuver that is understeered: the car does not turn enough and leaves the road. 46

4.3 A depiction of a car during a maneuver that is oversteered: the car turns more sharply than intended and could go into a spin. 46

4.4 An electronic stability control system can constrain the tire slip angles to lie within the linear operating region of the tire forces and thus be easily maneuverable by a driver. 47
4.5 A sketch of a vehicle with a lane detection system running to support safety systems such as Lane Keeping Systems. 47
4.6 Safety system architecture 48
4.7 A sketch of the structure of the ADAS designed in this thesis: the overall goal is to be more informed in order to make safer, smarter, and more coordinated decisions about the safety of the driver. 49

5.1 The initial set of states $\mathcal{X}_0^p$ and the final robust positive invariant set $\mathcal{Z}^p$ projected onto the e_y-e_ψ plane where $\mathcal{X}_0^p = \text{proj}_{e_y-e_\psi}(\mathcal{X}_0)$ and $\mathcal{Z}^p = \text{proj}_{e_y-e_\psi}(\mathcal{Z})$. 64

6.1 Architecture of the two-level MPC. The *motion primitive* or *spatial model* is used for high-level path planning. The four-wheel vehicle model (§2.1) is used for low-level path tracking. . 72
6.2 An example hybrid system with steady-state trims as discrete states and maneuvers execute transitions. 79
6.3 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. 81
6.4 A drifting maneuver compared to an aggressive turn. The turn is the sharpest allowable maneuver within the constraint of the slip angles of the tires. At this initial vehicle state ($\xi_g = [0, 0, 0]$, $\dot{x} = 11\text{m/s}$, q_1) the drifting maneuver is the only safe maneuver to avoid the obstacle, where the coefficient of friction $\mu = 0.55$. 83

6.5 A successfull navigation of the corner planned with the drifting primitives. The vehicle starts in trim q_1 (straight travel, $\dot{x} =$ 11m/s), takes maneuver p_5 to a drifting trim q_4, then takes maneuver p_6 back to q_1. The aggressive path is able to be tracked with no tracking error where the coefficient of friction $\mu = 0.55$. 87

6.6 Various lane change maneuvers are compared. As the vehicle approaches the obstacle the planned paths become more aggressive (high slip angles). 88

6.7 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. . . 89

6.8 Various candidate paths considered during the online optimization of the parameterized maneuver. 90

6.9 Vehicle successfully avoids the obstacle using maneuvers based on clothoids 92

6.10 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. 93

6.11 For the NMPC problem with steering and braking the obstacle avoidance problem may be posed as a complex mixed-integer program, shown here with 3 discrete constraint sets $\mathcal{X}_i,\ i \in \{1, 2, 3\}$, where the sampled points in time, $t_i,\ i \in \{1...N\}$, are not fixed in the spatial-direction s. 94

6.12 The spatial transformation reduces the complexity of the NMPC problem with steering and braking by a translation of the obstacle constraints to convex constraints on the state vector. The constraint set Σ_1 is then defined as interval constraints $\mathcal{X}_1 := \{e_y \,|\, e_{y,\min} \leq e_y \leq e_{y,\max}\}$ and the sampled points, $s_i,\ i \in \{1...N\}$, are fixed in the spatial-direction s. 95

6.13 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. 100

6.14 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. 101

6.15 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. 102

6.16 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. 103

6.17 Simulation result comparison with the controller proposed in [46]. 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. 104

7.1 A block diagram of the typical structure of a modular active safety system. 107

7.2 Simulation results. In these plots we show nominal driving behavior where the driver is not in need of assistance. These results show the behavior of the controller when the driver is capable of maintaining safety. The minimally invasive nature is illustrated by the controller calculating zero assistance in a safe situation. 113

7.3 Simulation results. The plots capture a snapshot of the prediction that detects a lane departure by excessive speed in a curve. The controller augments the driver's steering and braking to bring the vehicle back within the lane. 115

7.4 Simulation results. These plots capture a snapshot of the prediction where the driver will depart the lane, as shown in (a). The controller augments the driver's steering, (b), to satisfy the safety constraints and keep the vehicle within the lane. . . 117

7.5 Setup 120

7.6 The complete trajectory of the vehicle within the lane with two insets, A and B, detailing two interventions. The first occurs when the distracted driver approaches a curve and the second occurs when the driver lets the vehicle stray off the road. In both situations the controller corrected the steering to keep the vehicle safely on the road. 121

7.7 The upper plot shows the corrective steering inputs for the real-time simulation shown in Figure 7.6 inset A and the lower plot depicts the corrective steering for Figure 7.6 inset B. . . . 122

7.8 The trajectory of the vehicle during a roadway departure intervention. The driver is distracted and the vehicle drifts to the left of the road. An intervention occurs and the vehicle safely remains within the safety constraints imposed by the road bounds. The associated states and inputs can be seen in Figures 7.9 and 7.10, respectively. 123

7.9 A plot of 4 vehicle states, $[e_y, e_\psi, \dot{x}, \dot{\psi}]$, during the intervention depicted in Figure 7.8. The controller intervenes just before the vehicle reaches 800 m in the X-direction. The lateral offset, e_y, is constrained by the upper limit of the roadway, in this case 5 m. 124

7.10 A plot of the inputs added by the controller during the intervention depicted in Figure 7.8. The controller briefly adds corrective steering to keep the vehicle on the road. In this scenario very little braking action was required. 125

7.11 A situation where an attentive driver encounters an obstacle in the path. The driver is capable of avoiding the obstacle and for this reason, the MPC controller doesn't intervene. 126

7.12 The inputs from the optimization, $[\delta_c, \beta_r]$, the driver model, $\hat{\delta}_d$, and the closed-loop steering, δ. 127

7.13 The closed-loop vehicle trajectory where the driver is assisted by the MPC controller. The controller detects that the nominal driver behavior is no longer sufficient to maintain a safe vehicle trajectory and intervenes in order to avoid the road side obstacle. 128

7.14 The inputs required from the safety system, the controller slightly steers and brakes the vehicle to avoid the roadside obstacle and smoothly gives back control to the driver once the obstacle has been avoided. 129

7.15 The closed-loop trajectory of the vehicle. The inset shows a snapshot of the time instant an intervention was required. In this case a driver monitoring system has detected that the driver is distracted and consequently, we note that the intervention is activated earlier. 130

7.16 The input signals in the scenario shown in Figure 7.15. Since the intervention is issued early, the control action is smaller when compared to the input signals shown in Figure 7.14 . . . 131

7.17 The closed-loop trajectory of the vehicle in a situation with multiple obstacles. The inset shows a snapshot of the moment the second obstacle was encountered and intervention was required. 132

7.18 The inputs showing the corrective action, $[\delta_c, F_b]$, of the safety system required to keep the driver safe while navigating through multiple obstacles. 132

7.19 The paths predicted by the controller at each time instant. The initial condition is the circle marker, the dashed lines are the constraints imposed by the lane on the center of the vehicle, and the solid lines are the actual road bounds. This plot shows the time history leading up to the first intervention only at approximately $t = 6$ seconds. 134

7.20 A plot of 4 states of the vehicle, $[e_y, e_\psi, \dot{x}, \dot{\psi}]$, during the experiment in section 7.3.1. This plot details the moment the controller intervenes at approximately 6 seconds, as shown in Figure 7.19. The velocity is slightly reduced and the lateral offset is corrected to keep the vehicle inside the lane. The associated inputs are shown in Figure 7.21. 135

7.21 The corrective steering and braking action induced by the controller during the intervention depicted in Figures 7.19 and 7.20 during the experiment in section 7.3.1. The cost is also shown where the spike at approximately 6 seconds coincides with the soft constraint, ϵ, taking a small value. The inputs are shown correcting the constraint violation. 136

7.22 A plot of 4 states of the vehicle, $[e_y, e_\psi, \dot{x}, \dot{\psi}]$, during the experiment in section 7.3.2. The vehicle approaches the upper bound on e_y and the controller corrects the steering and braking to remain inside the lane. The associated inputs are shown in Figure 7.23. 138

7.23 The corrective steering and braking action induced by the controller during the intervention depicted in Figure 7.22 during the experiment in section 7.3.2. 139

7.24 Block diagram illustrating the structure of the simulated system. The actual plant is not shown. 143

7.25 Trajectory 1 is assisted by the controller to keep the driver safe. Trajectory 2 is the expected driver input and collides with the obstacle. 145

7.26 The inputs in Trajectory 1 and 2 as well as the corrective action $\bar{u}$ determined by the controller. 145

7.27 A plot of the nominal trajectory, $\bar{x}$, the disturbed trajectory, x, and a projection of the robust invariant set along the nominal trajectory. 146

7.28 The inputs showing the corrective action from the controller, $\bar{u}$, the input from the driver model augmented with the controller action, u_{nom}, the input with additive disturbance, u_{dis}, and the steering from the driver model, $\hat{\delta}_d$. 147

7.29 A snapshot of the prediction where the vehicle encounters an obstacle. The tightened constraints are labeled by γ_i. The vehicles in boxed-outline, Trajectory 2, is the trajectory with the driver model only and Trajectory 1 depicts the corrected trajectory to satisfy the safety constraints. 151

7.30 The inputs for the snapshot depicted in Figure 7.29. $v = K\xi + c$ is the minimization variable, c is the optimization variable, and $\hat{\delta}_d$ is the nominal driver input. 152

7.31 The upper bound γ on the disturbance propagation over the prediction horizon. 152

7.32 Plot of randomly simulated trajectories depicting the statistical behavior of the controller. The worst-case trajectories are marked in dashed line. 153

A.1 A diagram of the experimental setup. 154

A.2 Pictures of the Hyundai setup. 157

List of Tables

2.1 List of Pacejka tire model parameters. 14

6.1 Table of Trims 76
6.2 Table of Maneuvers 78
6.3 Tuning parameters for high-level Spatial MPC 99
6.4 Tuning parameters for low-level LTV MPC 100
6.5 Low Level Lateral Position (Y) Tracking Error Comparison. Simulation data corresponds to figure 6.17 and experimental data corresponds to figure 6.14. 101

7.1 Vehicle parameters 112
7.2 Design parameters 112
7.3 Real-Time Design parameters 119
7.4 Real-Time Design parameters 133
7.5 Vehicle parameters 144
7.6 Simulation Parameters 150