Title

Predictive Control under Uncertainty for Safe Autonomous Driving: Integrating Data-Driven Forecasts with Control Design

Permalink

https://escholarship.org/uc/item/8qz1c5nj

Author

Carvalho, Ashwin Mark

Publication Date

2016

Peer reviewed|Thesis/dissertation

Predictive Control under Uncertainty for Safe Autonomous Driving: Integrating Data-Driven Forecasts with Control Design

by

Ashwin Mark Carvalho

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

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

Fall 2016

Predictive Control under Uncertainty for Safe Autonomous Driving: Integrating Data-Driven Forecasts with Control Design

Abstract

Predictive Control under Uncertainty for Safe Autonomous Driving: Integrating Data-Driven Forecasts with Control Design

by

Ashwin Mark Carvalho

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Chair

Self-driving vehicles have attracted a lot of interest due to their potential to significantly reduce traffic fatalities and transform people's lives. The reducing costs of advanced sensing technologies and the increasing capabilities of embedded computing hardware have enabled the commercialization of highly automated driving features. However, the reliable operation of autonomous vehicles is still a challenge and a major barrier in the large scale acceptance and deployment of the technology.

This dissertation focuses on the challenges of designing safe control strategies for self-driving vehicles due to the presence of uncertainty arising from the non-deterministic forecasts of the driving scene. The overall goal is to unify elements from the fields of vehicle dynamics modeling, machine learning, real-time optimization and control design under uncertainty to enable the safe operation of self-driving vehicles. We propose a systematic framework based on Model Predictive Control (MPC) for the controller design, the effectiveness of which is demonstrated via applications such as lateral stability control, autonomous cruise control and autonomous overtaking on highways. Data collected from our experimental vehicles is used to build predictive models of the vehicle and the environment, and characterize the uncertainty therein. Several approaches for the control design are presented based on a worst-case or probabilistic view of the uncertain forecasts, depending on the application. The proposed control methodologies are validated by experiments performed on prototype passenger vehicles and are executed in real-time on embedded hardware with limited computational power. The experiments show the ability of the proposed framework to handle a variety of driving scenarios including aggressive maneuvers on low-friction surfaces such as snow and navigation in the presence of multiple vehicles.

I lovingly dedicate this dissertation to my family, Claire and her family, and everyone who has supported me throughout this journey.

Contents

Contents **ii**

List of Figures **iv**

List of Tables **viii**

List of Algorithms **ix**

1 Introduction **1**

1.1 Motivation and Background 1

1.2 Thesis Outline and Contributions 3

1.3 List of Publications 6

2 MPC for Control Under Uncertainty **8**

2.1 Notation 8

2.2 Model Predictive Control for Active Safety: An Illustrative Example 9

2.3 Problem Formulation for Control Under Uncertainty 12

3 Vehicle Dynamics Models **18**

3.1 Introduction 18

3.2 Dynamic Models 20

3.3 Kinematic Model 27

3.4 Longitudinal Point-Mass Model 30

3.5 Actuator Models 31

3.6 Uncertainty Characterization 33

3.7 Summary 35

4 Environment Models for Automated Driving **36**

4.1 Introduction 36

4.2 Multiple Model Approach 38

4.3 Lane Change Intention Estimation 45

4.4 Interaction-aware Car-Following Model 46

4.5 Collision Avoidance 49

5 Iterative Linearization for Real-Time Nominal MPC 53
5.1 Introduction 53
5.2 Nonlinear MPC Formulation 55
5.3 Iterative Linearization Approach 55
5.4 Simulation Results 58
5.5 Experimental Results 61
5.6 Conclusions 63

6 Robust Lateral Stability Control 65
6.1 Introduction 65
6.2 Piecewise Affine Vehicle Model 66
6.3 Uncertain Driver Model 68
6.4 Robust Control Design 69
6.5 Experimental Results 74
6.6 Conclusions 76

7 Safe Personalized Autonomous Cruise Control 78
7.1 Introduction 78
7.2 Robust Control Design 82
7.3 Personalized Driving 94
7.4 Experimental Results: Multi-lane Scenarios 96
7.5 Experimental Results: Personalization and Safety 100
7.6 Conclusions 102

8 Stochastic MPC for Automated Highway Driving 105
8.1 Introduction 105
8.2 Analytical Stochastic MPC 109
8.3 Sampling-based Stochastic MPC 115
8.4 Hybrid Stochastic Model Predictive Control 117
8.5 Simulation results 123
8.6 Experimental Results 127
8.7 Conclusions 131

A Experimental Vehicles 134
A.1 Jaguar S-Type 134
A.2 Hyundai Grandeur 134

B Measurement Uncertainty in Stochastic MPC 136
B.1 State Estimate Dynamics 137
B.2 Uncertainty Propagation 138

Bibliography 140

List of Figures

1.1 Outline of the dissertation. 4

2.1 Snapshots of simulation with the ego vehicle E in blue and the target vehicle T in red. The red line depicts the trajectory of the target vehicle, which is assumed to be known a-priori. 10
2.2 Driver and controller inputs during the simulation. The controller tracks the driver's acceleration until a violation of the safety distance is predicted around $t = 5$ s. The controller applies harder braking to keep the vehicle safe. The difference between the driver and controller inputs is shown by the black line. . 11
2.3 Open-loop predictions of the relative distance to the vehicle in front at $t = 6$ s. The red line depicts what would have happened had the controller not intervened. The controller plans to brake harder than the driver to prevent the relative distance going below the specified safety distance of 4 m. 11
2.4 Relative distance between lead and controlled vehicles. The red circle at $t = 4.8$ s indicates the instant at which the target vehicle is detected in front of the ego vehicle. 12

3.1 Notation used in the nonlinear vehicle model. 19
3.2 Notation used in the dynamic and kinematic bicycle models. 22
3.3 Variation of lateral tire force with slip angle using different tire modeling approaches. The gray dots indicate the values estimated during a test on a winding road and a 'figure 8' drift maneuver. The vehicle speed ranged from 10 to 35m/s. 26
3.4 Position $[\xi, \eta]^T$ and orientation ϕ in the road-aligned frame. 27
3.5 Percentage distributions of the 1 and 5-step open-loop position errors of the kinematic and dynamic bicycle models at time steps of 0.2 s. The mean and standard deviation of these errors are denoted by μ and σ, respectively. The subscripts K and D refer to the kinematic model and the dynamic model, respectively. 30
3.6 Percentage distributions of the one-step prediction error for the dynamic bicycle model with a linear tire model. The model discretization time is 0.1 s. The data was collected on the highways around Berkeley. 34

4.1 Lane change trajectories from 77 recorded datasets. Right lane change trajectories are flipped with respect to the lane center. 40
4.2 Gains K_1 and K_2 identified from 60 lane change trajectories and clustered into 5 modes depicted by the different marker types. 42
4.3 Lane change profiles corresponding to the 5 representative values of K_1 and K_2 identified from data. 42
4.4 Upper and lower boundaries $\eta_k^U(\xi_{k|t})$ and $\eta_k^L(\xi_{k|t})$, respectively, for the given scenario using the safety corridor approach. In general, the boundaries are given by piecewise affine functions of the EV's longitudinal position $\xi_{k|t}$. 50
4.5 Signed distance between convex shapes $\mathcal{A}$ and $\mathcal{B}$ (Figure similar to that in [109]) 51
4.6 Unsafe region for the depicted positions of the EV E and TV T in two scenarios. The EV's center of gravity must lie outside the gray shaded area for a collision to be avoided. 52

5.1 Simulation 1: The performance of the proposed controller (Controller 1) is similar to that of the controller which uses NPSOL (Controller 2) for the same sampling time, prediction horizon, environment scenario and constraints in the optimization problem. 60
5.2 Simulation 2: Controller 1 uses a much longer prediction horizon as compared to Controller 2. 61
5.3 Experimental test 1: The vehicle avoids a single obstacle with an entry speed of 80 km/hr. 62
5.4 Experimental test 2: The vehicle avoids two separated obstacles with an entry speed of 60 km/hr. 62
5.5 Experimental test 3: The vehicle avoids an obstacle moving at 36 km/hr with an entry speed of 80 km/hr. The positions of the vehicle and the obstacle at five time instants $t_1, ..., t_5$ during the test are shown. 62

6.1 Lateral tire forces and the PWA approximation 67
6.2 State-dependent constraints on δ_d validated with data from different sets of experiments and the same driver. 69
6.3 3–step controllable sets to the mode 1 RCI set $\mathcal{X}_\infty^1$, $\mu = 0.3$, $v_x = 50$ kph. The different colors correspond to the various steps. The red set depicts $\mathcal{X}_\infty^1$. 71
6.4 Maximal RCI set $\mathcal{X}_\infty^1$, $\mu = 0.3$, $v_x = 50$ kph. 73
6.5 Experiment 1: Double lane change maneuver with an entry speed of 60 kph. . . 75
6.6 Experiment 2: Driving on icy circular track ($\mu \approx 0.2$) of diameter 110 m at a speed of approximately 40 kph. 76
6.7 Experiment 2: Driving on icy circular track ($\mu \approx 0.2$) of diameter 110 m at a speed of approximately 40 kph. 77

7.1 Autonomous Cruise Control (ACC) system architecture as described in Section 7.1. 81

7.2 Projection of the maximal CI set $\mathcal{X}_{ep}$ in the ξ^r-v space for $d_{\text{safe}} = 5$ m and $v^{\max} = 25$ m/s. The black line depicts the direction of unboundedness of $\mathcal{X}_{ep}$. . . 86

7.3 Projection of the sets $\mathcal{R}_k$ corresponding to the target set $\mathcal{X}_{ep}$ in the ξ^r-v^e-v space for $k = 10, 20, 30, 40, 50$. Each two-dimensional slice in the ξ^r-v space corresponds to a particular value of v^e. 88

7.4 Comparison of the closed-loop acceleration profiles during a simulation of the scenario described in Section 7.4.3 of the proposed approach ($ACC1$, solid line) and the existing approach ($ACC2$, dashed line). $ACC1$ results in a smoother acceleration command that $ACC2$. 98

7.5 Results of the relevant target identification for a TV cut-in scenario from one of the datasets (described in Section 7.4.4). $ACC1$ identifies the RT 1.85 s before $ACC2$. 99

7.6 KS distances obtained for two driving style indicators, TTCi and VSP, for 5 different drivers (Figure from [75]) 100

7.7 Experiment 1: Switching from Driver Model 1 (conservative) to Driver Model 2 (aggressive). MRT denotes the most relevant TV while EV denotes the ego vehicle. 101

7.8 Experiment 2: Safety performance of controller. 103

7.9 Experiment 3: Acceleration commands provided by the driver model and controller. The controller applies maximum braking at around $t = 47$ s as the initial relative distance between the ego and preceding vehicles violates the safety distance. 104

8.1 Snapshot of simulation with worst-case target vehicle predictions. The red shaded rectangles show the set of all possible states of the target vehicle T_1 assuming constant speed. The red circles depict the closed-loop trajectory of T_1, while the blue circles depict the open-loop trajectory of the ego vehicle E. 106

8.2 Closed-loop longitudinal speed profile of the ego vehicle E. The ego vehicle initially speeds up to the reference speed, but slows down to satisfy the safety constraints due to the worst-case predictions. 107

8.3 Sampled predictions of the vehicle's lateral position starting from its current state at $t = 20$ s. The solid red line depicts its past trajectory while the dotted green line shows the actual future trajectory which is not known to the model. The red circles depict the means of the predictions at the corresponding time steps. . . . 118

8.4 PDFs of the distributions of the predicted lateral position of the vehicle at 3 different time steps of the horizon. The solid blue line shows the GMM characterization of the distribution while the dotted red line is the Gaussian approximation. . . . 118

8.5 Illustration of the proposed sample aggregation strategies. 121

8.6 Snapshots of simulation with the ego vehicle E in blue and the target vehicles T_1 and T_2 in red. The blue and red circles are the predicted positions of the ego and target vehicles, respectively, over the horizon. 125

8.7 Simulation showing the effect of varying the risk parameter ϵ_k on the conservatism of the controller. The notation is the same as that in Figure 8.6 126

8.8 Snapshots of the simulation at time $t = 10$ s. The blue and red circles are the open-loop predictions of E and T_1, respectively. The solid red line is the observed trajectory of T_1 so far while the dotted red line is its future trajectory which is not known to the forecast model. 128

8.9 Comparison of closed-loop speed profiles using traffic forecasts based on the most likely mode (Gaussian, dotted red line) and the multimodal distribution generated by the multiple model filter (solid blue line). 128

8.10 Snapshots of experiment with the ego vehicle E in blue and the target vehicle T_1 in red. The blue and red circles are the predicted positions of the ego and target vehicles, respectively, over the horizon. The vertical green lines indicate the safe region for E computed by the safety corridor approach, given the sampled predictions of T_1, ... 130

8.11 Snapshot of the experiment in which interaction between the target vehicle T_1 and the ego vehicle E is not considered. The notation is the same as that in Figure 8.10. 131

List of Tables

5.1 Tuning parameters for experiments 63

6.1 Tuning parameters for lateral stability control experiments 74

7.1 Means and standard deviations of the metrics defined in Section 7.4.2 for the proposed approach ($ACC1$) and the existing approach ($ACC2$) computed from 28 simulations. 97
7.2 Comparison of the metrics defined in Section 7.4.2 for $ACC1$ and $ACC2$ during a simulation of the scenario described in Section 7.4.3. 99

A.1 Parameters of the prototype Jaguar S-Type vehicle 135
A.2 Parameters of the prototype Hyundai Grandeur vehicle 135

List of Algorithms

4.1 Kalman filter equations for model $m^{(i)}$ 43
4.2 Sampled predictions of target vehicle state 44
5.1 Tailored MPC algorithm 58
6.1 *Pre* set computation 72
6.2 RCI set computation 72
7.1 Computation of ego vehicle stopping time 90
7.2 Computation of ego vehicle braking distance (Case 1) 91
7.3 Computation of minimum separation (Case 2) 92
8.1 FTOC recursion 112
B.1 Kalman Filter recursion 138

Acknowledgments

My extreme gratitude to my adviser, Francesco Borrelli, for his constant support and motivation throughout the course of my graduate study. This dissertation would not have been possible without his vision and meticulousness. I could not have asked for a better adviser, to whom I will always turn to for guidance. I would like to thank professors Karl Hedrick and Pieter Abbeel for being on my dissertation committee and providing valuable feedback on this thesis.

The many members of the Model Predictive Control lab over the years helped brighten the gloomy hallways of the Etcheverry Hall basement and made the lab a fun place to work. This dissertation is the outcome of some amazing collaborations; it was a pleasure to work with and learn from Giovanni Palmieri, Yiqi Gao, Andrew Gray, Chan Kyu Lee, Jason Kong, Stéphanie Lefèvre, Georg Schildbach, Ziya Ercan, Alek Williams and Gianluca Cesari. Sarah Koehler started and ended this journey with me and has been a great friend and classmate. Sergey Vichik, who also started graduate school with us, was an excellent lunch buddy and has been a role model. A shout-out to Tory Smith, Matthew Chong and Elena Carano for the fun times doing homework together, and to the members of the Vehicle Dynamics and Control lab with whom I had the opportunity to interact.

I would like to thank our collaborators at the Ford Motor Company, especially Eric Tseng and Vladimir Ivanovic, for hosting us at the winter test facility in Michigan, supporting our research and providing very valuable feedback. I would also like to thank the Hyundai Motor Company for its support via the Hyundai Center of Excellence and the test vehicle, without which the work in the lab would not have been half as much fun nor nearly as impactful.

The material in this dissertation is based upon work partially supported by the National Science Foundation under Grant No. 1239323; any findings and conclusions herein do not necessarily reflect its views. I would also like to acknowledge the support of the UC Berkeley Fellowship for Graduate Study and the John and Janet McMurtry Fellowship.

Of course, grad school would not have been as much fun without the many friends I made along the way; a heartfelt thanks to each of them. Lee-Huang Chen has been an awesome friend, and I am glad we could complete this journey together. Soccer and ultimate frisbee with the Mechanical Engineering Athletics Team (MEAT) provided a perfect ending to many a tough day.

My parents Rosemary and Stephen, and siblings Roshni and Pravin have been an incredible source of support and motivation even from so far away. Each of them continues to inspire me through their lives. I would not have made it here but for the values instilled in me by my parents. I was also fortunate to have my uncle Allwyn, aunt Cecille and cousin Smita live so close to Berkeley and provide a home away from home.

Finally, I was most fortunate to meet the love of my life, Claire Funke, during grad school. She has been an amazing source of fun, strength and motivation, and a constant during the many ups and downs over the past few years. I am excited to embark on the next phase of my life with her by my side. A special thanks to her parents Susan and Kevin, and siblings Katherine, Scott and Amy.

Chapter 1

Introduction

1.1 Motivation and Background

In the past decade, significant progress has been made in the research and commercial deployment of advanced driving assistance features such as autonomous cruise control, emergency braking and lane keeping assistance systems. These features augment existing active safety systems such as anti-lock braking and electronic stability control, and are expected to pave the way to fully self-driving cars. The technology has attracted a lot of attention both from the public and the government due to their potential to transform people's lives and the transportation industry at large. The main interest lies in the dramatic increase in road safety that can be achieved by means of self-driving vehicles. According to the National Highway Traffic Safety Administration (NHTSA), 35,092 people lost their lives in crashes on U.S. roadways in the year 2015, which is the largest percentage increase between consecutive years in nearly 50 years [93]. Approximately 90 people died each day and 6,400 were injured. 94% of the above crashes were attributed to human error [110]. The U.S. Department of Transportation, in its landmark policy on automated vehicles, has stressed the importance of the technology to mitigate that huge proportion of accidents [118]. It also emphasizes the need for the rigorous testing and validation before self-driving cars can be mass produced. Currently, the reliable operation of autonomous vehicles over long periods of time still remains a challenge. Systems implemented in research and production require the driver to be engaged at all times and the distance traveled between driver interventions is relatively low.

One of the main assumptions that would facilitate the control design for the safe operation of autonomous cars is the precise knowledge of the vehicle's location, and the current and future locations of objects around the vehicle. If this assumption is satisfied, control design for autonomous driving can be addressed using basic techniques, with extensive tuning and switching between local controllers to handle heterogeneous driving scenarios. This dissertation focuses on the challenges of control design when the above assumption cannot be satisfied, that is, there is uncertainty in the motion of the autonomous vehicle (also referred to as the *ego vehicle*) and the forecast locations of objects in the driving scene. Driver manuals around the world always warn drivers of wrong predictions. As an example, the driving rules in Australia require the driver to look around for “children (who) are small and can be unpredictable” [53]. The New Jersey driver manual [112] classifies “hazards . . . into three groups” with one of them being “pedestrians and animals” which “are characterized by unpredictability and quick movement.” Even drivers of other vehicles in traffic often exhibit a lack of predictability, especially in scenarios such as intersection and roundabout navigation and merging lanes on highways.

Undoubtedly, driving requires forecasts, which can be highly uncertain in some driving scenarios. The simplest way to deal with uncertain forecasts in autonomous driving is to reduce the vehicle speed and wait until the uncertainty becomes negligible. In general, this might not be always possible (for instance, when driving at high speed on a highway), may generate deadlock scenarios (waiting forever at an intersection) and is not the preferred driving mode of the vast majority of the drivers. For such scenarios, control design for autonomous driving is a real challenge.

In addition to the uncertainty in the driving scene, several features make the autonomous vehicle control problem non-trivial and interesting. The nonlinear dynamics of the vehicle, modeling errors and the presence of state and input constraints make the application of classical control schemes difficult. The consideration of the vehicle dynamics and the uncertainty therein is especially important while operating on low-friction surfaces such as snow where it is relatively easy for the vehicle to suddenly drift out of control. The use of classical controllers would entail several heuristics to assess the criticality of the situation and accordingly activate a control strategy which is finely tuned for that particular situation. This has been a standard approach in the automotive industry with regards to the development of active safety systems. However, this approach does not safely extend to the domain of self-driving cars.

The framework of Model Predictive Control (MPC) or Receding Horizon Control (RHC) provides a systematic approach to deal with the aforementioned challenges. Here, the control law is computed as the output of a Finite Time Constrained Optimal Control (CFTOC) problem. At each time step, a model of the system is used to propagate the effect of the control input on the future system states and the associated uncertainty over a finite prediction horizon. The optimal sequence of control inputs is determined such that the constraints on the state and input are satisfied and a chosen performance metric which is a function of the predicted states and inputs is optimized. The first input of the optimal plan is applied to the system, and the process is repeated at the next time step with the new measurements, thus yielding a feedback control strategy. Previous work has shown the effectiveness of MPC for the control of semi-autonomous and fully autonomous vehicles, for example, [8, 43, 45, 48, 49, 54, 55].

This dissertation focuses on the control challenges associated with the application of MPC-based strategies for safe semi-autonomous and fully autonomous driving in the face of uncertain forecasts of the vehicle, driver and environment subsystems. A systematic framework for the control design is proposed, the generality and effectiveness of which is demonstrated via several applications. The common themes across the applications presented in this dissertation are as follows.

- *Data-driven uncertainty characterization:* We use real driving data for the identification of the uncertainty in the models of the vehicle, driver and environment used for the predictive control design. The nature of the uncertainty informs the choice of the solution methodology used for the control design. For example, traffic prediction models yield a multimodal probabilistic description of the uncertainty, thus requiring a control strategy which can deal with the corresponding probabilistic predictions.

- *Navigation in complex scenarios:* Some of the presented applications involve the operation of the vehicle in challenging environments involving, for example, aggressive driving maneuvers on slippery surfaces such as snow. Other applications deal with general traffic environments involving the presence of multiple vehicles in the vicinity of the autonomous vehicle. The consequence is an increased level of uncertainty in the driving scene which must be accounted for by the controller.

- *Robustness to uncertainty:* The goal of the MPC-based controller is to compute a control policy which is robust to the uncertainty in the forecasts over the prediction horizon. We use several notions of robustness depending on the specific application and the aforementioned uncertainty characterization. In particular, worst-case approaches guarantee safety for all possible realizations of the uncertainty while probabilistic approaches provide an upper bound on the probability of constraint violation. The outcome in either case is a deterministic constrained optimization problem to be solved online.

- *Real-time optimization:* We place an emphasis on the real-time implementation of our proposed control strategies on embedded computing platforms. Tailored algorithms which exploit the structure of the problem are presented.

In summary, we combine elements from the fields of vehicle dynamics modeling, machine learning, real-time optimization and predictive control design under uncertainty to enable the safe operation of semi-autonomous and self-driving vehicles in complex driving scenarios.

1.2 Thesis Outline and Contributions

A schematic of this dissertation is depicted in Figure 1.1 and serves as a pictorial guide of how the chapters are linked. The contributions of the individual chapters are highlighted below.

Chapter 2 formally introduces the control problem that is addressed in this dissertation. First, the design of a simple of driver assistance feature using the framework of MPC is discussed to motivate the need for forecasts and to discuss the associated control challenges. We present the problem formulation for control in the presence of uncertainty due to the forecasts of the motion and behavior of the three main elements in the driving scene - the

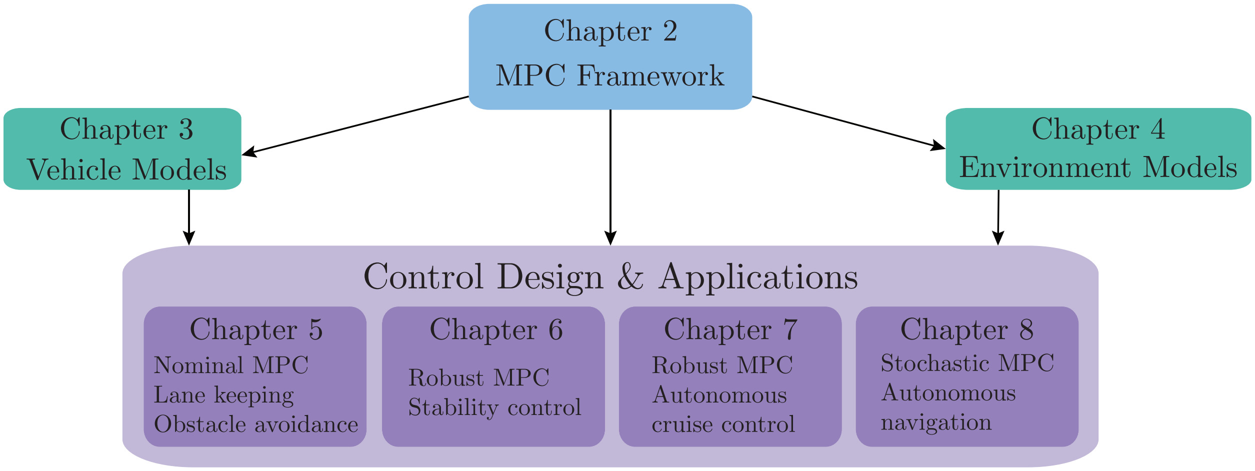


Figure 1.1: Outline of the dissertation.

vehicle, the driver and the environment. The latter consists of entities such as surrounding vehicles, bicyclists and pedestrians. Various characterizations of the uncertainty are discussed, ranging from deterministic approaches where no uncertainty is considered to robust ones where the worst-case effect of disturbances needs to be accounted for. Probabilistic models provide a trade-off between these two extremes. An overview of existing solution methodologies for the control problem under uncertainty is presented.

The above predictive control scheme relies on models of the vehicle and environment subsystems in order to make forecasts of the corresponding states over the prediction horizon. A detailed treatment of the various vehicle models used in this dissertation is provided in Chapter 3. The focus is on models which are useful from the perspective of model-based control design, where it is necessary to trade-off model accuracy for the computational complexity of the resulting control law. The choice of vehicle model is highly coupled with the specific application under consideration. For example, simple point-mass models suffice for the longitudinal controller presented in Chapter 7. However, for the experiments performed on snow in Chapter 5, a higher fidelity model is used which accounts for the highly nonlinear tire-road interaction at the four wheels. The characterization of the various sources of uncertainty in the vehicle model and the identification using real driving data is also discussed.

Chapter 4 presents forecast models of the positions and velocities of surrounding vehicles (referred to as *target vehicles*). Our focus is on highway driving applications and on approaches that can be run efficiently in real-time. First, a method based on the framework of multiple model filtering is proposed to predict the longitudinal and lateral motion of target vehicles on highways. The control actions of the vehicles are modeled as parametric state-feedback policies, the parameters of which are learned from real data collected using our experimental vehicle. We present a learning-based approach based on Hidden Markov Models (HMMs) to identify the lane changing intent of target vehicles, which is an input for the autonomous cruise control system developed in Chapter 7. A framework for modeling interactions between the ego and target vehicles in car-following situations is also presented. Finally, two approaches for formulating collision avoidance constraints between the ego and target vehicles based on their predicted positions are presented.

Chapters 5–8 deal with the control design for several semi-autonomous and fully autonomous driving applications using the framework presented in Chapter 2 and the vehicle and environment models of Chapters 3 and 4, respectively. For each application, we use a particular characterization of the system uncertainty and present strategies to solve the associated control problem.

Chapter 5 proposes a method to reduce the online computational burden of current nonlinear predictive control strategies used for autonomous lane keeping and obstacle avoidance when no uncertainty in the vehicle or environment is considered. The approach is based on Sequential Quadratic Programming (SQP) wherein a sequence of computationally cheaper quadratic optimization subproblems is solved at each time step. Each subproblem is constructed using a linearization of the system dynamics and safety constraints around the solution from the previous iteration resulting in an iterative refinement of the optimal trajectory. The proposed controller is validated via hardware-in-the-loop simulations and tests on a prototype vehicle. The experiments were performed on snow covered tracks where the nonlinear vehicle dynamics cannot be neglected. We show that the controller can yield smooth trajectories while avoiding static and moving obstacles. Moreover, the use of a longer prediction horizon than existing schemes is enabled which can be useful in emergency situations. The method is general in that any vehicle and environment model can be employed. Moreover, it can be used to solve the nonlinear CFTOC problems appearing in the subsequent chapters.

Chapter 6 proposes a robust control strategy that prevents the vehicle from drifting out of control while being operated by a driver, especially on slippery surfaces such as snow. The novelty of the approach is the systematic consideration of the uncertainty due to modeling errors and the actions of the driver. Set-theoretic methods are used for the control design, wherein a hybrid piecewise affine model of the vehicle dynamics is used and system uncertainties are modeled as set-valued functions of the state and input. The construction of the uncertainty sets is based on data collected from our prototype vehicle. Experiments performed on snow-covered and icy tracks show the ability of the controller to prevent the vehicle from drifting during aggressive maneuvers. While the method is presented for the semi-autonomous driving application with a driver in the loop, it easily extends to the autonomous case.

Chapter 7 proposes a robust predictive control strategy for autonomous cruise control systems. The main contribution is accounting for the uncertainty in the motion of the target vehicle in front of the ego (referred to as the *preceding vehicle*). A worst-case approach is used to predict the motion of the preceding vehicle. Set-theoretic and analytical approaches are presented to construct the robust safety constraints for the ego vehicle. Vehicles cutting-in to the path of the ego vehicle are accounted for by using the lane change intention estimator presented in Chapter 4. In addition, a method for personalizing the system is presented which learns driver preferred control actions from data collected from that driver. Experiments performed using our prototype vehicle on highways demonstrate the performance improvements due to the consideration of cutting-in vehicles. Moreover, the ability of the controller to keep the vehicle safe in certain hazardous situations is shown, where an aggressive braking maneuver is required to prevent a violation of the safety distance.

Chapter 8 uses a probabilistic description of the various system uncertainties and proposes approaches based on stochastic predictive control for automated highway driving applications. First, the motivation for probabilistic approaches over robust ones, especially while dealing with the uncertainty due to traffic forecasts, is provided by means of an illustrative example. Two approaches for stochastic MPC are presented, one where a Gaussian representation of the uncertainty is maintained and the other where samples are used to represent the uncertainty. A hybrid scheme is proposed which combines the benefits of the aforementioned approaches from the perspective of autonomous vehicle control. Simulations show the ability of our approach to systematically trade off risk for conservatism in a typical urban driving scenario. We discuss the implications of using a unimodal description of environmental uncertainty (via Gaussian distributions) versus a multimodal one (via samples) using a simulative example. Experiments on our prototype vehicle demonstrate the controller's ability to run in real-time on embedded platforms and handle a commonly occurring lane change scenario where the reaction of the target vehicle must be considered. Finally, we discuss how the robust control strategies presented in the previous chapters can be combined with the stochastic approach for autonomous navigation in urban driving scenarios.

1.3 List of Publications

The results presented in this dissertation have appeared in a number of publications coauthored by the author of the dissertation, in particular:

- Chapters 2, 3, 4, 8 are based on:
  - A. Carvalho, Y. Gao, S. Lefèvre, and F. Borrelli. "Stochastic predictive control of autonomous vehicles in uncertain environments". In: *12th International Symposium on Advanced Vehicle Control*. 2014,
  - A. Carvalho, S. Lefèvre, G. Schildbach, J. Kong, and F. Borrelli. "Automated driving: The role of forecasts and uncertainty - A control perspective". In: *European Journal of Control* 24 (2015). SI: 2015 European Control Conference, pp. 14–32.
- Chapters 3, 4, 5 are based on:
  - A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli. "Predictive control of an autonomous ground vehicle using an iterative linearization approach". In:

16th IEEE Conference on Intelligent Transportation Systems. 2013, pp. 2335–2340.

- Chapters 3, 6 are based on:
  - A. Carvalho, G. Palmieri, H. E. Tseng, L. Glielmo, and F. Borrelli. "Robust vehicle stability control with an uncertain driver model". In: *2013 European Control Conference*. 2013, pp. 440–445.
- Chapter 7 is based on:
  - S. Lefèvre, A. Carvalho, and F. Borrelli. "Autonomous car following: A learning-based approach". In: *2015 IEEE Intelligent Vehicles Symposium*. 2015,
  - S. Lefèvre, A. Carvalho, and F. Borrelli. "A learning-based framework for velocity control in autonomous driving". In: *IEEE Transactions on Automation Science and Engineering* 13.1 (Jan. 2016), pp. 32–42,
  - A. Carvalho, A. Williams, S. Lefèvre, and F. Borrelli. "Autonomous cruise control with cut-in target vehicle detection". In: *13th International Symposium on Advanced Vehicle Control*. 2016.

Chapter 2

Model Predictive Control for Control Under Uncertainty

In this chapter, the generic formulation and mathematical preliminaries of the predictive control problem for autonomous driving in the presence of uncertainty are presented. First, a simple example is shown to motivate the need for predictive control techniques which can account for uncertainty. Several approaches for the controller design are reviewed which involve transforming the problem into a deterministic one. Finally, an overview of the methods developed in this dissertation is given.

2.1 Notation

Throughout this dissertation, the superscripts v, d and e are used to denote quantities corresponding to the vehicle, driver and environment, respectively.

The subscript t represents discrete time steps. That is, $t+1$ denotes the time instant following t separated by a time step Δt. The sampling times Δt^v, Δt^d and Δt^e are allowed to be different from each other and to vary with time. The subscript k has a special meaning in terms of time. At any time instant t, the predicted value of a variable z at time $t+k$ is denoted by $z_{k|t}$. For simplicity of notation, $z_{k|t}$ is sometimes simply written as z_k; this is clear from the context. The true value (measured or estimated) of a variable z at time t is z_t which is equal to $z_{0|t}$. The notation $z_{t_1:t_2}$ denotes the sequence $\{z_t\}_{t=t_1}^{t_2}$ of values of the variable z from time instant t_1 to t_2.

In the most general case, the states of the vehicle and the environment subsystems are expressed in the belief space, that is, the space of probability distributions. For example, if the uncertainty in the state of the vehicle is characterized by a multivariate Gaussian random variable, the belief state consists of the mean and variance of this variable. The variables x_t^v and u_t^v denote the vehicle state and control input, respectively, at time t. The environment state and the driver's control input are denoted by x_t^e and u_t^d, respectively. Uncertain variables associated with the vehicle, environment and driver are concisely represented by the disturbance vector $d_t^\star$ with $\star \in \{v, e, d\}$.

Calligraphic symbols, for example $\mathcal{X}, \mathcal{U}$ denote sets. Usually, these sets are *polyhedra* or *polytopes*. A polyhedron $\mathcal{X}$ is an intersection of a finite set of closed half-spaces:

$$\mathcal{X} = \{x : a_i^T x \leq b_i,\ i = 1, \ldots, m\}. \tag{2.1}$$

A polytope is a bounded polyhedron.

If a continuous random variable z has a Probability Density Function (PDF) p, this is written as:

$$z \sim p. \tag{2.2}$$

$\mathcal{N}(\mu, \Sigma)$ represents a multivariate Gaussian (or normal) distribution with mean μ and covariance Σ. The corresponding PDF evaluated at a point z is denoted as $\mathcal{N}(z; \mu, \Sigma)$ and is given by:

$$\mathcal{N}(z; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{k}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(z - \mu)^T \Sigma^{-1} (z - \mu) \right), \tag{2.3}$$

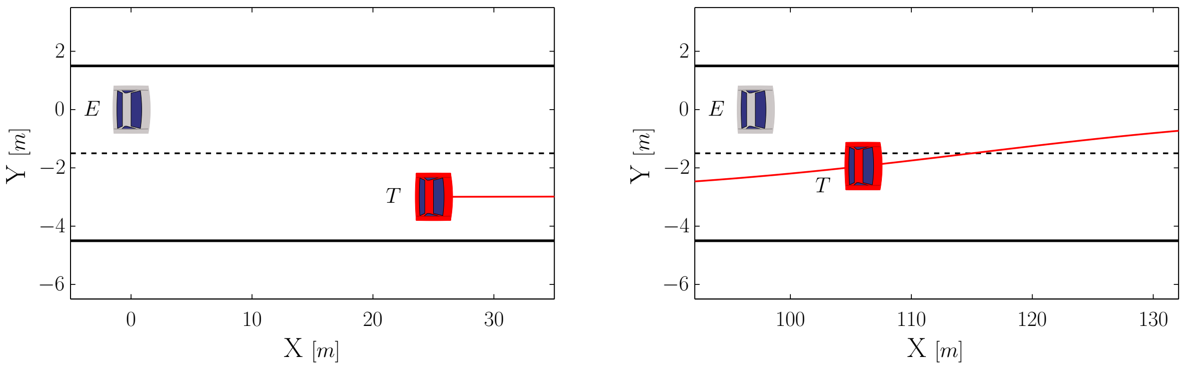
where $|\Sigma|$ is the determinant of Σ.

The symbols I_n and 0_n denote the identity and null (zero) matrices, respectively, of size $n \times n$. The same symbols without the subscript n denote the corresponding matrices of an appropriate size which is clear from the context.

2.2 Model Predictive Control for Active Safety: An Illustrative Example

The objective of this section is to motivate the use of forecasts in designing controllers for intelligent vehicles. We present the design of a linear MPC strategy for an emergency brake assist system. The goal is to track the driver's intended acceleration and intervene only when a violation of the preset safety distance to the vehicle in front is anticipated. Since the focus is on motivating the usefulness of predictive control approaches, the simplified example assumes that there is no uncertainty in the system. That is, the future positions of objects in the environment and the future inputs of the driver are known without uncertainty. In addition, the driver controls the steering such that the vehicle stays in the center of its lane.

The longitudinal motion of the vehicle is described by a double integrator model, where the control input is the longitudinal acceleration. We assume no mismatch between the simulated vehicle model and the one used in the predictive controller design. A learning-based driver model is used to predict the driver's acceleration inputs $\{u_{k|t}^d\}_{k=0}^{T-1}$ over the prediction horizon T (the approach will be described in detail in Chapter 7). At each time step t, the predictive controller solves the following Constrained Finite Time Optimal Control



(a) $t = 0$ s

(b) $t = 5.4$ s

Figure 2.1: Snapshots of simulation with the ego vehicle E in blue and the target vehicle T in red. The red line depicts the trajectory of the target vehicle, which is assumed to be known a-priori.

(CFTOC) problem:

$$\min_{u^c_{0:T-1|t}} \sum_{k=0}^{T-1} \left(u^v_{k|t} - u^d_{k|t}\right)^2, \tag{2.4a}$$

$$\begin{aligned} \text{subject to:}\quad & x^v_{k+1|t} = Ax^v_{k|t} + Bu^v_{k|t}, && (2.4b)\\ & x^v_{k+1|t} \leq x^{\max}_{k+1|t}, && (2.4c)\\ & u^{\min} \leq u^v_{k|t} \leq u^{\max}, && (2.4d)\\ & u^v_{k|t} = u^c_{k|t} + u^d_{k|t}, && (2.4e)\\ & (k = 0, \dots, T-1) \\ & x^v_{0|t} = x^v_t, && (2.4f) \end{aligned}$$

and implements the optimal input $u^v_t = u^c_{0|t} + u^d_{0|t}$ to the double integrator system $x^v_{t+1} = Ax^v_t + Bu^v_t$, where:

$$x^v_t = \begin{bmatrix} \xi_t \\ v_t \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & \Delta t^v \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \Delta t^v \end{bmatrix}. \tag{2.5}$$

The variables ξ_t and v_t denote the longitudinal position and velocity of the ego vehicle. The cost function in (2.4a) penalizes the deviation between the vehicle inputs $u^v_{k|t}$ and the predicted inputs $u^d_{k|t}$ given by the driver model. The variable $x^{\max}_{k|t}$ in (2.4c) is the upper bound on the state $x^v_{k|t}$ and consists of the position bound due to the vehicle in front and the speed limit. The inequalities in (2.4d) represent actuator limits, and (2.4f) is the initial condition for the state.

Two snapshots of the scenario for the simulations are shown in Figure 2.1. The ego vehicle E is moving in the left lane with a slower vehicle T (also referred to as the target vehicle) in

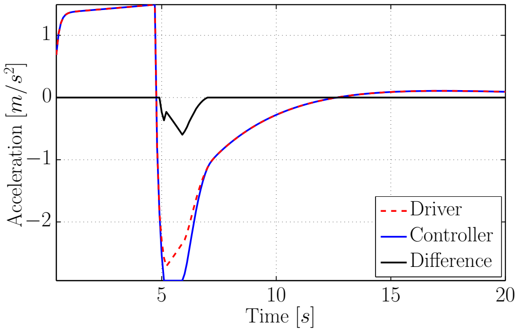


Figure 2.2: Driver and controller inputs during the simulation. The controller tracks the driver's acceleration until a violation of the safety distance is predicted around $t = 5$ s. The controller applies harder braking to keep the vehicle safe. The difference between the driver and controller inputs is shown by the black line.

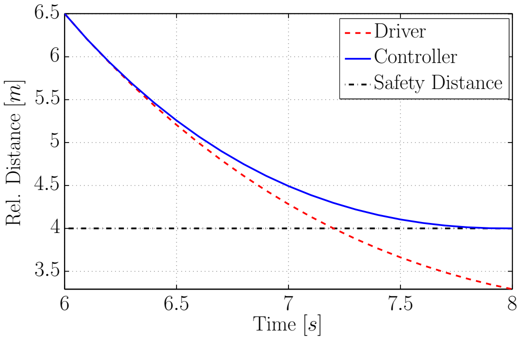


Figure 2.3: Open-loop predictions of the relative distance to the vehicle in front at $t = 6$ s. The red line depicts what would have happened had the controller not intervened. The controller plans to brake harder than the driver to prevent the relative distance going below the specified safety distance of 4 m.

the neighboring lane. As E approaches T, the latter changes lanes into the former's path, requiring a response from the driver of the ego vehicle.

The acceleration inputs applied by the driver and controller are shown in Figure 2.2. Initially, the driver accelerates as there is no vehicle ahead, and the controller matches the driver's input exactly. When T changes lanes into the path of E at $t = 4.8$ s, the driver begins to brake. However, the controller anticipates a violation of the safety distance to the vehicle in front and applies a higher value of braking than the driver. This is seen by the discrepancy between the driver's and the controller's inputs around $t = 5$ s in Figure 2.2. The difference between the two inputs is depicted by the black line.

The open-loop predictions of the relative distance to the vehicle in front at $t = 6$ s are shown in Figure 2.3. If the controller had not intervened, the driver's inputs would have caused the vehicle to violate the specified safety distance of 4 m at $t = 7.2$ s. Figure 2.4 shows that the controller does not allow the relative distance between the ego and target vehicles to go below the specified safety distance during the simulation.

In summary, the above example illustrates the use of predictions of the driver's inputs and the lead vehicle's positions to compute safe control actions for the ego vehicle. However, two important elements are not considered by this example. Firstly, the vehicle model in (2.4) is oversimplified and does not capture the non-linear vehicle dynamics nor the tire-road interaction. However, the use of a higher fidelity vehicle model would increase the complexity of the on-line optimization problem. This trade-off is discussed further in Chapter 3, which presents vehicle models that are suitable for real-time MPC.

The second element not considered is the uncertainty in the measured and predicted values of the vehicle, driver and environment states. The ability of the controller to prevent a collision in the above example relies on perfect knowledge of the vehicle states, driver's

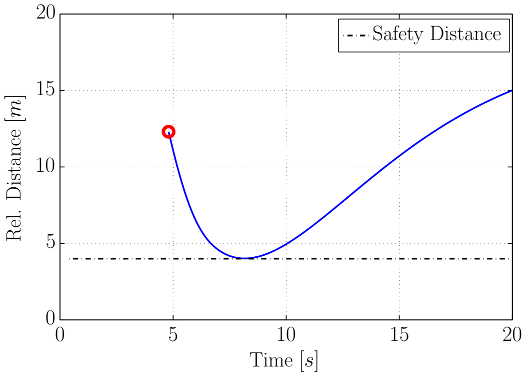


Figure 2.4: Relative distance between lead and controlled vehicles. The red circle at $t = 4.8$ s indicates the instant at which the target vehicle is detected in front of the ego vehicle.

input and position of the target vehicle. These assumptions are not realistic and can easily lead to unsafe control actions. In Chapters 3 and 4, we will present models of the vehicle and environment which include a description of the uncertainty in terms of set bounds or probability density functions (PDFs). This requires MPC design methodologies that account for uncertainty without being overly conservative, which are presented in Chapters 6–8.

In the following section, we present the general formulation of the predictive control problem which accounts for uncertainty at the design stage. Several uncertainty models and solution strategies are also discussed.

2.3 Problem Formulation for Control Under Uncertainty

The following CFTOC problem will be used to present the control methodologies developed in this dissertation:

$$\min_{\kappa^c_{0:T-1|t}(\cdot)} \sum_{k=0}^{T-1} E\left[J(x^v_{k+1|t}, x^{\text{ref}}_{k+1|t}, u^v_{k|t}, u^v_{k-1|t}, u^d_{k|t})\right], \tag{2.6a}$$

$$\begin{aligned}
\text{subject to: } \quad & x^v_{k+1|t} = f^v(x^v_{k|t}, u^v_{k|t}, d^v_{k|t}), && \text{(Vehicle model)} \quad (2.6\text{b})\\
& x^e_{k+1|t} = f^e(x^v_{k|t}, x^e_{k|t}, d^e_{k|t}), && \text{(Environment model)} \quad (2.6\text{c})\\
& u^d_{k|t} = f^d(x^v_{k|t}, x^e_{k|t}, d^d_{k|t}), && \text{(Driver model)} \quad (2.6\text{d})\\
& u^c_{k|t} = \kappa^c_{k|t}(x^v_{k|t}, x^e_{k|t}), && \text{(Control policy)} \quad (2.6\text{e})\\
& u^v_{k|t} = f^u(u^d_{k|t}, u^c_{k|t}), && \text{(Input mapping)} \quad (2.6\text{f})\\
& [u^v_{k|t}, u^v_{k-1|t}] \in \mathcal{U}, && \text{(Input constraints)} \quad (2.6\text{g})\\
& g(x^v_{k+1|t}, x^e_{k+1|t}) \leq 0, && \text{(Safety constraints)} \quad (2.6\text{h})
\end{aligned}$$

$$(k = 0, \dots, T-1)$$
$$x^\star_{0|t} = x^\star_t, \quad u^v_{-1|t} = u^v_{t-1}, \quad (\star \in \{v, e\}) \qquad \text{(Initial conditions)} \qquad (2.6\text{i})$$

where $E[z]$ denotes the expected value of z. Problem (2.6) seeks a set of feedback control policies $\kappa^c_{0:T-1|t}(\cdot) = \{\kappa^c_{k|t}(\cdot)\}_{k=0}^{T-1}$ which minimizes the expected cost (2.6a) and satisfies the safety and comfort constraints (2.6g)–(2.6h). The state-feedback policies in (2.6e) map the perceived vehicle and environment state to the control input. The reference trajectory $\{x^{\text{ref}}_{k|t}\}_{k=1}^{T}$ consists of information such as the desired speed and lane. The vehicle model $f^v(\cdot)$ in (2.6b) and the environment model $f^e(\cdot)$ in (2.6c) represent the uncertain dynamics of the corresponding state vectors. The driver model $f^d(\cdot)$ in (2.6d) is a mapping from the vehicle and environment states to the control actions of the driver. The mapping $f^u(\cdot)$ in (2.6f) defines how the inputs of the driver and controller are combined to obtain the input to the vehicle. The initial conditions based on the current state measurements and control input applied at the previous time step are given in (2.6i). In the following sections, we use z_k to denote the variable $z_{k|t}$.

2.3.1 Solution Methodologies

The solution to the CFTOC problem (2.6) depends on the characterization of the uncertainty $d^\star$ (with $\star \in \{v, e, d\}$) in the system models, which can be done in several ways. We denote by $\mathcal{D}^\star$ the set of all possible realizations of the uncertainty $d^\star$. Let $p^\star$ denote the probability density function (PDF) of $d^\star$ over the set $\mathcal{D}^\star$. Further, let $\bar{d}^\star \in \mathcal{D}^\star$ be a point estimate of $d^\star$, for example, the expected value $\hat{d}^\star$ of $d^\star$ under the distribution $p^\star$ over the set $\mathcal{D}^\star$. The above quantities are allowed to be time-varying in which case the subscript k is used, that is, $\mathcal{D}^\star_k$, $p^\star_k$, $\bar{d}^\star_k$.

The uncertainty $d^\star_k$ in the system models $f^\star(\cdot)$ result in the predicted states x^v_k and x^e_k being uncertain. Consequently, the safety constraints (2.6h) cannot be treated deterministically. Instead,

- constraint satisfaction can be enforced for a *nominal value* of the disturbance:

$$g(x^v_k, x^e_k) \leq 0 \text{ with } d^\star_j = \bar{d}^\star_j \;\; \forall j < k. \qquad (2.7)$$

  The constraints (2.7) are referred to as *nominal constraints* and the problem obtained by replacing constraints (2.6h) with (2.7) in (2.6) is called a *Nominal MPC problem.*

- constraint satisfaction can be enforced for *all admissible values* of the disturbance:

$$g(x^v_k, x^e_k) \leq 0 \qquad \forall d^\star_j \in \mathcal{D}^\star_j \;\; \forall j < k. \qquad (2.8)$$

  The constraints (2.8) are referred to as *robust constraints* and the problem obtained by replacing constraints (2.6h) with (2.8) in (2.6) is called a *Robust MPC (RMPC) problem.*

- we can allow for a small probability of constraint violation:

$$P\left(g(x_k^v, x_k^e) \leq 0\right) \geq 1 - \varepsilon_k \quad \text{with} \quad d_j^\star \sim p_j^\star \quad \forall j < k, \tag{2.9}$$

  where $P(A)$ is the probability that the event A is true. The notation $d_j^\star \sim p_j^\star$ implies that the random variable $d_j^\star$ has the PDF $p_j^\star$. The constraints (2.9) are referred to as *chance constraints* and the problem obtained by replacing constraints (2.6h) with (2.9) in (2.6) is called a *Stochastic MPC (SMPC) problem.*

The solution to the CFTOC problem (2.6) comprises three main steps:

(i) the translation of the optimization over control polices $\kappa_k^c(\cdot)$ into a finite-dimensional optimization problem,

(ii) the propagation of the uncertain system states over the prediction horizon in order to translate the robust constraints (2.8) or probabilistic constraints (2.9) into deterministic constraints, and

(iii) the solution of the resulting mathematical program.

With the exception of linear systems and special classes of disturbance sets $\mathcal{D}^\star$ (e.g., polyhedra) or probability distributions $p^\star$ (e.g., Gaussians), steps (i) and (ii) are non-trivial for the robust and stochastic MPC problems. Various methods have been presented in literature to handle steps (i) and (ii), and they affect the complexity of the mathematical program in step (iii) as well as the conservatism of the resulting solution (see, for example, [25, 63, 86, 129]).

There are two broad approaches for solving step (i) [14]:

- *Open-loop* prediction schemes solve for a sequence of control actions over the prediction horizon instead of a mapping $\kappa_k^c(\cdot)$. That is:

$$\kappa_k^c(x_k^v, x_k^e) = \hat{u}_k^c, \tag{2.10}$$

  where $\{\hat{u}_k^c\}_{k=0}^{T-1}$ are the optimization variables in (2.6) instead of $\{\kappa_k^c\}_{k=0}^{T-1}$.

- *Closed-loop* prediction schemes solve for a parametric representation of the mapping $\kappa_k^c(\cdot)$. That is:

$$\kappa_k^c(x_k^v, x_k^e) = \kappa_k^c(x_k^v, x_k^e; \theta_k^c), \tag{2.11}$$

  where $\{\theta_k^c\}_{k=0}^{T-1}$ are the optimization variables in (2.6) instead of $\{\kappa_k^c\}_{k=0}^{T-1}$.

Open-loop approaches are conservative as they look for one input sequence which has to counteract all possible future uncertainty realizations, without taking into account the fact that measurements will be available in the future. Closed-loop formulations overcome this issue but they can lead to computationally intractable problems as the space of control policies is infinite dimensional. A compromise lies in fixing the control structure (e.g., affine state or disturbance feedback policies), parameterizing the control sequence in the feedback gains, and optimizing over these parameters [70, 71]. The on-line computational complexity can be further reduced by computing the state or disturbance feedback gain off-line (e.g., using the Linear-Quadratic Regulator (LQR) algorithm).

The translation of the robust constraints (2.8) in step (ii) into deterministic ones on the nominal state (that is, the state of the system assuming a point estimate of the disturbances) involves propagating the worst-case effect of the disturbance set over the horizon. For linear systems with additive bounded disturbances described by polyhedral sets, the state constraints can be tightened using the reachable set of the system under the disturbance input [34]. In [27], a tube-based approach is developed where a nominal MPC problem is solved with tightened constraints. The amount of tightening is computed on the basis of a robust positively invariant set of the uncertain system around a nominal trajectory under a precomputed feedback law. This approach extends to nonlinear systems which are affine in the control input [129].

The chance constraints (2.9) in step (ii) are translated into deterministic ones by enforcing tightened constraints on the expected values of the states and inputs. The tightening offset is computed based on the tails of the disturbance probability distributions [60, 111]. In practice, the PDFs have a finite support and are non-Gaussian. In addition, the vehicle models are non-linear, making it difficult to propagate the distribution of the vehicle states over the prediction horizon. Exact solutions to the SMPC problem for non-linear systems subject to non-Gaussian disturbances are, in general, computationally intractable for real-time implementation on current vehicle platforms. Analytical methods are based on a linearization of the system dynamics and a Gaussian approximation of the uncertainty. Sampling-based methods provide an alternative approach to transform the chance constraints for non-Gaussian random variables (see, for example, [20, 24, 50]) The approach consists in transforming the chance constraints (2.9) into deterministic counterparts by evaluating them at a large number of disturbance samples.

Non-Gaussian and finitely supported disturbances have also been studied in [26, 27, 70, 71, 88, 113] in the context of SMPC. In [26, 27], a tube-based method is used to translate the chance constraints. The work in [70, 71] suggests to translate the chance constraints into deterministic ones by using tightening offsets. The offsets are computed off-line using numerical approximations of convolution integrals. In [88, 113], polynomial expansions are used to approximate probability distribution functions.

2.3.2 Contributions

With regards to solving the CFTOC problem (2.6) in the context of autonomous driving applications, the contributions of this dissertation are as follows:

- In Chapter 5, a tailored sequential quadratic programming (SQP) algorithm for solving the nonlinear Nominal MPC problem is developed to reduce the online computational burden.

- In Chapter 6, a Robust MPC approach for lateral stability control is presented which accounts for uncertainty in the driver's actions.
- In Chapter 7, a Robust MPC approach for Autonomous Cruise Control (ACC) systems is developed which accounts for the uncertainty in the motion of the preceding vehicle.
- In Chapter 8, analytical and sampling-based Stochastic MPC approaches are used for an automated highway driving application. Uncertainty arising from the modeling and state estimation errors and the motion of surrounding vehicles is considered.

The aforementioned methods rely on the vehicle and environment models presented in Chapters 3 and 4.

2.3.3 Practical Considerations

Interaction between vehicle, driver and environment

The CFTOC problem (2.6) accounts for the models of the environment and driver in the optimization problem and their interaction with the autonomous vehicle. From an implementation standpoint, this requires embedding the predictive models (2.6c) and (2.6d) of the environment and driver, respectively, in the numerical optimization solver. This implies evaluating the model outputs at each inner iteration of the optimization algorithm and their gradients with respect to the decision variables. In general, this is intractable due to the computations involved in these models. Moreover, they may not be differentiable making the use of gradient-based solvers impossible. We address this practical issue as follows.

Notice that the models (2.6c) and (2.6d) depend on the autonomous vehicle's state x_k^v which is not known ahead of time. If an estimate of x_k^v were available over the prediction horizon, the dynamics of the driver and environment can be decoupled from that of the vehicle and can be taken outside the CFTOC problem (2.6). Fortunately, as the finite horizon problem is solved at each time step, an estimate of the vehicle's expected future states is available based on the open-loop solution of (2.6) at the previous time step. Accordingly, we introduce the following assumption:

Assumption 2.1. *At time step t, the predicted states of the ego vehicle at time step $(t-1)$ are used to decouple the environment and driver models (2.6c) and (2.6d) from the vehicle state x_k^v. That is:*

$$x^e_{k+1|t} = f^e(x^{v^\star}_{k+1|t-1}, x^e_{k|t}, d^e_{k|t}), \tag{2.12a}$$

$$u^d_{k|t} = f^d(x^{v^\star}_{k+1|t-1}, x^e_{k|t}, d^d_{k|t}), \tag{2.12b}$$

where $\{x^{v^\star}_{k+1|t-1}\}_{k=0}^{T-1}$ is the optimal vehicle state sequence obtained from the solution of the CFTOC problem (2.6) at time step $(t-1)$.

This implies that the predicted environment state x^e_k and driver input u^d_k are computed outside the optimization problem and fed as inputs to the solver. The predictions are still allowed to be uncertain due to their dependence on the random variables d^e_k and d^d_k.

We also make a simplification regarding the feedback policy $\kappa^c_k(\cdot)$ to be optimized. In the most general form, $\kappa^c_k(\cdot)$ is a function of x^v_k and x^e_k. A meaningful policy for the autonomous driving task would be highly nonlinear and intractable to compute in a real-time optimization framework. Instead, we use a partially closed-loop approach (to be differentiated from the work and terminology in [116]) where $\kappa^c_k(\cdot)$ is only a function of the vehicle state x^v_k. This is formalized by the following assumption:

Assumption 2.2. *The feedback policy $\kappa^c_k(\cdot)$ is independent of the environment state x^e_k. That is:*

$$u^c_k = \kappa^c_k(x^v_k). \tag{2.13}$$

The above assumption is aligned with Assumption 2.1 by which the environment and driver models are decoupled from the control inputs u^c_k.

Combining problem (2.6) with Assumptions 2.1 and 2.2, the modified CFTOC can be written as:

$$\begin{aligned}
&\min_{\kappa^c_{0:T-1|t}(\cdot)} \sum_{k=0}^{T-1} E\left[J(x^v_{k+1|t}, u^v_{k|t}, u^v_{k-1|t}, x^{\text{ref}}_{k+1|t}, u^d_{k|t})\right], && \text{(2.14a)}\\
\text{subject to: } & x^v_{k+1|t} = f^v(x^v_{k|t}, u^v_{k|t}, d^v_{k|t}), && \text{(2.14b)}\\
& u^v_{k|t} = f^u(u^d_{k|t}, u^c_{k|t}), && \text{(2.14c)}\\
& u^c_{k|t} = \kappa^c_{k|t}(x^v_{k|t}), && \text{(2.14d)}\\
& [u^v_{k|t}, u^v_{k-1|t}] \in \mathcal{U}, && \text{(2.14e)}\\
& g(x^v_{k+1|t}, x^e_{k+1|t}) \leq 0, && \text{(2.14f)}\\
& (k = 0, \ldots, T-1) &&\\
& x^v_{0|t} = x^v_t, \quad u^v_{-1|t} = u^v_{t-1}. && \text{(2.14g)}
\end{aligned}$$

Control input mapping

With the exception of the semi-autonomous driving application presented in Chapter 6, this dissertation focuses on fully autonomous driving applications. In these cases, the driver does not apply any input to the system. Concretely, the mapping $f^u(\cdot)$ is given by:

$$u^v_k = f^u(u^d_k, u^c_k) = u^c_k. \tag{2.15}$$

In the semi-autonomous driving application in Chapter 6, the driver has the ability to directly augment the vehicle steering. In that case:

$$u^v_k = f^u(u^d_k, u^c_k) = u^d_k + u^c_k. \tag{2.16}$$

Chapter 3

Vehicle Dynamics Models

3.1 Introduction

In the context of path planning and control of autonomous vehicles, the goal of the vehicle dynamics model is to provide a relationship between physical inputs and the position and orientation of the vehicle. Vehicle models can be broadly classified, in an increasing order of complexity, into three categories:

1. *Point-mass models* treat the vehicle as a particle. Such models yield large tracking errors when used for path planning due to their inability to account for dynamic feasibility [48].

2. *Kinematic models* are a function of vehicle geometry, and can represent the vehicle motion in a range of conditions which does not involve highly dynamic maneuvers and/or tire force saturation [69].

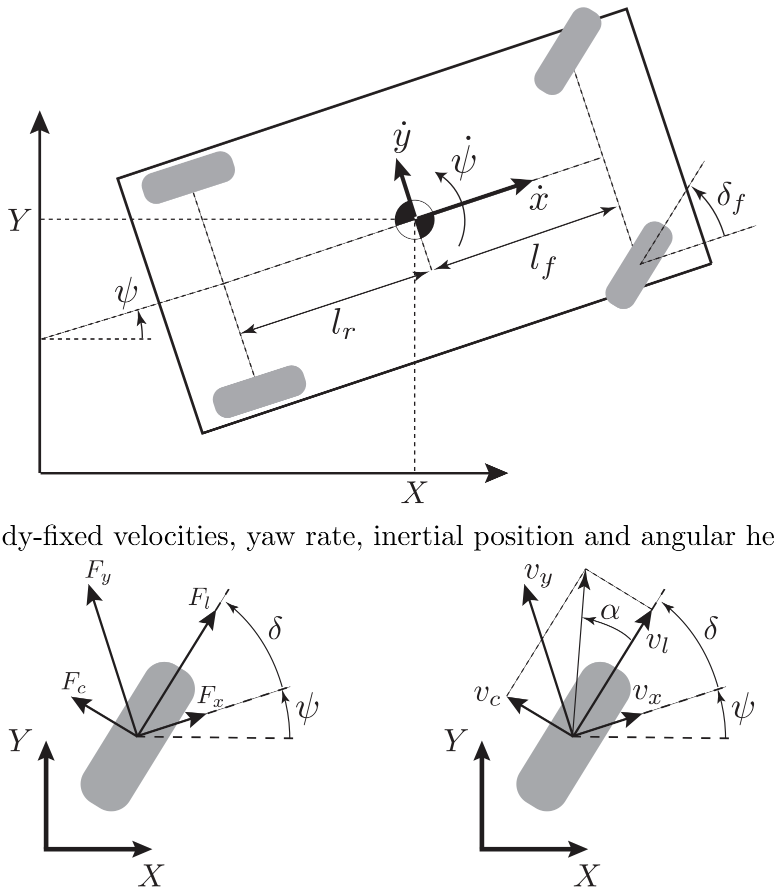
3. *Dynamic models* rely on tire models to describe the interaction between the vehicle and the road. In this case, the complexity arises from the non-linear relationship between the tire forces, and the vehicle states and inputs [96, 101].

This chapter focuses on vehicle models that are useful from a control perspective, that is, models that capture the essence of the dynamics while being computationally cheap for real-time model-based control strategies. Recall that the discrete-time state update equations corresponding to the vehicle model are compactly written as:

$$x^v_{t+1} = f^v(x^v_t, u^v_t, d^v_t), \tag{3.1}$$

where x^v_t and u^v_t are the vehicle state and input vectors, respectively. The variable d^v_t denotes the disturbance vector, which is introduced to represent parametric uncertainty and modeling errors. In formulations where the system dynamics depend on time-varying exogenous parameters (such as the road curvature), the vehicle model is written as:

$$x^v_{t+1} = f^v(x^v_t, u^v_t, d^v_t; p^v_t), \tag{3.2}$$



(a) Body-fixed velocities, yaw rate, inertial position and angular heading

(b) Body-fixed and tire longitudinal and lateral forces

(c) Body-fixed and tire longitudinal and lateral velocities

Figure 3.1: Notation used in the nonlinear vehicle model.

where p_t^v is the parameter vector. These must be differentiated from physical parameters such as the vehicle mass and dimensions which are assumed to be constant or to vary slowly with time. Without loss of generality, the parameter vector is omitted in the presentation of the control design in the later chapters.

The following sections detail the models that are used in the various applications presented in this dissertation. We first present nominal models which do not consider disturbances or modeling errors in Sections 3.2–3.5 and then present some characterizations of model uncertainty in Section 3.6.

3.2 Dynamic Models

3.2.1 Four Wheel Nonlinear Vehicle Model

The four wheel nonlinear model considers the effects of the tire forces at each of the four wheels. The notation used is depicted in Figure 3.1. In addition, the subscript $\star \in \{f, r\}$ is used to indicate variables corresponding to the front (f) and rear (r) axles of the vehicle while the subscript $\bullet \in \{l, r\}$ is used for the left (l) and right (r) sides. The following set of nonlinear differential equations are used to describe the motion of the vehicle in an inertial coordinate frame:

$$\ddot{x} = \dot{y}\dot{\psi} + \frac{1}{m} \sum_{\star \in \{f,r\}} \sum_{\bullet \in \{l,r\}} F_{x_{\star\bullet}}, \tag{3.3a}$$

$$\ddot{y} = -\dot{x}\dot{\psi} + \frac{1}{m} \sum_{\star \in \{f,r\}} \sum_{\bullet \in \{l,r\}} F_{y_{\star\bullet}}, \tag{3.3b}$$

$$\ddot{\psi} = \frac{1}{I_z}(l_f \sum_{\bullet \in \{l,r\}} F_{y_{f\bullet}} - l_r \sum_{\bullet \in \{l,r\}} F_{y_{r\bullet}} + l_w(-F_{x_{fl}} + F_{x_{fr}} - F_{x_{rl}} + F_{x_{rr}})), \tag{3.3c}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \tag{3.3d}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi, \tag{3.3e}$$

where m and I_z denote the vehicle mass and yaw inertia, respectively, l_f and l_r denote the distances from the vehicle's Center of Gravity (CoG) to the front and rear axles, respectively, and l_w denotes the vehicle's half track-width. $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral velocities in the body-fixed coordinate frame, respectively, and $\dot{\psi}$ denotes the yaw rate. X and Y denote the inertial coordinates of the vehicle, and ψ is the angular heading. The above variables are shown in Figure 3.1a.

The longitudinal and lateral forces in the body-fixed coordinate frame $F_{x_{\star\bullet}}$ and $F_{y_{\star\bullet}}$, respectively, are given by:

$$F_{x_{\star\bullet}} = F_{l_{\star\bullet}} \cos\delta_{\star\bullet} - F_{c_{\star\bullet}} \sin\delta_{\star\bullet}, \tag{3.4a}$$

$$F_{y_{\star\bullet}} = F_{l_{\star\bullet}} \sin\delta_{\star\bullet} + F_{c_{\star\bullet}} \cos\delta_{\star\bullet}, \tag{3.4b}$$

where the variables $F_{l_{\star\bullet}}$ and $F_{c_{\star\bullet}}$ are the longitudinal and lateral (cornering) forces in the coordinate frame aligned with the tire axes, and $\delta_\star$ is the steering angle between the wheel and the road at the corresponding wheel (see Figure 3.1b for an illustration). The following assumption is introduced on the steering angles:

Assumption 3.1. *Only the front wheels can be steered, and the steering angles at the front left and right wheels are equal, i.e.,* $\delta_{fl} = \delta_{fr} = \delta_f$, *and* $\delta_{rl} = \delta_{rr} = 0$.

The relationship between the lateral forces $F_{c_{\star\bullet}}$ and the vehicle states are described by *tire models* of the form:

$$F_{c_{\star\bullet}} = h_c(\alpha_{\star\bullet}, \sigma_{\star\bullet}, F_{z_{\star\bullet}}, \mu_{\star\bullet}), \tag{3.5}$$

where $\alpha_{\star\bullet}$ and $\sigma_{\star\bullet}$ denote the tire slip angle and slip ratio, respectively, $F_{z_{\star\bullet}}$ is the normal force and $\mu_{\star\bullet}$ is the friction coefficient between the tire and the road surface. Details of the tire models are presented in Section 3.2.4. The slip angles and slip ratios are a consequence of the distortion of the tire contact patch when the vehicle is steered and/or it accelerates or brakes. These quantities are formally defined below.

Definition 3.2 (Slip angles). *The slip angles $\alpha_{\star\bullet}$ are defined as the inverse tangent of the ratio of the lateral and longitudinal velocities:*

$$\alpha_{\star\bullet} = \arctan \frac{v_{c_{\star\bullet}}}{v_{l_{\star\bullet}}}, \tag{3.6}$$

where:

$$v_{c_{\star\bullet}} = v_{y_{\star\bullet}} \cos\delta_{\star\bullet} - v_{x_{\star\bullet}} \sin\delta_{\star\bullet}, \tag{3.7a}$$

$$v_{l_{\star\bullet}} = v_{y_{\star\bullet}} \sin\delta_{\star\bullet} + v_{x_{\star\bullet}} \cos\delta_{\star\bullet}, \tag{3.7b}$$

$$v_{y_{f\bullet}} = \dot{y} + l_f\dot{\psi}, \quad v_{x_{\star l}} = \dot{x} - l_w\dot{\psi}, \tag{3.7c}$$

$$v_{y_{r\bullet}} = \dot{y} - l_r\dot{\psi}, \quad v_{x_{\star r}} = \dot{x} + l_w\dot{\psi}. \tag{3.7d}$$

The slip angle is the effective direction of motion at the wheels with respect to the longitudinal axis of the wheel. This is illustrated in Figure 3.1c.

Definition 3.3 (Slip ratios). *The slip ratios are defined as:*

$$\sigma_{\star\bullet} = \begin{cases} \frac{R\omega_{\star\bullet}}{v_{l_{\star\bullet}}} - 1 & \text{if } v_{l_{\star\bullet}} \geq R\omega_{\star\bullet} \\ 1 - \frac{v_{l_{\star\bullet}}}{R\omega_{\star\bullet}} & \text{if } v_{l_{\star\bullet}} < R\omega_{\star\bullet} \end{cases}, \tag{3.8}$$

where R is the wheel radius and $\omega_{\star\bullet}$ is the angular velocity of the wheel.

The slip ratio $\sigma_{\star\bullet} = -1$ corresponds to the case when the wheels are locked (i.e. $\omega_{\star\bullet} = 0, v_{l_{\star\bullet}} \neq 0$), while $\sigma_{\star\bullet} = 1$ corresponds to a "spin-out" (i.e. $v_{l_{\star\bullet}} = 0, \omega_{\star\bullet} \neq 0$). The dependence of the slip ratio on the angular velocity $\omega_{\star\bullet}$ necessitates the consideration of the rotational dynamics of the wheels in addition to the vehicle dynamics in (3.3). From the perspective of model-based control design, the state vector x^v gets extended by four elements (corresponding to the angular velocities at the four wheels), significantly increasing the controller's computational complexity. Hence, we introduce the following assumption:

Assumption 3.4. *For the computation of the lateral tire forces $F_{c_{\star\bullet}}$ by* (3.5)*, the slip ratios $\sigma_{\star\bullet}$ at each of the four wheels is assumed to be equal to zero.*

In general, the longitudinal tire forces $F_{l_{\star\bullet}}$ also have a nonlinear dependence on the slip angles and ratios as in the case of the lateral forces in (3.5). However, as discussed above, this would require a consideration of the wheel dynamics which is not desirable. Moreover, the assumption of zero slip ratio (Assumption 3.4) is not valid in the longitudinal case as

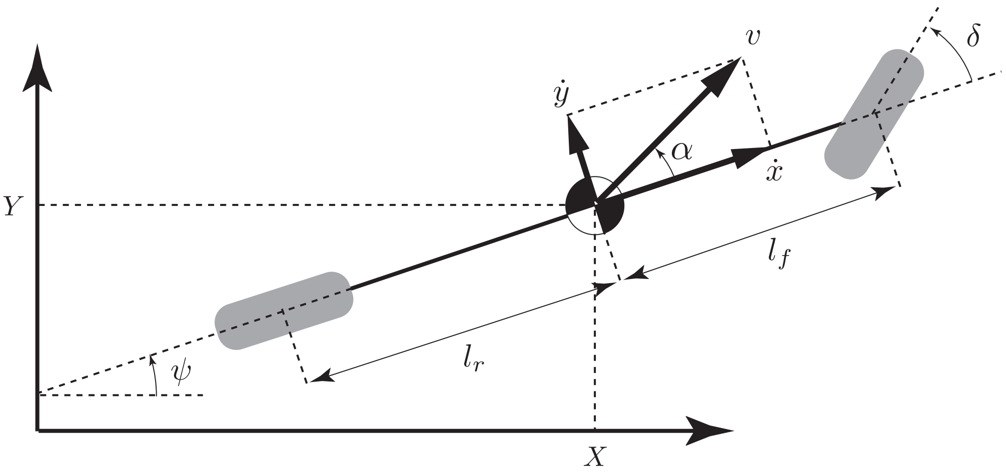


Figure 3.2: Notation used in the dynamic and kinematic bicycle models.

the slip ratio is the primary reason for the generation of longitudinal tire forces. Instead, we model the longitudinal forces $F_{l_{\star\bullet}}$ in (3.4) as:

$$F_{l_{\star\bullet}} = \beta_{\star\bullet}\mu_{\star\bullet}F_{z_{\star\bullet}}, \tag{3.9}$$

where $\beta_{\star\bullet}$ is referred to as the braking ratio. Intuitively, it is the ratio of the longitudinal tire force $F_{l_{\star\bullet}}$ to the total available force $\mu_{\star\bullet}F_{z_{\star\bullet}}$. As with the slip ratio, $\beta_{\star\bullet} = -1$ corresponds to the case when the wheels are locked, while $\beta_{\star\bullet} = 1$ corresponds to a spin-out. The existence of a lower level controller is assumed which controls the individual wheel braking torques and the engine throttle in order to track a given value of $\beta_{\star\bullet}$. Hence, the braking ratio $\beta_{\star\bullet}$ is treated as the control input.

Summary

The discrete-time model (3.1) is obtained from (3.3) by the first-order forward Euler method. The state and input vectors for the four wheel nonlinear model are defined as $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, X_t, Y_t, \psi_t]^T$ and $u_t^v = [\delta_{f_t}, \beta_{fl_t}, \beta_{fr_t}, \beta_{rl_t}, \beta_{rr_t}]^T$, respectively.

3.2.2 Dynamic bicycle model

Bicycle models represent the left and right wheels of the vehicle by a single wheel. This approximation is a common approach in developing models suitable for control design (see e.g. [31, 47, 54, 101]). The notation used is depicted in Figure 3.2. The nonlinear differential equations in this case are given by:

$$\ddot{x} = a_x, \tag{3.10a}$$

$$\ddot{y} = -\dot{x}\dot{\psi} + \frac{2}{m}\left(F_{c_f}\cos\delta_f + F_{c_r}\right), \tag{3.10b}$$

$$\ddot{\psi} = \frac{2}{I_z}\left(l_f F_{c_f}\cos\delta_f - l_r F_{c_r}\right), \tag{3.10c}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \tag{3.10d}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi, \tag{3.10e}$$

where a_x is the external longitudinal acceleration, and the variables F_{c_f} and F_{c_r} are the lateral tire forces at the front and rear axles, respectively, given by tire models of the form (detailed in Section 3.2.4):

$$F_{c_\star} = h_c(\alpha_\star, \sigma_\star, F_{z_\star}, \mu_\star) \quad \text{for } \star \in \{f, r\}. \tag{3.11}$$

A modified definition of the slip angles $\alpha_\star$ is used in the case of the bicycle model.

Definition 3.5 (Slip angles)**.** *The slip angles $\alpha_\star$ are defined as:*

$$\alpha_\star = \arctan\frac{v_{c_\star}}{v_{l_\star}}, \tag{3.12}$$

where:

$$v_{c_\star} = v_{y_\star}\cos\delta_\star - v_{x_\star}\sin\delta_\star, \tag{3.13a}$$

$$v_{l_\star} = v_{y_\star}\sin\delta_\star + v_{x_\star}\cos\delta_\star, \tag{3.13b}$$

$$v_{y_f} = \dot{y} + l_f\dot{\psi}, \quad v_{y_r} = \dot{y} - l_r\dot{\psi}, \tag{3.13c}$$

$$v_{x_\star} = \dot{x} - l_w\dot{\psi}. \tag{3.13d}$$

with $\delta_r = 0$.

Note that the body slip angle α shown in Figure 3.2 is given by:

$$\alpha = \arctan\frac{\dot{y}}{\dot{x}}, \tag{3.14}$$

and is used later in the kinematic bicycle model (Section 3.3.1). Under the assumption of small steering angle, lateral velocity and yaw rate, the tire slip angles can be approximated as:

$$\alpha_f = -\delta_f + \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}, \tag{3.15a}$$

$$\alpha_r = \frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}. \tag{3.15b}$$

This is useful as the slip angles are a linear function of the states and inputs under the additional assumption of constant longitudinal velocity $\dot{x}$. We exploit this property later in Chapter 6.

Summary

The discrete-time model (3.1) is obtained from (3.10) by the first-order forward Euler method. The state and input vectors for the dynamic bicycle model are defined as $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, X_t, Y_t, \psi_t]^T$ and $u_t^v = [\delta_{ft}, a_{xt}]^T$, respectively.

3.2.3 Lateral Model for Yaw Moment Control

The models presented above in Sections 3.2.1 and 3.2.2 describe both the longitudinal and lateral motion of the vehicle. Moreover, as they include the kinematics (i.e., the evolution of the position and orientation), they are suitable for the purpose of path planning and control. For applications such as lateral stability control (discussed in Chapter 6), only a description of the lateral dynamical states (i.e., excluding the kinematics) is necessary. The following modified nonlinear bicycle model is used [67, 101]:

$$\ddot{y} = -\dot{x}\dot{\psi} + \frac{2}{m}(F_{c_f} + F_{c_r}), \tag{3.16a}$$

$$\ddot{\psi} = \frac{2}{I_z}(l_f F_{c_f} - l_r F_{c_r}) + \frac{1}{I_z} M_z, \tag{3.16b}$$

where M_z is the external yaw moment which is generated by means of the slip ratio controllers used in the nonlinear vehicle model in Section 3.2.1. The lateral forces $F_{c_\star}$ are computed by the tire models presented in Section 3.2.4. The lateral dynamics model is based on the following assumptions:

Assumption 3.6. *For the evaluation of the tire forces F_{c_f} and F_{c_r}, the friction coefficient μ and the normal force F_z are known, constant and identical for both wheels.*

Assumption 3.7. *The longitudinal velocity $\dot{x}$ is assumed to be known and constant over the prediction horizon for the purpose of control design.*

The discrete-time model (3.1) is obtained from (3.16) by the first-order forward Euler method. The state and input vectors for the dynamic bicycle model are defined as $x_t^v = [\dot{y}_t, \dot{\psi}_t]^T$ and $u_t^v = [\delta_{ft}, M_{zt}]^T$, respectively. As the longitudinal speed $\dot{x}_t$ is time-varying and not controlled, it is a part of the parameter vector p_t^v in (3.2).

3.2.4 Tire Models

As seen in Sections 3.2.1–3.2.3, tire models provide a mathematical relationship between the lateral tire forces F_c and the vehicle states:

$$F_{c_{\star\bullet}} = h_c(\alpha_{\star\bullet}, \sigma_{\star\bullet}, F_{z_{\star\bullet}}, \mu_{\star\bullet}). \tag{3.17}$$

In the subsequent discussion, the subscripts $\star\bullet$ corresponding to the wheel are dropped for simplicity of notation. Several approaches for modeling tire behavior have been presented in the literature. The *Pacejka tire model* is a semi-empirical model commonly used in high-fidelity vehicle simulation models [96]. Its main limitations are the large number of parameters to be identified and the black-box nature of the model. An analytical approximation of the Pacejka model is given by:

$$F_c = D \sin\left(C \arctan\left(B(1-E)\alpha + E \arctan\left(B\alpha\right)\right)\right), \tag{3.18}$$

where B, C, D, E are constants which are identified from experimental data. Note that this approach ignores the effect of σ, F_z and μ. Typically, a bank of such models is used, each valid in a small range of σ, normal F_z and μ.

The *Fiala tire model* is more suited for control design due to its explicit representation involving few parameters [62]:

$$F_c = \begin{cases} -C_\alpha \tan\alpha + \frac{C_\alpha^2}{3\mu F_z\sqrt{1-\beta^2}}|\tan\alpha|\tan\alpha - \frac{C_\alpha^3}{27(1-\beta^2)\mu^2 F_z^2}\tan^3\alpha, & \text{if } |\alpha| < \alpha_{sl}, \\ -\sqrt{1-\beta^2}\mu F_z \text{sign } \alpha, & \text{if } |\alpha| \geq \alpha_{sl}, \end{cases} \tag{3.19}$$

where C_α denotes the tire cornering stiffness and $\alpha_{sl} = \tan^{-1}\left(\frac{3\mu F_z}{C_\alpha}\sqrt{1-\beta^2}\right)$ is the tire slip angle at which maximum lateral force is obtained. Recall that β is the braking ratio which was introduced in (3.9).

A common approach used in automotive active safety applications is the *linear tire model*, wherein $h(\cdot)$ in (3.17) is a linear function of the slip angle:

$$F_c = -C_\alpha \alpha. \tag{3.20}$$

As in the case of the Pacejka model, a bank of such models can be used to account for effects due to varying σ, F_z and μ. A benefit of the linear tire model (3.20) is that combined with the slip angle approximation (3.15), a Linear Parameter Varying (LPV) approximation of the dynamic bicycle model (3.10) can be derived [101]. This allows the use of techniques from linear systems theory for the controller design.

Figure 3.3 shows the variation of lateral tire force with slip angle for the aforementioned tire models, for a given slip ratio, normal force and friction coefficient. The gray dots depict the lateral force values estimated from data collected during experiments on a winding road and a ‘figure 8’ drift maneuver. The speed of the vehicle ranged from 10 to 35 m/s during these experiments. The parameters of each of the tire models are identified from the experimental data using a standard nonlinear least-squares regression approach. That is, the parameters are chosen to minimize the error (measured by the vector 2-norm) between the state predicted by the vehicle model (3.1) and that measured during the experiment. It is seen that the linear tire model fits the data well for low slip angles. At higher slip angles, however, the nonlinear effects introduced due to tire saturation are not captured by the linear tire model. In these regimes, the Fiala and Pacejka tire models are seen to perform better.

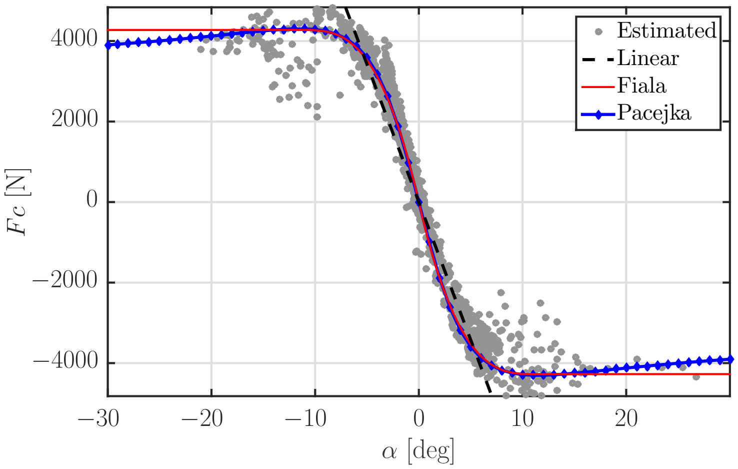


Figure 3.3: Variation of lateral tire force with slip angle using different tire modeling approaches. The gray dots indicate the values estimated during a test on a winding road and a 'figure 8' drift maneuver. The vehicle speed ranged from 10 to 35m/s.

Summary

In this dissertation, we use the linear tire model for applications such as highway driving (Chapter 8) where the vehicle operates at relatively low slip angles. The Fiala and Pacejka models are used in applications involving road surfaces such as snow (Chapter 5 and 6) where the tires are operating close to their limits.

3.2.5 Road-aligned coordinate frame

A simple transformation allows us to express the position and orientation of the vehicle in a coordinate frame aligned with the road or lane centerline [101]. The vector of position and orientation coordinates at time t is given by:

$$[\xi_t, \eta_t, \phi_t]^T, \tag{3.21}$$

where ξ is the longitudinal position of the vehicle along the road, and η and ϕ denote the lateral position error and angular error with respect to the lane centerline, respectively. The notation is depicted in Figure 3.4.

The differential equations corresponding to the position and orientation in (3.10) are modified as:

$$\dot{\xi} = \frac{1}{1 - \kappa^r(\xi)\eta} \left(\dot{x}\cos\phi - \dot{y}\sin\phi\right), \tag{3.22a}$$

$$\dot{\eta} = \dot{x}\sin\phi + \dot{y}\cos\phi, \tag{3.22b}$$

$$\dot{\phi} = \dot{\psi} - \kappa^r(\xi)\dot{\xi}, \tag{3.22c}$$

where the function $\kappa^r(\xi)$ provides the road curvature at the longitudinal position ξ.

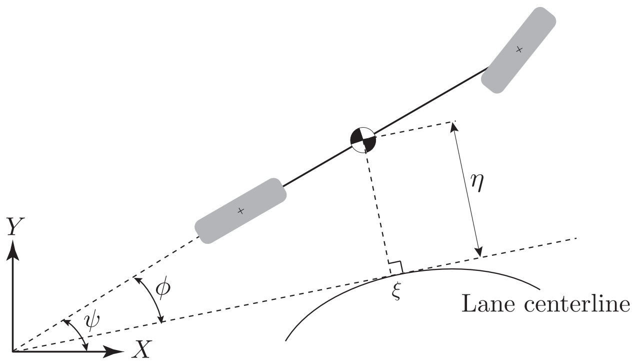


Figure 3.4: Position $[\xi, \eta]^T$ and orientation ϕ in the road-aligned frame.

Remark 3.8. *The mapping $\kappa^r(\cdot)$ can be obtained from an on-board sensor such as a forward looking camera and is most commonly parameterized as a cubic polynomial:*

$$\kappa^r(\xi) = \sum_{j=0}^{3} c_j \xi^j. \tag{3.23}$$

The above transformation to a road-aligned frame is especially useful in lane-keeping and obstacle avoidance applications for several reasons. Firstly, the goal of tracking the lane centerline can be achieved by setting the reference position to $\eta^{\text{ref}} = 0$. Secondly, the safety constraints enforced by the lane boundaries and obstacles can be expressed as simple lower and upper bounds on the state η (see e.g. [55],[56]). Moreover, under the assumptions of constant speed, low road curvature and a linear tire model, the resulting dynamic bicycle model in the road-aligned frame (3.22) is linear in the states and inputs[101]. As a consequence of the aforementioned properties, the resulting MPC optimization problem is a Quadratic Program (QP) which can be solved efficiently in real-time.

Replacing the coordinates $[X_t, Y_t, \psi_t]^T$ with $[\xi_t, \eta_t, \phi_t]^T$ in the nonlinear vehicle model (3.3) and the dynamic bicycle model (3.10) yields the modified state vector $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, \xi_t, \eta_t, \phi_t]^T$. The curvature mapping $\kappa_t^r(\cdot)$ is added to the parameter vector p_t^v in (3.2).

3.3 Kinematic Model

The main issue with dynamic models is that they are undefined at low speeds. Specifically, the lateral tire forces in (3.17) are a function of the slip angles which become singular at low speeds due to the velocity term in the denominator (see Definitions 3.2 and 3.5). Kinematic models preclude this issue and have been shown to be effective for autonomous vehicle control design [69]. The derivation of the kinematic model is based on the following assumption:

Assumption 3.9. *The slip angle at each wheel is zero. That is, the velocity vector at the wheel is aligned with the wheel.*

While the above assumption is valid at low speeds, in practice, it is a reasonable assumption even at higher speeds for applications such as highway driving where the slip angles are small.

3.3.1 Kinematic bicycle model

The notation used in the kinematic bicycle model is shown in Figure 3.2. The following set of differential equations describe the motion of the vehicle in the inertial frame:

$$\dot{X} = v\cos(\psi + \alpha), \tag{3.24a}$$

$$\dot{Y} = v\sin(\psi + \alpha), \tag{3.24b}$$

$$\dot{\psi} = \frac{v}{l_r}\sin\alpha, \tag{3.24c}$$

$$\dot{v} = a, \tag{3.24d}$$

$$\alpha = \arctan\left(\frac{l_r}{l_f + l_r}\tan\delta_f\right), \tag{3.24e}$$

where v and a denote the velocity and acceleration of the CoG, respectively, and α denotes the angle between the velocity vector and the longitudinal axis, referred to as the body slip angle. Whereas the slip angles are functions of the vehicle states and steering angle in the case of the dynamic models presented in Section 3.2, in the kinematic bicycle model, the body slip angle α is only a function of the vehicle geometry and steering angle.

The discrete-time model (3.1) is obtained from (3.24) by the first-order Euler method. The state and input vectors for the kinematic bicycle model are defined as $x_t^v = [X_t, Y_t, \psi_t, v_t]^T$ and $u_t^v = [\delta_{f_t}, a_t]^T$, respectively.

3.3.2 Road-aligned coordinate frame

As in the case of the dynamic bicycle model, the position and orientation of the vehicle can be expressed in a curvilinear coordinate frame aligned to the road (see Figure 3.4). The differential equations corresponding to the position and orientation in (3.24) are modified as:

$$\dot{\xi} = \frac{1}{1 - \kappa^r(\xi)\eta} v\cos(\phi + \alpha), \tag{3.25a}$$

$$\dot{\eta} = v\sin(\phi + \alpha), \tag{3.25b}$$

$$\dot{\phi} = \frac{v}{l_r}\sin\alpha - \kappa^r(\xi)\dot{\xi}. \tag{3.25c}$$

Replacing the coordinates $[X_t, Y_t, \psi_t]^T$ with $[\xi_t, \eta_t, \phi_t]^T$ in the kinematic bicycle model (3.24) yields the modified state vector $x_t^v = [\xi_t, \eta_t, \phi_t, v_t]^T$. The curvature mapping $\kappa_t^r(\cdot)$ is added to the parameter vector p_t^v in (3.2).

3.3.3 Comparison with the dynamic bicycle model

We use data collected from our experimental vehicle in order to compare the accuracy of the kinematic bicycle model presented above and the dynamic bicycle model presented in Section 3.2.2. A detailed analysis can be found in [28, 68, 69]. For a given model, the accuracy is quantified by the *multi-step open-loop prediction errors*, which are defined as follows:

Definition 3.10. *The k-step open-loop prediction $x^v_{k|t}$ of the state vector at time t is recursively defined as:*

$$x^v_{k|t} = f^v(x^v_{k-1|t}, u^v_{t+k-1}, \bar{d}^v_{k-1|t}), \qquad \forall k = 1, 2, \dots \tag{3.26}$$

initialized with:

$$x^v_{0|t} = x^v_t. \tag{3.27}$$

The k-step open-loop prediction error $\tilde{x}^v_{k|t}$ is defined as:

$$\tilde{x}^v_{k|t} = \|x^v_{t+k} - x^v_{k|t}\|_2, \tag{3.28}$$

where $\|z\|_2 = \sqrt{z^T z}$. By definition, $\tilde{x}^v_{0|t} = 0$.

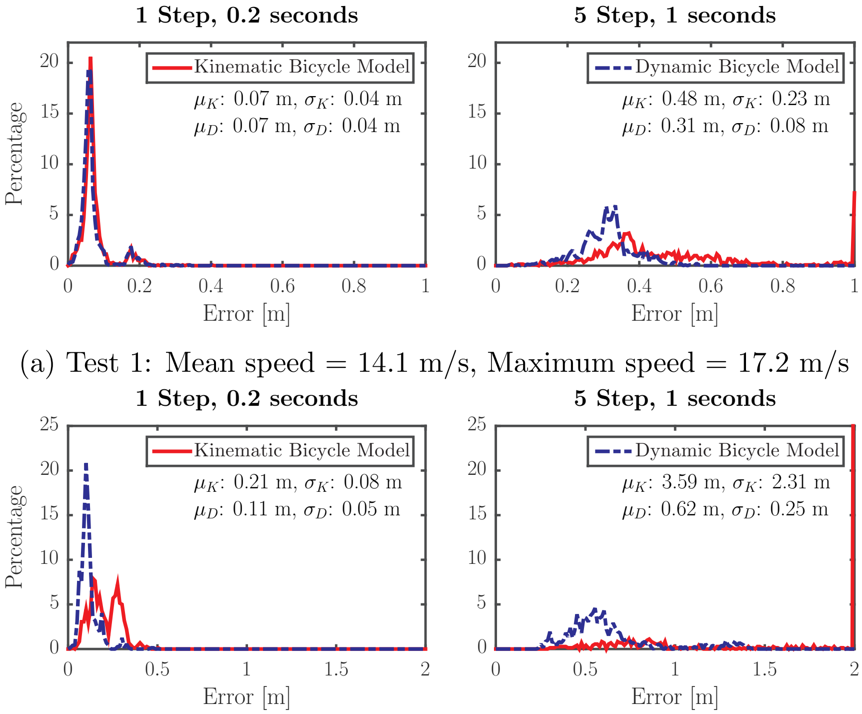
In particular, note the use of the actual (measured) input u^v_{t+k-1} from the dataset and a nominal value of the disturbance $\bar{d}^v_{k-1|t}$ (usually equal to zero) in (3.26).

Figure 3.5 shows the percentage distributions of the 1 and 5 step open-loop prediction errors of the inertial position for the kinematic and dynamic bicycle models with a discretization time of 0.2 s. The latter uses the linear tire model (3.20) to compute the lateral tire forces. The data was collected at our test facility on a winding track modeled after a racing track. During Test 1 which involved lower speeds (Figure 3.5a), it is seen that both models perform similarly well in predicting the states of the vehicle, although the mean and variance for the dynamic model are lower in general. However, at higher speeds (Test 2, Figure 3.5b), the prediction errors of the dynamic model are significantly lower than those of the kinematic model.

A benefit of kinematic models is that they only depend on the geometric parameters of the vehicle. On the other hand, dynamic models include additional parameters such as mass, rotational inertia, tire stiffness coefficients and the road friction coefficient. These parameters vary with operating conditions and require online non-linear parameter identification techniques [64].

Dynamic models are able to capture the vehicle dynamics while driving at the limits of handling [61]. Emergency collision avoidance maneuvers on low-friction surfaces such as snow in [29] show that the tire slip angles during such maneuvers are close to, and often exceed, their saturation limits.

To summarize, the choice of vehicle model is highly dependent on the application under consideration. In our implementation, we use a hybrid approach where the model used in the predictive controller is switched based on the current operating point. Specifically, we use the kinematic bicycle model for speeds lower than a given threshold to deal with stop-and-go situations and the dynamic bicycle model for higher speeds.



(a) Test 1: Mean speed = 14.1 m/s, Maximum speed = 17.2 m/s

(b) Test 2: Mean speed = 24.7 m/s, Maximum speed = 37.8 m/s

Figure 3.5: Percentage distributions of the 1 and 5-step open-loop position errors of the kinematic and dynamic bicycle models at time steps of 0.2 s. The mean and standard deviation of these errors are denoted by μ and σ, respectively. The subscripts K and D refer to the kinematic model and the dynamic model, respectively.

3.4 Longitudinal Point-Mass Model

In general, point-mass models are not useful for the design of lateral controllers due to their large modeling errors [48]. However, such models are sufficient for longitudinal control applications such as autonomous cruise control (Chapter 7). In our work, we model the longitudinal motion as a *double integrator system* described by the following equations:

$$\dot{\xi} = v, \tag{3.29a}$$
$$\dot{v} = a. \tag{3.29b}$$

The discrete-time model (3.1), obtained by the zero-order hold method, can be written as:

$$\xi_{t+1} = \xi_t + \Delta t^v v_t + \frac{1}{2}\Delta t^{v^2} a_t, \tag{3.30a}$$
$$v_{t+1} = v_t + \Delta t^v a_t. \tag{3.30b}$$

The state and input vectors are $x_t^v = [\xi_t, v_t]^T$ and $u_t^v = a_t$, respectively.

3.5 Actuator Models

The vehicle models presented above assume that the inputs δ_f and a_x (or a) can be controlled directly. In practice, however, low-level controllers are used to transform the aforementioned commands into physical control signals. For example, in the case of the prototype Hyundai vehicle described in Appendix A.2, steering angle control is achieved by means of controlling the torque reference sent to the Electronic Power Steering (EPS) system. Similarly, the vehicle's longitudinal motion is controlled by means of a desired acceleration command to the cruise-control system installed on the vehicle. These low-level controllers introduce delays in the overall control loop. Including the dynamics of these actuators in the vehicle model allows us to better account for the effect of the control inputs on the vehicle states. Moreover, directly optimizing on physical inputs in the MPC problem mitigates issues due to tracking errors of the low-level control systems. We present models of the acceleration and steering actuators in Sections 3.5.1 and 3.5.2, respectively.

Remark 3.11. *The dynamics of the actuators that control the slip ratios $\sigma_{\star\bullet}$ and the yaw moment M_z are not considered here. It is assumed that they have relatively fast dynamics. Reasonable tracking errors can be obtained by imposing suitable constraints on the rate of change of inputs in the MPC problem.*

3.5.1 Acceleration lag model

In the case of the longitudinal acceleration, a synthetic input a_x^{des} introduced which is mapped to the actual longitudinal acceleration of the car by a first-order delay:

$$\dot{a}_x = -\frac{1}{\tau_a} a_x + a_x^{\text{des}}, \tag{3.31}$$

where τ_a is the actuator time constant. As mentioned earlier, the input a_x^{des} is the reference command sent to the vehicle's cruise control system. When (3.31) is added to the vehicle dynamics (3.1), the state vector x^v is augmented with the longitudinal acceleration a_x and the input vector u^v consists of a_x^{des} instead of a_x.

3.5.2 Steering torque model

Steering angle control is achieved by means of the vehicle's EPS system where the physical input is the motor torque command τ_s. The steering system including the EPS motor, steering column, rack-and-pinion and linkages is modeled by a lumped inertia wherein the second-order equations of motion are written as:

$$\ddot{\delta}_f = -\frac{\beta_s}{J_s}\dot{\delta}_f - \frac{K_s}{J_s}\delta_f + \frac{1}{J_s}\tau_s, \tag{3.32}$$

where J_s, β_s, K_s denote the equivalent rotational inertia, damping and stiffness coefficients, respectively. In this case, when (3.32) is added to the vehicle dynamics (3.1), the state vector x^v is augmented with $[\delta_f, \dot{\delta}_f]^T$ and the input vector u^v consists of the steering torque τ_s instead of δ_f.

Angle vs. torque control

The choice of the steering angle δ_f or the torque τ_s as the control input is largely dictated by actuator setup on the experimental vehicle used. For example, in the case of the prototype Jaguar vehicle described in Appendix A.1, angle control can be achieved directly as the vehicle is equipped with an Active Front Steering (AFS) system. On the other hand, the prototype Hyundai vehicle described in Appendix A.2 consists of a Motor Driven Power Steering (MDPS) system which takes an overlay torque command as the input. In this case, we can either directly optimize on the torque τ_s or optimize on the angle δ_f and use a lower level feedback controller to generate the torque command.

3.5.3 Parameter identification

For the purpose of controller design, the actuator models (3.31) and (3.32) are linear approximations of the underlying nonlinear system dynamics. This is because the systems themselves consist of low-level controllers and elements such as lookup tables which make them hard to model analytically. To account for the model mismatch, online identification techniques can be used to update the parameters in these models (see [42] for details related to the steering model identification).

3.5.4 Input constraints

Physical limits on the actuators and comfort requirements impose bounds on the control input and its rate of change:

$$u^v_{\min} \leq u^v_t \leq u^v_{\max}, \tag{3.33a}$$

$$\Delta t^v \cdot \dot{u}^v_{\min} \leq u^v_t - u^v_{t-1} \leq \Delta t^v \cdot \dot{u}^v_{\max}. \tag{3.33b}$$

The values of the bounds $u^v_{\min}$, $u^v_{\max}$, $\dot{u}^v_{\min}$ and $\dot{u}^v_{\max}$ are detailed in Appendix A for our experimental vehicles. The constraints (3.33) are concisely written (as in (2.6g)) as:

$$[u^v_{k|t}, u^v_{k-1|t}] \in \mathcal{U}, \tag{3.34}$$

where $\mathcal{U}$ is a polytope defined by the inequalities in (3.33).

3.6 Uncertainty Characterization

The models presented above do not include any description of the uncertainty d_t^v appearing in the vehicle model (3.1). This section describes the various sources of uncertainty and a mathematical characterization.

3.6.1 Measurement noise

Noisy sensory measurements and partial state feedback lead to uncertainty in the current state of the vehicle x_t^v. We use state observers to estimate the state x_t^v conditioned on the history of measurements up to time t. Approaches for nonlinear state estimation such as the Extended and Unscented Kalman Filter (see [115] for details) model x_t^v as a multivariate Gaussian random variable:

$$x_t^v \sim \mathcal{N}(\hat{x}_t^v, \Sigma_t^v). \tag{3.35}$$

The parameters $\hat{x}_t^v$ and Σ_t^v are recursively updated at each time step based on the measurements received.

3.6.2 Model mismatch

As the vehicle model (3.1) is a mathematical representation of the underlying system dynamics, it is necessary to account for the error in the state-update equations. Model mismatch also occurs from the discretization of the continuous time dynamics. In our work, we use an *additive disturbance* formulation to account for modeling errors:

$$x_{t+1}^v = f^v(x_t^v, u_t^v) + D^v d_t^v, \tag{3.36}$$

where D^v denotes the disturbance-to-state transition matrix, usually chosen to be the identity matrix.

In robust control approaches such as that in Chapter 6, the disturbance d_t^v is constrained to lie in a polyhedral set:

$$d_t^v \in \mathcal{D}_t^v. \tag{3.37}$$

In addition, $\mathcal{D}_t^v$ is allowed to be a set-valued mapping of the state x_t^v or input u_t^v. Details on the construction of the set $\mathcal{D}_t^v$ can be found in Chapter 6 for the lateral stability control application.

In probabilistic control approaches such as that in Chapter 8, we use a Gaussian representation of the disturbance:

$$d_t^v \sim \mathcal{N}(\hat{d}_t^v, \Sigma_t^d). \tag{3.38}$$

To motivate the choice of a Gaussian, we use data collected from our experimental vehicle to quantify the predictions errors in the vehicle model. As an example, Figure (3.6) shows the percentage distributions of the *one-step prediction error* of the lateral dynamical states $\dot{y}$ and $\dot{\psi}$ and the lane-relative position η and orientation ϕ computed using the dynamic bicycle model in 3.2.2. The one-step prediction error $\tilde{x}^v_{1|t}$ at time t is defined as:

$$\tilde{x}^v_{1|t} = x^v_{t+1} - f^v(x^v_t, u^v_t), \tag{3.39}$$

where x^v_t and u^v_t refer to the measured values of the state and input vectors, respectively. The data was collected while driving on the highways around Berkeley. The distributions of the prediction errors are seen to exhibit unimodal behavior, thus justifying the Gaussian representation.

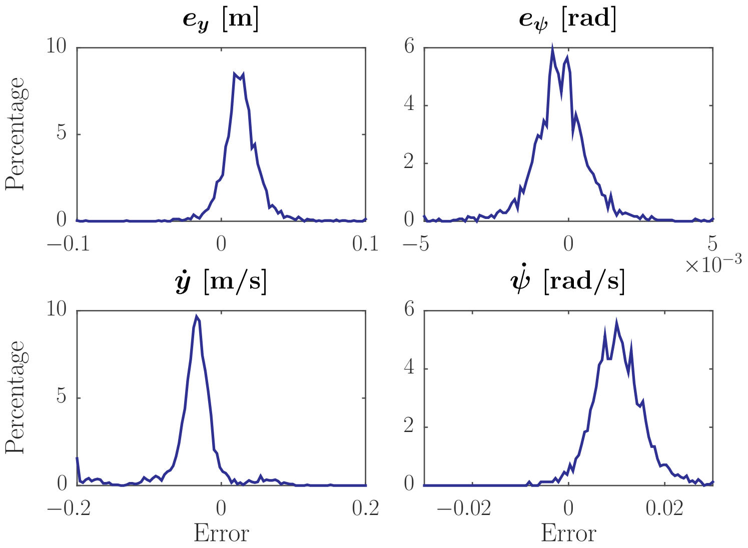


Figure 3.6: Percentage distributions of the one-step prediction error for the dynamic bicycle model with a linear tire model. The model discretization time is 0.1 s. The data was collected on the highways around Berkeley.

3.6.3 Imperfect parameter knowledge

As noted in Section 3.2.4, tire model parameters such as the stiffness coefficients are estimated from experimental data. Similarly, while physical parameters such as the mass, rotational inertia and vehicle dimensions can be measured, parameter identification techniques are used to account for their variation over time and to reduce modeling errors [64]. Statistical techniques for parameter identification typically yield a point estimated and an associated variance. Thus, a Gaussian representation of the uncertainty is a natural choice in this case. In fact, when parametric uncertainty is considered in the control design problem, we augment the state vector x^v_t with the uncertain parameters. Formally, let $x^{v,v}_t$ and $x^{v,p}_t$ denote the vehicle state vector (constituting, for example, the velocity, position, orientation) and the vector of uncertain parameters, respectively. The modified state vector x_t^v is defined as:

$$x_t^v = [x_t^{v,v}, x_t^{v,p}]^T. \tag{3.40}$$

The parameters in $x_t^{v,p}$ are assumed to evolve according to a *constant parameter model.* Concretely, the modified discrete-time state update equations are given by:

$$x_{t+1}^v = f^v(x_t^v, u_t^v, d_t^v), \tag{3.41}$$

with:

$$f^v(x_t^v, u_t^v, d_t^v) = \begin{bmatrix} x_{t+1}^{v,v} \\ x_{t+1}^{v,p} \end{bmatrix} = \begin{bmatrix} f^{v,v}(x_t^v, u_t^v, d_t^{v,v}) \\ x_t^{v,p} + d_t^{v,p} \end{bmatrix}. \tag{3.42}$$

Note the coupling introduced via the dynamics model $f^{v,v}(\cdot)$. The variables $d_t^{v,v}$ and $d_t^{v,p}$ denote the disturbances related to the discrete-time updates of $x_t^{v,v}$ and $x_t^{v,p}$, respectively, and together constitute the disturbance vector d_t^v. The Gaussian uncertainty models (3.35) and (3.36) apply. With this formulation, the control design in the presence of parametric uncertainty can be handled in a similar manner as that with model mismatch. This is discussed later in Chapter 8.

3.7 Summary

In this chapter, we discussed the vehicle models used in the various applications in this dissertation. The model choice is dictated by the expected operating conditions of the vehicle and by the sensors and actuators present on the vehicle. For example, the nonlinear Fiala tire model is employed for the experiments performed on snow in Chapter 5 as the slippery surface necessitates consideration of the nonlinear tire effects. Moreover, the four-wheel model is used as the experimental vehicle enables the actuation of the individual braking torques at the four wheels. Finally, a concrete formulation of the disturbance models (related to the vehicle model) used in the control design in this dissertation is presented.

Chapter 4

Environment Models for Automated Driving

4.1 Introduction

Autonomous vehicles navigating in the presence of other objects such as cars and pedestrians, must anticipate their future behavior in order to plan safe trajectories. A challenge with predicting the motion of target vehicles is that unlike the ego vehicle, useful indicators of intent such as the steering wheel angle cannot be measured. Instead, one must rely on measurements such as relative positions and velocities of target vehicles from sensor onboard the ego vehicle.

An overview of existing approaches for target vehicle motion modeling can be found in [76]. *Physics-based models* extrapolate the vehicle's movement based on a simplified model of the vehicle dynamics with some assumptions on the driver's inputs. These are computationally cheap to implement but only suitable for short prediction horizons. Longer-term predictions require *maneuver-based models*, which exploit the structure of the driving problem to represent the various maneuvers of drivers by motion primitives [10, 114]. Trajectory prediction is performed by comparing the trajectory executed so far with the set of motion primitives. An alternate approach is to first estimate the intent of the target vehicle (e.g. keep or change lanes), and generate a trajectory which is consistent with the intent [107, 114]. *Interaction-aware models* yield an even higher level of realism as they base their predictions on an understanding of the interactions between vehicles. Most methods use Dynamic Bayesian Networks to model dependencies between vehicles in traffic scenarios [52]. These models perform better for long-term predictions as compared to physics and maneuver based models at the cost of computational complexity.

From the perspective of this dissertation, a more relevant classification is that based on the nature of the forecasts in terms of the associated uncertainty. Specifically, within each of the above categories, environment models can be further classified into *(i)* deterministic, *(ii)* set-based, *(iii)* stochastic, and *(iv)* scenario-based models. Deterministic models provide a single predicted trajectory for each object in the scene. While this simplifies the control design process, deterministic models cannot capture the uncertainty associated with the future actions of human drivers, especially over longer time periods.

Set-based predictions over-approximate the future occupancy of target vehicles, given a prediction model and bounds on the drivers' actions [6]. While this allows for a formal verification of safety [5], we show in Chapter 8 that such a worst-case approach leads to an extremely conservative control policy. It is, however, suitable for applications such as autonomous cruise control as shown in Chapter 7.

Stochastic models use standard probability distribution functions to model the driver's behavior [35]. A deterministic trajectory can easily be obtained as the most likely outcome of the stochastic model. Gaussian Mixture Models (GMMs) are typically used to characterize the distributions over the future positions or trajectories of target vehicles [66]. In general, it is difficult to include traffic rules and the interaction between vehicles in such models because it increases their complexity.

Scenario-based models can overcome these limitations by not stating the probability distributions explicitly [4, 9]. Instead, the uncertainty is described via a finite number of samples, each representing one possible future outcome. This facilitates the modeling of vehicle interactions, by excluding traffic scenarios that are unrealistic, e.g. if they cause a collision.

The choice of a deterministic, set-based or probabilistic environment model is dictated by the strategy used for the control design. For example, a deterministic approach is used for nominal MPC problem in Chapter 5, a worst-case approach is used for the design of the robust controller in Chapter 7, and probabilistic models are used for the stochastic MPC problems in Chapter 8. This chapter presents details of the environment models used in the various applications in this dissertation. In particular:

1. A probabilistic multiple model filtering approach is proposed in Section 4.2 to estimate the lane changing intent of target vehicles and the associated lane change trajectory.

2. Section 4.3 presents a learning-based method to estimate the lane change probability of a target vehicle given the measurements of its lateral position.

3. Preliminary work in order to account for the reaction of target vehicles in adjacent lanes to the ego vehicle's lane change intention is discussed in Section 4.4.

4. Section 4.5 discusses two approaches for formulating collision avoidance constraints on the state of the ego vehicle based on the predictions of the environment models.

The evolution of the environment over time is described by the following discrete-time state-space representation:

$$x^e_{k+1} = f^e(x^v_k, x^e_k, d^e_k), \tag{4.1}$$

where the disturbance d^e_k is introduced to model the uncertainty in the environment dynamics. Typically, the state vector x^e_k contains the position, orientation and velocity of each target vehicle in the scene and includes a representation of the uncertainty therein. The control inputs to the target vehicles are implicitly modeled by making some assumptions or as possibly non-linear functions of the system state. Hence, they do not explicitly appear in (4.1).

In the remainder of this dissertation, the acronyms EV and TV stand for Ego Vehicle and Target Vehicle, respectively.

4.2 Multiple Model Approach

In this section, we propose a probabilistic model based on the framework of multiple model filtering for the prediction of the lateral and longitudinal motion of target vehicles. The method combines physics and maneuver-based models. The main idea is to model each target vehicle as a *hybrid dynamical system*, that is, a system consisting of both continuous and discrete states. This is a natural choice for traffic modeling [76]. Specifically, the discrete states (also referred to as *modes*) correspond to maneuvers such as lane keeping and lane changing. Within each of the modes, the continuous dynamics describe the motion of the vehicle corresponding to that mode. The problem of estimating the mode of the vehicle and predicting its future positions can then be posed as a state estimation problem for the hybrid system. In this work, we use the framework of multiple model filtering which has been successfully used in target tracking applications [87].

4.2.1 Modeling

The dynamics of the j^{th} TV are described by a Markov jump affine system:

$$x^e_{j_{t+1}} = A^{(i)}_t x^e_{j_t} + D^{(i)}_t d^{e(i)}_t + E^{(i)}_t, \tag{4.2a}$$

$$y^e_{j_t} = H^{(i)}_t x^e_{j_t} + w^{e(i)}_t, \tag{4.2b}$$

where $x^e_{j_t}$ and $y^e_{j_t}$ denote the TV state and measurement vectors, respectively, at time t. The process noise $d^{e(i)}_t$ and measurement noise $w^{e(i)}_t$ are assumed to be independent and identically distributed (i.i.d.) as $\mathcal{N}(0, Q^{(i)})$ and $\mathcal{N}(0, R^{(i)})$, respectively. The superscript (i) in (4.2) refers to the model $m^{(i)}$ in the model set $\mathcal{M} = \{m^{(1)}, m^{(2)}, \ldots, m^{(M)}\},$. Transitions between modes have fixed probabilities given by a matrix $\pi \in \mathbb{R}^{M \times M}$, such that:

$$\pi_{ij} = P(m_{t+1} = m^{(j)} | m_t = m^{(i)}), \tag{4.3}$$

where m_t denotes the mode at time t. For the prediction problem, we decouple the longitudinal and lateral motion of the TV. The models used for each are described below. The subscript j and superscript e are dropped for visual clarity.

4.2.2 Longitudinal models

We choose $x = [\xi, \dot{\xi}, \ddot{\xi}]^T$ and $y = [\xi, \dot{\xi}]^T$ for the longitudinal state and measurement vectors, respectively. As we do not model traffic interactions using the multiple model framework, two modes based on the common assumptions of constant velocity and acceleration are used to model the longitudinal motion as described below.

Constant velocity

$$A_t^{(1)} = \begin{bmatrix} 1 & \Delta t^e & \frac{\Delta t^{e^2}}{2} \\ 0 & 1 & \Delta t^e \\ 0 & 0 & 0 \end{bmatrix}, \quad D_t^{(1)} = \begin{bmatrix} \frac{\Delta t^{e^2}}{2} \\ \Delta t^e \\ 1 \end{bmatrix}, \quad E_t^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.4}$$

where Δt^e is the sampling time.

Constant acceleration

$$A_t^{(2)} = \begin{bmatrix} 1 & \Delta t^e & \frac{\Delta t^{e^2}}{2} \\ 0 & 1 & \Delta t^e \\ 0 & 0 & 1 \end{bmatrix}, \quad D_t^{(2)} = \begin{bmatrix} \frac{\Delta t^{e^2}}{2} \\ \Delta t^e \\ 1 \end{bmatrix}, \quad E_t^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.5}$$

The measurement matrix in both the above cases is given by:

$$H_t^{(1)} = H_t^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.6}$$

4.2.3 Lateral models

We use parametric state-feedback models to represent the lateral dynamics in typical lane keeping and lane changing maneuvers. It is assumed that the TV changes one lane at a time and that it prefers staying close to the lane center. The TV lateral state and measurement are chosen to be $x = [\eta, \dot{\eta}]^T$ and $y = \eta$, respectively. The lateral acceleration of the TV with respect to the lane is assumed to evolve as:

$$\ddot{\eta}_t \approx \frac{\dot{\eta}_{t+1} - \dot{\eta}_t}{\Delta t^e} = -K_2(\eta_t - \eta^{\text{ref}}) - K_1\dot{\eta}_t. \tag{4.7}$$

Thus, η_t is assumed to have a second-order response. The gains K_1 and K_2 affect the nature of this response and $\eta_t \to \eta^{\text{ref}}$ as $t \to \infty$ if the gains are chosen such that the system (4.7) is stable. The corresponding system matrices are given by:

$$A_t^{(i)} = \begin{bmatrix} 1 & \Delta t^e \\ -\Delta t^e K_2^{(i)} & 1 - \Delta t^e K_1^{(i)} \end{bmatrix}, \qquad D_t^{(i)} = \begin{bmatrix} \Delta t^{e^2}/2 \\ \Delta t^e \end{bmatrix}, \tag{4.8a}$$

$$E_t^{(i)} = \begin{bmatrix} 0 \\ \Delta t^e K_2^{(i)} \eta^{\mathrm{ref}^{(i)}} \end{bmatrix}, \qquad H_t^{(i)} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{4.8b}$$

where the superscript (i) denotes the mode. The main idea of the proposed approach is to capture the variation in the manner in which human drivers change lanes by means of a finite number of representative gains $\{K_1^{(i)}, K_2^{(i)}\}_{i=1}^M$. In general, the higher the values of the gains, the more aggressive is the driving style. Our goal is to identify these values from real data. This is presented in the next section.

Remark 4.1. *The choice of $x^e = [\eta, \dot{\eta}]$ and $y^e = \eta$ is based on the assumption that only the lateral position of the TV with respect to its lane can be estimated. This is done by fusing the lane information from the vision system or a digital map with the relative position of the TV obtained from sensors on-board the EV such as the radar, camera and lidar. If additional states such as the orientation and yaw rate can be estimated, these can be augmented to the state x^e and models such as the kinematic bicycle model of Section 3.3.1 can be used instead of the point-mass model in* (4.8)*.*

4.2.4 Parameter identification

We use data collected from our experimental vehicle for the identification of the gains $\{K_1^{(i)}, K_2^{(i)}\}_{i=1}^M$ used to model the lateral motion of TVs. The number of modes M is treated as a tuning parameter.

The identification is performed as follows. We first extract lane change trajectories from multiple datasets recorded using our vehicle, which are shown in Figure 4.1. Left and right lane changes are treated identically due to their symmetry with respect to the lane center. Without loss of generality, we flip right lane change trajectories with respect to the lane center, hence, only left lane changes are seen in Figure 4.1. Note that the time duration

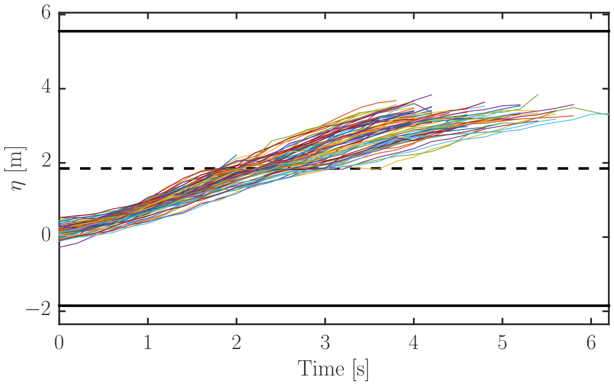


Figure 4.1: Lane change trajectories from 77 recorded datasets. Right lane change trajectories are flipped with respect to the lane center.

of each trajectory is allowed to be different. For each trajectory, a nonlinear least squares regression is performed to identify values of K_1 and K_2 such that the prediction error of the model over the duration of the lane change (4.8) is minimized. Concretely, the following nonlinear optimization problem is solved for a given trajectory:

$$\min_{K_1,K_2} \sum_{t=1}^{T} (y_t - \hat{y}_t)^2, \tag{4.9a}$$

$$\text{subject to: } \hat{x}_{t+1} = A_t \hat{x}_t + E_t, \tag{4.9b}$$

$$\hat{y}_{t+1} = H_t \hat{x}_{t+1}, \tag{4.9c}$$

$$(t = 0, \ldots, T-1)$$

$$\hat{x}_0 = x_0, \tag{4.9d}$$

where T is the trajectory length. The cost function (4.9a) to be minimized is the sum of squares of the difference between the measurement y_t and its predicted value $\hat{y}_t$ given by the model (4.9b)–(4.9c). The system matrices F_t, E_t and H_t in (4.9b)–(4.9c) are given by (4.8) and depend on the optimization variables K_1 and K_2. The value of η^{ref} (which appears in E_t) is equal to the lane width estimated by the vision system. For the purpose of identification, the full initial state $x_0 = [\eta_0, \dot{\eta}_0]^T$ is assumed to be measured in (4.9d).

The above procedure yields a set of gains $\{K_1^{(i)}, K_2^{(i)}\}_{i=1}^{N_{\text{traj}}}$, one per trajectory ($N_{\text{traj}}$ is total number of trajectories). Next, we use K-means clustering to identify representative values of the gains, where the number of clusters is equal to the desired number of modes M. This yields the set $\{K_1^{(i)}, K_2^{(i)}\}_{i=1}^{M}$.

The results of the gain identification and clustering are shown in Figure 4.2 for the case of $M = 5$ using the 77 lane change trajectories shown in Figure 4.1. We observe a wide range of behaviors as seen by the large range of values of K_1 and K_2. The identified clusters are depicted by the different colors and marker types in Figure 4.2. The lane change profiles corresponding to the 5 modes are shown in Figure 4.3.

4.2.5 Interacting Multiple Model Kalman Filter

At each time step t, our goal is to estimate the following quantities conditioned on the history of measurements $y_{1:t}$:

1. Model probability:

$$\mu_t^{(i)} = P(m_t = m^{(i)} | y_{1:t}). \qquad (i = 1, \ldots, M) \tag{4.10}$$

   This yields a distribution over the discrete modes and provides an estimate of which model in the set $\mathcal{M}$ is in effect at time t. In the case of the lateral models (Section 4.2.3), the model probability is directly linked to the TV's intent such as lane keeping or changing.

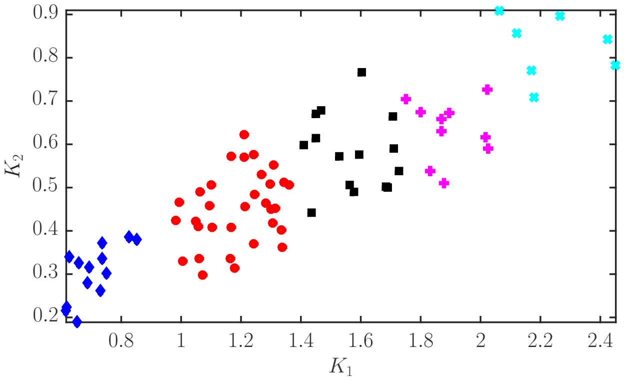


Figure 4.2: Gains K_1 and K_2 identified from 60 lane change trajectories and clustered into 5 modes depicted by the different marker types.

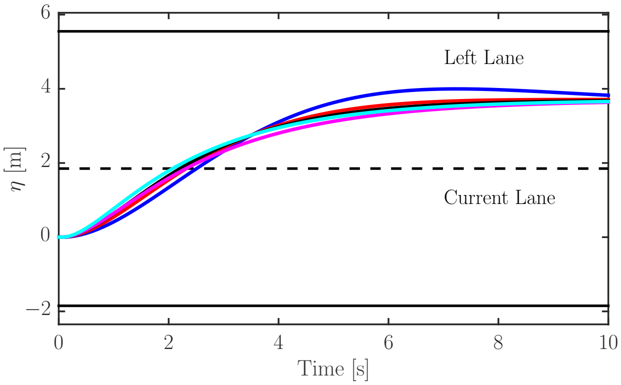


Figure 4.3: Lane change profiles corresponding to the 5 representative values of K_1 and K_2 identified from data.

2. Posterior mean and covariance:

$$\hat{x}_t = E[x_t|y_{1:t}], \qquad \Sigma_t = \text{Var}[x_t|y_{1:t}]. \tag{4.11a}$$

This is the estimated state of the vehicle at time t accounting for the distribution over the discrete modes.

In the above equations, $E[v]$ and $\text{Var}[v]$ denote the expected value and variance of v, respectively. The problem of estimating the above quantities is posed as a hybrid system state estimation problem. The Interacting Multiple Model Kalman Filter (IMM-KF) developed in [95] is a computationally efficient algorithm for this problem which involves running a bank of M Kalman filters in parallel. We use the implementation of the algorithm described in [99]. At each time step t, the following steps are performed:

Step 1: Evaluate mixing probabilities:

$$\mu_{t-1}^{j|i} = \frac{\pi_{ji}\mu_{t-1}^{(j)}}{\mu_{1|t-1}^{(i)}}, \qquad \text{where } \mu_{1|t-1}^{(i)} = \sum_{j=1}^{M} \pi_{ji}\mu_{t-1}^{(j)}. \tag{4.12}$$

Mixing estimates and covariances:

$$\bar{x}_{t-1}^{(i)} = \sum_{j=1}^{M} \hat{x}_{t-1}^{(j)}\mu_{t-1}^{j|i}, \qquad \bar{\Sigma}_{t-1}^{(i)} = \sum_{j=1}^{M} \left[\Sigma_{t-1}^{(j)} + (\bar{x}_{t-1}^{(i)} - \hat{x}_{t-1}^{(j)})(\bar{x}_{t-1}^{(i)} - \hat{x}_{t-1}^{(j)})^T\right] \mu_{k-1}^{j|i}. \tag{4.13}$$

Step 2: For each model $m^{(i)}$, run Kalman filter with inputs $(\bar{x}_{t-1}^{(i)}, \bar{\Sigma}_{t-1}^{(i)})$ using Algorithm 4.1 to obtain the following mode-conditioned estimates:

$$(\hat{x}_t^{(i)}, \Sigma_t^{(i)}, \tilde{y}_t^{(i)}, S_t^{(i)}).$$

Step 3: Update the probability of each model:

$$\mu_t^{(i)} = \frac{\mu_{1|t-1}^{(i)} \mathcal{N}(\tilde{y}_t^{(i)}; 0, S_t^{(i)})}{\sum\limits_{j=1}^{M} \mu_{1|t-1}^{(j)} \mathcal{N}(\tilde{y}_t^{(j)}; 0, S_t^{(j)})}. \tag{4.14}$$

Step 4: Compute the posterior estimates,

$$\hat{x}_t = \sum_{j=1}^{M} \hat{x}_t^{(i)} \mu_t^{(i)}, \qquad \Sigma_t = \sum_{i=1}^{M} \left[\Sigma_t^{(i)} + (\hat{x}_t - \hat{x}_t^{(i)})(\hat{x}_t - \hat{x}_t^{(i)})^T \right] \mu_t^{(i)}. \tag{4.15}$$

Algorithm 4.1 Kalman filter equations for model $m^{(i)}$

```
Input: $(\bar{x}_{t-1}^{(i)}, \bar{\Sigma}_{t-1}^{(i)})$
    Prediction step:
    $\hat{x}_{1|t-1}^{(i)} = A_{t-1}^{(i)} \bar{x}_{t-1}^{(i)} + E_{t-1}^{(i)}$
    $\Sigma_{1|t-1}^{(i)} = A_{t-1}^{(i)} \bar{\Sigma}_{t-1}^{(i)} A_{t-1}^{(i)^T} + D_{t-1}^{(i)} Q_{t-1}^{(i)} D_{t-1}^{(i)^T}$
    Update step:
    $\tilde{y}_t^{(i)} = y_t - H_t^{(i)} \hat{x}_{1|t-1}^{(i)}$
    $S_t^{(i)} = H_t^{(i)} \Sigma_{1|t-1}^{(i)} H_t^{(i)^T} + R_t^{(i)}$
    $K_t^{(i)} = \Sigma_{1|t-1}^{(i)} H_t^{(i)^T} S_t^{(i)^{-1}}$
    $\hat{x}_t^{(i)} = \hat{x}_{1|t-1}^{(i)} + K_t^{(i)} \tilde{y}_t^{(i)}$
    $\Sigma_t^{(i)} = \Sigma_{1|t-1}^{(i)} - K_t^{(i)} S_t^{(i)} K_t^{(i)^T}$
Output: $(\hat{x}_t^{(i)}, \Sigma_t^{(i)}, \tilde{y}_t^{(i)}, S_t^{(i)})$
```

4.2.6 Target State Prediction

The main goal of the environment model is to forecast the state of the TV and the associated uncertainty over the prediction horizon of the controller. The method used to propagate the uncertainty of the TV over the horizon depends on the probabilistic control strategy employed. In particular, for the analytical stochastic MPC approach in Section 8.2, a Gaussian representation of the uncertainty is desired, while for the sampling-based and hybrid SMPC approaches in Sections 8.3 and 8.4, respectively, samples of the predicted states over the horizon are required.

Gaussian (unimodal) predictions

In order to obtain a Gaussian estimate of the future uncertainty, we base the predictions on the *most likely model* m_t^{ML}, defined as the model $m^{(i)}$ with the highest model probability $\mu_t^{(i)}$. Concretely:

$$m_t^{ML} = \arg \max_{m^{(i)} \in \mathcal{M}} \mu_t^{(i)}. \tag{4.16}$$

The predictions of the mean $\hat{x}_{1:T|t}$ and variance $\Sigma_{1:T|t}$ of the state of the TV are then given by the dynamics in (4.2a) corresponding to m_t^{ML}. Concretely:

$$\hat{x}_{k+1|t} = A_k^{(ML)} \hat{x}_{k|t} + E_k^{(ML)}, \tag{4.17a}$$

$$\Sigma_{k+1|t} = A_k^{(ML)} \Sigma_{k|t} (A_k^{(ML)})^T + D_k^{(ML)} Q^{(ML)} (D_k^{(ML)})^T, \tag{4.17b}$$

$$(k = 0, \ldots, T-1)$$

where the superscript (ML) refers to the model m_t^{ML}. The above recursion is initialized with $\hat{x}_{0|t} = \hat{x}_t$, $\Sigma_{0|t} = \Sigma_t$, which are obtained from the IMM-KF.

Sampled (multimodal) predictions

In order to generate samples of the predicted TV states, we utilize the distribution over the discrete modes given by $\{\mu_t^{(i)}\}_{i=1}^M$. The process for generating a set of S sampled states $\{x_{1:T|t}^{(s)}\}_{s=1}^S$ over the prediction horizon is detailed in Algorithm 4.2 below.

Algorithm 4.2 Sampled predictions of target vehicle state

```
for s = 1 to $S$:
    Sample model $m^{(j)}$ with probability $\mu_t^{(j)}$
    Compute predicted mean $\hat{x}_{1:T|t}$ and variance $\Sigma_{1:T|t}$ consistent with model $m^{(j)}$:
        $\hat{x}_{k+1|t} = A_k^{(j)} \hat{x}_{k|t} + E_k^{(j)}$,
        $\Sigma_{k+1|t} = A_k^{(j)} \Sigma_{k|t} (A_k^{(j)})^T + D_k^{(j)} Q^{(j)} (D_k^{(j)})^T$,
            $(k = 0, \ldots, T-1)$
        initialized with $\hat{x}_{0|t} = \hat{x}_t$, $\Sigma_{0|t} = \Sigma_t$.
    At prediction step $k$, sample from $\mathcal{N}(\hat{x}_{k|t}, \Sigma_{k|t})$ to obtain sampled state $x_{k|t}^{(s)}$ $(k = 1, \ldots, T)$.
```

Note that the longitudinal and lateral state predictions are performed separately and later combined for the control design.

4.3 Lane Change Intention Estimation

This section presents a learning-based approach to infer the lane changing intent of vehicles in the neighboring lanes. Such an approach is useful in applications where *(i)* it is only necessary to determine if a target vehicle is likely to change lanes and the actual execution of the path is not relevant, or *(ii)* the lane change trajectory prediction is conditioned on the intent of the target vehicle. For example, in the Autonomous Cruise Control (ACC) application presented in Chapter 7, the intent estimator is used to identify relevant targets. The framework was proposed in [78] for the lane change detection and trajectory prediction of the ego vehicle and is applied here for the motion prediction of target vehicles.

4.3.1 Modeling and training

Based on the work of [78], we model the lane change decision making process for a given TV as a Hidden Markov Model (HMM). We define the following variables:

- $m_t \in \mathcal{M} = \{LK, LCL, LCR\}$ is the hidden mode or latent variable at time instant t (LK = lane keeping, LCL = lane change left, LCR = lane change right).
- z_t is the observed variable at time instant t. In our case, $z_t = \eta^e_t$, where η^e is the lateral position of the TV with respect to the centerline of the nearest lane. The variable η^e is estimated using a combination of the position of the TV relative to the EV obtained from the sensor fusion system and the road geometry information provided by the camera on the EV.

Remark 4.2. *Additional features such as the relative orientation of the TV with respect to its lane can be used to improve the performance of the lane change intention estimation. The choice of a single observed variable η in our work is due to limitations imposed by the current sensing systems onboard our experimental vehicle.*

The joint probability distribution of the modes $m_{0:t} = \{m_0, \ldots, m_t\}$ and the observations $z_{1:t} = \{z_1, \ldots, z_t\}$ is given by:

$$P(m_{0:t}, z_{1:t}) = P(m_0) \prod_{k=1}^{t} P(m_k | m_{k-1}) P(z_k | m_k), \tag{4.19}$$

where the *emission* probability density function $P(z_k|m_k)$ is modeled as a Gaussian distribution. The parameters which characterize the *prior* and *transition* probability mass functions $P(m_0)$ and $P(m_k|m_{k-1})$, respectively, and the means and covariances of the Gaussian emission density function are learned from data collected from our test vehicle using the Expectation-Maximization (EM) algorithm and the Bayesian Information Criterion [78].

4.3.2 Inference

During online operation, at time t, inference on the HMM gives a probability distribution over the hidden mode m_t for each TV conditioned on the history of observations $z_{1:t}$. That is, $P(m_t = i|z_{1:t})$ can be recursively computed as:

$$P(m_t = i|z_{1:t}) \propto P(z_t|m_t = i) \sum_{j\in\mathcal{M}} P(m_{t-1} = j|z_{1:t-1})P(m_t = i|m_{t-1} = j), \tag{4.20}$$

initialized with the prior distribution $P(m_0)$. In certain applications (for example, the Autonomous Cruise Control (ACC) system presented in Chapter 7), the most likely intent of the surrounding TVs is of interest. The most likely mode or intent m_t^{ML} at time t is defined as:

$$m_t^{ML} = \arg\max_{i\in\mathcal{M}} P(m_t = i|z_{1:t}). \tag{4.21}$$

The lane change intention estimator presented in this section is applied to the problem of identifying relevant target vehicles for the purpose of ACC in Chapter 7. The ability of the approach to detect lane changes earlier than an existing method is shown via experiments performed on our test vehicle. The results of the performance improvements due to the integration of the lane change estimator and controller are also discussed.

4.4 Interaction-aware Car-Following Model

The models presented above do not account for interactions between the various traffic entities. For example, the intent of TVs to change lanes is usually motivated by the presence of a slower moving vehicle in their own lane and the ability to speed up by moving to the adjacent lane. In this section, we present a simplified approach to account for the interaction between the EV and surrounding TVs. The approach is used for the experimental study in Section 8.6 for a very specific scenario described below.

The goal of the interaction-aware car-following model is to account for the reaction of a TV in the adjacent lane to the EV's intent to change lanes in front of it. Consider a scenario wherein a TV approaches the EV from behind. If the EV indicates its intent to change lanes by means of the turn signal or by a lateral movement towards the TV's lane, the TV may either respond by slowing down to allow the EV to change lanes or not respond and maintain its current behavior. In the former case, we would like the EV to recognize the compliance of the TV and change lanes in front of the TV. Our intuition is that not considering the interaction between the TV and the EV would yield a conservative control policy by the EV where it does not change lanes. However, in such a commonly observed scenario in everyday urban driving, human drivers tend to anticipate the motion of the TV and change lanes if they are confident that the TV is compliant. The aim of the proposed environment model is to:

(i) estimate the intent of the TV to yield to the EV, and,

(ii) generate forecasts of the longitudinal motion of the TV consistent with the intent.

In combination with the stochastic MPC-based controller presented in Chapter 8, we expect the EV to replicate the decision making process of a typical human driver.

4.4.1 Problem formulation

We introduce the following notation:

- $m_t^e \in \mathcal{M}^e = \{\text{Yield}(Y), \text{Do Not Yield}(N)\}$ is the behavior of the TV which cannot be observed directly. If $m_t^e = Y$, the TV yields to the EV when the latter indicates its intent to change lanes. If $m_t^e = N$, the TV does not respond to the EV and continues tracking the reference speed v^{ref} or the vehicle in its own lane.
- $x_t^e = [\xi_t^e, v_t^e]^T$ is the measured or estimated state of the TV at time t.
- $u_t^e = a_t^e$ is the acceleration input applied by the TV at time t. This is treated as unknown but an estimate $\hat{a}_t^e$ of the current acceleration is assumed to be available.
- $z_t^e = [\xi_t^r, v_t^r]^T$ is the feature vector measured by the TV. It consists of the relative position and velocity of the *primary target relative to the TV*, which is the vehicle that the TV considers as the one to follow. As we assume no communication with the TV, the feature vector z_t^e is unknown.

The acceleration input u_t^e is assumed to be given by a car-following model of the TV:

$$u_t^e = \kappa^e(x_t^e, z_t^e, v^{\text{ref}}) + d_t^e, \tag{4.22}$$

where d_t^e is an additive disturbance normally distributed as $\mathcal{N}(0, \Sigma^e)$. In our implementation, we use the Intelligent Driver Model (IDM) which was proposed in [117] for microscopic traffic simulations. The acceleration of the TV is given by:

$$u_t^e = a_{\text{max}}^e \left[1 - \left(\frac{v_t^e}{v^{\text{ref}}} \right)^{\delta} - \left(\frac{\xi_t^{r,\text{des}}}{\xi_t^r} \right)^2 \right], \tag{4.23}$$

where the desired following distance $\xi_t^{r,\text{des}}$ is given by:

$$\xi_t^{r,\text{des}} = \xi_{\text{safe}}^r + \max \left\{ 0, v_t^e T_h + \frac{v_t^e v_t^r}{2\sqrt{a_{\text{max}}^e a_{\text{brake}}^e}} \right\}. \tag{4.24}$$

In the absence of a primary target vehicle, $\xi_t^{r,\text{des}}$ is set to 0 and the last term in (4.23) drops out. This is referred to as the *free-flow* mode. The parameters that define the IDM for a particular driver are the maximum acceleration a^e_{max}, reference speed v^{ref}, acceleration exponent δ, minimum following distance ξ^r_{safe}, time headway T_h and desired deceleration a^e_{brake}. The TV is assumed to follow the longitudinal point-mass model described in Section 3.4, concisely written as:

$$x^e_{t+1} = f^e(x^e_t, u^e_t). \tag{4.25}$$

We introduce the following assumption:

Assumption 4.3. *The reference speed v^{ref}, the parameters defining $\kappa^e(\cdot)$ and the variance Σ^e of the disturbance d^e_t are assumed to be known.*

If the feature vector z^e_t could be measured, we would be able to predict the mean and variance of u^e_t and hence x^e_t, using (4.22) and the point-mass model (4.25). Instead, we assume that z^e_t cannot be observed directly, but is a known function of the hidden mode m^e_t. In particular, if $m^e_t = Y$, z^e_t consists of the relative distance and velocity of the EV (assuming the EV would like to change lanes), and if $m^e_t = N$, z^e_t consists of the relative measurements of the vehicle in front of it in its own lane.

4.4.2 Intent estimation

The problem of estimating the probability that the TV will yield to the EV given the estimate of its acceleration $\hat{a}^e_t$ is posed as a Bayesian estimation problem. Concretely:

$$P(m^e_t = Y|\hat{a}^e_t) \propto P(\hat{a}^e_t|m^e_t = Y)P(m^e_t = Y), \tag{4.26}$$

where $P(m^e_t)$ is the prior over the set $\mathcal{M}$ which assumed to be a uniform distribution, that is, $P(m^e_t = Y) = P(m^e_t = N) = 0.5$. The conditional probability $P(\hat{a}^e_t|m^e_t)$ is computed as $\mathcal{N}(\hat{a}^e_t - \kappa^e(z^e_t(m^e_t), v^{\text{ref}}); 0, \Sigma^e)$. The notation $z^e_t(m^e_t)$ indicates the dependence of the feature z^e_t on the hidden mode m^e_t, that is, z^e_t is known for a given $m^e_t \in \mathcal{M}$. Note that $P(m^e_t = N|\hat{a}^e_t) = 1 - P(m^e_t = Y|\hat{a}^e_t)$.

4.4.3 Prediction

The interaction-aware car-following model presented here is used with the SMPC approach in Chapter 8 where sampled predictions of the future state of the TV are desired. This is achieved as follows. At time t, we first sample from the distribution $P(m^e_t|\hat{a}^e_t)$ to obtain $m^{e^{(s)}}_t$ (the superscript denotes the s^{th} sample). Then, we sample the disturbance d^e_t in (4.22) and compute the sampled input $u^{e^{(s)}}_t$. This is propagated through the dynamics (4.25) to obtain the predicted sampled state $x^{e^{(s)}}_{1|t}$. A full-horizon sample $x^{e^{(s)}}_{1:T|t}$ can be obtained by iteratively running the above procedure assuming the mode is invariant, that is, $m^{e^{(s)}}_{0:T-1|t} = m^{e^{(s)}}_t$.

Remark 4.4. *The method presented here is specific to the scenario described above and is only used in Section 8.6 to demonstrate the ability of the proposed control strategy to handle multimodal traffic forecasts. In the experiment, the parameters of the model $\kappa^e(\cdot)$ are assumed to be known, but the primary target that the TV is trying to track is unknown.*

4.5 Collision Avoidance

In this section, we present two approaches for collision avoidance which are used in this dissertation. In the context of constrained predictive control, the goal is to construct constraints of the form $g(x^v_{k|t}, x^e_{k|t}) \leq 0$ on the predicted vehicle and environment states $x^v_{1:T|t}$ and $x^e_{1:T|t}$, respectively, over the prediction horizon. While the vehicle states $x^v_{1:T|t}$ are a function of the control inputs $u^v_{0:T-1|t}$, the environment states $x^e_{1:T|t}$ are computed by the forecast models described above in this chapter and are independent of the vehicle inputs $u^v_{0:T-1|t}$ (refer to the discussion in Section 2.3.3).

4.5.1 Safety corridor approach

This method for formulation collision avoidance constraints is similar to the approaches in [8, 49]. The main idea is to map static and moving obstacles in the environment into a region of the road in which the vehicle can move safely. At a given time step k of the prediction horizon, the boundaries of the safe region are a function of the predicted obstacle positions at that time instant. Hence, in general, the boundaries are a time-varying function of the longitudinal position $\xi_{k|t}$ of the EV along the road. As in [49], we define the safe region in the road-aligned coordinate frame, as this results in simple upper and lower bounds on the lateral position $\eta_{k|t}$ of the EV. Let $\eta^L_k(\xi_{k|t})$ and $\eta^U_k(\xi_{k|t})$ denote the lateral positions of the lower and upper boundaries of the safe region, respectively, with respect to the road centerline. These are illustrated in Figure 4.4 for the simple case of one obstacle. In our work, we express the boundaries as piecewise affine functions of the longitudinal position $\xi_{k|t}$ of the EV. This is seen by the lower boundary $\eta^L_k(\cdot)$ in Figure 4.4.

The collision avoidance constraints using the safety corridor approach are given by:

$$\eta^L_k(\xi_{k|t}) \leq \eta_{k|t} \leq \eta^U_k(\xi_{k|t}). \tag{4.27}$$

The dependence of the bounds on the predicted positions of the TVs is implicitly accounted for by their dependence on the predicted time step k. Note that the constraints are independent of the road curvature as they are expressed in the road-aligned coordinate frame.

In general, the constraints given by (4.27) are non–convex and non–differentiable due to the nonlinear dependence of the bounds η^L_k and η^U_k on $\xi_{k|t}$, which in turn depends on the control inputs $u^v_{k|t}$ yet to be determined. This is not an issue if general purpose nonlinear solvers (such as NPSOL [51]) are used to solve the resulting optimization problem. However, for the linearization-based approach presented in Chapter 5, a convex approximation of the

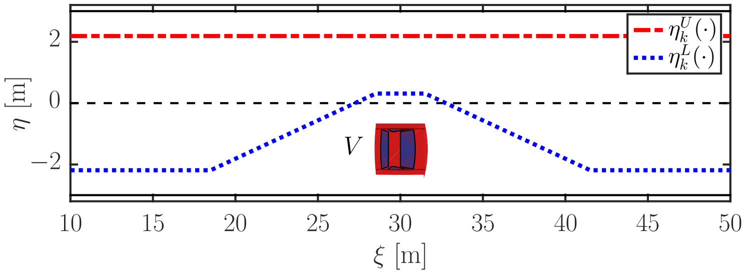


Figure 4.4: Upper and lower boundaries $\eta_k^U(\xi_{k|t})$ and $\eta_k^L(\xi_{k|t})$, respectively, for the given scenario using the safety corridor approach. In general, the boundaries are given by piecewise affine functions of the EV's longitudinal position $\xi_{k|t}$.

safety constraint is desired. The linearization of (4.27) around a nominal EV state $\bar{x}_{k|t}^v$ is given by:

$$\eta_{k|t} \leq \eta_k^U(\bar{\xi}_{k|t}) + g_k^U(\xi_{k|t} - \bar{\xi}_{k|t}), \tag{4.28a}$$

$$\eta_{k|t} \geq \eta_k^L(\bar{\xi}_{k|t}) + g_k^L(\xi_{k|t} - \bar{\xi}_{k|t}), \tag{4.28b}$$

with:

$$g_k^U = \nabla_{\xi_{k|t}} \eta_k^U(\xi_{k|t})|_{\bar{\xi}_{k|t}}, \tag{4.29a}$$

$$g_k^L = \nabla_{\xi_{k|t}} \eta_k^L(\xi_{k|t})|_{\bar{\xi}_{k|t}}. \tag{4.29b}$$

Note that $\bar{\xi}_{k|t}$ is the element of the nominal state vector $\bar{x}_{k|t}^v$ corresponding to the longitudinal position along the road. Fortunately, the chosen parameterization of the mappings $\eta_k^L(\cdot)$ and $\eta_k^U(\cdot)$ implies that their derivatives with respect to $\xi_{k|t}$ are piecewise constant. The constraint (4.28) can be rewritten in the standard form of a linear equality:

$$G_k^e(x_{k|t}^v - \bar{x}_{k|t}^v) \leq h_k^e. \tag{4.30}$$

4.5.2 Signed distance function

The safety corridor approach presented above is tailored for collision avoidance in structured applications such as highway driving. A more general collision avoidance formulation is presented in [109], which is based on the *signed distance function* between objects.

Definition 4.5 (Signed distance)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be convex shapes. The signed distance $sd(\mathcal{A}, \mathcal{B})$ is defined as the magnitude of the smallest translation that puts the two shapes $\mathcal{A}$ and $\mathcal{B}$* (1) *in contact, if they are currently separated or* (2) *out of contact, if they are currently in contact.*

This is illustrated in Figure 4.5. Intuitively, when the objects are separated, the signed distance is the smallest distance between any two points on the objects. On the other hand,

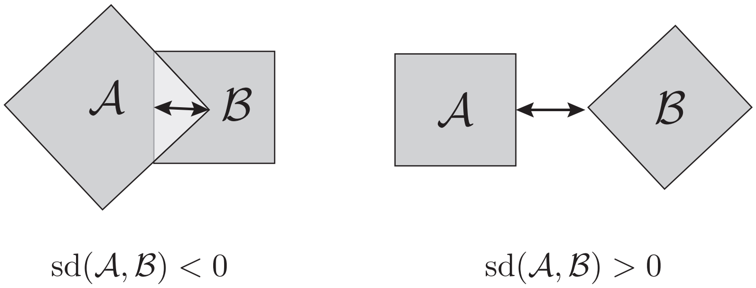

$\mathcal{A}$
$\mathcal{B}$
$\mathcal{A}$
$\mathcal{B}$
sd($\mathcal{A}, \mathcal{B}$) < 0
sd($\mathcal{A}, \mathcal{B}$) > 0

Figure 4.5: Signed distance between convex shapes $\mathcal{A}$ and $\mathcal{B}$ (Figure similar to that in [109])

when the objects are in collision, the signed distance (also known as the *penetration depth* in this case) is the minimum distance by which either object has to be moved in order to resolve the collision. Details about the signed distance and its computation are presented in [109]. Without loss of generality, we consider the case of one moving TV. Let the TV be represented by a convex shape $\mathcal{T}_{k|t}$, and the EV by $\mathcal{E}_{k|t}$. Note that $\mathcal{T}_{k|t}$ and $\mathcal{E}_{k|t}$ are functions of $x^e_{k|t}$ and $x^v_{k|t}$, respectively, and the vehicle geometries. Clearly, collision avoidance is achieved by constraining the signed distance to be greater than the minimum safety distance d_{safe}. That is:

$$\text{sd}(\mathcal{T}_{k|t}, \mathcal{E}_{k|t}) \geq d_{\text{safe}}. \tag{4.31}$$

The above constraint is nonlinear and nonconvex in general. Moreover, the computations involved make it formidable for use with a real-time nonlinear predictive control scheme. In practice, (4.31) is linearized around a nominal EV state $\bar{x}^v_{k|t}$ to obtain:

$$G^e_k(x^v_{k|t} - \bar{x}^v_{k|t}) \leq h^e_k, \tag{4.32}$$

with:

$$G^e_k = -\nabla_{x^v_{k|t}} \text{sd}(\mathcal{T}_{k|t}, \mathcal{E}_{k|t})|_{\bar{x}^v_{k|t}}, \tag{4.33a}$$

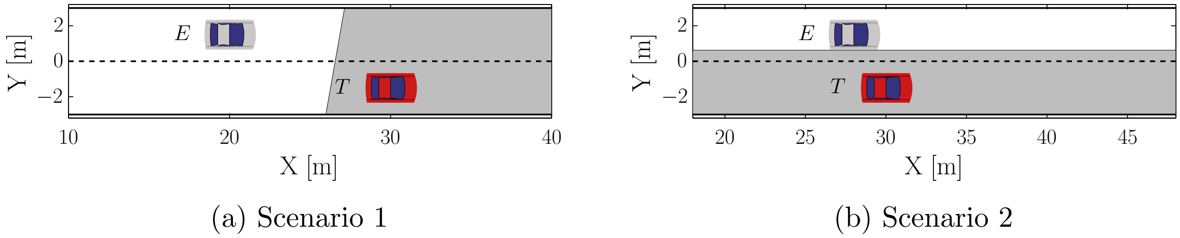
$$h^e_k = \text{sd}(\mathcal{T}_{k|t}, \bar{\mathcal{E}}_{k|t}) - d_{\text{safe}}. \tag{4.33b}$$

$\bar{\mathcal{E}}_{k|t}$ denotes the convex shape corresponding to $\bar{x}^v_{k|t}$. The polyhedral safety region obtained by the linearized signed distance function (4.32) for two configurations of $\bar{x}^v_{k|t}$ and $x^e_{k|t}$ is depicted in Figure 4.6. The shaded area is the unsafe region for the EV's center of gravity. The efficient computation of the gradient $\nabla_{x^v_{k|t}} \text{sd}(\mathcal{T}_{k|t}, \mathcal{E}_{k|t})$ is discussed in [109] and can be obtained as an output of a collision checking library.

4.5.3 Summary

In general, the collision avoidance constraints in (4.32) and (4.27) formulated using the approaches in Sections 4.5.2 and 4.5.1, respectively, are concisely written as:

$$g(x^v_k, x^e_k) \leq 0. \tag{4.34}$$



(a) Scenario 1

(b) Scenario 2

Figure 4.6: Unsafe region for the depicted positions of the EV E and TV T in two scenarios. The EV's center of gravity must lie outside the gray shaded area for a collision to be avoided.

The above constraints, linearized about a nominal state $\bar{x}_k^v$, are concisely written as:

$$G_k^e(x_{k|t}^v - \bar{x}_{k|t}^v) \leq h_k^e. \tag{4.35}$$

Chapter 5

Iterative Linearization for Real-Time Nominal Model Predictive Control

5.1 Introduction

This chapter focuses on the design of a real-time path-following and obstacle avoidance controller for the nominal or deterministic predictive control problem where uncertain variables assume their mean values. In this context, the controller design is challenging due to: (1) the nonlinearity of the vehicle dynamics, and (2) the presence of time–varying nonconvex state constraints while navigating in a dynamic environment. Because of its ability to handle system nonlinearities and constraints in a unified manner, MPC has been shown to be an attractive control methodology in this class of applications [8, 43, 45, 48, 49, 54, 55].

The main limitation in using optimization–based control strategies, such as MPC, is the computational burden of solving the optimization problem in real–time. The complexity arises from the nonlinearity of the vehicle dynamics and the non–convexity of the constraints [49]. While linearizing the system dynamics leads to some of the constraints becoming convex, the collision avoidance constraints are still non–convex. Moreover, a linearized vehicle model is a good approximation only in a small region around the reference state and input vectors about which the linearization is performed. The use of nonlinear MPC for vehicle control has been proposed in [45, 48, 49, 54, 55]. General purpose solvers for nonlinear optimization, such as NPSOL [51], are used to solve the resulting non–convex optimization problem.

Low-complexity MPC schemes in the context of automotive control applications have been proposed in several works in literature. The approach in [44] still solves a nonlinear optimization problem at each time step, but the number of control inputs (and hence, decision variables) is reduced by exploiting the structure of the four-wheel nonlinear model in (3.3). A linearization-based MPC approach to solve the nonlinear optimization problem is proposed in [43]. A Linear Time-Varying (LTV) model of the vehicle dynamics over the prediction horizon is computed based on the assumption of constant control inputs. The resulting

CFTOC problem to be solved online is a Quadratic Program (QP). A similar approach is used in [82], but the LTV model is obtained by using a shifted sequence of optimal inputs from the previous time step. In [130], the *real-time iteration* scheme (proposed in [39] by the same authors) is applied to the collision avoidance problem for autonomous vehicles using nonlinear MPC. This approach is based on a single linearization and the solution of a QP at each sampling time. A drawback of these approaches is that if the optimal input sequence differs significantly from the one used for linearization (for example, due to the sudden appearance of an obstacle), the LTV model used for the control design is no longer a good approximation of the vehicle dynamics. This motivates the need for an iterative linearization procedure where a sequence of approximate problems is solved at each time step.

In this chapter, we present a tailored algorithm based on sequential quadratic programming (SQP) which exploits the structure of the autonomous driving control problem. SQP involves the iterative solution of a convex approximation to the original problem [18, 58, 94]. We use a similar approach to that in [109] wherein a method based on sequential convex programming for trajectory optimization is proposed. At each iteration of the optimization procedure, a convex approximation of the obstacle-free space is computed and the equality constraints due to the system dynamics are linearized. This results in a sequence of computationally cheaper QPs to be solved at each time step.

The features of the presented approach are listed below.

1. The algorithm is general in that it can be used with any of the nonlinear vehicle models presented in Chapter 3. In the examples in this chapter, we use the high-fidelity four-wheel nonlinear vehicle model of Section 3.2.1.

2. We validate the approach in challenging driving situations where a consideration of the nonlinear tire-road interaction is especially important. In particular, we perform autonomous driving experiments at high speeds on snow where the friction coefficient is around 0.3.

3. The algorithm is run in real-time on embedded computing platforms with relatively low processing power. We show the ability of our tailored approach in reducing the online computational time as compared to a generic nonlinear solver currently employed.

4. Although we solve a Nominal MPC in this chapter (that is, system uncertainties are assumed to take on their mean values), the approach easily extends to the solution of the Robust and Stochastic MPC problems. This is due to the fact that the solution of these problems involves their translation into deterministic ones.

This chapter is organized as follows. Section 5.2 presents the formulation of the MPC problem, Section 5.3 details the proposed algorithm, Section 5.4 shows results from hardware–in–the–loop (HIL) simulations, experimental results are presented in Section 5.5, and concluding remarks are made in Section 5.6.

5.2 Nonlinear MPC Formulation

The nominal open-loop CFTOC problem for autonomous navigation and obstacle avoidance is given by:

$$\min_{u^v_{0:T-1|t}} \sum_{k=0}^{T-1} \|x^v_{k+1|t} - x^{\mathrm{ref}}_{k+1|t}\|^2_Q + \|u^v_{k|t}\|^2_R + \|\Delta u^v_{k|t}\|^2_P, \tag{5.1a}$$

$$\begin{aligned}
\text{subject to:}\quad & x^v_{k+1|t} = f^v(x^v_{k|t}, u^v_{k|t}, \bar{d}^v_{k|t}), && (5.1\text{b})\\
& g(x^v_{k|t}, \bar{x}^e_{k|t}) \leq 0, && (5.1\text{c})\\
& \Delta u^v_{k|t} = u^v_{k|t} - u^v_{k-1|t}, && (5.1\text{d})\\
& [u^v_{k|t}, u^v_{k-1|t}] \in \mathcal{U}, && (5.1\text{e})\\
& (k = 0, ..., T-1) && \\
& x^v_{0|t} = x^v_t, \quad u^v_{-1|t} = u^v_{t-1}, && (5.1\text{f})
\end{aligned}$$

where the notation $\|z\|^2_Q = z^{\mathrm{T}}Qz$. The cost function (5.1a) penalizes the deviation of the state from a given reference trajectory $\{x^{\mathrm{ref}}_{k|t}\}^T_{k=1}$ in addition to penalizing the magnitude and the rate of the control inputs. Note the use of the mean value of the disturbance $\bar{d}^v_k$ in the dynamics (5.1b) and the nominal predicted environment state $\bar{x}^e_k$ in the safety constraints (5.1c). We now present a tailored algorithm for solving (5.1). The presented methodology is applicable to any nonlinear dynamics model $f^v(\cdot)$ and any formulation of the safety constraints $g(\cdot)$. The models used in our work are made clear later in Section 5.4. In the following sections, we use z_k to denote the variable $z_{k|t}$.

5.3 Iterative Linearization Approach

The proposed algorithm is similar in structure to an SQP approach [18, 58, 94]. Recall that the optimization problem (5.1) is nonconvex due to the nonlinearity of the dynamics map $f^v(\cdot)$ in (5.1b) and the nonconvexity of the collision avoidance constraints (5.1c). Hence, we analytically linearize the vehicle dynamics model, and simplify the collision avoidance constraints to obtain a QP approximation to the original problem. Such an approach is significantly faster than a numerical convexification scheme.

We introduce the following notation. Let $\bar{\mathcal{U}}^v_t = \{\bar{u}^v_k\}^{T-1}_{k=0}$ denote the current candidate solution to the optimization problem (5.1), that is, the current guess for the optimal open-loop control sequence. Let $\bar{\mathcal{X}}^v_t = \{\bar{x}^v_k\}^T_{k=1}$ be the sequences of predicted states generated by the nonlinear vehicle model (5.1b) with the input sequence $\bar{\mathcal{U}}^v_t$ and $\bar{x}^v_{0|t} = x_t$ as the initial condition. We first find a convex local approximation to the original problem around $\bar{\mathcal{X}}^v_t$ and $\bar{\mathcal{U}}^v_t$. The resulting QP is then solved to obtain a new candidate solution. This process of convexification and optimization is iterated till a convergence criterion is satisfied. A method of approximately convexifying constraints, given $\bar{\mathcal{U}}^v_t$ and $\bar{\mathcal{X}}^v_t$, is presented below.

5.3.1 Convexification of constraints

Vehicle dynamics

The nonlinear model (5.1b) can be linearized about $\bar{\mathcal{U}}_t^v$ and $\bar{\mathcal{X}}_t^v$ to yield a linear time-varying (LTV) state-space representation:

$$x_{k+1}^{\delta,v} = A_k x_k^{\delta,v} + B_k u_k^{\delta,v}, \qquad (k = 0, ..., T-1) \tag{5.2a}$$

$$x_0^{\delta,v} = 0, \tag{5.2b}$$

where:

$$x_k^{\delta,v} = x_k^v - \bar{x}_k^v, \tag{5.3a}$$

$$u_k^{\delta,v} = u_k^v - \bar{u}_k^v, \tag{5.3b}$$

and:

$$A_k = \nabla_{x_k^v} f^v(x_k^v, u_k^v, d_k^v)|_{(\bar{x}_k^v, \bar{u}_k^v, \bar{d}_k^v)}, \tag{5.4a}$$

$$B_k = \nabla_{u_k^v} f^v(x_k^v, u_k^v, d_k^v)|_{(\bar{x}_k^v, \bar{u}_k^v, \bar{d}_k^v)}. \tag{5.4b}$$

Analytical expressions for the matrices A_k and B_k as a function of the nominal trajectory are determined offline for the specific vehicle model employed. The analytical linearization is computationally much less expensive as compared to any numerical linearization scheme, especially when the dimensions of the state and input space, and the prediction horizon are large. Automatic differentiation solvers can also be used to reduce implementation errors (see, for example, [98]).

Safety constraints

Recall that the collision avoidance constraints as formulated in (5.1c) are non-differentiable in general. The linearized safety constraints are given by:

$$G_k^e x_k^{\delta,v} - h_k^e \leq 0, \qquad (k = 0, ..., T-1) \tag{5.5}$$

where:

$$G_k^e = \nabla_{x_k^v} g(x_k^v, x_k^e)|_{(\bar{x}_k^v, \bar{x}_k^e)}, \tag{5.6a}$$

$$h_k^e = -g(\bar{x}_k^v, \bar{x}_k^e). \tag{5.6b}$$

The collision avoidance constraints presented in Section 4.5 are particularly amenable to the above linearization procedure. In the case of the signed distance function presented in Section 4.5.2, the gradient with respect to the ego vehicle's state can be computed analytically as shown in [109]. Similarly, the piecewise affine nature of the safety corridor constraint formulation in Section 4.5.1 allows us to easily compute the gradients of the upper and lower bounds of the corridor with respect to the vehicle states.

5.3.2 QP formulation

The convexification methodology presented in Section 5.3.1 transforms the nonconvex optimization problem (5.1) into a QP given by:

$$
\begin{aligned}
\min_{u^{\delta,v}_{0:T-1}} \quad & \sum_{k=0}^{T-1} \|\bar{x}^v_{k+1} + x^{\delta,v}_{k+1} - x^{\text{ref}}_{k+1}\|^2_Q + \|\bar{u}^v_k + u^{\delta,v}_k\|^2_R \\
& \qquad\qquad + \|u^{\delta,v}_k\|^2_{R^\delta} + \|\Delta u^v_k\|^2_P + M\epsilon_{k+1}, && \text{(5.7a)} \\
\text{subject to:} \quad & x^{\delta,v}_{k+1} = A_k x^{\delta,v}_k + B_k u^{\delta,v}_k, && \text{(5.7b)} \\
& G^e_k x^{\delta,v}_k \leq h^e_k, && \text{(5.7c)} \\
& \Delta u^v_k = (\bar{u}^v_k + u^{\delta,v}_k) - (\bar{u}^v_{k-1} - u^{\delta,v}_{k-1}), && \text{(5.7d)} \\
& [\bar{u}^v_k + u^{\delta,v}_k, \bar{u}^v_{k-1} + u^{\delta,v}_{k-1}] \in \mathcal{U}, && \text{(5.7e)} \\
& \|u^{\delta,v}_k\| \leq \epsilon_{u,tr} && \text{(5.7f)} \\
& (k = 0, ..., T-1) \\
& x^{\delta,v}_0 = x^v_t - \bar{x}^v_t, \quad u^{\delta,v}_{-1} = u^v_{t-1} - \bar{u}^v_{t-1}. && \text{(5.7g)}
\end{aligned}
$$

The additional term $\|u^{\delta,v}_k\|^2_{R^\delta}$ in the objective function (5.7a) penalizes the deviation $u^{\delta,v}_k$ from $\bar{u}^v_k$ in order to improve the convergence of the algorithm. A trust region constraint (5.7f) is introduced to ensure the accuracy of the linear model (5.7b) around the nominal trajectory $\bar{\mathcal{U}}^v_t$ and $\bar{\mathcal{X}}^v_t$. In practice, the linearized safety constraints (5.7c) are imposed as soft constraints by introducing a slack variable which is heavily penalized in the cost function (5.7a) by a large number M.

The computational complexity of the MPC problem can be further reduced by keeping the input vectors constant for every T_b time steps in the prediction horizon. That is:

$$
u^v_{i\cdot T_b + k} = u^v_{i\cdot T_b} \qquad \forall\ k = 0, ..., (T_b - 1), \quad \forall\ i = 0, ..., (T/T_b). \tag{5.8}
$$

This is equivalent to holding the deviations $u^{\delta,v}_k$ constant for every T_b steps and reduces the number of optimization variables in (5.7) by a factor of T_b. We refer to the parameters T and T_b as the *prediction horizon* and the *input blocking factor*, respectively.

5.3.3 Overall algorithm

The nonlinear constrained finite-time optimal control problem (5.1) is solved in a receding horizon manner by using Algorithm 5.1. The variable $N_{\max}$ limits the maximum number of iterations of the iterative linearization procedure. The function warm_start($\cdot$) uses the optimal input sequence $\bar{\mathcal{U}}^v_{t-1}$ from the previous time step to generate an initial candidate input sequence $\bar{\mathcal{U}}^v_t$ for the current time step as follows:

$$
\bar{u}_{k|t} = \begin{cases} \bar{u}_{k|t-1}, & \text{if } k = 0, 1, ..., (T-2) \\ \bar{u}_{T-2|t-1}, & \text{if } k = T-1. \end{cases} \tag{5.9}
$$

The function simulate$(\cdot)$ generates the state sequence $\bar{\mathcal{X}}_t^v$ using the nonlinear vehicle model (5.1b). The function solve_QP$(\cdot)$ formulates the QP (5.7) using $\bar{\mathcal{U}}_t^v$ and $\bar{\mathcal{X}}_t^v$ (as presented in Section 5.3.1) and yields the perturbed input sequence $\mathcal{U}_t^{\delta,v} = \{u_k^{\delta,v}\}_{k=0}^{T-1}$. The algorithm terminates when the norm of the perturbations is below a specified threshold ϵ_u or the iteration limit $N_{\max}$ is reached.

Algorithm 5.1 Tailored MPC algorithm

```
Initialize:
$\bar{\mathcal{U}}_0^v = 0$
while $t \geq 0$ do:
    $x_{t|t} = x_t$                                                    ▷ Measure state
    $\bar{\mathcal{U}}_t^v = \text{warm\_start}(\bar{\mathcal{U}}_{t-1}^v)$        ▷ Initialize candidate solution
    iter = 0                                                            ▷ Iteration number
    while $\|\mathcal{U}_t^{\delta,v}\| > \epsilon_u$ and iter $< N_{\max}$ do:
        $\bar{\mathcal{X}}_t^v = \text{simulate}(x_{t,t}, \bar{\mathcal{U}}_t^v)$     ▷ Simulate system using (5.1b)
        $\mathcal{U}_t^{\delta,v} = \text{solve\_QP}(\bar{\mathcal{U}}_t^v, \bar{\mathcal{X}}_t^v)$   ▷ Solve QP (5.7)
        $\bar{\mathcal{U}}_t^v = \bar{\mathcal{U}}_t^v + \mathcal{U}_t^{\delta,v}$             ▷ Update candidate
        iter = iter + 1
    $u_t^\star = \bar{u}_{t|t}^v$                                      ▷ Apply optimal input
    $t = t + 1$
```

Remark 5.1. *The main feature of the algorithm is that it generates a sequence of states and inputs ($\bar{\mathcal{X}}_t^v$ and $\bar{\mathcal{U}}_t^v$, respectively) that are dynamically feasible according to the nonlinear vehicle model* (5.1b)*. In fact, dynamic feasibility is maintained at every iteration of the algorithm.*

Remark 5.2. *As the linearization is performed analytically, the time-limiting step is the solution of the QP* (5.7)*. This can be sped up by using tailored embedded QP solvers [40, 41].*

5.4 Simulation Results

We compare the performance of the proposed controller with that of the controller which uses the general purpose nonlinear solver NPSOL. Note that NPSOL implements a sequential quadratic programming (SQP) algorithm for nonlinear optimization [51] and that the internal QP solver used by NPSOL (i.e. LSSOL) is the same as that used by the proposed algorithm. The main difference is that NPSOL computes the convex approximation of the nonlinear problem using numerical differentiation which is slow, especially for a large problem. Moreover, NPSOL does not utilize any problem specific information to convexify the nonlinear program.

5.4.1 Simulation setup

HIL simulations of the controller are performed on a dSPACE rapid prototyping system consisting of a DS1401 MicroAutoBox (IBM PowerPC 750FX processor, 800 MHz) and a DS1006 processor board (Quad-core AMD Opteron processor, 2.8 GHz). The controller runs on the MicroAutoBox, and the DS1006 board simulates the vehicle dynamics using a nonlinear four wheel vehicle model with a Pacejka tire model. The simulation model is of a higher fidelity than that of the four wheel vehicle model presented in Section 3.2.1 as it also considers the rotational dynamics of the wheels. The physical parameters correspond to those of the prototype Jaguar vehicle described in Appendix A.1. The control actions are executed at 10 Hz.

5.4.2 Technical approach

Vehicle model: The nonlinear four wheel vehicle model presented in Section 3.2.1 with a Fiala tire model is used for the control design. The position and orientation of the vehicle are expressed in the inertial frame. Hence, $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, X_t, Y_t, \psi_t]^T$ and $u_t^v = [\delta_{f_t}, \beta_{fl_t}, \beta_{fr_t}, \beta_{rl_t}, \beta_{rr_t}]^T$. In order to reduce the number of optimization variables and hence the computational demand, we use a similar approach as in [44] and introduce the following assumption on the braking ratios.

Assumption 5.3. *The braking ratio at the front left (right) wheel is equal to that at the rear left (right) wheel, i.e., $\beta_{fl} = \beta_{rl} = \beta_l$, and $\beta_{fr} = \beta_{rr} = \beta_r$.*

The modified input vector is given by $u_t^v = [\delta_{f_t}, \beta_{l_t}, \beta_{r_t}]^T$. The road is assumed to be straight and aligned with the inertial axes. A single lane road is considered with the centerline defined by $Y = 0$.

Safety constraints: The collision avoidance constraints are formulated using the safety corridor approach discussed in Section 4.5.1.

Environment model: The environment state x_t^e consists of the inertial positions of all TVs in the scene. We assume that the future positions of all TVs are known ahead of time without any uncertainty. Thus, the nominal values $\bar{x}_t^e$ are known over the duration of the simulation.

5.4.3 Results and discussion

We denote the controller which uses the proposed iterative linearization scheme as Controller 1, and the controller which uses NPSOL as Controller 2. Note that both controllers are solving the nonlinear problem (5.1) with identical safety constraints. That is, the complexity of the optimization problem solved by both controllers is the same in terms of the number of optimization variables and constraints. The simulation results are summarized in Figures 5.1–5.2.

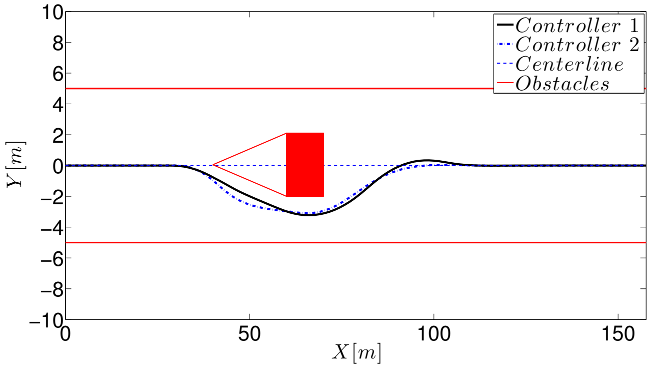


Figure 5.1: Simulation 1: The performance of the proposed controller (Controller 1) is similar to that of the controller which uses NPSOL (Controller 2) for the same sampling time, prediction horizon, environment scenario and constraints in the optimization problem.

The scenario consists of a single obstacle in the path of the vehicle. The edge of the obstacle is at a distance of 2 m from the road centerline. In order to account for the width of the vehicle, a tolerance of 1 m is included in the obstacle and lane bounds. A triangular zone is appended to the front of the obstacle to avoid aggressive maneuvers. We assume the existence of a high–level decision making module which decides the side of the obstacle on which the vehicle must pass. The entry speed of the vehicle is 15 m/s. In our tests, the high–level module uses a simple heuristic to choose the side with greater room to pass. The goal of the controller is to track the road centerline (that is, $Y^{\text{ref}} = 0$) while avoiding a collision with the obstacle.

In Simulation 1, the performance of the two controllers is compared when the same prediction horizon and input blocking factor are used ($T = 9$, $T_b = 3$). The path of the vehicle is shown in Figure 5.1. The performance of the two controllers is observed to be similar.

In Simulation 2, the prediction horizon for controller 1 is increased to $T = 30$, while that for controller 2 is increased to $T = 11$. The path of the vehicle is shown in Figure 5.2.

Remark 5.4. *The prediction horizon used in Simulation* 2 *is the maximum allowable given the* 100 *ms sampling time.*

It is seen that the vehicle stays very close to the obstacle while using controller 1, and returns to the road centerline with almost no overshoot. However, controller 2 does not improve its performance compared to Simulation 1. Moreover, the proposed methodology allows the use of a much longer prediction horizon as compared to the approach that uses NPSOL. This is desirable for the early consideration of hazards.

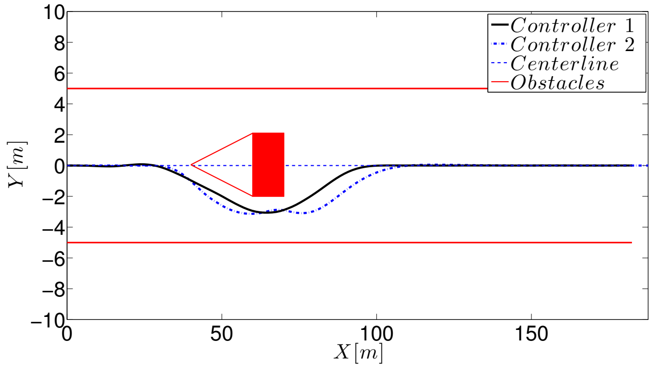


Figure 5.2: Simulation 2: Controller 1 uses a much longer prediction horizon as compared to Controller 2.

5.5 Experimental Results

5.5.1 Experimental setup

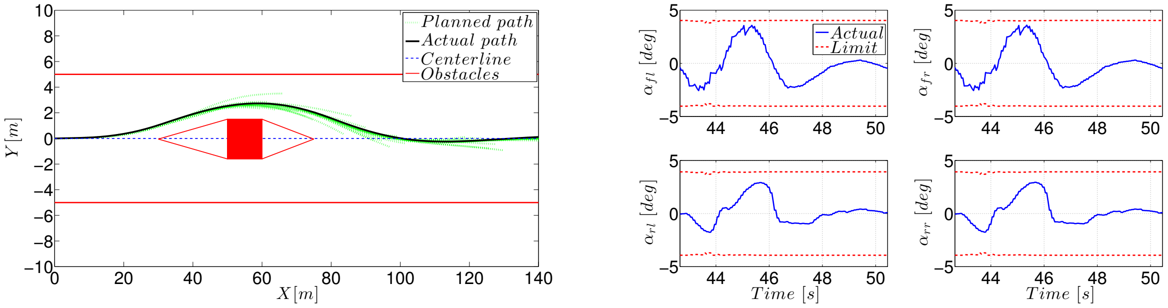
The experiments were performed on the prototype Jaguar vehicle described in Appendix A.1 at the Smithers winter testing center in Michigan on tracks covered with packed snow ($\mu \approx 0.3$). The goal is to validate the performance of the proposed iterative linearization approach in relatively challenging situations where a consideration of the nonlinear vehicle dynamics is essential. The choice of the vehicle and environment models and the safety constraint formulation is the same as that in the simulations (see Section 5.4.2 for details). The control actions are executed at 10 Hz.

5.5.2 Results and discussion

The following parameters are used for the controller: $T = 21$, $T_b = 3$, $N_{max} = 8$. The experimental results are presented in Figures 5.3–5.5. The green lines depict the open–loop planned paths while the black line denotes the actual path of the vehicle. In the scenarios considered, the road is straight and the edge of each obstacle is at a distance of 1.5 m from the road centerline. A triangular zone is appended to the front and back of each obstacle to avoid aggressive avoidance maneuvers. As in the case of the simulations, we assume that a high-level module decides the side of the obstacle on which the vehicle must pass.

Figure 5.3a plots the path of the vehicle while avoiding a single obstacle at an entry speed of 80 km/hr. The vehicle stays close to the obstacle, and returns to the road centerline with a low overshoot. The slip angles at the four wheels are shown in Figure 5.3b. The controller is able to keep the slip angles within the limits. It is seen that at around $t = 45$ s, the slip angles at the front wheels are close to the limiting values (≈ 4 deg).

Figure 5.4 shows the path of the vehicle while avoiding two obstacles separated by a distance of 70 m. The vehicle passes the two obstacles on opposite sides due to the high–



(a) Vehicle path

(b) Slip angles at the four wheels

Figure 5.3: Experimental test 1: The vehicle avoids a single obstacle with an entry speed of 80 km/hr.

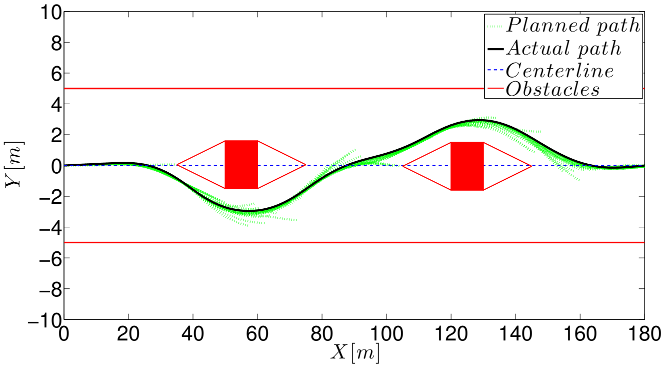


Figure 5.4: Experimental test 2: The vehicle avoids two separated obstacles with an entry speed of 60 km/hr.

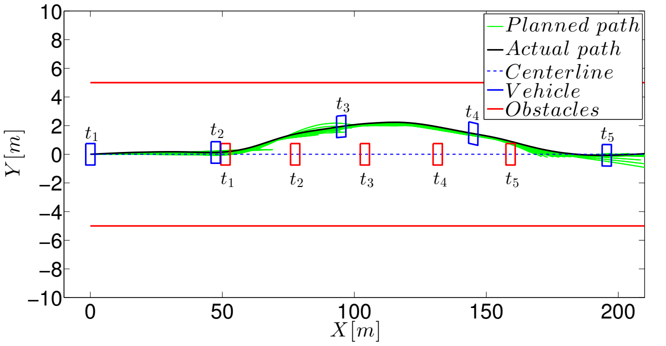


Figure 5.5: Experimental test 3: The vehicle avoids an obstacle moving at 36 km/hr with an entry speed of 80 km/hr. The positions of the vehicle and the obstacle at five time instants $t_1, ..., t_5$ during the test are shown.

level decision making heuristic. An important point to be noted is that the number of constraints and the complexity of the optimization problem (5.7) is independent of the number of obstacles. This is due to the consideration of obstacles by the safety corridor approach.

Figure 5.5 plots the path of the vehicle while avoiding an obstacle moving at a constant speed of 36 km/hr. The positions of the vehicle and obstacle at time instants $t_1, ..., t_5$ during the test are shown. The edge of the obstacle is 1 m away from the road centerline. The lateral position of the obstacle remains constant, and it is assumed that the controller knows the present and future positions of the obstacle.

5.5.3 Controller tuning

We discuss the tuning of the controller keeping the parameters listed in Section 5.5.2 fixed from an implementation perspective. Recall that the parameter R^δ was introduced in (5.7a) to penalize the perturbation from the sequence $\bar{\mathcal{U}}_t$. If R^δ is of the order of magnitude of Q and R, the iterative procedure takes longer to converge. On the other hand, if $R^\delta \gg Q, R$, the convergence is faster at the cost of not meeting the original objectives. We deal with this trade–off by using the following heuristics:

- For $i = 0$: $R^\delta \ll Q, R \lll c$
- For $i < N_{sat}$: $R^\delta \leftarrow \beta R^\delta$, where $\beta > 1$
- For $N_{sat} \leq i < N_{max}$: $Q, R \ll R^\delta \ll c$

where i is the iteration number, and N_{sat} is an additional tuning parameter. Based on the above heuristics, the set of tuning parameters used in the experiments is given in Table 5.1.

Table 5.1: Tuning parameters for experiments

Parameter	**Value**	**Parameter**	**Value**
Q	$\text{diag}\,(0.01, 0.02, 0.05, 0.1, 0.1, 0)$	R	$\text{diag}\,(0.05, 0.1, 0.1)$
Initial R^δ	$\text{diag}\,(0.001, 0.001, 0.001)$	c	$\text{diag}\,(10^4, 10^5)$
β	100	N_{sat}	3

5.6 Conclusions

In this chapter, we presented a tailored algorithm for the nonlinear nominal MPC problem based on an iterative linearization of the vehicle dynamics and a convex approximation of the safety constraints. It reduces the online computational complexity of the nonlinear MPC scheme. Simulative and experimental results show the ability of the proposed algorithm to run in real-time on embedded computing platforms. Moreover, much longer prediction horizons as compared to general purpose nonlinear solvers can be used. The experiments involve aggressive maneuvers at high speeds on snow and demonstrate the ability of the controller to keep the vehicle safe in scenarios involving multiple static and moving obstacles. While uncertainty is not considered in this chapter, the presented approach is general and easily extends to the robust and stochastic MPC problems presented later in Chapters 7 and 8, respectively.

Chapter 6

Robust Lateral Stability Control with an Uncertain Driver Model

6.1 Introduction

Modern passenger vehicles are equipped with active safety systems which assist drivers in order to stabilize the vehicle and prevent accidents. In this chapter, we focus on the integration of two active safety systems, Electronic Stability Control (ESC) and Active Front Steering (AFS), which enhance vehicle stability by applying braking torques to the wheels and correcting the front steering, respectively [3, 122]. This problem has been studied in the past by several authors. For instance, optimal control methods have been studied in [84], generalized predictive control in [92] and state feedback linearization in [19]. In [38, 97], the authors presented a model–based predictive control technique in which a piecewise affine (PWA) vehicle dynamics model was used for the control design.

Four elements make the control problem nontrivial: the vehicle model uncertainty, the nonlinear tire characteristics, the presence of state and input constraints, and the uncertainty in driver behavior. We present a systematic approach to design a robust lateral stability controller which addresses all four challenges in a unified framework during the control design process. In particular, the nonlinear vehicle dynamics are modeled as a PWA system in which the states and inputs are subject to hard constraints. Model mismatch is captured by introducing an additive uncertainty in the input. Finally, the driver's steering input is treated as a bounded disturbance in the model, whose bounds are a set–valued function of the vehicle states.

The principal goal of our control strategy is to enlarge the vehicle's stability region when it undertakes aggressive maneuvers. Such maneuvers are primarily determined by the front wheel steering angle which is a combined effect of the driver's steering input and the AFS correction. In order to take both contributions into account, we propose the use of a predictive control framework which considers the driver's input as a measured disturbance, and the AFS correction as a constrained control input. Our main motivation is to ensure robustness to the uncertainty in the driver's inputs at each time step. In this work, we extend the methodology presented in [97] to account for this uncertainty. Existing approaches in literature do not address the uncertain driver behavior and constraint satisfaction challenges in a systematic way. In particular, in [38], the driver's input is assumed to be constant over the prediction horizon. In [46], the output of the controller is the desired total steering angle (driver + AFS), which could lead to the possibility of an infeasible AFS command. In [92], no constraints are imposed on the steering corrections that can be applied by the AFS system.

We use set–theoretic methods for the control synthesis [16]. The theoretical framework developed in [97] forms the basis for our control design. In particular, we compute the robust control invariant (RCI) set for the PWA vehicle model using results on min–max and max–min reachability [12, 16, 102]. Such a RCI set constitutes the target set of a robust time–optimal control algorithm. Experimental tests at high speeds on ice with aggressive driver maneuvers show the effectiveness of the proposed scheme.

The chapter is organized as follows. Section 6.2 describes the derivation of the PWA model starting from the nonlinear bicycle model. The construction of the set–valued bounds on the driver steering is shown in Section 6.3. In Section 6.4, we discuss the theoretical framework used for the design of the robust controller. We present experimental results in Section 6.5, followed by final remarks in Section 6.6.

6.2 Piecewise Affine Vehicle Model

The design of the robust stability controller is based on the lateral dynamics model (3.16) in Section 3.2.3. The PWA approximation of the nonlinear lateral tire force function $h_{c_\star}(\cdot)$ in (3.17) with $\star \in \{f, r\}$ is given by:

$$h_{c_\star}^{\text{pwa}}(\alpha_\star) = \begin{cases} c_{s_\star}\alpha_\star + (c_{l_\star} + c_{s_\star})\hat{\alpha}_\star & \text{if } \alpha_\star \leq -\hat{\alpha}_\star, \\ -c_{l_\star}\alpha_\star & \text{if } -\hat{\alpha}_\star \leq \alpha_\star \leq \hat{\alpha}_\star, \\ c_{s_\star}\alpha_\star - (c_{l_\star} + c_{s_\star})\hat{\alpha}_\star & \text{if } \alpha_\star \geq \hat{\alpha}_\star, \end{cases} \tag{6.1}$$

where $c_{l_\star}$ and $c_{s_\star}$ are the cornering stiffnesses of the tires in the linear and saturated regions, respectively, and $\hat{\alpha}_\star$ is the slip angle at which the lateral force is maximum. Figure 6.1 shows the lateral force plotted against the slip angle for the Pacejka tire model (with slip ratio $\sigma = 0$), the PWA approximation. The gray dots depict the lateral force values estimated from data collected during experiments on a winding road and a 'figure 8' drift maneuver.

The hybrid bicycle model is obtained by combining the lateral dynamics model (3.16), the slip angle approximation (3.15), the assumption of constant longitudinal speed $\dot{x}$ (see Assumption 3.7) and the PWA approximation (6.1). It is written as:

$$\begin{bmatrix} \ddot{y} \\ \ddot{\psi} \end{bmatrix} = A_i \begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix} + B_i \begin{bmatrix} \delta_f \\ M_z \end{bmatrix} + f_i \quad \text{if} \quad (\dot{y}, \dot{\psi}, \delta_f) \in \mathcal{R}_i. \qquad (i = 1, \ldots, 9) \tag{6.2}$$

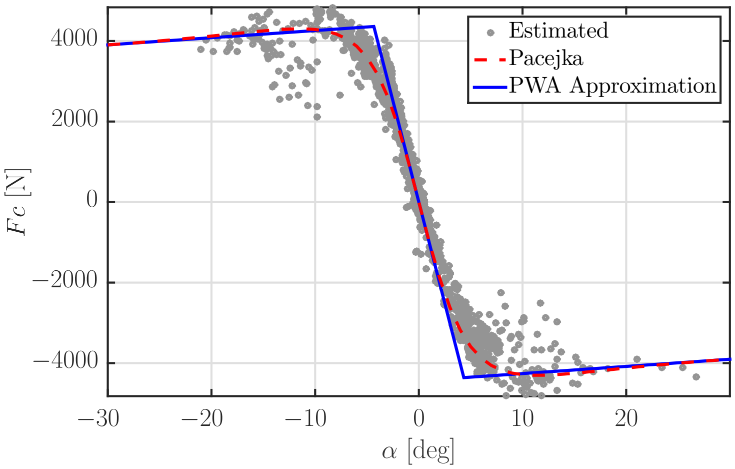


Figure 6.1: Lateral tire forces and the PWA approximation

Model (6.2) has nine modes resulting from all possible combinations of the three modes for the front wheels and the three modes for the rear wheels. The polyhedral regions $\{\mathcal{R}_i\}_{i=1}^{9}$ define the state and input constraints corresponding to these nine modes. The regions are obtained from the slip angle inequalities associated with each of the cases in (6.1).

It is important to note that in a vehicle equipped with an AFS system, the front wheel steering angle δ_f is the sum of the steering angles resulting from two independent inputs: (i) the driver's input at the steering wheel, and (ii) the AFS input. That is:

$$\delta_f = \delta_d + \delta_{\text{AFS}}. \tag{6.3}$$

We also introduce an additive uncertainty d_u^v in the input to account for unmodeled actuator dynamics and input delays. The modified PWA bicycle model can then be written in a discrete–time state–space form as:

$$\begin{aligned} &x_{t+1}^v = A_i^d x_t^v + B_i^d u_t^v + D_i^d \delta_{d_t} + f_i^d + B_i^d d_{u_t}^v, \\ &(x_t^v,\, u_t^v, \delta_{d_t}) \in \mathcal{Q}_i, \qquad (i = 1, 2, ..., 9) \\ &d_{u_t}^v \in \mathcal{D}_u^v(u_t^v), \end{aligned} \tag{6.4}$$

where $x_t^v := [\dot{y}_t, \dot{\psi}_t]^T$ and $u_t^v := [\delta_{\text{AFS}_t}, M_{z_t}]^T$. The collection of polyhedral regions $\{\mathcal{Q}_i\}_{i=1}^{9}$ is obtained from $\{\mathcal{R}_i\}_{i=1}^{9}$ in (6.2), but lies in a higher dimensional space due to the decoupling of the steering input by (6.3). $\mathcal{D}_u^v(\cdot)$ is a set–valued mapping which defines bounds on d_u^v. Note that $\mathcal{D}_u^v(\cdot)$ is chosen to be a set–valued function as the uncertainty in the input may depend on its value. In our implementation, we assume an uncertainty of 10% in each of the control inputs. The model (6.4) has the standard form of the additive disturbance model presented in Section 3.6.2 with $d_t^v = [\delta_{d_t}, d_{u_t}^v]^T$. Reformulating (6.2) as (6.4) has several advantages:

(i) It allows us to impose bounds (arising from design constraints) directly on δ_{AFS} as opposed to imposing bounds on δ_f. This ensures that the input commanded by the controller can actually be attained.

(ii) It enables us to guarantee vehicle stability for any anticipated behavior of the driver. This is achieved by treating δ_d in (6.4) as a measured disturbance which is bounded by a function of the vehicle states. We design the controller to be robust to this disturbance.

In the remainder of the chapter, the superscript v corresponding to the vehicle is dropped for better visual clarity.

6.3 Uncertain Driver Model

In this section, we construct constraints of the form $\delta_d \in \mathcal{D}_x(x)$, where $\mathcal{D}_x(\cdot)$ is a set–valued mapping of the state x. One approach in determining bounds on δ_d is to assume a worst–case scenario. That is, $|\delta_d| \leq \delta_d^{\max}$, where $\delta_d^{\max}$ is the bound on δ_d due to the physical limits of the steering wheel. Such constraints, however, ignore the relationship between the yaw rate and the front wheel steering angle, and hence, are too conservative. We derive less conservative bounds on δ_d based on a steady-state cornering analysis of the linear bicycle model [89, 101]. At steady-state, the relationship between $\dot{\psi}$ and δ_f is given by:

$$\dot{\psi}_{ss} = \frac{\dot{x}}{L(1+\dot{x}^2/v_{\text{ch}}^2)}\delta_{f,ss} =: \frac{\delta_{f,ss}}{K_{\psi,ss}} =: G_{\psi,ss}\delta_{f,ss}, \tag{6.5}$$

where $L = l_f + l_r$ and v_{ch} is the characteristic speed [101]. $G_{\psi,ss} := 1/K_{\psi,ss}$ represents the steady-state yaw rate gain. Inverting (6.5) and assuming no control ($\delta_{\text{AFS},ss} = 0$) at steady-state gives:

$$\delta_{f,ss} = \delta_{d,ss} = K_{\psi,ss}\dot{\psi}_{ss}. \tag{6.6}$$

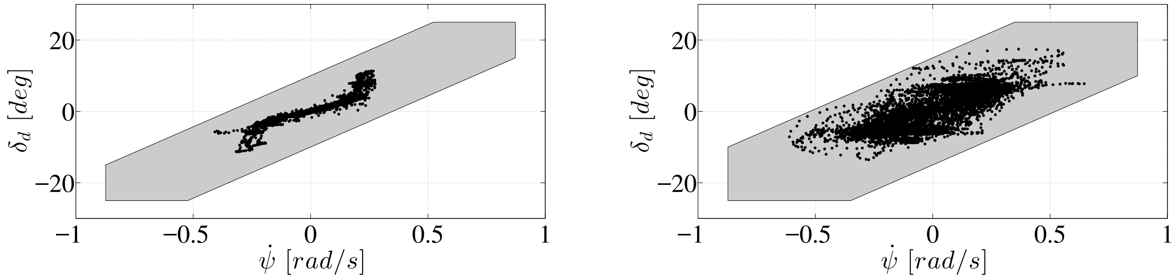
We use the value of $\delta_{d,ss}$ obtained from (6.6) as a linear state–dependent estimate of the driver's steering input. We then assume that the actual value of δ_d lies in an interval centered at $\delta_{d,ss}$. Thus:

$$\mathcal{D}_x(x) = \left\{\delta_d : |\delta_d - K_{\psi,ss}\dot{\psi}| \leq \epsilon, |\delta_d| \leq \delta_d^{\max}\right\}, \tag{6.7}$$

where ϵ is a non–negative parameter which must be chosen. The constraint $\delta_d \in \mathcal{D}_x(x)$ can also be expressed in terms of a polytopic constraint in $\mathbb{R}^3$. That is:

$$\delta_d \in \mathcal{D}_x(x) \Leftrightarrow (x, \delta_d) \in \mathcal{D}_d \subseteq \mathbb{R}^3. \tag{6.8}$$

Note that the bounds on δ_d are derived assuming the vehicle is cornering in a steady–state condition. Therefore, in practice, the average gain $(\delta_d/\dot{\psi})$ differs from $K_{\psi,ss}$. We account



(a) $K_\psi = 0.5$ sec, $\epsilon = 10$ deg

(b) $K_\psi = 0.5$ sec, $\epsilon = 15$ deg

Figure 6.2: State-dependent constraints on δ_d validated with data from different sets of experiments and the same driver.

for this by using a modified gain K_ψ in (6.7). This is illustrated in Figure 6.2 in which we plot the values of δ_d vs $\dot{\psi}$ obtained from experimental tests. The empirical value of K_ψ was computed to be $K_\psi \approx 0.5$ sec. The projection of the polytope $\mathcal{D}_d$ defined in (6.8) on the δ_d–$\dot{\psi}$ space is also plotted in Figure 6.2. Note that the choice of K_ψ and ϵ is critical to our analysis. For example, in Figure 6.2b, a higher value of ϵ is needed to ensure constraint satisfaction. The choice of ϵ is a trade–off between being too conservative on one hand, and having the possibility of violating constraints on the other.

6.4 Robust Control Design

The objective of the robust stability control system is to keep the front and rear tires in the linear region *for all admissible values* of δ_d and d_u. This linear mode of the vehicle is denoted as "Mode 1". If the vehicle goes outside mode 1, we want the controller to compute a feasible input which ensures that the vehicle goes back into mode 1 in a finite number of time steps. Moreover, when the vehicle is in mode 1, the controller action should be such that the predicted state at the next time step should also lie in mode 1. These notions of *reaching* a specified *target set* in the state–space and *staying* within the target set *for all admissible values of the disturbance variables* can be formalized using the framework of robust reachability analysis [16, 103].

6.4.1 Robust reachability framework

We make use of the notion of robust controllable or backward reachable sets which are defined below.

Definition 6.1. *The one–step robust backward reachable set to a given target set $\mathcal{X}$ in the state–space is defined as:*

$$Pre(\mathcal{X}) := \{x : \forall \delta_d \in \mathcal{D}_x(x), \exists u \text{ such that } (x, u, \delta_d) \in \mathcal{Q}, x^+ \in \mathcal{X}, \forall d_u \in \mathcal{D}_u(u)\}, \quad (6.9)$$

where x^+ is the predicted state at the next time step given by (6.4) *and* $\mathcal{Q} := \bigcup_{i=1}^{9} \mathcal{Q}_i$.

The $Pre(\cdot)$ mapping gives us the set of states from which there exists at least one feasible input that can ensure that the predicted state is inside the target set, for all admissible values of the disturbances δ_d and d_u. In addition to the one–step controllable sets, we introduce the notions of the robust control invariant (RCI) set associated with mode 1, and the corresponding control mapping. Let $\mathcal{P}_1 = \text{Proj}_x(\mathcal{Q}_1)$, where $\mathcal{Q}_1 \subseteq \mathbb{R}^5$ defines the state and input constraints for mode 1.

Definition 6.2. *A set $\mathcal{X}_i^1 \subseteq \mathcal{P}_1$ is called a mode 1 RCI set if for every $x \in \mathcal{X}_i^1$ and each $\delta_d \in \mathcal{D}_x(x)$, there exists a control u such that $(x, u, \delta_d) \in \mathcal{Q}_1$ and $x^+ \in \mathcal{X}_i^1$, for all $d_u \in \mathcal{D}_u(u)$.*

Definition 6.3. *The maximal mode 1 RCI set $\mathcal{X}_\infty^1$ is the RCI set which contains all mode 1 RCI sets $\mathcal{X}_i^1$.*

Definition 6.4. *The control mapping $\mathcal{U}_\infty^1(\cdot)$ corresponding to $\mathcal{X}_\infty^1$ is defined as*

$$\mathcal{U}_\infty^1(x, \delta_d) := \left\{u : (x, u, \delta_d) \in \mathcal{Q}_1, x \in \mathcal{X}_\infty^1, x^+ \in \mathcal{X}_\infty^1 \ \forall d_u \in \mathcal{D}_u(u)\right\}. \tag{6.10}$$

If the state x of the vehicle lies in $\mathcal{X}_\infty^1$, then for each $\delta_d \in \mathcal{D}_x(x)$, any choice of input u from $\mathcal{U}_\infty^1(x, \delta_d)$ would cause the predicted state at the next time step to remain in $\mathcal{X}_\infty^1$. However, it is possible for the state of the vehicle to go outside $\mathcal{X}_\infty^1$ due to unmodeled factors such as sudden disturbances, change in surface friction coefficient, etc. In such a scenario, we would like to drive the state of the vehicle back into the mode 1 RCI set in a finite number of time steps. This leads to the notion of N–step backward reachable sets or N–step controllable sets.

Definition 6.5. *The N–step backward reachable sets $\mathcal{X}_N$ to a given target set $\mathcal{X}$ are recursively defined as:*

$$\mathcal{X}_k = Pre(\mathcal{X}_{k-1}), \quad (k = 1, ..., N) \tag{6.11a}$$

$$\mathcal{X}_0 = \mathcal{X}. \tag{6.11b}$$

If the state of the vehicle lies in $\mathcal{X}_N$, there exists a sequence of control inputs $\{u_k\}_{k=0}^{N-1}$ which ensures that the predicted state of the vehicle will lie inside the target set $\mathcal{X}$ in N steps. We can compute $\mathcal{X}_N$ by using Algorithm 6.1 (discussed in Section 6.4.2) to recursively compute $Pre(\mathcal{X}_k), (k = 0, ..., N)$. Figure 6.3 shows the 3–step backward reachable sets to $\mathcal{X}_\infty^1$ for $\mu = 0.3$ and $\dot{x} = 50$ kph. Note that $\mathcal{X}_N$ is non–convex in general as it is a union of convex sets. We now define the control mappings corresponding to the k–step backward reachable sets.

Definition 6.6. *The control mapping $\mathcal{U}_k(\cdot)$ corresponding to $\mathcal{X}_k$ is defined as:*

$$\mathcal{U}_k(x, \delta_d) :=\{u : (x, u, \delta_d) \in \mathcal{Q}, x \in \mathcal{X}_k, x^+ \in \mathcal{X}_{k-1}, \forall d_u \in \mathcal{D}_u(u)\}. \tag{6.12}$$

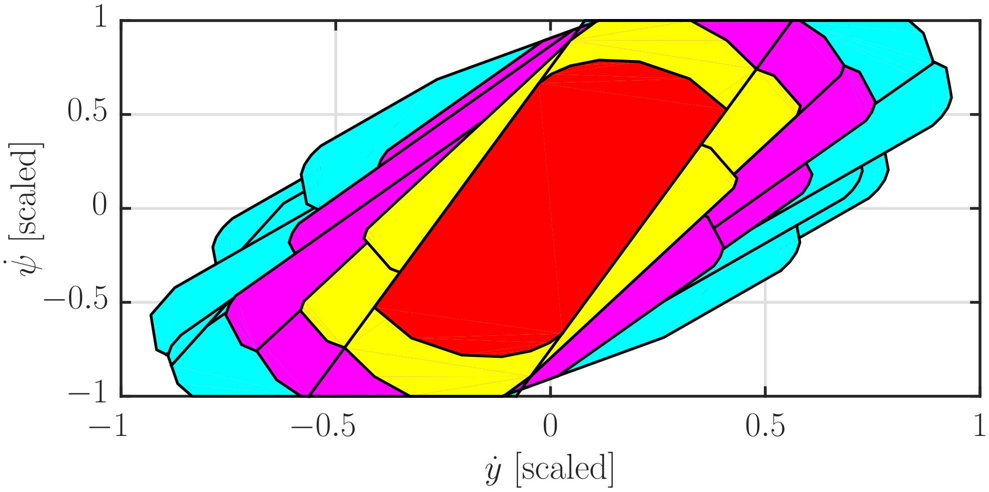


Figure 6.3: 3–step controllable sets to the mode 1 RCI set $\mathcal{X}_\infty^1$, $\mu = 0.3$, $v_x = 50$ kph. The different colors correspond to the various steps. The red set depicts $\mathcal{X}_\infty^1$.

When the state of the vehicle goes outside the RCI set, our control strategy identifies the smallest k such that $x \in \mathcal{X}_k$. For each $\delta_d \in \mathcal{D}_x(x)$, choosing any input from $\mathcal{U}_k(x, \delta_d)$ ensures that the predicted state of the vehicle will lie inside $\mathcal{X}_{k-1}$. Repeating this procedure at every time step ensures that the controller will eventually drive the state back into $\mathcal{X}_\infty^1$ in a finite number of time steps. Note that the analysis guarantees robustness to the uncertainty d_u as well as to the disturbance δ_d so long as the constraints $d_u \in \mathcal{D}_u(u)$ and $\delta_d \in \mathcal{D}_x(x)$ are satisfied.

6.4.2 Reachability analysis of the PWA bicycle model

We apply the algorithm developed in [97] for the robust reachability analysis of the PWA hybrid bicycle model with input and state-dependent disturbances. The algorithm for the computation of the one-step backward reachable set and RCI set can be derived in terms of set operations on the target set. The reader is referred to [97] for details of the proof. It is useful to define the following sets which represent state, input and disturbance constraints, and their projections:

$$\begin{aligned}
\Omega &:= \{(x, u, \delta_d, d_u) : (x, u, \delta_d) \in \mathcal{Q}, \\
&\qquad \delta_d \in \mathcal{D}_x(x), d_u \in \mathcal{D}_u(u)\} \subseteq \mathbb{R}^7, && \text{(6.13a)} \\
\Omega_{x,u,\delta} &:= \mathrm{Proj}_{x,u,\delta}(\Omega), \quad \Omega_{x,\delta} := \mathrm{Proj}_{x,\delta}(\Omega), \\
\Omega_x &:= \mathrm{Proj}_x(\Omega). && \text{(6.13b)}
\end{aligned}$$

The steps for the computation of $Pre(\mathcal{X})$ are given by Algorithm 6.1. Based on this method, the iterative procedure to compute the mode 1 RCI set is given by Algorithm 6.2. Note the use of $Pre^1(\mathcal{X})$ instead of $Pre(\mathcal{X})$ in Algorithm 6.2. This notation implies that we perform

Algorithm 6.1 *Pre* set computation

```
$\Phi(\mathcal{X}) = \{(x, u, \delta_d, d_u) \in \Omega : x^+ \in \mathcal{X}\}$
$\Delta_1 = \Omega \backslash \Phi(\mathcal{X})$,
$\Psi = \text{Proj}_{x,u,\delta}(\Delta_1)$,
$\Sigma_1 = \Omega_{x,u,\delta} \backslash \Psi$,
$\Sigma_2 = \text{Proj}_{x,\delta}(\Sigma_1)$,
$\Delta_2 = \Omega_{x,\delta} \backslash \Sigma_2$,
$\Delta_3 = \text{Proj}_x(\Delta_2)$,
$Pre(\mathcal{X}) = \Omega_x \backslash \Delta_3$
Output: $Pre(\mathcal{X})$
```

Algorithm 6.2 RCI set computation

```
Initialize:
$i = 0$
$\mathcal{X}_i = \mathcal{P}_1$
repeat
    $i = i + 1$
    $\mathcal{X}_i = Pre^1(\mathcal{X}_{i-1}) \cap \mathcal{X}_{i-1}$
until $\mathcal{X}_i == \mathcal{X}_{i-1}$
$\mathcal{X}^1_\infty = \mathcal{X}_i$
Output: $\mathcal{X}^1_\infty$
```

the one–step reachability analysis using the constraints and dynamics associated with mode 1 only. Figure 6.4 shows the output of Algorithm 6.2 for $\mu = 0.3$ and $v_x = 50$ kph. As the model (6.4) is affine, and the state and input constraints are polyhedral, the sets $Pre(\mathcal{X})$ and $\mathcal{X}^1_\infty$ are also polyhedral by construction [103, 97]. The Multi–Parametric Toolbox (MPT) [73] in MATLAB can efficiently perform set operations on polytopes, and hence was used for the reachability analysis and RCI set computations.

6.4.3 Robust control design with reference tracking

The stability control system provides front steering corrections and yaw moment commands in order to track a given reference in the state–space. The reference signals, $r = [\dot{y}^{\text{ref}}, \dot{\psi}^{\text{ref}}]^T$, are a function of the current states and the driver's steering input. We use a standard reference generator for ESC systems [67] which is based on a steady–state cornering analysis of the linear bicycle model. The optimal control input, $u^\star = [\delta_{\text{AFS}}, M]^T$, is computed by minimizing a quadratic cost function as:

$$u^\star = \underset{u \in \mathcal{U}_\star(x, \delta_d)}{\arg\min} \ (x^+ - r)^T Q (x^+ - r) + (u - u^{\text{pre}})^T R (u - u^{\text{pre}}), \tag{6.14}$$

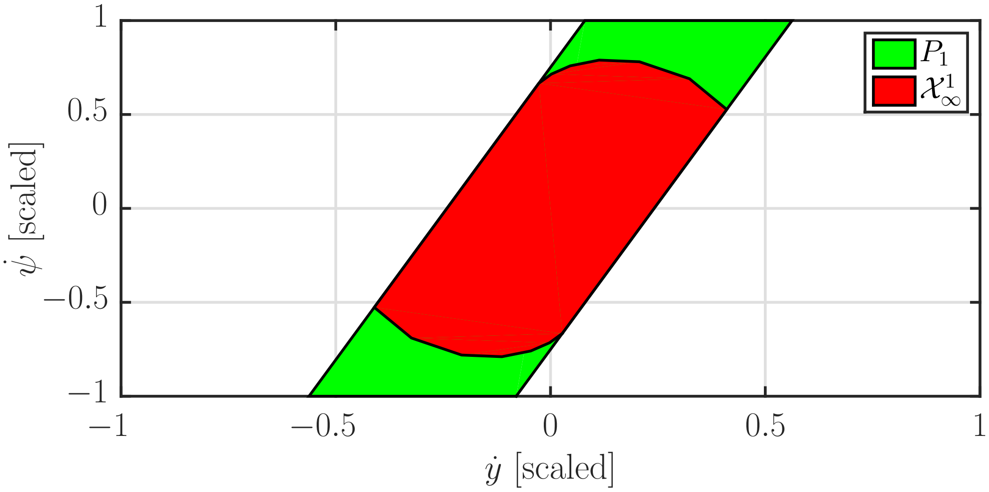


Figure 6.4: Maximal RCI set $\mathcal{X}_\infty^1$, $\mu = 0.3$, $v_x = 50$ kph.

where Q and R are suitably chosen positive definite matrices, x^+ is the predicted state at the next time step given by model (6.4), and u^{pre} is the control input commanded at the previous time step. Depending on the current value of the state x, the set of admissible control inputs $\mathcal{U}_\star(x, \delta_d)$ is given by either (6.10) or (6.12). In particular, if $x \in \mathcal{X}_\infty^1$, then $\mathcal{U}_\star(x, \delta_d) = \mathcal{U}_\infty^1(x, \delta_d)$, otherwise $\mathcal{U}_\star(x, \delta_d) = \mathcal{U}_k(x, \delta_d)$, where k is the smallest positive integer for which $\mathcal{U}_k(x, \delta_d)$ is not empty. Note that the controller is switched based on the polytopes $\mathcal{U}_\infty^1$ and $\mathcal{U}_k$ which are computed offline.

The cost function in (6.14) is used only when the vehicle is in mode 1 or when the front tires are not saturated. If the front tires are saturated, we impose a penalty on the high positive or negative slip angle in order to drive it to that value at which maximum lateral force is obtained. Since the AFS only provides control of the front slip angle, we ignore the rear slip angle in the modified cost function. In this case, we obtain the control input $u^\star$ as the optimizer of the following problem:

$$u^\star = \underset{u \in \mathcal{U}_\star(x,\delta_d)}{\arg\min}\ (x^+ - r)^T Q (x^+ - r) + (u - u^{\text{pre}})^T R (u - u^{\text{pre}}) + P(\alpha_f^+ - \hat{\alpha}_f)^2, \qquad (6.15)$$

where $P > 0$, α_f^+ is the linear approximation of the predicted front slip angle at the next time step given by (3.15) and $\hat{\alpha}_f$ is the value of the front tire slip angle at which the lateral force is maximum. The definition of $\mathcal{U}_\star(x, \delta_d)$ is the same as that for the optimization problem (6.14). The problems (6.14) and (6.15) are quadratic programs as x^+ and α^+ are affine in u, and the sets $\mathcal{U}_\star(x, \delta_d)$ defining the constraints on u are polyhedral. Note that the optimal control input $u^\star$ has two components: i) the front wheel steering angle $\delta_{\text{AFS}}^\star$ and ii) the yaw moment $M_z^\star$. While we can directly set the AFS input to $\delta_{\text{AFS}}^\star$, the yaw moment command must be converted to four individual braking torques that can be applied to the four wheels. The wheel braking torques are computed from $M_z^\star$ using the algorithm presented in [46].

6.5 Experimental Results

6.5.1 Experimental setup

The prototype Jaguar vehicle described in Appendix A.1 was used for the experimental validation of our approach. The sampling time for the experiments was $T_s = 0.05$ s.

6.5.2 Results

We performed several tests on an icy surface ($\mu \approx 0.3$), and present the results from two relevant experiments in this work. In the first experiment, the driver performed a double lane change maneuver with an entry speed of 60 kph and maintained an almost constant position of the accelerator pedal. The tuning parameters are listed in Table 6.1. Note the different sets of parameters used depending on whether the vehicle state lies within the RCI set $\mathcal{X}_\infty^1$ or not.

Table 6.1: Tuning parameters for lateral stability control experiments

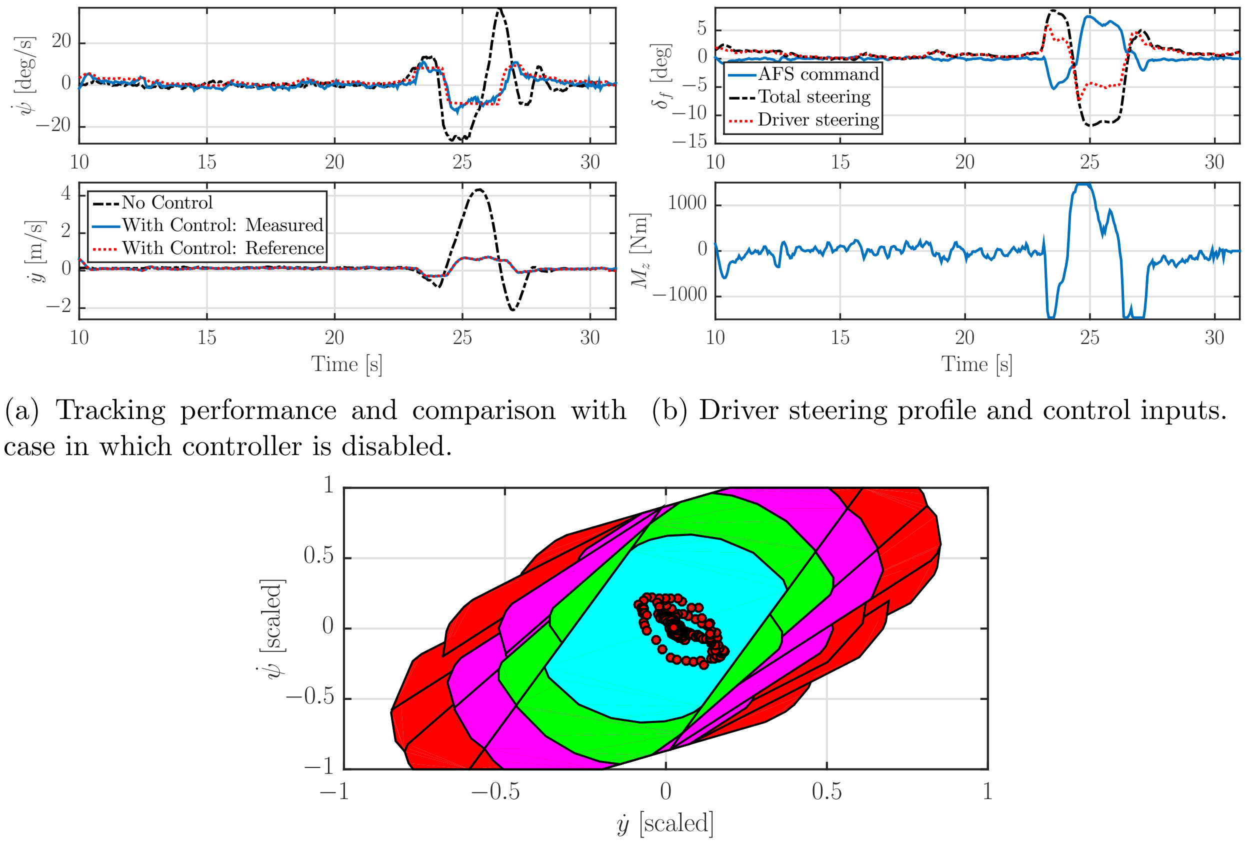
Parameter	**Value**	**Parameter**	**Value**
N	4		
Inside $\mathcal{X}_\infty^1$		**Outside** $\mathcal{X}_\infty^1$	
Q	$\text{diag}\,(0.1, 30)$	Q	$\text{diag}\,(0.1, 10)$
R	$\text{diag}\,(2, 0.1)$	R	$\text{diag}\,(2, 0.1)$
		P	0.07

The vehicle response is shown in Figures 6.5a–6.5c. In Figure 6.5a, we show the performance of the controller in tracking yaw rate and lateral velocity signals. We also compare the vehicle states with the controller being active to the states when the active safety system is disabled. We obtain a satisfactory tracking performance and are able to prevent the vehicle from attaining high values of lateral velocity and yaw rate.

In the upper plot of Figure 6.5b, we plot the driver's steering input, the AFS command and the measured total front wheel steering angle. The controller imposes a counter–steering correction which results in vehicle stabilization. The AFS command is opposite to the driver's input in order to generate a counter–yaw moment to avoid a spin–out. In addition, the braking moment command supports this action as shown in the bottom part of Figure 6.5b.

Figure 6.5c depicts the evolution of the state trajectory superimposed on the 3–step backward reachable set which were computed offline. The controller is able to compute an input command such that the vehicle states always lie in $\mathcal{X}_\infty^1$.

The second experiment was performed on an icy circular track ($\mu \approx 0.2$) of diameter 110 m at a speed of approximately 40 kph. The aim of this test was to verify the controller action *i*) when the vehicle simultaneously experiences front and rear tire saturation and *ii*) when



(a) Tracking performance and comparison with case in which controller is disabled.

(b) Driver steering profile and control inputs.

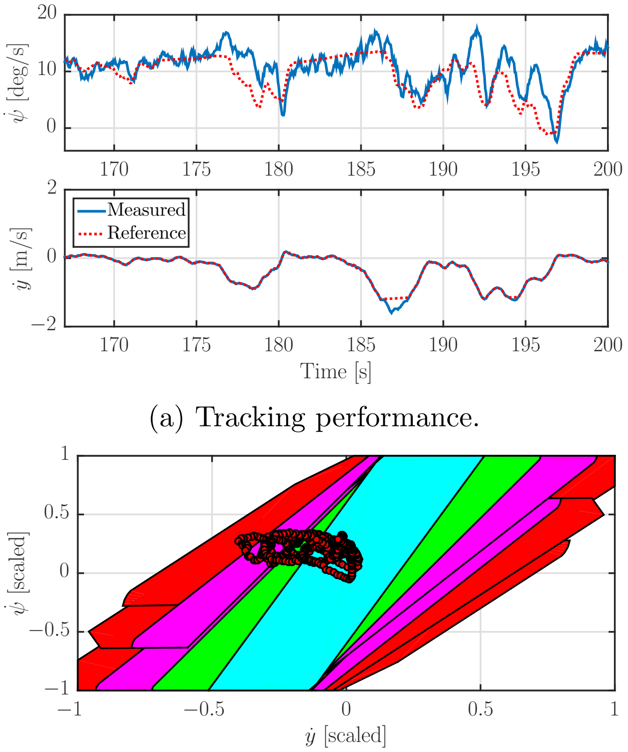
(c) Experiment 1: Evolution of state trajectory. The states always stay inside $\mathcal{X}_\infty^1$.

Figure 6.5: Experiment 1: Double lane change maneuver with an entry speed of 60 kph.

model mismatch occurs as a result of any of the assumptions of known friction coefficient (Assumption 3.6) and constant speed (Assumption 3.7) not being satisfied. In particular, our control model assumes a friction coefficient of 0.3, which is different from the actual friction coefficient observed in this experiment. The tuning parameters are the same as those used in the first experiment. The vehicle response is shown in Figures 6.6a–6.7b.

Figure 6.6b shows the state trajectory superimposed on the 3–step controllable set. Although the state of the vehicle goes outside the mode 1 RCI set $\mathcal{X}_\infty^1$, the controller is able to bring the state back into $\mathcal{X}_\infty^1$ within a short time. In Figure 6.6a, we note that the tracking performance is affected by the excursion of the state outside $\mathcal{X}_\infty^1$. The main reason for the relatively poor tracking in such a situation is that the reference trajectory is generated using a steady–state cornering analysis.

We illustrate the effectiveness of using the modified cost function (6.15)) in Figures 6.7a and 6.7b. In the top part of Figure 6.7a, we plot the driver's steering input, the AFS command and the measured total front wheel steering angle. Figure 6.7b shows that around



(a) Tracking performance.

(b) Evolution of state trajectory: state goes outside $\mathcal{X}_{\infty}^{1}$ during circular maneuver.

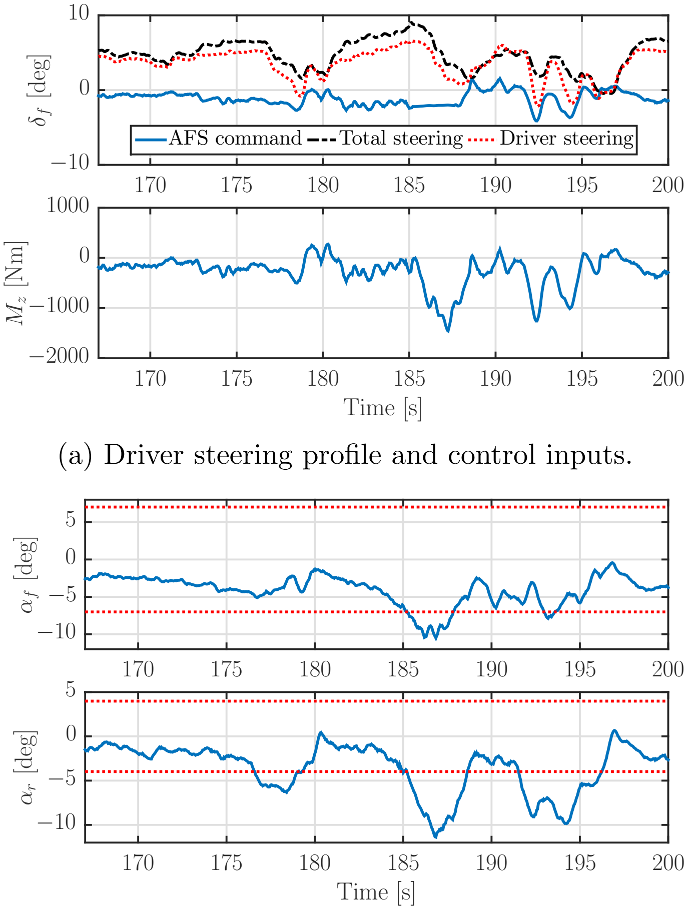
Figure 6.6: Experiment 2: Driving on icy circular track ($\mu \approx 0.2$) of diameter 110 m at a speed of approximately 40 kph.

$t = 185$ s, the absolute values of both, the front and rear tire slip angles become greater than $\hat{\alpha}_f$ and $\hat{\alpha}_r$, respectively. In order to stabilize the vehicle in this case, the controller generates a negative yaw moment by the combination of a negative AFS command and a negative yaw moment command, as desired. This causes the front tire slip angle to return to the linear region at $t = 187$ s.

6.6 Conclusions

In this chapter, we presented the design of a robust vehicle stability controller which uses AFS and differential braking. The control approach explicitly considers the driver's intent in the control design process, taking into account the uncertainty in both, the driver behavior and the commanded input. The experimental results illustrate the effectiveness of the proposed controller, which is able to guarantee convergence into the maximal RCI set despite model mismatch and the presence of state and input constraints.

While the presented application involves the driver in the loop, the method is general and



(a) Driver steering profile and control inputs.

(b) Computed front (top) and rear (bottom) slip angles. The dotted line denotes $\hat{\alpha}_\star$.

Figure 6.7: Experiment 2: Driving on icy circular track ($\mu \approx 0.2$) of diameter 110 m at a speed of approximately 40 kph.

can be applied to the autonomous driving case as well. In fact, this simplifies the control design as the state-dependent disturbance δ_d (that is, the driver's input) is zero. In this case, the set-valued mapping $\mathcal{U}_\star(\cdot)$ (defined in Section 6.4.3) is only a function of the vehicle state x^v. The goal of the autonomous controller is to choose a control input u_t^v from the set $\mathcal{U}_\star(x_t^v)$ at time t to guarantee the lateral stability of the vehicle. The vehicle state x_t^v, the system dynamics (6.4) and the cost functions in (6.14)–(6.15) would have to be appropriately modified to encode additional objectives such as lane keeping.

Chapter 7

Safe Personalized Autonomous Cruise Control

7.1 Introduction

Autonomous or Adaptive Cruise Control (ACC) systems were introduced in the automotive industry in the early 1990s and are present in many passenger vehicles today [119]. By controlling the speed of the ego vehicle to match a desired reference such as the road speed limit and adapting the speed to that of the car in front, these systems help prevent rear-end collisions and relieve the driver of a tedious task. Moreover, they can easily be combined with an autonomous lane-keeping system such as the one presented in Chapter 5 to provide full autonomy on highways. In this chapter, we focus on three aspects of ACC which are detailed below.

Guaranteed Safety

The main requirement of ACC systems is to prevent collisions with the vehicle in front of it (referred to as the *preceding vehicle*) or target vehicles which may cut-in to its path. The problem of guaranteeing safety is complicated by the fact that the preceding vehicle's future motion is not known. Common approaches for designing safety constraints in car-following applications are based on certain assumptions about the preceding vehicle's behavior such as constant velocity or constant acceleration. The predicted positions and velocities of the preceding vehicle are then used to compute spacing constraints based on, for example, a desired time-to-collision (see e.g. [11, 81]). While such approaches perform well in practice, they do not guarantee safety for all possible actions of the preceding vehicle. For instance, a situation in which the preceding vehicle brakes suddenly to come to a stop is likely to result in a rear end collision. In the context of vehicle platooning, the authors in [7] construct a set of states from which there exists a safe control law such that collisions with the leader platoon are avoided for all possible maneuvers of the leader. We aim to construct a similar safe set based on the notion of robust reachability analysis. In [127], the *envelope of opportunity* for drivers in car-following situations is analyzed, that is, when and how drivers must brake to avoid rear-end collisions. Our goal is to incorporate such envelopes directly as constraints in the optimal control problem for safe decision making.

Personalization

In addition to safety, ACC systems and fully autonomous cars in general must generate smooth control actions for the comfort of their passengers. However, the notion of comfort varies greatly from driver to driver. Current ACC systems are based on *constant time-headway* policies where the desired speed of the vehicle is proportional to the distance to the vehicle in front [65]. The time-headway is defined as the ratio of the relative distance between the ego and preceding vehicles to the speed of the ego vehicle. In addition to knowing the driver's desired time-headway, a challenge for ACC systems is designing controllers that can achieve this headway with an acceleration profile which feels comfortable to the driver. For example, some drivers may like to aggressively close the gap between themselves and the preceding vehicle while others prefer using a low value of acceleration. Current systems are based on extensive tuning to achieve a comfortable behavior which is hard to customize to drivers' preferences. We propose using a learning-based approach to learn a model of the driver's longitudinal behavior from driving data collected from that particular driver. This requires minimal tuning. Moreover, in combination with the robust controller, collisions are prevented if the expected driver's actions are not safe for the given scenario.

Several approaches for learning driving styles of individual drivers have been presented in the literature. *Parametric* approaches fix the structure (functional form) of the model and identify the corresponding parameters from data. Examples include the constant velocity and acceleration models, the Intelligent Driver Model (IDM) [117] and the SUMO simulator model [13]. *Non-parametric* methods, on the other hand, learn the model structure and parameters from data. In this class, the frameworks of Artificial Neural Networks (ANNs) [100], imitation learning using dataset aggregation [104] and Gaussian Mixture Regression [22, 23] have been used to learn control actions for robots from demonstrations by a human. Inverse Optimal Control (also known as Inverse Reinforcement Learning) techniques instead learn a representation of the cost function from which humans derive their control actions and are applied to the driving problem in [1, 2, 80, 72]. A comparison of parametric and non-parametric approaches in [77] shows that the latter outperforms the former for long-term predictions of driver behavior. We use a combination of Hidden Markov Models (HMMs) and Gaussian Mixture Regression (GMR) in our work, which was presented in [74] for the longitudinal control task.

Multi-lane target vehicles

Current ACC systems use a combination of on-board radar and camera to identify a *Primary Target (PT)* - the car in front of the ego vehicle in its own lane. The relative distance and speed of the PT are then passed to the control algorithm, which determines an acceleration command for the ego vehicle such that a safe following distance is maintained. If a PT is not detected, the controller tracks a desired speed such as the road speed limit.

A challenging scenario for current ACC systems is to deal with target vehicles in neighboring lanes cutting in to the lane of the ego vehicle. The authors in [91] recognize the vehicle cutting in as the PT when the Time to Lane-Crossing (TLC) of the target vehicle is within a threshold. In [90], a fuzzy logic approach is used to compute the probability of a target vehicle entering into the ego vehicle's lane. These approaches are rule-based and not robust to noisy measurements. Moreover, they assume the existence of a single PT. In the case of multiple target vehicles, existing approaches choose the most relevant one based on some heuristics.

The authors in [108] develop a model to estimate the probability that a vehicle in the neighboring lane cuts-in to the lane of the ego vehicle. Their approach is based on the headway times of the vehicle changing lanes with respect to the vehicles surrounding it. A drawback of this approach is that drivers' tendencies to initiate lane changes vary significantly. Specifically, for a given traffic situation, some drivers may attempt to change lanes while others may not.

Probabilistic approaches for detecting lane changes of target vehicles are proposed in [36, 106]. In [36], a probabilistic network is constructed where the input nodes are an extensive set of features extracted from the driving scene and include inter-vehicle relationships. A combination of Gaussian functions of the input nodes and thresholding is used to identify the lane change intent. The work in [106] identifies key features for lane change prediction of target vehicles starting from a larger feature set. A naive Bayes classifier is used to compute the lane change intent. In our work, we use the learning-based intent estimation algorithm presented in Chapter 4 (Section 4.3), which was shown to be effective for the intent prediction of the ego vehicle [78]. We apply this method to the prediction of target vehicle intent where only a limited set of measurements or features of target vehicles can be reliably estimated using existing sensors. Our goal is to study the effect of cut-in detection of surrounding vehicles on the performance of the proposed ACC system.

Summary of contributions

In this chapter, we focus on advanced autonomous cruise control systems with the following features:

1. The personalization of the car-following behavior of the autonomous vehicle using data collected from the driver. A learning-based approach is used to build a model of the driver's longitudinal acceleration given the history of driving situations.

2. Identification of relevant target vehicles in the scene for the purpose of ACC based on their lane change probabilities. The probability of a target vehicle changing lanes is estimated by a learning-based method trained on real driving data.

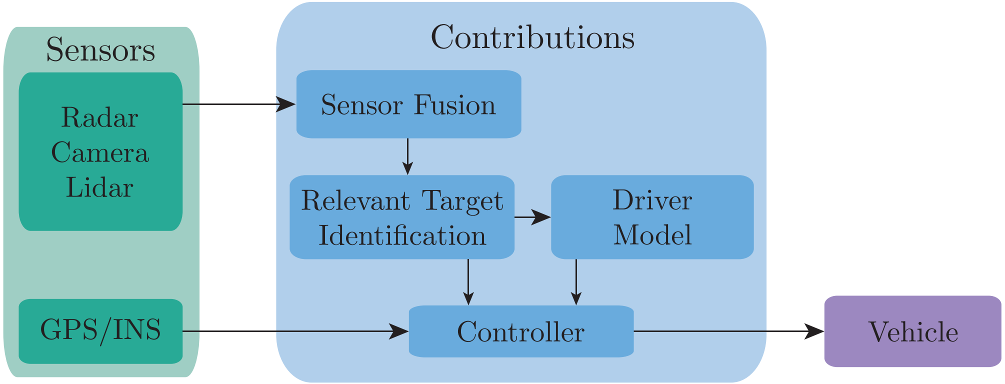


Figure 7.1: Autonomous Cruise Control (ACC) system architecture as described in Section 7.1.

3. The design of a controller based on robust predictive control which accounts for the uncertainty in the future motion of the relevant target vehicles. This allows the autonomous vehicle to be safe even under aggressive braking maneuvers by the vehicles in front of it. The controller can handle multiple vehicles without the need for heuristics to choose the most safety-critical one.

The effectiveness of the proposed approach is demonstrated by offline simulations using data collected from our test vehicle and by real-time experiments on highways.

A schematic of the system architecture is shown in Figure 7.1. The "Sensor Fusion" module combines the measurements from the sensors such as radar, camera, lidar and GPS/INS to provide estimates of the relative positions and velocities of target vehicles (TVs) in front of the ego vehicle (EV). The "Relevant Target Identification" module computes lane change probabilities of the detected TVs and identifies relevant TVs for ACC. These are denoted as *Relevant Targets* (RTs) as opposed to a single PT in the case of conventional ACC systems. Based on an assessment of the *Most Relevant Target* (MRT), the "Driver Model" generates a reference acceleration for the "Controller" which computes the desired acceleration command for the "Vehicle".

The remainder of the chapter is organized as follows. We first present the design of the robust controller and the relevant target identification in Section 7.2. The driver model used for the personalization of the ACC system is presented in 7.3. Off-line simulation results using real data and real-time experimental results are presented in Sections 7.4 and 7.5, respectively, to demonstrate the effectiveness of the proposed approach. Finally, concluding remarks are made in Section 7.6.

Note on sensor fusion

While the sensor fusion algorithm is not a focus of this dissertation, it is a key element for the robust detection of objects in front of the ego vehicle. Our sensor fusion approach was developed by the author in [126] and is summarized below.

The experimental vehicle used for this application (Vehicle B described in Appendix A) is equipped with a radar, camera and lidar. Each sensor reports the relative positions and velocities of the TVs in front of the EV with respect to its body fixed coordinate frame. The main challenge of sensor fusion is data association, that is, deciding which measurements from the three sensors originate from the same TV. In our case, the individual sensors perform their own tracking, so we use a globally optimal probabilistic track-to-track association algorithm to determine which tracks pertain to the same real-world object. Based on the work of [37], we formulate the problem as a variation of the 3-dimensional assignment problem. Measurements from each of the three sensor tracks are grouped into triples to represent the hypothesis that those measurements have a common origin. The output of the algorithm is the optimal assignment which minimizes a chosen cost function. The cost of a triple is defined to be the negative log of the ratio of the probability that all three measurements in the triple pertain to the same object to the probability that all three measurements are spurious. The cost function is then the sum of the costs of all possible triples. The resulting optimization problem is a Mixed Integer Linear Program (MILP) which can be solved online. We use the open-source solver ECOS for solving the optimization problem [40].

A fused measurement of each associated TV's states is obtained by a convex combination of the measurements from each sensor. Additionally, a Kalman filter is used to estimate the relative positions and velocities of the TVs with respect to the EV.

7.2 Robust Control Design

In this section, we present the design of the robust predictive controller. First, the analysis with a single RT (known as the preceding vehicle) is presented. It is extended for the case of multiple RTs.

7.2.1 Vehicle models

Ego vehicle

The longitudinal point-mass model introduced in Section 3.4 is used to model the ego vehicle motion. Compactly, the linear time-invariant (LTI) model is written in state-space form as:

$$x_{t+1}^v = Ax_t^v + Bu_t^v, \tag{7.1}$$

with $x_t^v = [\xi_t, v_t]^T$ and $u_t^v = a_t$.

Remark 7.1 (Actuator dynamics). *As noted in Section 3.5, it is straightforward to include the acceleration lag in the ego vehicle model* (7.1). *For the sake of clarity, we do not explicitly consider the actuator dynamics in the subsequent controller synthesis. The analysis is general as the LTI model structure* (7.1) *is retained even when the acceleration lag model* (3.31) *is included in the system dynamics. The main difference is in the increase in the dimension of the state vector by one element.*

Preceding vehicle motion

The motion of the preceding vehicle is also assumed to be governed by the model which describes the ego vehicle motion. That is:

$$x^e_{t+1} = Ax^e_t + Bd^e_t, \tag{7.2}$$

with $x^e_t = [\xi^e_t, v^e_t]^T$ and $d^e_t = a^e_t$. As the future behavior of the preceding vehicle is not known, the proposed RMPC scheme assumes the acceleration d^e_t of the preceding vehicle at time t to be a disturbance whose bounds are described as:

$$d^e_t \in \mathcal{D} := \{d : a^e_{\min} \leq d \leq a^e_{\max}\}, \tag{7.3}$$

where $a^e_{\min}$ and $a^e_{\max}$ are the estimated minimum and maximum accelerations, respectively, of the preceding vehicle. The goal of the RMPC design is to satisfy the safety constraints discussed below for all $d^e_t \in \mathcal{D}$.

7.2.2 Safety constraints

The safety requirement of avoiding collisions for all possible realizations of the disturbance d^e_t in the set $\mathcal{D}$ is enforced by the following constraint on the relative distance between the preceding and ego vehicles over the prediction horizon:

$$\xi^e_{k|t} - \xi_{k|t} \geq d_{\text{safe}} \quad \forall d^e_{j|t} \in \mathcal{D} \quad \forall j = 0, \ldots, k-1, \tag{7.4}$$

where d_{safe} is the minimum safe following distance.

Due to the simple nature of the dynamics (7.2), disturbance bounds (7.3) and relative distance constraints (7.4), robust satisfaction of (7.4) can be achieved by assuming that the disturbance $d^e_{k|t}$ takes on its lower bound $a^e_{\min}$ at every time step in the prediction horizon. Intuitively, if the preceding vehicle actually accelerates at a value greater than $a^e_{\min}$, the relative distance will be greater than that computed with the worst-case value $a^e_{\min}$. Hence, (7.4) will be satisfied for all possible values of the disturbance, as is shown formally below.

The predicted worst-case states of the preceding vehicle, denoted by $\bar{x}^e_{k|t} = [\bar{\xi}^e_{k|t}, \bar{v}^e_{k|t}]$, evolve as:

$$\bar{x}^e_{k+1|t} = A\bar{x}^e_{k|t} + Ba^e_{\min}. \tag{7.5}$$

In addition, we constrain the speed of the preceding vehicle to be non-negative. The predicted positions $\bar{\xi}^e_{k|t}$ from (7.5) are used to formulate the safety distance constraints over the prediction horizon:

$$\bar{\xi}^e_{k|t} - \xi_{k|t} \geq d_{\text{safe}}. \tag{7.6}$$

From the system dynamics in (7.5), it is easy to show that:

$$\bar{\xi}^e_{k|t} \leq \xi^e_{k|t} \quad \forall d^e_{j|t} \in \mathcal{D} \quad \forall j = 0, \ldots, k-1. \tag{7.7}$$

Hence, satisfaction of (7.6) ensures that (7.4) is satisfied.

In addition to maintaining a safe following distance to the preceding vehicle, the controller must ensure that the EV respects the speed limit:

$$v_{k|t} \leq v_{\max}, \tag{7.8}$$

where $v_{\max}$ is the speed limit of the road. The state constraints (7.6) and (7.8) are concisely expressed as:

$$g(x^v_{k|t}, \bar{x}^e_{k|t}) \leq 0. \tag{7.9}$$

In addition, we introduce constraints on the control input (defined in (3.34)):

$$[u^v_{k|t}, u^v_{k-1|t}] \in \mathcal{U} \tag{7.10}$$

7.2.3 Persistent feasibility

In general, there is no guarantee that the safety constraints (7.9) and (7.10) will be satisfied in closed-loop. This problem of persistent (or recursive) feasibility is well studied in the literature (see [85] for a survey). In RMPC, persistent feasibility can be ensured by introducing a suitably chosen terminal set in which the system state at the end of the horizon is constrained to lie. A sufficient condition for recursive feasibility is that the terminal set is a Robust Control Invariant (RCI) set, which were introduced in Section 6.4 and defined below for the current application:

Definition 7.2. *Consider the system $x_{t+1} = f(x_t, u_t, d_t)$, where the state $x_t \in \mathcal{X}$, the input $u_t \in \mathcal{U}$ and the disturbance $d_t \in \mathcal{D}$. A set $\mathcal{R} \subseteq \mathcal{X}$ is said to be an RCI set for the system if:*

$$x_t \in \mathcal{R} \implies \exists u_t \in \mathcal{U} \text{ s.t. } f(x_t, u_t, d_t) \in \mathcal{R} \quad \forall d_t \in \mathcal{D}. \tag{7.11}$$

Remark 7.3. *In the absence of disturbances in the system, the corresponding set is called a Control Invariant (CI) set.*

Definition 7.4. *The RCI (resp. CI) set which contains all other RCI (resp. CI) sets is called the maximal RCI (resp. CI) set.*

An RCI set is the set of states from which there exists a feasible input such that the state at the next time step lies within the set for all possible values of the disturbance. Computing an RCI set for the car-following problem is not trivial due to the fact that the motion of the preceding vehicle cannot be controlled. The set of reachable states corresponding to the model (7.5) given the disturbance bounds (7.3) includes negative speeds of the preceding vehicle. Hence, standard methods for computing the RCI set for linear systems (see [17] for a survey) would result in a conservative or empty RCI set. Theoretically, the preceding vehicle behaves as a switched (or hybrid) system where the permissible values of the acceleration are such that the vehicle does not attain negative speeds. This, however, significantly complicates the computation of the RCI set. Moreover, the resulting set usually consists of a union of convex sets, and is non-convex. In this work, we propose a method of computing a polyhedral (hence, convex) terminal constraint for the RMPC problem.

We use the same assumption about the future behavior of the preceding vehicle as is used in the design of the safety constraints (7.6). Starting from its state $x^e_t = [\xi^e_t, v^e_t]$ at time instant t, the preceding vehicle is assumed to apply maximum braking. Let $(t + k_s)$ denote the time at which the preceding vehicle comes to a stop. The model (7.5) allows us to compute the predicted worst-case states $\{\bar{x}^e_{k|t} = [\bar{\xi}^e_{k|t}, \bar{v}^e_{k|t}]^T\}^{k_s}_{k=0}$. Note that $\bar{v}^e_{k|t} = 0$ for $k \geq k_s$. We will show that the worst-case assumption is sufficient to compute the required RCI set.

The computation of the RCI terminal set is broken down into two steps. In the first step, we compute the maximal CI set (denoted as $\mathcal{X}_{ep}$) for the predicted ego and preceding vehicle states at time $(t + k_s)$. This is the set of safe states for the ego vehicle assuming that the preceding vehicle has already come to a stop. The second step is to compute the set $\mathcal{X}_T$ of states $[x^v_{T|t}, \bar{x}^e_{T|t}]$ at the end of the controller prediction horizon that can be driven into $\mathcal{X}_{ep}$ in $(k_s - T)$ number of time steps. This ensures the infinite horizon safety of the terminal vehicle state $x^v_{T|t}$. Concretely, starting from $x^v_{T|t}$, the controller can bring the state into $\mathcal{X}_{ep}$ in a finite number of steps, and once there, it can keep the state in $\mathcal{X}_{ep}$ by the definition of the maximal CI set. The formal definitions of the sets $\mathcal{X}_{ep}$ and $\mathcal{X}_T$ are introduced below.

The maximal CI set $\mathcal{X}_{ep}$ for the predicted ego and preceding vehicle states at time $(t+k_s)$ is defined as follows:

$$\begin{aligned}\mathcal{X}_{ep} = \{[x^v_{k_s|t},\ \bar{x}^e_{k_s|t}] \mid \exists u^v_{k_s|t} \in \mathcal{U} \text{ s.t. } [x^v_{k_s+1|t},\ \bar{x}^e_{k_s+1|t}] \in \mathcal{X}_{ep},\ x^v_{k_s+1|t} = Ax^v_{k_s|t} + Bu^v_{k_s|t},\\ \bar{x}^e_{k_s+1|t} = \bar{x}^e_{k_s|t},\ g(x^v_{k_s|t}, \bar{x}^e_{k_s|t}) \leq 0\}.\end{aligned} \tag{7.12}$$

The absence of the worst-case disturbance $a^e_{\min}$ in (7.12) is due to the fact that $\bar{v}^e_{k|t} = 0$ for $k \geq k_s$. Therefore, the preceding vehicle cannot decelerate further. This also allows us to set $\bar{x}^e_{k_s+1|t} = \bar{x}^e_{k_s|t}$ in (7.12). Note that $\mathcal{X}_{ep}$ is a polyhedron as the system dynamics are linear in the states and inputs and the safety constraints are defined by a set of linear inequalities.

Remark 7.5. *The input constraint set $\mathcal{U}$ in* (7.12) *only includes the bounds on the control input u^v_k defined in* (3.33a) *and does not include the rate constraints* (3.33b)*. Later, in Section 7.2.4, we will show how the analysis can be extended to include the aforementioned rate bounds.*

We use the Multi-Parametric Toolbox (MPT) in MATLAB is used to compute $\mathcal{X}_{ep}$ [59]. The projection of $\mathcal{X}_{ep}$ in the ξ^r-v space is shown in Figure 7.2, where $\xi^r = \xi^e - \xi$ is the relative distance between the preceding and ego vehicles. As expected, the maximum admissible ego speed increases with the relative distance. The black line depicts the direction of unboundedness of $\mathcal{X}_{ep}$. Recall that $\mathcal{X}_{ep}$ is a polyhedron although it does not appear so from Figure 7.2 at first glance. This is due to the fact that the approximate analytical expression for the left

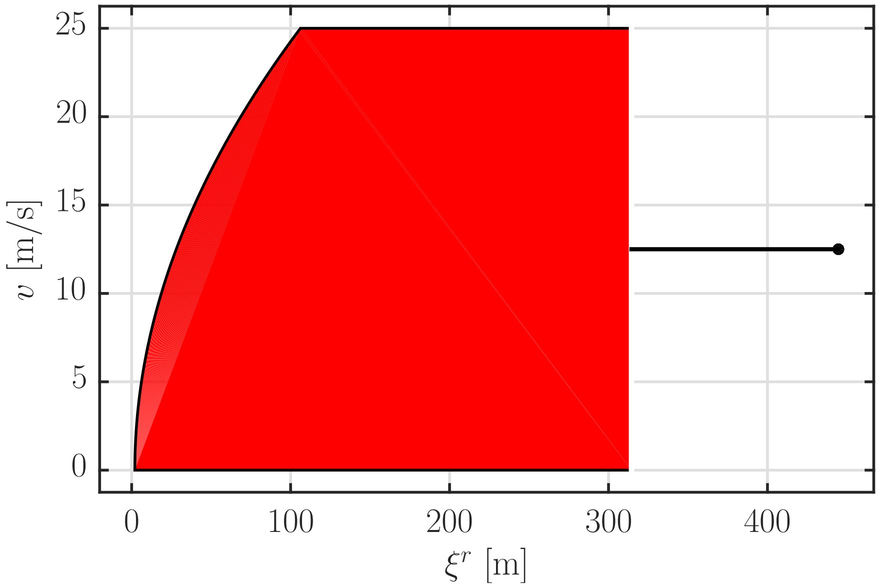


Figure 7.2: Projection of the maximal CI set $\mathcal{X}_{ep}$ in the ξ^r-v space for $d_{\text{safe}} = 5$ m and $v^{\max} = 25$ m/s. The black line depicts the direction of unboundedness of $\mathcal{X}_{ep}$.

boundary of $\mathcal{X}_{ep}$ is quadratic in the speed v of the ego vehicle as will be shown later in Section 7.2.5. The analytical derivation is based on the continuous-time representation (3.29) of the longitudinal dynamics whereas the above analysis uses the discretized representation (3.30). This results in the quadratic boundary being approximated by a set of linear inequalities.

Recall that the next step is to compute the set $\mathcal{X}_T$ of terminal states $[x^v_{T|t}, \bar{x}^e_{T|t}]$ that can be driven into $\mathcal{X}_{ep}$ in $(k_s - T)$ number of time steps. The computation of the terminal set $\mathcal{X}_T$ is based on the notion of backward reachable sets, which were introduced in Section 6.4 and are defined below for the current application.

Definition 7.6. *Consider the system $x_{t+1} = f(x_t, u_t)$, where the state $x_t \in \mathcal{X}$, and the input $u_t \in \mathcal{U}$. The one-step backward reachable set to a given target set $\mathcal{T} \subseteq \mathcal{X}$ is defined as:*

$$Pre(\mathcal{T}) = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U} \ s.t. \ f(x,u) \in \mathcal{T}\}. \tag{7.13}$$

Definition 7.7. *The N-step backward reachable sets $\mathcal{R}_N$ to a given target set $\mathcal{T}$ are recursively defined as:*

$$\mathcal{R}_k = Pre(\mathcal{R}_{k-1}) \quad (k = 1, \dots, N), \quad \mathcal{R}_0 = \mathcal{T}. \tag{7.14}$$

For our problem, the set $Pre(\mathcal{T})$ for a given target set $\mathcal{T}$ takes the form:

$$Pre(T) = \{[x^v_{k|t},\ \bar{x}^e_{k|t}] \mid \exists u^v_{k|t} \in \mathcal{U} \text{ s.t. } [x^v_{k+1|t},\ \bar{x}^e_{k+1|t}] \in \mathcal{T},\ x^v_{k+1|t} = Ax^v_{k|t} + Bu^v_{k|t},$$

$$\bar{x}^e_{k+1|t} = A\bar{x}^e_{k|t} + Ba^e_{\min},\ g(x^v_{k|t}, \bar{x}^e_{k|t}) \leq 0\}. \tag{7.15}$$

Note the use of the worst-case disturbance $a^e_{\min}$ in (7.15). This is due to the assumption that the acceleration of the preceding vehicle takes on its minimum value $a^e_{\min}$ from time instant $(t+T)$ to $(t+k_s)$.

We now state and prove the following theorem which allows us to compute a suitable terminal constraint for the RMPC problem.

Theorem 7.8. *Let $\mathcal{X}_T = \mathcal{R}_{k_s-T}$, where $\mathcal{R}_{k_s-T}$ is the (k_s-T)-step backward reachable set to $\mathcal{X}_{ep}$. The RMPC problem is persistently feasible with respect to the safety constraints* (7.10) *and* (7.9) *if the terminal constraint $[x^v_{T|t}, \bar{x}^e_{T|t}] \in \mathcal{X}_T$ is introduced.*

Proof: Let $[x^v_{T|t}, \bar{x}^e_{T|t}] \in \mathcal{X}_T = \mathcal{R}_{k_s-T}$ as defined above. From (7.14) and (7.15), there exists a feasible sequence of control inputs $\{a^v_{k|t}\}^{k_s-1}_{k=T}$ such that the sequence of states $\{[x^v_{k|t}, \bar{x}^e_{k|t}]\}^{k_s}_{k=T+1}$ satisfies the state constraints (7.9) and $[x^v_{k_s|t}, \bar{x}^e_{k_s|t}] \in \mathcal{X}_{ep}$ for $d^e_{k|t} = a^e_{\min}$ $(k = T, \ldots, k_s-1)$. As $a^e_{\min}$ is the worst-case acceleration of the preceding vehicle, the above statement holds for all possible $d^e_{k|t} \in \mathcal{D}$ $(k = T, \ldots, k_s-1)$. Moreover, by (7.12), $[x^v_{k_s|t}, \bar{x}^e_{k_s|t}] \in \mathcal{X}_{ep}$ implies that for $k \geq k_s$, there exists a feasible control input $a^v_{k|t}$ which keeps the ego vehicle safe. In summary, the satisfaction of the constraints (7.9) and the terminal constraint is guaranteed in closed-loop for all possible values of $d^e_{k|t} \in \mathcal{D}$ $(k \geq 0)$. □

As the backward reachability analysis in (7.14) is performed in discrete-time, the k-step backward reachable set $\mathcal{R}_k$ to the target set $\mathcal{X}_{ep}$ corresponds to a specific value of $\bar{v}^e$. In particular, $\mathcal{R}_k$ corresponds to $\bar{v}^e = -a^e_{\min}k\Delta t^v$. Figure 7.3 shows the projection of the sets $\mathcal{R}_k$ corresponding to the target set $\mathcal{X}_{ep}$ in the ξ^r-v^e-v space for $k = 10, 20, 30, 40, 50$. Each $\mathcal{R}_k$ is a two-dimensional plane in the ξ^r-v space. In order to account for continuous values of $\bar{v}^e$, we set:

$$X_T = \mathcal{R}_{k^\star}, \tag{7.16}$$

where:

$$k^\star = \max_{k=1,2,\ldots} k \ \text{ s.t. } \bar{v}^e_{T|t} \geq -a^e_{\min}k\Delta t^v. \tag{7.17}$$

Intuitively, for a given value of $\bar{v}^e_{T|t}$, we choose a k-step backward reachable set corresponding to a value of $\bar{v}^e$ less than $\bar{v}^e_{T|t}$. Persistent feasibility is still guaranteed as the above choice is conservative.

Remark 7.9. *If $k_s \leq T$, $\mathcal{X}_T = \mathcal{X}_{ep}$ is sufficient to guarantee persistent feasibility. This is the case if the preceding vehicle is expected to come to a full stop with maximum deceleration $a^e_{\min}$ within the prediction horizon of the controller.*

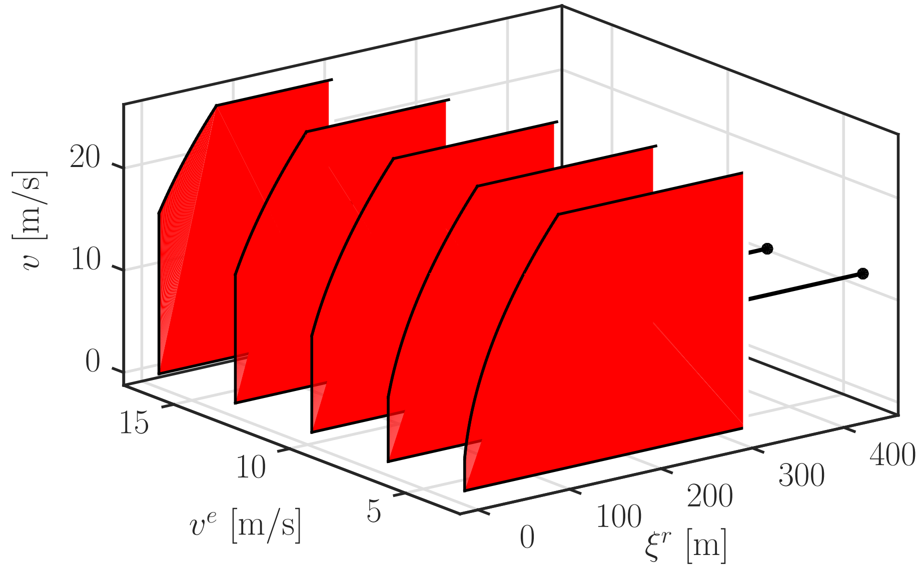


Figure 7.3: Projection of the sets $\mathcal{R}_k$ corresponding to the target set $\mathcal{X}_{ep}$ in the ξ^r-v^e-v space for $k = 10, 20, 30, 40, 50$. Each two-dimensional slice in the ξ^r-v space corresponds to a particular value of v^e.

7.2.4 Dealing with actuator rate constraints

We now extend the above analysis to include the input rate constraints defined in (3.33b). We define the auxiliary control input $\Delta u^v_{k|t} = u^v_{k|t} - u^v_{k-1|t}$. Note that $u^v_{-1|t}$ is initialized to the input u^v_{t-1} commanded at the previous time step. Further, the state vector is augmented to include the control input at the previous time step. Concretely, we define the augmented state vector as:

$$\tilde{x}^v_{k|t} = \begin{bmatrix} x^v_{k|t} \\ u^v_{k-1|t} \end{bmatrix}. \tag{7.18}$$

The corresponding system dynamics are given by:

$$\tilde{x}^v_{k+1|t} = \widetilde{A}\tilde{x}^v_{k|t} + \widetilde{B}\tilde{u}^v_{k|t}, \tag{7.19}$$

with:

$$\tilde{u}^v_{k|t} = \Delta u^v_{k|t}, \tag{7.20a}$$

$$\widetilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \qquad \widetilde{B} = \begin{bmatrix} B \\ I \end{bmatrix}. \tag{7.20b}$$

Expressing the dynamics in the standard form of a LTI system allows us to employ the terminal set computations developed in Section 7.2.3 with the following modifications:

1. As the augmented state vector $\tilde{x}^v_{k|t}$ now contains the control input $u^v_{k-1|t}$, the modified state constraints $g(\tilde{x}^v_{k|t}, \bar{x}^e_{k|t})$ include the actuator bounds (3.33a).

2. As the modified control input $\tilde{u}^v_{k|t}$ is the change $\Delta u^v_{k|t}$ in inputs between consecutive time steps, the polytope $\mathcal{U}$ used in the terminal set computations in Section 7.2.3 is defined by the actuator rate bounds (3.33b).

Consideration of the actuator rate constraints increases the conservativeness of the terminal set as the stopping distance of the ego vehicle increases.

7.2.5 Analytical approaches for computing the terminal set

The methodology for computing the robust control invariant set presented in Section 7.2.3 yields a polyhedral representation of the terminal set $\mathcal{X}_T$. The advantage of the approach based on robust reachability analysis is that it allows for the systematic consideration of the actuator dynamics and rate constraints. Moreover, as the computed set is convex, it can be used with a large number of convex optimization solvers for solving the resulting CFTOC problem. A potential drawback from the standpoint of real-time implementation is that the number of halfspaces defining the polyhedron $\mathcal{X}_T$ can be large, resulting in an equivalent number of state constraints in the online optimization problem. In this section, we derive an analytical expression for the terminal set which can account for the input rate constraints but not the actuator dynamics. Nevertheless, the analysis provides us with a reasonable terminal set.

A similar analysis is performed in [127] in the context of threat assessment for human driven vehicles with the goal of providing warnings to drivers. It is assumed that the driver of the ego vehicle can instantaneously apply maximum braking after a certain reaction time. We generalize this approach by accounting for actuator rate constraints.

The computation is based on the following intuition: in the worst-case event that the preceding vehicle applies maximum braking starting from its current state, the best response of the ego vehicle is to also do the same. In order to account for the rate bounds on the control input, we assume that the ego vehicle can decrease its acceleration at a maximum rate of $|\dot{u}^v_{\min}|$. As discussed in [127], two cases need to be considered. In Case 1, the preceding vehicle comes to a full stop before the ego vehicle. In this case, the separation of the vehicles after they have both stopped is relevant, and must be greater than the safety distance d_{safe}. In Case 2, the ego vehicle comes to a full stop before the preceding vehicle. Here, the minimum separation between the vehicles is attained before they have both stopped, in particular, the time at which the velocities of both vehicles is equal. In this case, the minimum separation must be greater than the safety distance d_{safe}. We first analyze the conditions under which Case 1 or 2 applies.

Let $x^v_{T|t} = [\xi_{T|t}, v_{T|t}]$ and $u^v_{T-1|t} = a_{T-1|t}$ be the state and control input at the end of the horizon. The computation of the time to stop t_s for the ego vehicle is given by Algorithm 7.1. If the acceleration a of the vehicle is allowed to be unbounded, t_s is given by the expression on Line 2. However, as the acceleration a is lower bounded by $u^v_{\min}$, applying a constant jerk of $\dot{u}^v_{\min}$ causes this lower bound to be reached at time t_{md}. After t_{md}, the ego vehicle can brake at a constant acceleration $u^v_{\min}$ to come to a stop. t_s is then given by Line 5. The time to stop for the preceding vehicle, assuming it can achieve its maximum deceleration $a^e_{\min}$ instantaneously, is given by:

$$t^e_s = -\frac{\bar{v}^e_{T|t}}{2a^e_{\min}}. \tag{7.21}$$

Hence, Case 1 applies when $t^e_s \leq t_s$ and Case 2 applies otherwise. We now present the design the RCI terminal set for the two cases.

Algorithm 7.1 Computation of ego vehicle stopping time

```
Input: $\xi := \xi_{T|t}, v := v_{T|t}, a := a_{T-1|t}$    ▷ State and input at end of horizon
1: $t_{\text{md}} = \frac{u^v_{\min} - a}{\dot{u}^v_{\min}}$    ▷ Time to achieve minimum deceleration $u^v_{\min}$ with jerk $\dot{u}^v_{\min}$
2: $t_s = \frac{1}{\dot{u}^v_{\min}}\left(-a - \sqrt{a^2 - 2v\dot{u}^v_{\min}}\right)$    ▷ Time to stop with constant jerk $\dot{u}^v_{\min}$
3: if $t_{\text{md}} < t_s$:
4:     $v_{\text{md}} = v + at_{\text{md}} + \frac{1}{2}\dot{u}^v_{\min}t^2_{\text{md}}$    ▷ Velocity at $t_{\text{md}}$
5:     $t_s = t_{\text{md}} - \frac{v_{\text{md}}}{u^v_{\min}}$    ▷ Time to stop with constant jerk $\dot{u}^v_{\min}$ followed by constant acceleration $u^v_{\min}$
Output: $t_s$
```

Case 1: $t^e_s \leq t_s$

In this case, the distances traveled by the ego and preceding vehicles in the process of coming to a halt need to be computed. The distance d^v_{brake} traveled by the ego vehicle by applying the minimum jerk $\dot{u}^v_{\min}$ is given by Algorithm 7.2. As in the case of the stopping time computations in Algorithm 7.1, if the acceleration a is allowed to be unbounded, d^v_{brake} is given by the expression on Line 8. However, as the acceleration a is lower bounded by $u^v_{\min}$, applying a constant jerk of $\dot{u}^v_{\min}$ causes the lower bound to be reached at time t_{md}. After t_{md}, the ego vehicle can brake at a constant acceleration $u^v_{\min}$ to come to a stop.

The braking distance d^e_{brake} for the preceding vehicle, assuming it can achieve its maximum deceleration $a^e_{\min}$ instantaneously, is given by:

$$d^e_{\text{brake}} = -\frac{(\bar{v}^e_{T|t})^2}{2a^e_{\min}}. \tag{7.22}$$

Note the use of the predicted worst-case speed $\bar{v}^e_{T|t}$ in (7.22). Robust persistent feasibility in Case 1 is guaranteed by defining the following terminal set:

$$\mathcal{X}_T = \{[\xi_{T|t}, v_{T|t}, a_{T-1|t}] : \xi_{T|t} + d^v_{\text{brake}} + d_{\text{safe}} \leq \bar{\xi}^e_{T|t} + d^e_{\text{brake}}\}. \tag{7.23}$$

Algorithm 7.2 Computation of ego vehicle braking distance (Case 1)

```
Input: $\xi := \xi_{T|t}, v := v_{T|t}, a := a_{T-1|t}$    ▷ State and input at end of horizon
1: $t_{\text{md}} = \frac{u^v_{\min}-a}{\dot{u}^v_{\min}}$    ▷ Time to achieve minimum deceleration $u^v_{\min}$ with jerk $\dot{u}^v_{\min}$
2: $t_s = \frac{1}{\dot{u}^v_{\min}}\left(-a-\sqrt{a^2-2v\dot{u}^v_{\min}}\right)$    ▷ Time to stop with constant jerk $\dot{u}^v_{\min}$
3: if $t_{\text{md}} < t_s$:
4:     $v_{\text{md}} = v + at_{\text{md}} + \frac{1}{2}\dot{u}^v_{\min}t^2_{\text{md}}$    ▷ Velocity at $t_{\text{md}}$
5:     $\xi_{\text{md}} = \xi + vt_{\text{md}} + \frac{1}{2}at^2_{\text{md}} + \frac{1}{6}\dot{u}^v_{\min}t^3_{\text{md}}$    ▷ Position at $t_{\text{md}}$
6:     $d^v_{\text{brake}} = \xi_{\text{md}} - \xi - \frac{v^2_{\text{md}}}{2u^v_{\min}}$    ▷ Braking distance with constant jerk $\dot{u}^v_{\min}$ followed by constant acceleration $u^v_{\min}$
7: else:
8:     $d^v_{\text{brake}} = vt_s + \frac{1}{2}at^2_s + \frac{1}{6}\dot{u}^v_{\min}t^3_s$    ▷ Braking distance with constant jerk $\dot{u}^v_{\min}$
Output: $d^v_{\text{brake}}$
```

Case 2: $t^e_s > t_s$

In this case, the separation between the ego and preceding vehicles at the time when their velocities are equal needs to be computed. This yields the minimum separation between the vehicles. Let $v = v_{T|t}$, $a = a_{T-1|t}$, $v^e = \bar{v}^e_{T|t}$. If the acceleration of the ego vehicle is allowed to be unbounded, the time $t^{(1)}_{\text{ms}}$ at which the minimum separation is achieved is given by:

$$t^{(1)}_{\text{ms}} = \{t \in \mathbb{R}^+ | v + at + \frac{1}{2}\dot{u}^v_{\min}t^2 = v^e + a^e_{\min}t\}. \tag{7.24}$$

The condition on the RHS of (7.24) equates the velocities of the vehicles. If $t^{(1)}_{\text{ms}} > t_{\text{md}}$ (t_{md} is defined on Line 1 of Algorithm 7.2), the modified time to minimum separation $t^{(2)}_{\text{ms}}$ is computed as:

$$t^{(2)}_{\text{ms}} = \{t \in \mathbb{R}^+ | v_{\text{md}} + u^v_{\min}(t - t_{\text{md}}) = v^e + a^e_{\min}t\}, \tag{7.25}$$

where the ego velocity v_{md} at t_{md} is defined on Line 4 of Algorithm 7.2. Equation (7.25) accounts for the fact that the ego vehicle will achieve its maximum deceleration $u^v_{\min}$ at t_{md} and continue braking at constant acceleration from then onwards. The minimum separation ξ^r_{ms} between the vehicles is then given by Algorithm 7.3 and is computed along the same lines as the braking distance in Algorithm 7.2.

Robust persistent feasibility in Case 2 is guaranteed by defining the following terminal set:

$$\mathcal{X}_T = \{[\xi_{T|t}, v_{T|t}, a_{T-1|t}] : \xi^r_{\text{ms}} \geq d_{\text{safe}}\}. \tag{7.26}$$

Remark 7.10. *It is possible that solutions to* (7.24) *and/or* (7.25) *do not exist. This implies that the minimum separation is either attained before time* $t + T$ *(that is, the end of the*

Algorithm 7.3 Computation of minimum separation (Case 2)

```
Input: $\xi := \xi_{T|t}, v := v_{T|t}, a := a_{T-1|t}, \xi^e := \bar{\xi}^e_{T|t}, v^e := \bar{v}^e_{T|t}$   ▷ State and input at end of horizon
$t_{\mathrm{md}} = \frac{u^v_{\min} - a}{\dot{u}^v_{\min}}$   ▷ Time to achieve minimum deceleration $u^v_{\min}$ with jerk $\dot{u}^v_{\min}$
if $t_{\mathrm{md}} < t^{(1)}_{\mathrm{ms}}$:
    $v_{\mathrm{md}} = v + at_{\mathrm{md}} + \frac{1}{2}\dot{u}^v_{\min}t^2_{\mathrm{md}}$   ▷ Velocity at $t_{\mathrm{md}}$
    $\xi_{\mathrm{md}} = \xi + vt_{\mathrm{md}} + \frac{1}{2}at^2_{\mathrm{md}} + \frac{1}{6}\dot{u}^v_{\min}t^3_{\mathrm{md}}$   ▷ Position at $t_{\mathrm{md}}$
    $\xi^v_{\mathrm{ms}} = \xi_{\mathrm{md}} + v_{\mathrm{md}}(t^{(2)}_{\mathrm{ms}} - t_{\mathrm{md}}) + \frac{1}{2}u^v_{\min}(t^{(2)}_{\mathrm{ms}} - t_{\mathrm{md}})^2$   ▷ Position at $t^{(2)}_{\mathrm{ms}}$ with constant jerk $\dot{u}^v_{\min}$ followed by constant acceleration $u^v_{\min}$
    $\xi^e_{\mathrm{ms}} = \xi^e + v^e t^{(2)}_{\mathrm{ms}} + \frac{1}{2}a^e_{\min}(t^{(2)}_{\mathrm{ms}})^2$   ▷ Preceding vehicle position at $t^{(2)}_{\mathrm{ms}}$ with constant acceleration $a^e_{\min}$
else:
    $\xi^v_{\mathrm{ms}} = \xi + vt^{(1)}_{\mathrm{ms}} + \frac{1}{2}a(t^{(1)}_{\mathrm{ms}})^2 + \frac{1}{6}\dot{u}^v_{\min}(t^{(1)}_{\mathrm{ms}})^3$   ▷ Position at $t^{(1)}_{\mathrm{ms}}$ with constant jerk $\dot{u}^v_{\min}$
    $\xi^e_{\mathrm{ms}} = \xi^e + v^e t^{(1)}_{\mathrm{ms}} + \frac{1}{2}a^e_{\min}(t^{(1)}_{\mathrm{ms}})^2$   ▷ Preceding vehicle position at $t^{(1)}_{\mathrm{ms}}$ with constant acceleration $a^e_{\min}$
$\xi^r_{\mathrm{ms}} = \xi^e_{\mathrm{ms}} - \xi^v_{\mathrm{ms}}$
Output: $\xi^r_{\mathrm{ms}}$
```

prediction horizon) or at time $t + t_s$ (that is, when the ego vehicle comes to a stop). In this case, the following terminal constraint is sufficient for persistent feasibility:

$$\bar{\xi}^e_{T|t} - \xi_{k|t} \geq d_{\mathrm{safe}}, \tag{7.27a}$$

$$\bar{\xi}^e_{t_s|t} - \xi_{t_s|t} \geq d_{\mathrm{safe}}, \tag{7.27b}$$

where $\bar{\xi}^e_{t_s|t}$ and $\xi_{t_s|t}$ are the positions of the preceding and ego vehicles, respectively, when the latter comes to a stop. $\bar{\xi}^e_{t_s|t}$ and $\xi_{t_s|t}$ can be computed using Algorithm 7.2 and (7.22)*, respectively.*

Summary

In summary, the RCI terminal sets are given by (7.23) or (7.26) depending on the relative stopping times of the preceding and ego vehicles. While the computations involved (Algorithms 7.1, 7.2, 7.3) are nonlinear, they can be embedded in optimization solvers such as NPSOL [51] which rely on function evaluations to compute the values and gradients of the constraints. For convex optimization solvers, we use the terminal set obtained by the robust reachability approach presented in Section 7.2.3. Here, even in the presence of a large number of inequalities defining $\mathcal{X}_T$, we can achieve efficient real-time performance as most of the inequalities are expected to be inactive.

A practical approximation which yields a much simpler expression for the terminal constraint is to only consider Case 1 (that is, $t^e_s \leq t_s$) and assume that the ego vehicle can achieve its maximum deceleration instantaneously. In this case, the terminal constraint is given by:

$$\bar{\xi}^e_{T|t} - \frac{(\bar{v}^e_{T|t})^2}{2a^e_{\min}} - \left(\xi_{T|t} - \frac{v^2_{T|t}}{2u^v_{\min}}\right) \geq d_{\text{safe}}. \tag{7.28}$$

That is, we constrain the distance between the ego and preceding vehicles when they come to a stop to be greater than the safety distance d_{safe}. Note that the set of states defined by (7.28) is not a RCI set.

7.2.6 Safety margin

The inequality in (7.28) introduces the notion of a safety margin. We define the *braking distance margin (BDM)* at time t as follows:

$$BDM_t = \bar{\xi}^e_{T|t} - \frac{(\bar{v}^e_{T|t})^2}{2a^e_{\min}} - \left(\xi_{T|t} - \frac{v^2_{T|t}}{2u^v_{\min}}\right) - d_{\text{safe}}. \tag{7.29}$$

We use this metric to determine the most relevant TV for the driver model presented in Section 7.3. That is, among the detected relevant target vehicles (discussed below in Section 7.2.7), the vehicle with the smallest BDM is chosen as the most relevant TV. This metric is also used to compare the performance of our algorithm against an existing approach in Section 7.4.

7.2.7 Dealing with multi-lane targets

The above analyses assume the existence of a single TV in front of the EV, referred to as the preceding vehicles. One of our main objectives is to account for TVs cutting in to the EV's lane. We use the lane change intention estimator developed in Section 4.3 to identify such Relevant Targets (RTs).

Recall that, for a given TV at time t, the intention estimator provides an estimate of its most likely mode $m^{ML}_t \in \mathcal{M} = \{LK, LCL, LCR\}$ conditioned on the history of measurements. Among all TVs in front of the EV, we define the RTs as:

- the TVs in the EV's lane,
- the TVs in the adjacent left lane whose most likely mode $m^{ML}_t = LCR$,
- the TVs in the adjacent right lane whose most likely mode $m^{ML}_t = LCL$.

Let N_{RT} denote the number of RTs detected at time t, and $x^{e_i}_t$ the state of the i^{th} RT at time t ($i = 1, \ldots, N_{RT}$). We enforce the safety constraints (7.9) and the terminal constraint for each RT. Concretely:

$$g(x^v_{k|t}, x^{e_i}_{k|t}) \leq 0, \qquad (k = 1, \ldots, T-1) \tag{7.30a}$$

$$
\begin{aligned}
&[x^v_{T|t}, x^{e_i}_{T|t}] \in \mathcal{X}_T. && (7.30b)\\
&(i = 1, \dots, N_{RT})
\end{aligned}
$$

Remark 7.11. *The above relevant target selection strategy is not truly "robust" in a worst-case sense due to the use of the most likely mode. This is due to the fact that a worst-case analysis would cause the ego vehicle to slow down frequently and unnecessarily even for very small lane change probabilities of the neighboring vehicles. We show this by means of a simulative example later in Chapter 8.*

7.2.8 Online optimization problem

The CFTOC problem to be solved online at time t is given by:

$$
\begin{aligned}
&\min_{u^v_{0:T-1|t}, \epsilon} \sum_{k=0}^{T-1} \left(\|v_{k+1|t} - v^{\text{ref}}\|^2_Q + \|u^v_{k|t} - u^d_{k|t}\|^2_R + \|\Delta u^v_{k|t}\|^2_P \right) + M\epsilon && (7.31a)\\
&\text{subject to:} \quad x^v_{k+1|t} = Ax^v_{k|t} + Bu^v_{k|t}, && (7.31b)\\
&\qquad g(x^v_{k|t}, x^{e_i}_{k|t}) \leq \epsilon, \quad (i = 1, \dots, N_{RT}) && (7.31c)\\
&\qquad \Delta u^v_{k|t} = u^v_{k|t} - u^v_{k-1|t}, && (7.31d)\\
&\qquad [u^v_{k|t}, u^v_{k-1|t}] \in \mathcal{U}, && (7.31e)\\
&\qquad (k = 0, \dots, T-1)\\
&\qquad [x^v_{T|t}, x^{e_i}_{T|t}] \in \mathcal{X}_T, \qquad (i = 1, \dots, N_{RT}) && (7.31f)\\
&\qquad x_{0|t} = x_t, \quad u^v_{-1|t} = u^v_{t-1}, \quad \epsilon \geq 0. && (7.31g)
\end{aligned}
$$

The safety constraints (7.31c) are imposed as soft constraints where the slack variable ϵ is penalized by a large number M. The reference acceleration $u^d_{k|t}$ in the cost function (7.31a) is the driver desired acceleration which is generated by the driver model presented next in Section 7.3. The above formulation allows the controller to replicate the driver's driving style while still maintaining safety guarantees in the presence of uncertainty. In general, the value of R is much larger than that of Q to encourage the controller to track $u^d_{k|t}$. However, in the absence of any RTs, a *free-flow* mode is triggered wherein the controller tracks the road speed limit or a user-defined reference speed v^{ref} as in conventional cruise control systems. In this case, the value of Q is increased.

7.3 Personalized Driving

In this section, we present the driver model used for personalizing the driving behavior of the ego vehicle. The goal of the driver model is to generate a sequence of reference accelerations $u^d_{0:T-1|t}$ for the MPC-based controller for the given driving situation. The idea is to learn the driver's control input $u^d_{k|t}$ from the data collected from that particular driver. By tracking this input, the controller can yield a driving style similar to that of the driver, thus *personalizing* the autonomous vehicle behavior.

We use the approach developed in [74] which employs the frameworks of Hidden Markov Models (HMMs) and Gaussian Mixture Regression (GMR). Similar to the lane change intention estimation algorithm presented in Section 4.3, the car-following behavior of the driver is modeled as a HMM. In this case, however, the discrete modes do not hold a physical meaning such as lane keeping or changing in the lateral case. Instead, the modes allow for a systematic identification of the areas of the feature space where the driver likes to operate. We define the following variables:

- $m_t \in \{1, \dots, M\}$ is the hidden mode at time instant t, where M is the number of hidden modes.
- $z_t = [\xi^r_t, v^r_t, v_t]$ is the vector of observations at time t, where $\xi^r_t = \xi^e_t - \xi_t$ and $v^r_t = v^e_t - v_t$ are the relative distance and relative velocity of the most relevant TV, respectively.
- a_t is the acceleration of the EV at time instant t.

The joint distribution of the modes $m_{0:t} = \{m_0, \dots, m_t\}$, observations $z_{1:t} = \{z_1, \dots, z_t\}$ and accelerations $a_{1:t} = \{a_1, \dots, a_t\}$ is given by:

$$P(m_{0:t}, z_{1:t}, a_{1:t}) = P(m_0) \prod_{k=1}^{t} P(m_k|m_{k-1}) P(z_k, a_k|m_k), \tag{7.32}$$

where the emission probability density function $P(z_k, a_k|m_k = i)$ is modeled as a Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$. The parameters that characterize the prior and transition probability mass functions in (7.32) and the means and covariances of the Gaussian emission density function are learned from data collected from a single driver using the Expectation-Maximization (EM) algorithm and the Bayesian Information Criterion [74].

During online operation, the reference acceleration u^d_t at time t is computed using Gaussian Mixture Regression (GMR) as:

$$u^d_t = E[a_t|z_{1:t}] = \sum_{i=1}^{M} P(m_t = i|z_{1:t})[\mu^a_i + \Sigma^{az}_i (\Sigma^{zz}_i)^{-1}(z_t - \mu^z_i)], \tag{7.33}$$

where:

$$\mu_i = \begin{bmatrix} \mu^z_i \\ \mu^a_i \end{bmatrix}, \qquad \Sigma_i = \begin{bmatrix} \Sigma^{zz}_i & \Sigma^{za}_i \\ \Sigma^{az}_i & \Sigma^{aa}_i \end{bmatrix}. \tag{7.34}$$

The conditional distribution over modes $P(m_t = i|z_{1:t})$ is computed recursively as:

$$P(m_t = i|z_{1:t}) \propto P(z_t|m_t = i) \sum_{j \in \mathcal{M}} P(m_{t-1} = j|z_{1:t-1}) P(m_t = i|m_{t-1} = j), \tag{7.35}$$

initialized with the prior distribution $P(m_0)$.

A sequence of reference accelerations $u^d_{0:T-1|t}$ over the prediction horizon T of the controller can be obtained by iteratively simulating $u^d_{k|t}$ through the vehicle dynamics (7.1) and running (7.33) using the simulated features. In order to compute the predicted values of the features, the most relevant TV is assumed to move at a constant velocity over the horizon.

7.4 Experimental Results: Multi-lane Scenarios

This section demonstrates the performance of the proposed ACC system in traffic situations involving target vehicle cut-ins. While the system is capable of running in real-time on our experimental vehicle (see Section 7.5), it was tested in offline simulations using sensory data collected from our vehicle for a fair comparison with existing approaches. Specifically, we simulate the effect of the control input on the motion of the EV but use real data to reconstruct the environment. It is assumed that the EV has no influence on TVs that cut-in to its path. The data collection, scene reconstruction and controller evaluation processes are explained in more detail below.

7.4.1 Data collection and scene reconstruction

We collected manual driving data from our on-board vehicle sensors (radar, camera, lidar, GPS/INS) on highways. In order to accurately reconstruct the evolution of the environment, the inertial motion of the EV is estimated using the GPS/INS measurements. Combining this with the relative motion of the TVs perceived by the outward looking sensors allows us to estimate the inertial motion of the TVs. It is then straightforward to compute the apparent relative motion of the TVs with respect to the EV during a simulation involving the proposed controller. The TV data is passed to the sensor fusion and RT identification algorithms. The resulting RT data is used as an input to the MPC-based controller. Finally, the EV motion is simulated using (3.30), hence closing the loop.

We focus specifically on short segments of the datasets (around 10-15 seconds each) which involve a cut-in maneuver by a TV in the neighboring lane. The segment start time is chosen to be a few seconds before the TV initiates the lane change and the end time is when the TV completes the lane change.

7.4.2 Evaluation methodology

We compare the performance of the proposed ACC approach (denoted as $ACC1$) with the existing ACC system implemented on our experimental vehicle (denoted as $ACC2$). Note that the existing system uses a combination of radar and camera without any lane change intent estimation to identify a single PT. The PT identification is based on whether a candidate TV lies in a *virtual lane* constructed on the basis of the longitudinal velocity and yaw rate of the EV. The main difference between $ACC1$ and $ACC2$ lies in the identification of

Metric	Unit	$ACC1$	$ACC2$
μ_a	m/s^2	1.00 ± 0.46	1.91 ± 0.91
p_a	m/s^2	2.00 ± 1.12	3.01 ± 1.17
μ_j	m/s^3	2.04 ± 1.83	2.20 ± 1.68
p_j	m/s^3	8.69 ± 2.45	9.53 ± 1.29
BDM	m	42.8 ± 15.9	32.2 ± 17.1
$TTCi$	s^{-1}	0.20 ± 0.12	0.81 ± 1.24

Table 7.1: Means and standard deviations of the metrics defined in Section 7.4.2 for the proposed approach ($ACC1$) and the existing approach ($ACC2$) computed from 28 simulations.

the RTs or the PT. For a fair evaluation, the same control strategy (presented in Section 7.2) is used for both approaches.

For each segment of the dataset involving a simulation of either $ACC1$ or $ACC2$, we compute the following metrics:

1. Absolute values of the mean and maximum deceleration, μ_a and p_a, respectively.

2. Absolute values of the mean and maximum negative jerk, μ_j and p_j, respectively.

3. Minimum braking distance margin, BDM, as defined in (7.29), computed for the most relevant TV.

4. Maximum inverse time-to-collision, $TTCi$, defined as the ratio of the relative speed of the EV with respect to the most relevant TV and the relative distance.

Metrics 1 and 2 characterize the comfort of the system, where lower values correspond to a smoother behavior. Metrics 3 and 4 characterize the safety. A higher value of the BDM is desired while a lower value of $TTCi$ implies lower collision risk.

7.4.3 Results

For a given approach ($ACC1$ or $ACC2$), the mean and standard deviation of the quantities described in Section 7.4.2 were computed for 28 segments from 4 datasets and are tabulated in Table 7.1. We see that $ACC1$ yields lower values of the mean and maximum deceleration as compared to $ACC2$ resulting in a more comfortable experience for the driver. The difference in the mean and peak negative jerk is less marked. This can be attributed to the fact that only aggressive cut-in maneuvers were selected for the analysis which required a sudden change in the acceleration commanded by the controller. The average values of the BDM

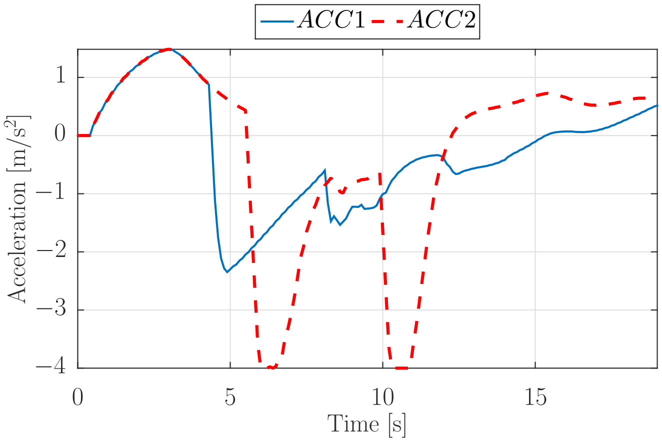


Figure 7.4: Comparison of the closed-loop acceleration profiles during a simulation of the scenario described in Section 7.4.3 of the proposed approach ($ACC1$, solid line) and the existing approach ($ACC2$, dashed line). $ACC1$ results in a smoother acceleration command that $ACC2$.

and $TTCi$ show that $ACC1$ yields a safer behavior than $ACC2$, validating the consideration of TVs' lane change intentions.

We further analyze an interesting and challenging situation recorded in one of the datasets where two TVs changed lanes in front of the EV within a short span of time. At time $t = 5.4$ s (from an arbitrary $t = 0$), the first TV ($TV1$) crosses the right boundary of the EV's lane, followed by the second TV ($TV2$) at time $t = 9.2$ s. The small relative distance and lower speeds of the TVs necessitates a braking maneuver by the EV. Figure 7.4 shows a comparison of the acceleration profiles for $ACC1$ and $ACC2$ when this scenario is reconstructed in simulation. It is seen that due to the lane change estimation of $TV1$, $ACC1$ starts braking at $t = 4.3$ s, 1.2 s before $ACC2$, resulting in a smoother acceleration profile. Later, when $TV2$ cuts in, $ACC1$ responds at $t = 8.1$ s, 1.8 s before $ACC2$. The deceleration command saturates in the case of $ACC2$ but not in the case of $ACC1$. A comparison of the metrics defined in Section 7.4.2 for both approaches is shown in Table 7.2.

7.4.4 Target vehicle cut-in detection

Finally, we compare the performance of the proposed RT identification methodology presented in Section 7.2.7 ($ACC1$) against the existing approach ($ACC2$). The metric used is the time difference between the two approaches recognizing a cutting-in TV as a RT or PT. From the 28 cut-in scenarios experienced during our experiments, we found that our method identifies a RT 1.28 ± 0.26 s before the existing approach.

The trajectory of a TV's lateral position η while it performs a cut-in maneuver from one of the datasets is shown in top plot of Figure 7.5. The middle plot shows the estimated most likely mode (defined in Section 7.2.7). We see that the estimator switches to the lane change right (LCR) mode 1.4 s before the estimated center of mass of the TV crosses the

Metric	Unit	$ACC1$	$ACC2$
μ_a	$\mathrm{m/s^2}$	0.95	2.05
p_a	$\mathrm{m/s^2}$	2.35	4
μ_j	$\mathrm{m/s^3}$	1.42	1.70
p_j	$\mathrm{m/s^3}$	10	10
BDM	m	33.4	20.3
$TTCi$	$\mathrm{s^{-1}}$	0.26	0.52

Table 7.2: Comparison of the metrics defined in Section 7.4.2 for $ACC1$ and $ACC2$ during a simulation of the scenario described in Section 7.4.3.

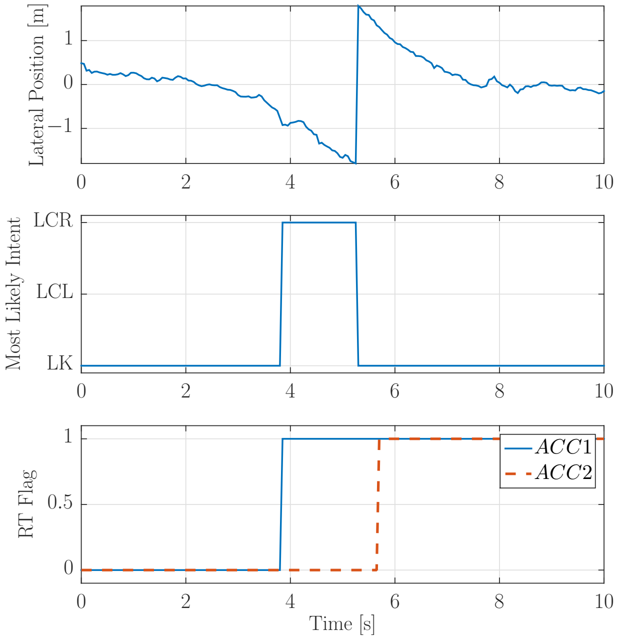


Figure 7.5: Results of the relevant target identification for a TV cut-in scenario from one of the datasets (described in Section 7.4.4). $ACC1$ identifies the RT 1.85 s before $ACC2$.

lane boundary. The bottom plot of Figure 7.5 depicts the binary variable indicating whether or not the TV has been identified as a RT for both methods $ACC1$ and $ACC2$. It is seen that $ACC1$ identifies the RT 1.85 s before $ACC2$ in this scenario.

7.5 Experimental Results: Personalization and Safety

In this section, we show the ability of the proposed ACC system to reproduce different driving styles and keep the vehicle safe in scenarios where the reference deceleration generated by the driver model is not sufficient to prevent a rear-end collision.

As driver modeling is not the main focus of this dissertation, we only briefly review the results of studies performed with different drivers. Readers are referred to [74, 75] for more details of this analysis. In order to evaluate the ability of the driver model presented in Section 7.3 to learn different driving styles, we collected highway driving data from 5 drivers. For each driver, two models are learned, *(i)* a *personalized* model which is trained using the data collected from that driver, and *(ii)* an *average* model trained using the data collected from the remaining drivers. Each model is evaluated offline in simulation and presented with the same scenarios from the relevant datasets. In order to quantify the similarity between the behavior of a driver model in simulation and the real driver from which the data was collected, the Kolmogorov-Smirnov (KS) distance of the empirical distributions of two key indicators are computed. These are the inverse Time-To-Collision $TTCi$ (defined in Section 7.4) and the Vehicle Specific Power (VSP), which is an indicator of the instantaneous engine load of the vehicle. A smaller KS distance implies a greater similarity between two distributions. For the data collected from each driver, we run the offline evaluation using both the personalized and average models, and compute the aforementioned metrics. The results are shown in Figure 7.6. It is seen that the personalized models perform consistently better than the average ones from the smaller KS distance. On average, the decrease in the KS distance is 27.0% for the VSP and 49.5% for the $TTCi$.

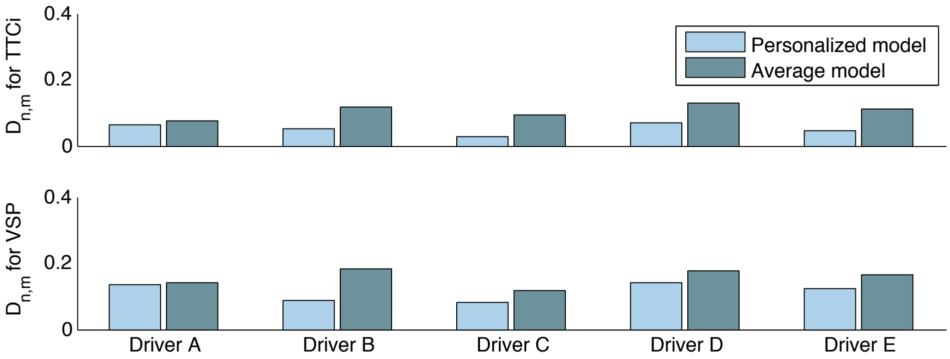


Figure 7.6: KS distances obtained for two driving style indicators, TTCi and VSP, for 5 different drivers (Figure from [75])

The following experiments were performed on the prototype Hyundai Grandeur described in Appendix A.2.

In the first experiment (referred to as Experiment 1), we show the effect of changing the driver model used to generate the reference acceleration for the controller. Two models are used. Driver Model 1 was learned from a conservative driver who preferred keeping a larger time-headway (hence, relative distance) to the preceding vehicle. On the other hand, Driver Model 2 was learned from a relatively aggressive driver who maintained smaller time-headways to the preceding vehicle.

The results from Experiment 1 are shown in Figure 7.7. Initially, the vehicle was operating with the ACC system enabled using Driver Model 1 to generate the reference input. At around time $t = 20$ s, the operator used a button on the steering wheel to switch the model to that of Driver 2. The top plot depicts the acceleration applied by the controller and the accelerations that would have been applied by Driver Model 1 and 2 for the driving situation at the corresponding time. We see that before $t = 20$ s, the controller tracks the reference provided by Driver Model 2. During the same time period, the acceleration reference given by Driver Model 1 is much higher and saturates the upper bound of the control input. After $t = 20$ s, the controller initially tracks the reference given by Driver Model 2, but eventually reduces the acceleration command as the road speed limit is reached. The speed profiles of the EV and most relevant TV (i.e, the MRT) are shown in the middle plot while the bottom plot shows the relative distance between the MRT and EV. It is seen that Driver 2 prefers closing the gap between itself and the vehicle in front to yield a much smaller time-headway as compared to Driver 1.

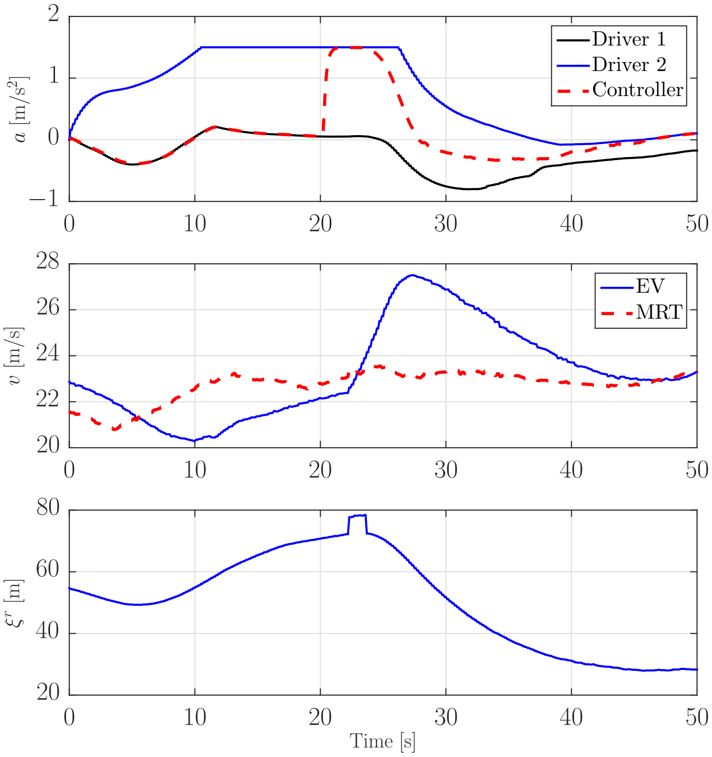


Figure 7.7: Experiment 1: Switching from Driver Model 1 (conservative) to Driver Model 2 (aggressive). MRT denotes the most relevant TV while EV denotes the ego vehicle.

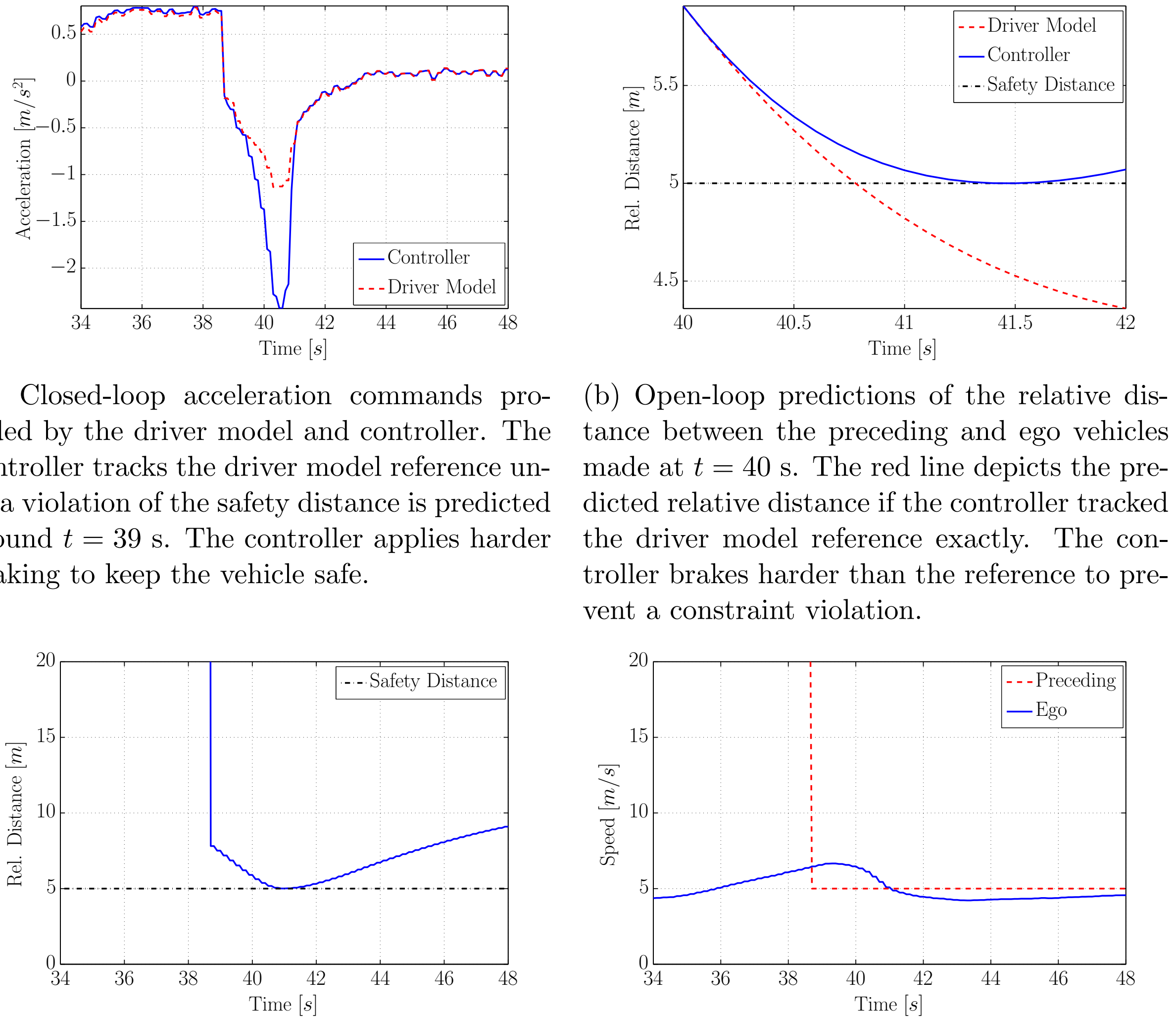
In the following experiments, we demonstrate the ability of the controller to enforce the safety constraints in scenarios where simply following the reference acceleration provided by the driver model would lead to a rear-end collision. For safety reasons, the experiments were performed at low speed using virtual preceding vehicles. An average driver model built using data from multiple drivers was used for the reference generation. We performed the following two experiments, referred to as Experiment 2 and 3.

The results for Experiment 2 are summarized in Figures 7.8a–7.8d. The vehicle is initially moving with a speed of about 4.3 m/s with no vehicle in front of it. The driver model provides a positive acceleration reference which the controller matches closely. At around $t = 38$ s, when the ego vehicle's speed is roughly 6.5 m/s, we introduce a virtual preceding vehicle at a relative distance of 8 m moving at 5 m/s. Although unlikely, such a situation could arise in practice if a vehicle in a neighboring lane makes a sudden lane change into the ego vehicle's lane. The driver model responds by commanding a negative acceleration. However, the controller anticipates a violation of the safety distance and applies a stronger braking. This is seen by the discrepancy between the driver model and controller inputs in Figure 7.8a. The open-loop predictions of the relative distance to the preceding vehicle made by the controller at $t = 40$ s are shown in Figure 7.8b. If the driver model reference was matched exactly, the safety distance of 5 m would be violated at $t = 40.8$ s. This is seen by the red line going below the black line in Figure 7.8b. The controller plans to deviate from the reference input to keep the predicted relative distance (blue line) greater than the safety distance (black line). Figure 7.8c shows that the controller does not allow the relative distance to go below the specified safety distance in closed-loop operation. The speeds of the ego and preceding vehicles during the experiment are shown in Figure 7.8d.

In Experiment 3, we show the ability of the controller to handle situations in which the virtual preceding vehicle initially violates the safety constraints. We introduce the virtual preceding vehicle at a relative distance of 4 m to the ego vehicle moving at a speed of 5 m/s. The initial relative distance is less than the specified safety distance of 5 m. Moreover, the ego vehicle is moving faster than the preceding vehicle. As shown in Figure 7.9, the controller commands the maximum braking $|a_{\min}|$ as expected.

7.6 Conclusions

In this chapter, we show the application of robust predictive control to the Autonomous Cruise Control (ACC) problem where the motion of the surrounding vehicles is unknown. A data-based lane change intention estimator identifies relevant targets for the purpose of ACC. Safety constraints are formulated accounting for the worst-case actions of the potential target vehicles. Moreover, a method of personalizing the car-following behavior of the system based on demonstrations from a human driver is presented. The results show the ability of the controller to reproduce various driving styles and to keep the vehicle safe in scenarios such as cut-in maneuvers by surrounding vehicles.



(a) Closed-loop acceleration commands provided by the driver model and controller. The controller tracks the driver model reference until a violation of the safety distance is predicted around $t = 39$ s. The controller applies harder braking to keep the vehicle safe.

(b) Open-loop predictions of the relative distance between the preceding and ego vehicles made at $t = 40$ s. The red line depicts the predicted relative distance if the controller tracked the driver model reference exactly. The controller brakes harder than the reference to prevent a constraint violation.

(c) Closed-loop relative distance between the preceding and ego vehicles. The high values before $t = 38.6$ s indicate the absence of a preceding vehicle in that duration.

(d) Closed-loop speed profiles of the ego and preceding vehicles.

Figure 7.8: Experiment 2: Safety performance of controller.

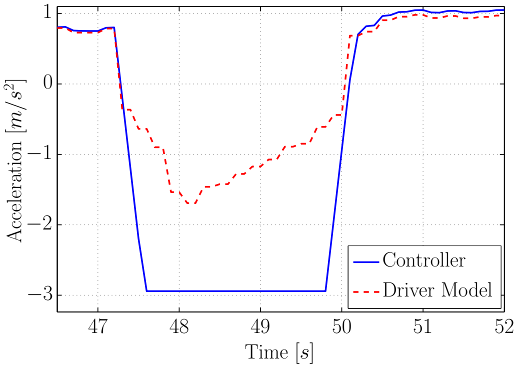


Figure 7.9: Experiment 3: Acceleration commands provided by the driver model and controller. The controller applies maximum braking at around $t = 47$ s as the initial relative distance between the ego and preceding vehicles violates the safety distance.

Chapter 8

Stochastic Predictive Control for Automated Highway Driving

8.1 Introduction

The control design in Chapters 6 and 7 is based on robust MPC where disturbances are assumed to be bounded and have a uniform distribution over their support sets. While such a worst-case approach is suitable for the applications presented in the respective chapters, our intuition is that it does not extend to general traffic scenarios where lateral and longitudinal predictions of target vehicles must be considered. This chapter focuses on stochastic approaches where we utilize knowledge of the distribution of the uncertainty to reduce the conservatism of the resulting control policy. We first motivate the need for probabilistic approaches over robust ones, specifically while dealing with the uncertainty arising from traffic predictions.

Motivating Example

We consider a simple scenario. The ego vehicle E is driving in the left lane, initially at a speed of 10 m/s. The target vehicle T_1 is driving at a constant speed of 10 m/s in the right lane, and keeps its speed and lane during the simulation. The goal for E is to track a reference speed of 15 m/s and pass T_1 in doing so.

At each time step, we use a worst-case approach to predict the set of reachable states of T_1 over the prediction horizon. We assume that T_1 will keep its speed constant over the horizon and only focus on the uncertainty associated with its future lateral motion. At time t, the kinematic bicycle model (3.24) is used to simulate the motion of T_1 over the horizon with the *worst-case inputs* $\bar{u}^e_{0:T-1|t}$ given the current state x^e_t. The worst-case control inputs are defined as the extreme points of the set $\mathcal{U}$ which defines the actuator limit. Concretely, the extreme points are given by two sequences $\bar{u}^{\min,e}_{0:T-1|t}$ and $\bar{u}^{\max,e}_{0:T-1|t}$, defined as:

$$\bar{u}^{\min,e}_{k+1|t} = \max\{\bar{u}^{\min,e}_{k|t} + \Delta t^e \dot{u}^e_{\min}, u^e_{\min}\}, \tag{8.1a}$$

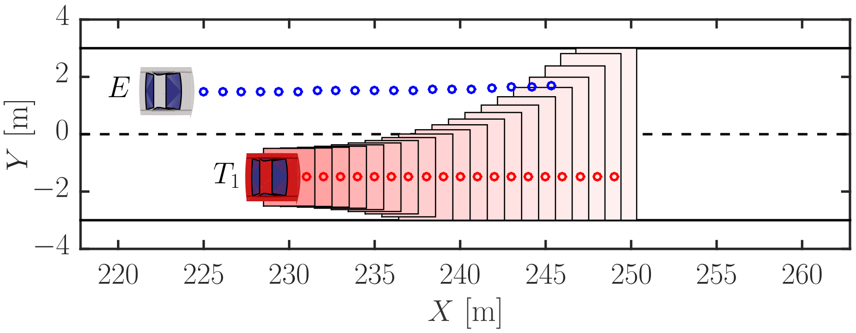


Figure 8.1: Snapshot of simulation with worst-case target vehicle predictions. The red shaded rectangles show the set of all possible states of the target vehicle T_1 assuming constant speed. The red circles depict the closed-loop trajectory of T_1, while the blue circles depict the open-loop trajectory of the ego vehicle E.

$$\bar{u}^{\max,e}_{k+1|t} = \min\{\bar{u}^{\max,e}_{k|t} + \Delta t^e \dot{u}^e_{\max}, u^e_{\max}\}, \tag{8.1b}$$

initialized with $\bar{u}^{\min,e}_{0|t} = \bar{u}^{\max,e}_{0|t} = 0$ as the control inputs of T_1 cannot be measured. The corresponding extreme states of T_1 over the horizon are denoted as $\bar{x}^{\min,e}_{0:T-1|t}$ and $\bar{x}^{\max,e}_{0:T-1|t}$. At each time step k of the prediction horizon, the reachable set of T_1 is obtained by taking the convex hull of the vehicle shapes corresponding to $\bar{x}^{\min,e}_{k|t}$ and $\bar{x}^{\max,e}_{k|t}$. The safety constraints are formulated by using the signed distance approach of Section 4.5.2 using the the aforementioned reachable set and the predicted position $x^v_{k|t}$ of E. No uncertainty in the motion of E is considered for the sake of simplicity.

A snapshot at the end of the simulation (when the overall vehicle-environment system reaches a steady-state) is shown in Figure 8.1. The red shaded rectangles depict the reachable sets of T_1 at the discrete time steps of the prediction horizon. Each rectangle corresponds to one time step and lighter rectangles correspond to later time steps. The blue circles depict the open-loop trajectory of the ego vehicle E. It is seen that the predicted occupancy of T_1 over the horizon spans the entire width of the roadway, thus preventing E from passing it. The closed-loop speed profile of E is shown in Figure 8.2. We see that the ego vehicle initially tries to reach the reference speed, but is forced to slow down to satisfy the safety constraints due to the worst-case predictions.

The stochastic predictive control strategies presented in this chapter resolve the above issue as follows. Based on the motion of the target vehicles, a probability distribution over their future actions (and hence, states) is estimated. The distributions are either represented explicitly as Gaussians or using samples. This assigns low probabilities to events such as extreme steering maneuvers by the surrounding vehicles. In the event that a target vehicle does exhibit aggressive behavior, we rely on the environment model to detect this and estimate the corresponding distributions of the future target vehicle predictions.

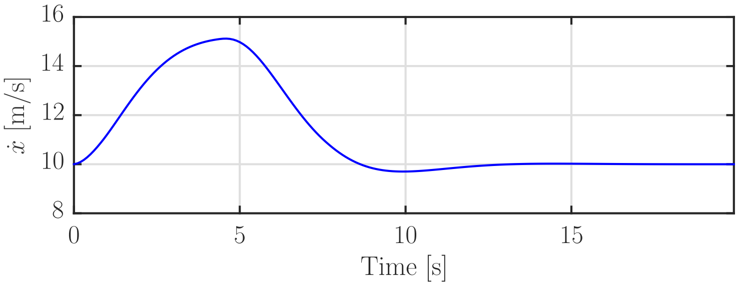


Figure 8.2: Closed-loop longitudinal speed profile of the ego vehicle E. The ego vehicle initially speeds up to the reference speed, but slows down to satisfy the safety constraints due to the worst-case predictions.

Related work

The broad problem of control in the presence of stochastic uncertainty is well-studied in literature. In the predictive control literature, the problem is referred to as *Stochastic or Chance-constrained MPC*, while in the robotics literature, the term *Belief-space Planning* is also used. In the latter, additional uncertainty due to partial observability of the system state is considered. The problem then lies in the class of Partially Observable Markov Decision Processes (POMDPs) where a dual objective exists of reaching a specified goal while performing information gathering actions to reduce the uncertainty of the state. From the perspective of autonomous vehicle control, full state feedback is usually a reasonable assumption except in the cases where parametric uncertainty in the system dynamics is considered (as discussed in Appendix B). Here, we review prior work in optimization-based stochastic control.

As discussed in Section 2.3.1, the main challenges in SMPC are the characterization of the future uncertainty in the state and the translation of the chance-constraints into deterministic ones. Efficient solution strategies exist for linear systems affected by additive Gaussian disturbances with polyhedral chance-constraints [71, 124]. Analytical expressions for the predicted variance of the state and the constraint tightening offset can be obtained. Finitely supported non-Gaussian disturbances are also considered in [71] where a numerical scheme is used to evaluate the multivariate convolution integral required to tighten the chance-constraints.

For nonlinear systems, a linearization-based approach is commonly used. In [120], the nonlinear system dynamics are linearized around a nominal trajectory. The goal is to quantify the distribution of the state along the trajectory assuming that a given feedback controller and state estimator will be used for the control and estimation, respectively. The approach is used to assess the safety of candidate trajectories provided by a high-level planner algorithm, but not to compute the optimal trajectory itself. The work is extended in [121] where an iterative linearization method is used to compute a locally optimal policy for the unconstrained finite horizon problem starting from a given feasible trajectory. Obstacles are encoded as costs in the objective function and the probability of not colliding with an obstacle is maximized. However, probabilistic constraint satisfaction is not guaranteed and no uncertainty in the obstacles is considered.

Uncertainty in the parameters defining the polyhedral chance-constraints is accounted for in [125]. The constraint parameters are modeled as Gaussian random variables in [125] and a tractable approximation is presented to obtain a deterministic convex optimization problem.

Sampling-based methods have shown to be effective for dealing with the issues of nonlinear system dynamics, non-Gaussian disturbances and nonconvex constraints with uncertain parameters. In [123], the uncertain constraint parameters and disturbances in the system are sampled. A convex bounding approach is used to approximate the chance-constraint, which may lead to an overly conservative solution. In [50], samples of the disturbance are drawn and used to enforce the state constraints. However, this may be computationally expensive due to the large number of equality and inequality constraints in the optimization problem.

For the specific application of autonomous driving, optimization-based planning under uncertainty has been studied in [33, 79, 105, 128]. A linearized vehicle model is used in [79] to propagate the state uncertainty over the horizon and chance-constraints are formulated for collision avoidance. However, the environment is assumed to be deterministic. Sampling-based methods with reduced complexity are presented in [33, 105] for autonomous highway driving applications by only considering uncertainty in the forecasts of the environment. In [128], a similar approach as in [120] is used to assess the safety of a set of trajectories provided by a motion planner, but no modification of the trajectories is performed.

Contributions

The focus of our work is on developing a framework for the stochastic predictive control of autonomous vehicles in order to systematically deal with uncertainty in the forecasts of the vehicle and the environment. The approach must be able to handle the nonlinear vehicle dynamics and potentially non-Gaussian uncertainty arising from traffic predictions. In this chapter, we first review the following two approaches for solving the SMPC problem (2.6) with the chance-constraints (2.9):

- *Analytical SMPC*, where the distribution of the uncertain variables is characterized by multivariate Gaussians. A linearization of the system dynamics around a nominal trajectory is used to propagate the uncertainty in the state over the horizon. The chance-constraints are tightened based on the predicted variance of the state vector. The control design for the autonomous driving application using analytical SMPC is presented in Section 8.2.

- *Sampling-based SMPC*, where the distribution of the uncertain variables is represented by means of samples. The constraints are enforced for all sampled state variables thus guaranteeing chance-constraint satisfaction. The control design using sampling-based SMPC is discussed in Section 8.3.

In Section 8.4, the aforementioned strategies are compared from the perspective of autonomous vehicle control and a hybrid SMPC approach is proposed which combines the strengths of either method. In particular, the analytical SMPC approach is used to propagate the uncertainty in the vehicle state based on a linearization of the dynamics and a Gaussian characterization of the uncertainty. A sampled representation of the uncertain traffic forecasts along with a sample aggregation scheme is used to handle environmental uncertainty. Simulation and experimental results are presented in Sections 8.5 and 8.6, respectively, which illustrate the following:

1. The trade-off between risk and conservatism that can be obtained by the use of a probabilistic control framework.
2. The value of accounting for multimodal distributions of environmental uncertainty by means of a sampled representation.
3. The ability of the presented control schemes to run in real-time on embedded hardware.

8.2 Analytical Stochastic MPC

This section provides the background for the analytical SMPC strategy. Recall the notation introduced in Chapter 2. The subscript t denotes the current time step whereas the subscripts k and $k|t$ refer to the predicted time step $t+k$ at time t. The symbol $z_{t_1:t_2}$ (or $z_{k_1:k_2|t}$) denotes the sequence $\{z_t\}_{t=t_1}^{t_2}$ (or $\{z_{k|t}\}_{k=k_1}^{k_2}$).

8.2.1 Preliminaries

Analytical SMPC approaches are restricted to linear systems with normally distributed additive disturbances. The fact that linear transformations of Gaussian random variables are themselves normally distributed allows for the easy propagation of the uncertainty through the system dynamics. Moreover, an analytical expression for the chance-constraint tightening can be obtained. Accordingly, the following assumptions are introduced.

Assumption 8.1. *The random variable x_t^v corresponding to the ego vehicle's state vector at time t is modeled as a multivariate Gaussian:*

$$x_t^v \sim \mathcal{N}(\hat{x}_t^v, \Sigma_t^v). \tag{8.2}$$

The disturbance d_t^v in the vehicle dynamics $f^v(\cdot)$ is an i.i.d. multivariate Gaussian:

$$d_t^v \sim \mathcal{N}(\hat{d}_t^v, \Sigma_t^d). \tag{8.3}$$

The predicted states $x_{1:T|t}^v$ (simply denoted as $x_{1:T}^v$) are also modeled as Gaussians:

$$x_{k|t}^v := x_k^v \sim \mathcal{N}(\hat{x}_k^v, \Sigma_k^v). \qquad (k = 1, \dots, T) \tag{8.4}$$

Estimates of the mean $\hat{x}_t^v$ and variance Σ_t^v at time t can be obtained by a state observer. The above assumption is aligned with the stochastic uncertainty characterization in the vehicle model presented in Section 3.6.1. The main challenge with the analytical SMPC approach is characterizing the predicted means $\hat{x}_{1:T}^v$ and variances $\Sigma_{1:T}^v$ over the prediction horizon for the nonlinear vehicle dynamics model $f^v(\cdot)$. In our work, we adopt a linearization-based approach to efficiently predict the means and variances of the state vector. The following assumption is introduced:

Assumption 8.2. *The ego vehicle dynamics model $f^v(\cdot)$ is linearized about a nominal open-loop trajectory $\{\bar{x}_{0:T}^v, \bar{u}_{0:T-1}^v, \bar{d}_{0:T-1}^v\}$ of the ego vehicle to obtain the following Linear Time-Varying (LTV) representation:*

$$x_{k+1}^{\delta,v} = A_k x_k^{\delta,v} + B_k u_k^{\delta,v} + D_k d_k^{\delta,v}, \tag{8.5}$$

where the deviations from the nominal trajectory are denoted as:

$$\begin{aligned} x_k^{\delta,v} &= x_k^v - \bar{x}_k^v, \\ u_k^{\delta,v} &= u_k^v - \bar{u}_k^v, \\ d_k^{\delta,v} &= d_k^v - \bar{d}_k^v, \end{aligned}$$

and the system matrices are given by:

$$\begin{aligned} A_k &= \nabla_{x_k^v} f^v(x_k^v, u_k^v, d_k^v)|_{(\bar{x}_k^v, \bar{u}_k^v, \bar{d}_k^v)}, \\ B_k &= \nabla_{u_k^v} f^v(x_k^v, u_k^v, d_k^v)|_{(\bar{x}_k^v, \bar{u}_k^v, \bar{d}_k^v)}, \\ D_k &= \nabla_{d_k^v} f^v(x_k^v, u_k^v, d_k^v)|_{(\bar{x}_k^v, \bar{u}_k^v, \bar{d}_k^v)}. \end{aligned}$$

The nominal input sequence $\bar{u}_{0:T-1}^v$ can be obtained from the open-loop solution of the SMPC problem at the previous time step. This is similar to the warm start procedure used in the iterative linearization algorithm in Section 5.3.3. The state sequence $\bar{x}_{1:T}^v$ is computed by a forward simulation of the vehicle dynamics $f^v(\cdot)$ using $\bar{u}_{0:T-1}^v$ and the initial state $\bar{x}_{0|t}^v = \hat{x}_t^v$. Note that (8.5) is consistent with the additive disturbance formulation in Section 3.6.2. The nominal disturbance $\bar{d}_{k|t}^v$ is identically equal to $\hat{d}_t^v$. Thus, $d_k^{\delta,v} \sim \mathcal{N}(0, \Sigma_k^d)$.

In order to make the treatment of the chance-constraints (2.9) tractable, we make the following assumption on the formulation of the safety constraints:

Assumption 8.3. *The mapping $g(\cdot)$ in the probabilistic safety constraints* (2.9) *is affine in the vehicle state. Concretely:*

$$g(x_k^v, x_k^e) = G_k^e x_k^{\delta,v} - h_k^e. \tag{8.6}$$

That is, we approximate the nonconvex safe region $g(x_k^v, x_k^e) \leq 0$ for the ego vehicle by the polyhedral region $G_k^e x_k^{\delta,v} - h_k^e \leq 0$. Uncertainty in the environment is accounted for by modeling h_k^e as a multivariate Gaussian:

$$h_k^e \sim \mathcal{N}(\hat{h}_k^e, \Sigma_k^h). \tag{8.7}$$

By the above assumption, the chance-constraints (2.9) can be rewritten as:

$$P(G_k^e x_k^{\delta,v} \leq h_k^e) \geq 1 - \epsilon_k. \tag{8.8}$$

The linearization of the constraints to obtain G_k^e and h_k^e is performed in a similar manner as in the iterative linearization procedure in Chapter 5 (Section 5.3.1). In general, G_k^e and h_k^e are functions of the nominal states $\bar{x}_k^v$ and $\bar{x}_k^e$.

The validity and generality of the above assumptions is analyzed later in Section 8.4 while comparing the analytical and sampling-based SMPC approaches.

8.2.2 Closed-loop approach

As mentioned in Section 2.3.1, we would like to adopt a closed-loop formulation where a parametric representation of the control policy $\kappa^c(\cdot)$ is computed. In this work, we use the following affine representation of the control policy:

$$u_k^{\delta,v} = K_k x_k^{\delta,v} + c_k^v. \tag{8.9}$$

This is a commonly chosen parameterization in the literature [71, 120, 124]. Note that we have implicitly assumed that $u_k^v = u_k^c$ in (8.9) as this chapter focuses on fully autonomous applications without the driver in the loop (see Section 2.3.3 for details).

Due to the linearization in Assumption 8.2 and the presence of constraints in the problem, the chosen control policy parameterization in (8.9) is not necessarily optimal for the original SMPC problem. Nevertheless, the state feedback (8.9) has useful implications which are explained in Section 8.2.3.

While it is possible to optimize over the feedback gains K_k in the SMPC optimization problem, we assume fixed feedback gains that are computed a-priori. We solve an unconstrained finite time optimal control (FTOC) problem (also known as the Linear Quadratic Regulator (LQR) problem) to obtain a sequence of gains $K_{0:T-1}$ over the prediction horizon. The FTOC solves the following minimization problem:

$$\min_{v_0,\ldots,v_{T-1}} E\left[\sum_{k=0}^{T-1} x_{k+1}^{\delta,v} \tilde{Q} x_{k+1}^{\delta,v} + v_k \tilde{R} v_k\right], \tag{8.10a}$$

$$\text{subject to: } x_{k+1}^{\delta,v} = A_k x_k^{\delta,v} + B_k v_k, \quad (k = 0, \ldots, T-1) \tag{8.10b}$$

where the matrices $\tilde{Q}$ and $\tilde{R}$ are tunable parameters. Note that in general, the cost function (8.10a) differs from the cost (2.6a) in the SMPC problem. Intuitively, the FTOC problem tries to keep the trajectory close to the nominal one, while the SMPC problem (2.6) computes a sequence of *offsets* $c_{0:T-1}^v$ to achieve the objectives specified by (2.6a) while satisfying the state and input constraints. The optimal solution to (8.10) is given by the state-feedback law:

$$v_k = K_k x_k^{\delta,v}. \tag{8.11}$$

The feedback gains are computed by the backward recursion in Algorithm 8.1.

Algorithm 8.1 FTOC recursion

```
1: Initialize:
   $S_T = \tilde{Q}$
2: for k = $T-1$ to 0:
3:     $K_k = -(B_k^T S_{k+1} B_k + \tilde{R})^{-1} B_k^T S_{k+1} A_k$
4:     $S_k = \tilde{Q} + A_k^T S_{k+1} A_k + A_k^T S_{k+1} B_k K_k$
Output: $\{K_k\}_{k=0}^{T-1}$
```

8.2.3 Uncertainty propagation

The key element of the SMPC approach is the characterization of the uncertainty in the predicted states $x_{1:T}^{\delta,v}$. By the feedback policy (8.9), the inputs $u_{0:T-1}^{\delta,v}$ are a function of the uncertain states $x_{0:T-1}^{\delta,v}$. Hence, $u_{0:T-1}^{\delta,v}$ are also stochastic and the uncertainty therein must be characterized.

Substituting for $u_k^{\delta,v}$ in the LTV dynamics (8.5) using the feedback policy (8.9) yields:

$$\begin{aligned} x_{k+1}^{\delta,v} &= \Phi_k x_k^{\delta,v} + B_k c_k^v + D_k d_k^{\delta,v}, \\ \Phi_k &= A_k + B_k K_k. \end{aligned} \tag{8.12}$$

As the initial state $x_{0|t}^{\delta,v} = x_t^{\delta,v}$ and the disturbance $d_k^{\delta,v}$ are assumed to be normally distributed (by Assumption 8.1), the mean $\hat{x}_k^{\delta,v}$ and variance Σ_k^v of the state evolve as:

$$\hat{x}_{k+1}^{\delta,v} = \Phi_k \hat{x}_k^{\delta,v} + B_k c_k^v, \tag{8.13a}$$

$$\Sigma_{k+1}^v = \Phi_k \Sigma_k^v \Phi_k^T + D_k \Sigma_k^d D_k^T, \tag{8.13b}$$

initialized with $\hat{x}_{0|t}^{\delta,v} = 0$ and $\Sigma_{0|t}^v = \Sigma_t^v$. As the control policy in (8.9) is affine in the state, the input $u_k^{\delta,v}$ is distributed as:

$$u_k^{\delta,v} \sim \mathcal{N}(\hat{u}_k^{\delta,v}, \Sigma_k^u), \tag{8.14}$$

with:

$$\hat{u}_k^{\delta,v} = K_k \hat{x}^{\delta,v} + c_k^v, \tag{8.15a}$$

$$\Sigma_k^u = K_k \Sigma_k^v K_k^T. \tag{8.15b}$$

Effect of state feedback

From (8.13b), we see that the closed-loop formulation (8.9) has an interesting consequence. By allowing the future control input $u_k^{\delta,v}$ to be a function of the future measured state $x_k^{\delta,v}$, we assume that measurements in the future will be available. This is opposed to the open-loop approach where $K_{0:T-1}$ is set to zero and $c_{0:T-1}^v$ is the open-loop control sequence to be optimized. If $K_{0:T-1}$ stabilizes the LTV system (8.5), the predicted variance $\Sigma_{1:T}^v$ is smaller than in the open-loop case with $K_{0:T-1} = 0$.

Remark 8.4. *By* (8.13b)*, the predicted variance $\Sigma^v_{1:T}$ over the horizon is independent of the modified control inputs $c^v_{0:T-1}$ due to the linearity of the system dynamics (Assumption 8.2) and the Gaussian characterization of the uncertainty (Assumption 8.1). This implies that the uncertainty propagation can be performed a-priori, that is, outside the numerical optimization problem where the inputs $c^v_{0:T-1}$ are computed [120, 124]. The above property is further discussed in Section 8.2.5.*

8.2.4 Constraint tightening

The characterization of the state uncertainty allows us to write the probabilistic constraints (8.8) as deterministic constraints on the mean $\hat{x}^{\delta,v}_k$. We first consider the case of single chance-constraints where G^e_k is a row vector and h^e_k is a univariate Gaussian random variable.

Theorem 8.5 (Chance-constraint tightening)**.** *A necessary and sufficient condition for the chance-constraint* (8.8) *to be satisfied is:*

$$G^e_k \hat{x}^{\delta,v}_k \leq h^e_k - \gamma(\epsilon_k), \tag{8.16}$$

$$\gamma(\epsilon_k) = \mathrm{erf}^{-1}(1-2\epsilon_k)\sqrt{2\begin{bmatrix} G^e_k & -1 \end{bmatrix}\begin{bmatrix} \Sigma^v_k & 0 \\ 0 & \Sigma^h_k \end{bmatrix}\begin{bmatrix} (G^e_k)^T \\ -1 \end{bmatrix}}. \tag{8.17}$$

The proof is detailed in [71]. The main idea is to compute the tightening offset $\gamma(\cdot)$ based on the tail of the univariate normal random variable $G^e_k x^{\delta,v}_k - h^e_k$.

In the case of joint chance-constraints where G^e_k is a matrix and h^e_k is a vector, we employ Boole's inequality to decompose the original constraint into several univariate constraints.

Theorem 8.6 (Joint chance-constraint tightening)**.** *Let n_c be the number of constraints. That is, $h^e_k \in \mathbb{R}^{n_c \times 1}$. A sufficient condition for the chance-constraint* (8.8) *to be satisfied is the existence of $\{\epsilon_{k_i}\}_{i=1}^{n_c}$ such that:*

$$G^e_{k_i} \hat{x}^{\delta,v}_k \leq h^e_{k_i} - \gamma(\epsilon_{k_i}), \quad (i = 1, \ldots, n_c) \tag{8.18a}$$

$$\sum_{i=1}^{n_c} \epsilon_{k_i} \leq \epsilon_k, \tag{8.18b}$$

$$\epsilon_{k_i} \geq 0, \quad (i = 1, \ldots, n_c) \tag{8.18c}$$

where $\gamma(\cdot)$ is defined in (8.17)*. $G^e_{k_i}$ and $h^e_{k_i}$ denote the i^{th} row of G^e_k and h^e_k, respectively.*

Intuitively, the variable ϵ_{k_i} determines the amount of the total risk ϵ_k is allocated to the i^{th} constraint. The constraint (8.18b) ensures that the total risk is less than the maximum allowed risk ϵ_k.

Remark 8.7. *The variables $\{\epsilon_{k_i}\}_{i=1}^{n_c}$ can be added as optimization variables to the SMPC problem to determine the optimal risk allocation [83]. In our implementation, however, we treat the aforementioned variables as tunable parameters. A common choice is:*

$$\epsilon_{k_i} = \frac{\epsilon_k}{n_c}, \tag{8.19}$$

which corresponds to a uniform risk allocation between constraints.

8.2.5 Deterministic optimization problem

In summary, the analytical SMPC approach is based on the following steps:

Step 1: Linearization of the system dynamics around a nominal trajectory (Assumption 8.2).

Step 2: Affine parameterization of the feedback control policy (Section 8.2.2)

Step 3: Propagation of the mean and variance of the state x_k^v using the linearized dynamics, affine policy and a Gaussian representation of the uncertainty (Assumption 8.1, Section 8.2.3)

Step 4: Tightening of the linearized chance-constraints (8.8) using the predicted variance of the state (Assumption 8.3, Section 8.2.4).

The above steps yield the following deterministic CFTOC problem:

$$\begin{aligned}
&\min_{c^v_{0:T-1}} \sum_{k=0}^{T-1} J(\hat{x}^v_{k+1}, \hat{u}^v_k, \hat{u}^v_{k-1}, x^{\text{ref}}_{k+1}, u^d_k), && \text{(8.20a)}\\
&\text{subject to: } \hat{x}^{\delta,v}_{k+1} = \Phi_k \hat{x}^{\delta,v}_k + B_k c^v_k, && \text{(8.20b)}\\
&\hat{u}^v_k = \bar{u}^v_k + K_k \hat{x}^{\delta,v}_k + c^v_k, && \text{(8.20c)}\\
&[\hat{u}^v_{k|t},\ \hat{u}^v_{k-1|t}] \in \mathcal{U}, && \text{(8.20d)}\\
&G^e_{k_i} \hat{x}^{\delta,v}_k \le h^e_{k_i} - \gamma(\epsilon_{k_i}), \quad (i = 1, \dots, n_c) && \text{(8.20e)}\\
&\hat{x}^{\delta,v}_k = \hat{x}^v_k - \bar{x}^v_k, && \text{(8.20f)}\\
&(k = 0, \dots, T-1) &&\\
&\hat{x}^v_{0|t} = \hat{x}^v_t, \quad \hat{u}^v_{-1|t} = u^v_{t-1}. && \text{(8.20g)}
\end{aligned}$$

The variables $\{\epsilon_{k_i}\}_{i=1}^{n_c}$ are assumed to satisfy the risk allocation constraints (8.18b)–(8.18c).

Recall that the computation of the predicted variance $\Sigma^v_{1:T}$ is independent of the decision variables $c^v_{0:T-1}$ (see Remark 8.4). Combined with the analytical tightening of the chance-constraints, it can be seen that the complexity of the resulting numerical optimization problem is the same as in the nominal case where $\Sigma^v_{1:T}$ is set to zero. The additional effort comes from the computation of the gains $K_{0:T-1}$ (by Algorithm 8.1) and the predicted variance $\Sigma^v_{1:T}$ (by (8.13b)) which are performed outside the optimization loop.

In the autonomous driving application considered in this work, the cost function $J(\cdot)$ is quadratic in the vehicle states and inputs. This allows us to use the means $\hat{x}^v_k$ and $\hat{u}^v_k$ to compute the expected value of the cost function in (8.20a). With the aforementioned choice of the cost function, the CFTOC problem (8.20) is a QP, which makes it amenable to the iterative linearization procedure developed in Chapter 5. Concretely, the problem (8.20) can be solved iteratively at each sampling time, where the nominal trajectory $\{\bar{x}^v_{0:T}, \bar{u}^v_{0:T-1}, \bar{d}^v_{0:T-1}\}$ is updated at each iteration using the solution at the previous iteration. An additional trust-region constraint and penalty on the deviation from the nominal trajectory can be added as in the algorithm presented in Section 5.3.3. The benefit of the iterative procedure is that encourages the optimal trajectory to be close to the nominal one, thus maintaining the validity of the LTV model around the optimal trajectory. Simulation results showing the application of the analytical SMPC approach to the autonomous navigation problem are discussed in Section 8.5.

Remark 8.8. *The analytical SMPC approach presented above assumes that the full state of the vehicle can be directly measured at every time step, as seen by the control policy parameterization in* (8.9). *In partially observable environments, additional uncertainty due to measurement noise must be considered. This affects the closed-loop policy and uncertainty propagation steps in Sections 8.2.2 and 8.2.3, respectively. A detailed treatment of the control design in the presence of measurement uncertainty is provided in Appendix B and is based on the work in [120, 124].*

8.3 Sampling-based Stochastic MPC

This section reviews the recent developments in sampling-based SMPC or Scenario MPC (SCMPC) with a focus on application to the autonomous driving domain [50, 105]. We use an open-loop formulation where we directly optimize on the control inputs u^v_k. The approach can be generalized to optimize parameterized feedback policies $\kappa^c_k(\cdot)$ as in the case of the analytical SMPC approach. Moreover, we assume full state feedback as the consideration of measurement uncertainty is non-trivial.

8.3.1 Sampling-based predictions

In SCMPC, uncertain variables in the problem are represented by *scenarios*, which are sampled realizations of the uncertainty. We focus only on the uncertainty associated with the vehicle and the environment as the subsequent applications do not involve a driver in the loop. Let $\{d^{v^{(s)}}_k\}^S_{s=1}$ and $\{d^{e^{(s)}}_k\}^S_{s=1}$ denote S samples of the random variables d^v_k and d^e_k, respectively. A scenario $\omega^{(s)}$ is defined as the full horizon sample:

$$\omega^{(s)} = \{(d^{v^{(s)}}_0, d^{e^{(s)}}_0), \dots, (d^{v^{(s)}}_{T-1}, d^{e^{(s)}}_{T-1})\}. \tag{8.21}$$

Consequently, by the dynamics of the vehicle and the environment, we get samples of the states $\{x_k^{v(s)}\}_{s=1}^S$ and $\{x_k^{e(s)}\}_{s=1}^S$. Note that while the sampled vehicle states $\{x_k^{v(s)}\}_{s=1}^S$ are deterministic, they are functions of the control input $\{u_k^v\}_{k=0}^{T-1}$. That is:

$$x_{k+1}^{v(s)} = f^v(x_k^{v(s)}, u_k^v, d_k^{v(s)}) \quad \forall s = 1, \ldots, S. \quad (k = 0, \ldots, T-1) \tag{8.22}$$

The main idea of SCMPC is to impose the nonlinear safety constraints (4.34) for all possible scenarios:

$$g(x_k^{v(s)}, x_k^{e(s)}) \leq 0 \quad \forall s = 1, \ldots, S. \quad (k = 1, \ldots, T) \tag{8.23}$$

8.3.2 Deterministic optimization problem

Using the sampled dynamics (8.22) and constraints (8.23), we obtain the following deterministic CFTOC problem:

$$\begin{aligned}
&\min_{u_{0:T-1}^v} \sum_{k=0}^{T-1} \frac{1}{S} \sum_{s=1}^{S} J(x_{k+1}^{v(s)}, u_k^v, u_{k-1}^v, x_{k+1}^{\text{ref}}, u_k^d), && \text{(8.24a)}\\
\text{subject to: } & x_{k+1}^{v(s)} = f^v(x_k^{v(s)}, u_k^v, d_k^{v(s)}) \quad \forall s = 1, \ldots, S, && \text{(8.24b)}\\
& [u_{k|t}^v,\ u_{k-1|t}^v] \in \mathcal{U}, && \text{(8.24c)}\\
& g(x_k^{v(s)}, x_k^{e(s)}) \leq 0 \quad \forall s = 1, \ldots, S, && \text{(8.24d)}\\
& (k = 0, \ldots, T-1) && \\
& x_{0|t}^{v(s)} = \hat{x}_t^v \quad \forall s = 1, \ldots, S, \quad u_{-1|t}^v = u_{t-1}^v. && \text{(8.24e)}
\end{aligned}$$

The above problem is nonconvex in general and can be solved using a nonlinear optimization package.

8.3.3 Choice of sample size

Recent results in SCMPC have developed a relationship between the number of scenarios S and the satisfaction of the chance-constraint (2.9) [50]. Intuitively, the larger the number of samples, the lower the probability ϵ of violating the safety constraints $g(x_k^v, x_k^e) \leq 0$. The work in [50] formalizes this relationship, which is based on the notion of the *support rank* of the chance-constraint (2.9). The support rank ρ_k is defined as the dimension of the linear subspace spanned by the constraint $g(x_k^v, x_k^e) \leq 0$.

Let V_t denote the probability that the constraint (8.24d) is violated at the next time step $(t+1)$ with the optimal control input u_t^v obtained by solving the CFTOC (8.24). Given a sample size S, the expected value of V_t is bounded above as [50]:

$$E[V_t] \leq \frac{\rho_1}{S+1}. \tag{8.25}$$

Thus, in order to achieve a closed-loop violation probability of ϵ, a sample size $S \geq (\rho_1/\epsilon) - 1$ is required, which is proportional to the support rank. Fortunately, due to the nature of the collision avoidance constraints that define $g(\cdot)$, the support rank is low. For example, in [105], it is shown that the support rank is 1 for the specific formulation of the safety constraints. In [33], the support rank is shown to be bounded by 2 for the case of an overtaking scenario. An upper bound on the number of support constraints exists for the general autonomous driving scenario due to the limited number of vehicles that can surround the ego vehicle at any given time.

Scenario removal

The SCMPC approach allows for the removal of scenarios after they have been sampled. This helps protect against outliers generated from the random sampling process which may cause an abrupt change in the control policy. The discarding of scenarios affects the relationship between the number of remaining samples and the probability of constraint violation (see [24, 50] for details). Intuitively, if the number of remaining scenarios is the same as the original number (that is, with no removal), the closed-loop constraint violation probability is higher. In general, systematic scenario removal requires the CFTOC problem (8.24) to be solved several times at each time step which is computationally challenging for real-time implementation. However, simpler heuristics for determining the most extreme scenarios can be employed in the autonomous driving problem. For example, when dealing with forecasts of a target vehicle, if its lane change probability is estimated to be sufficiently low, scenarios corresponding to the lane change maneuver can be discarded. We use a similar domain-specific scheme in our implementation.

8.4 Hybrid Stochastic Model Predictive Control

In the next section, we compare the analytical and sampling-based SMPC approaches and propose an approach that combines the benefits of both methods. We refer to this as the *hybrid* SMPC approach.

8.4.1 Analytical vs. sampling-based SMPC

The main limitation of the analytical SMPC approach presented in Section 8.2 is that the characterization of all system uncertainties is restricted to Gaussian distributions. While this may be a reasonable assumption for modeling errors and disturbances in the vehicle dynamics (see Section 3.6), this is not the case for uncertainty in the forecasts of surrounding vehicles. For example, a target vehicle in the process of changing lanes induces a multimodal distribution on its future position. This is shown using data collected from our experimental vehicle. We analyzed about 50 minutes of driving data on highways around Berkeley which included lane keeping and lane changing maneuvers. Using the data, a Gaussian Mixture

Model (GMM) was learned which models the joint probability distribution between the vehicle's current lateral position and velocity, and the lateral position at a future time step. In order to use the model to predict the distribution of the future positions of the vehicle, we use Gaussian Mixture Regression (GMR) [21] to compute the conditional distribution of the future lateral position given the current lateral position and velocity.

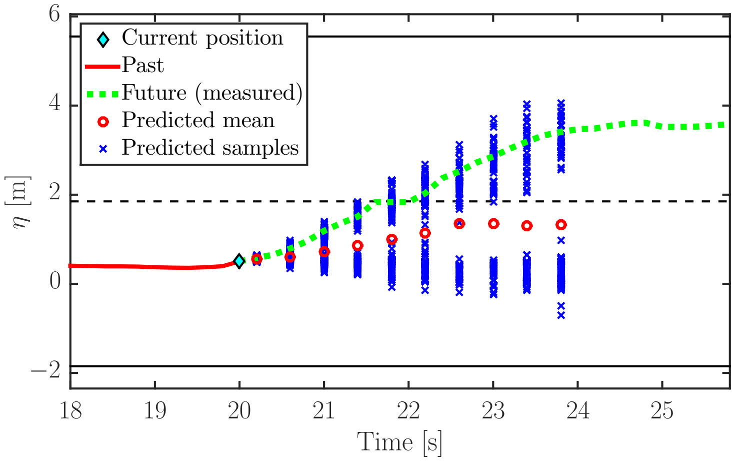


Figure 8.3: Sampled predictions of the vehicle's lateral position starting from its current state at $t = 20$ s. The solid red line depicts its past trajectory while the dotted green line shows the actual future trajectory which is not known to the model. The red circles depict the means of the predictions at the corresponding time steps.

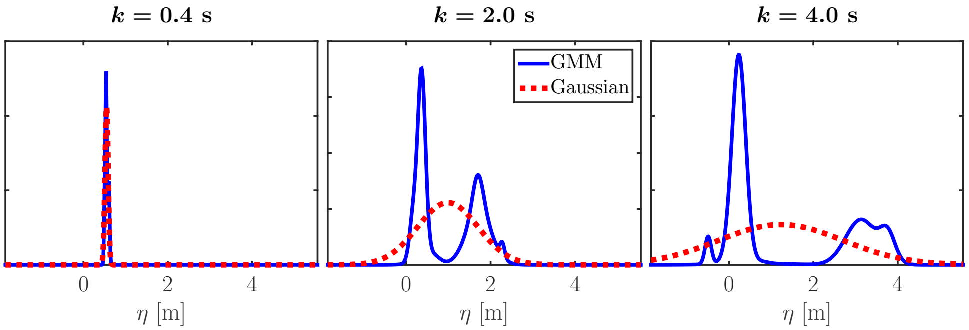


Figure 8.4: PDFs of the distributions of the predicted lateral position of the vehicle at 3 different time steps of the horizon. The solid blue line shows the GMM characterization of the distribution while the dotted red line is the Gaussian approximation.

Figures 8.3 and 8.4 show the results of the prediction for a situation where the vehicle is about to initiate a lane change. In Figure 8.3, the sampled predictions of the vehicle's lateral position over a horizon of 4 s are shown, starting from its current position at $t = 20$ s. Based on the driving data, our model predicts that the vehicle could either change lanes (reflected by the samples going towards the left) or keep its current lane (reflected by the samples returning towards the lane center). The actual future trajectory of the vehicle is a lane change (depicted by the green line), but this is not known to the prediction model. The red circles show the means of the predictions over the horizon assuming a Gaussian approximation of the distribution. Clearly, the predicted trajectory does not make physical sense as it lies in between the lane keeping and changing maneuvers. Figure 8.4 shows the PDFs of the distributions of the predicted lateral position of the vehicle at 3 different time steps of the horizon. Our conclusion is that the Gaussian assumption is not a valid characterization of the uncertainty in traffic forecasts. Another situation in which the Gaussian assumption breaks down is in the case of a vehicle approaching an uncontrolled intersection where it may decide to go straight or turn left or right. Thus, any distribution on its future position must be multimodal.

The sampling-based SMPC in Section 8.3 resolves this issue by representing system uncertainties by samples. The only requirement of the system model is that it can generate samples of the future states. This is an additional benefit as compared to the analytical approach where an explicit characterization of the mean and variance over the horizon is required. The sampled representation, however, does come at a computational cost. As seen from (8.22) and (8.23), there is one equality constraint (corresponding to the dynamics) and one inequality constraint (corresponding to the safety constraints) per scenario for each time step of the horizon. This yields S times the number of equality and inequality constraints as compared to the nominal MPC problem, making it computationally formidable for real-time applications. Our aim is to combine the benefits of the analytical and sampling-based SMPC approaches to obtain a real-time control scheme capable of handling uncertainty in the vehicle, driver and environment states. Our algorithm is similar in spirit to the work in [123], where the uncertain parameters of the linear constraints are sampled and a Gaussian representation of the system state is used to tighten the chance-constraints.

8.4.2 Aggregating samples due to environment uncertainty

We make the following observation regarding the sampled constraints (8.23). Assume no uncertainty in the vehicle state x_k^v. That is, d_k^v is equal to a nominal value $\bar{d}_k^v$ with probability one, and there is one sample $x_k^{v^{(1)}}$ corresponding to the nominal value $\bar{x}_k^v$.

Our claim is that, in most cases, it is possible to *aggregate* the samples $\{x_k^{e^{(s)}}\}_{s=1}^S$ (at each predicted time step k) into a reduced set of samples by exploiting the structure of the driving problem. We provide an illustration by means of Figure 8.5. Consider the scenario in Figure 8.5a where the target vehicle T_1 is in the process of changing lanes. Two nominal predicted trajectories are shown; the blue and red lines correspond to the left lane change and lane keeping maneuvers, respectively. The rectangles represent the sampled predictions of T_1 at the end of the prediction horizon of 15 time steps as given by the environment model. The blue and red rectangles correspond to the left lane change and lane keeping maneuvers, respectively. We draw 5 samples for each of the maneuvers, corresponding to a lane change probability of 50%. Sampled predictions for the other time steps in the horizon are not shown for the sake of clarity.

Figures 8.5b and 8.5c show two approaches for aggregating samples arising from environment uncertainty, referred to as Methods 1 and 2, respectively. In Method 1, samples corresponding to the same maneuver as combined together. This can be done by computing a rectangular bounding box aligned with the road as depicted by the green rectangle in Figure 8.5b. Another approach is to compute the convex hull of the relevant samples [33]. Method 1 yields one aggregated sample per maneuver, hence the reduced number of samples is equal to the number of maneuvers. In our illustration, we get 2 aggregated samples (corresponding to the lane change and lane keeping maneuvers), which is depicted by the green polygons in Figure 8.5b.

In Method 2 (Figure 8.5c), all samples of the target vehicle's predicted pose all combined together. Again, this can be done by computing a rectangular bounding box aligned with the road (green rectangle in Figure 8.5c) or by taking the convex hull. Method 2 results in one aggregated sample per time step of the horizon, but is clearly more conservative than Method 1. The former is suitable for highway driving situations such as that in Figure 8.5a. Here, a single aggregated sample which spans the width of the roadway makes physical sense as this would cause the ego vehicle to stay behind T_1 till it completes the lane change maneuver. Method 1 is suitable for urban driving scenarios such as intersection navigation. For example, in the case of a target vehicle approaching an intersection, the environment model yields predictions corresponding to multiple maneuvers such as going straight, turning left or right. Here, a single aggregated sample would not make physical sense and would be overly conservative.

We formalize the notion of sample aggregation by introducing the following assumption:

Assumption 8.9 (Sample aggregation)**.** *At each time step k of the prediction horizon, the samples $\{x_k^{e^{(s)}}\}_{s=1}^{S}$ corresponding to environment uncertainty can be aggregated into a reduced sample set $\{\tilde{x}_k^{e^{(s)}}\}_{s=1}^{\widetilde{S}}$ by the methods presented above such that:*

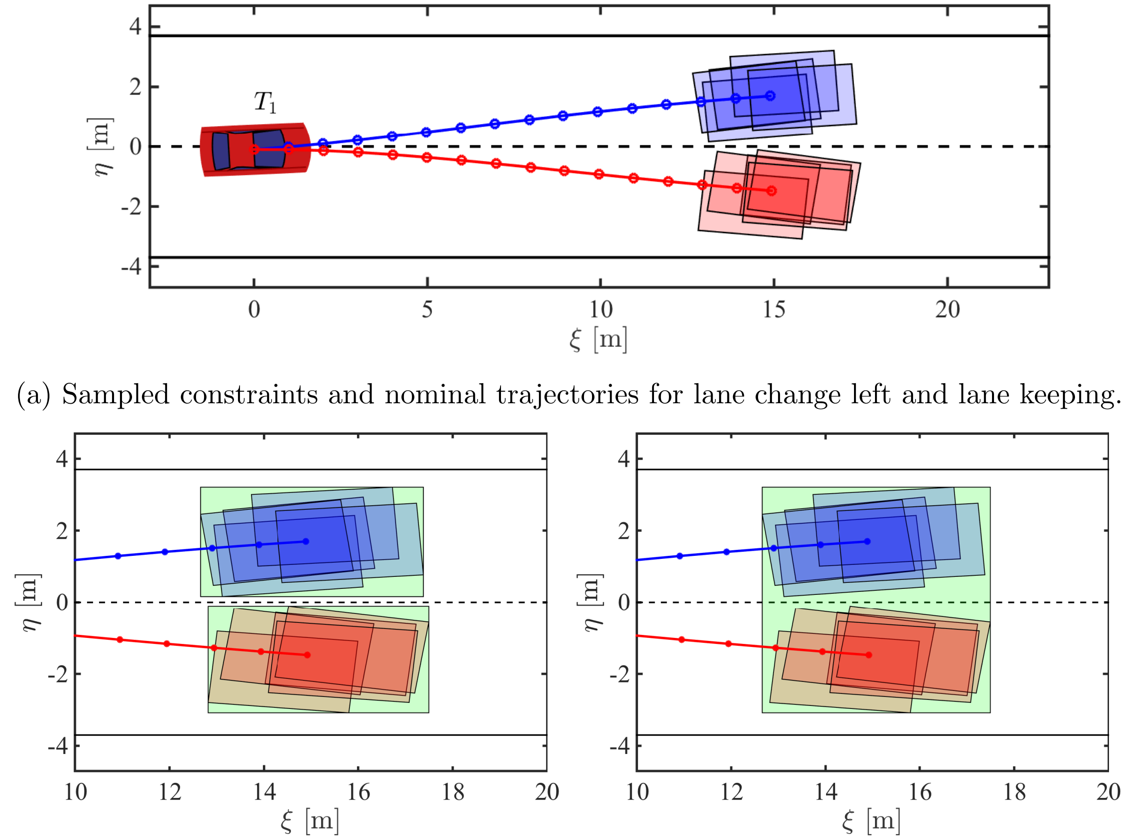
$$g(x_k^v, \tilde{x}_k^{e^{(s)}}) \leq 0 \quad \forall s = 1, \dots, \widetilde{S} \quad \Longrightarrow \quad g(x_k^v, x_k^{e^{(s)}}) \leq 0 \quad \forall s = 1, \dots, S. \tag{8.26}$$

Typically, $\widetilde{S} \ll S$.

Equation (8.26) follows from the fact that Methods 1 and 2 perform a conservative aggregation of the samples via the convex hull or bounding box. Note that the sampled states $\{x_k^{e^{(s)}}\}_{s=1}^{S}$ and $\{\tilde{x}_k^{e^{(s)}}\}_{s=1}^{\widetilde{S}}$ not only contain information such as the position and orientation of the target vehicles, but also their geometries. The effective dimensions of $\tilde{x}^{e^{(s)}}$ are larger than those of the samples $x^{e^{(s)}}$ they contain.

8.4.3 Accounting for uncertainty in the vehicle dynamics

The approach presented in Section 8.4.2 for dealing with uncertainty in the environment forecasts does not consider the uncertainty in the vehicle state x_k^v. In the presence of this



(a) Sampled constraints and nominal trajectories for lane change left and lane keeping.

(b) Method 1 for sample aggregation: Samples corresponding to each maneuver are combined into a single sample by computing a rectangular bounding box. The number of reduced samples is equal to the number of maneuvers (2 in this case).

(c) Method 2 for sample aggregation: All samples are combined into a single sample by computing a rectangular bounding box. The number of reduced samples is equal to 1.

Figure 8.5: Illustration of the proposed sample aggregation strategies.

uncertainty, we use the analytical SMPC approach of Section 8.2 to propagate the mean and variance of the state x_k^v over the horizon. This is based on a Gaussian characterization of the uncertainty (Assumption 8.1) and a LTV model of the vehicle dynamics (Assumption 8.2). Using the approach presented in 8.2.3, we can compute the mean $\hat{x}_k^v$ and variance Σ_k^v of the state x_k^v over the horizon as a function of the inputs c_k^v.

8.4.4 Constraint tightening

The final step in the hybrid SMPC approach is the tightening of the sampled constraints 8.26 for the reduced set of samples $\widetilde{S}$. As in Assumption 8.3, for a tractable determination of the tightening offsets $\gamma(\cdot)$, we linearize the nonlinear constraint mapping $g(\cdot)$. Specifically, at time step k of the prediction horizon, the convex approximation of the safe region $g(x_k^v, \tilde{x}_k^{e^{(s)}})$ corresponding to the s^{th} aggregated sample is given by:

$$G_k^{e^{(s)}}(x_k^v - \bar{x}_k^v) \leq h_k^{e^{(s)}}, \quad \forall s = 1, \dots, \widetilde{S}, \tag{8.27}$$

with:

$$G_k^{e^{(s)}} = \nabla_{x_k^v} g(x_k^v, x_k^e)|_{\bar{x}_k^v, \tilde{x}_k^{e^{(s)}}}, \tag{8.28a}$$

$$h_k^{e^{(s)}} = -g(\bar{x}_k^v, \tilde{x}_k^{e^{(s)}}). \tag{8.28b}$$

The constraint tightening procedure presented in Section 8.2.4 can now be applied to obtain the following deterministic constraint on the mean $\hat{x}_k^v$:

$$G_k^{e^{(s)}}(\hat{x}_k^v - \bar{x}_k^v) \leq h_k^{e^{(s)}} - \gamma(\epsilon_k^{(s)}). \tag{8.29}$$

The risk allocation constraints: $\sum_{s=1}^{\widetilde{S}} \epsilon_k^{(s)} \leq \epsilon_k,\ \epsilon_k^{(s)} \geq 0$, are assumed to be satisfied.

8.4.5 Hybrid approach

In summary, the outline of the hybrid SMPC approach is as follows:

Step 1: Use the analytical SMPC methodology to characterize and propagate the uncertainty in the vehicle state x_k^v over the horizon. In order to do this efficiently, we use a Gaussian representation of the uncertainty in the vehicle dynamics (Assumption 8.1) and a linearized vehicle model (Assumption 8.2).

Step 2: Use a sampled representation of the environment state x_k^e and the aggregation procedure in 8.4.2 to obtain a reduced number of sampled constraints per time step of the horizon.

Step 3: Tighten the aggregated safety constraints as a function of the predicted variance in the vehicle state x_k^v. This requires a linearization of the safety constraints computed with respect to the aggregated samples (Assumption 8.3).

The above approach yields the following deterministic CFTOC problem:

$$\min_{c_{0:T-1}^v} \sum_{k=0}^{T-1} J(\hat{x}_{k+1}^v, \hat{u}_k^v, \hat{u}_{k-1}^v, x_{k+1}^{\text{ref}}, u_k^d), \tag{8.30a}$$

$$\text{subject to: } \hat{x}_{k+1}^{\delta,v} = \Phi_k \hat{x}_k^{\delta,v} + B_k c_k^v, \tag{8.30b}$$

$$\hat{u}_k^v = \bar{u}_k^v + K_k \hat{x}_k^{\delta,v} + c_k^v, \tag{8.30c}$$

$$[\hat{u}_{k|t}^v,\ \hat{u}_{k-1|t}^v] \in \mathcal{U}, \tag{8.30d}$$

$$G_k^{e^{(s)}} \hat{x}_k^{\delta,v} \leq h_k^{e^{(s)}} - \gamma(\epsilon_k^{(s)}), \quad (s = 1, \dots, \widetilde{S}) \tag{8.30e}$$

$$\hat{x}_k^{\delta,v} = \hat{x}_k^v - \bar{x}_k^v, \tag{8.30f}$$
$$(k = 0, \dots, T-1)$$
$$\hat{x}_{0|t}^v = \hat{x}_t^v, \quad \hat{u}_{-1|t}^v = u_{t-1}^v. \tag{8.30g}$$

Problem (8.30) is a QP. To improve the quality of the solution, the iterative linearization procedure developed in Chapter 5 can be applied as discussed in Section 8.2.5. In addition to linearizing the dynamics and safety constraints, each iteration of this procedure would involve computing the gains $K_{0:T-1}$, variances $\Sigma_{1:T}^v$ and tightening offsets $\gamma(\epsilon_k^{(s)})$. The computational complexity can be further reduced by keeping the aforementioned quantities (that is, the gains, variances and tightening offsets) fixed over the iterations performed at a given time step. This approach is used in the experiments performed in Section 8.6.

Remark 8.10 (Safety guarantees)**.** *Note that due to the hybrid analytical and sampling approach to deal with the chance-constraints, no formal guarantees of the constraint violation probability as function of the number of scenarios S and the parameter ϵ_k can be obtained. However, our method allows for a systematic treatment of the uncertainty in the vehicle and environment forecasts.*

8.5 Simulation results

This section demonstrates the application of the analytical SMPC scheme of Section 8.2 to the autonomous navigation problem in urban environments. We explore the trade-off between risk and conservatism that can be obtained by tuning the constraint violation probability used in the controller design. The effect of using multimodal distributions of the future locations of target vehicles as opposed to Gaussian distributions is also studied.

8.5.1 Simulation setup

Simulations are performed in MATLAB and the CFTOC problem (8.20) is solved using Gurobi [57]. The controller is connected in closed–loop with a higher fidelity four–wheel nonlinear model (which uses a Pacejka tire model) to simulate model mismatch. The physical parameters of the prototype Hyundai vehicle described in Appendix A.2 are used.

8.5.2 Test scenario

The scenario for the simulation is shown in Figure 8.6a. The ego vehicle E is moving at a speed of 10 m/s in the left lane with the objective of staying in the left lane and increasing its speed to 15 m/s. The target vehicles T_1 and T_2 are in the right lane at the start of the simulation. T_1 moves at a speed of 8 m/s and is in the process of changing its lane to the left lane. Its lane change trajectory is generated using a sigmoid function. T_2 moves in a straight line at a speed of 10.5 m/s and stays in the right lane for the entire duration of the simulation.

8.5.3 Technical approach

Vehicle model: The dynamic bicycle model described in Section 3.2.2 with a linear tire model is used for the control design. The position and orientation of the vehicle are expressed in the road-aligned frame. Hence, $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, \xi_t, \eta_t, \phi_t]^T$ and $u_t^v = [\delta_{f_t}, a_{x_t}]^T$.

Reference generation: A simple reference generator which provides speed and lateral position set-points is used in this example. These constitute the reference state sequence $x_{1:T|t}^{\text{ref}}$. In particular, the reference speed is 15 m/s while the desired lateral position is the center of the left lane.

Environment model: The multiple-model filtering approach presented in Section 4.2 is used to estimate and predict the positions of the target vehicles. For the predictions, the dynamics corresponding to the most likely mode are used to propagate the mean and variance of the state of each target vehicle over the horizon (see Section 4.2.6 for details).

Controller: We use the analytical SMPC approach presented in Section 8.2. The collision avoidance constraints are formulated using the signed distance approach of Section 4.5.2. The SMPC problem is solved with a constraint violation probability $\epsilon_k = 0.004$ in (8.8). This corresponds to a constraint satisfaction probability of 99.6%. The prediction horizon is $T = 20$ time steps with a discretization time of 0.1 s, resulting in a preview time of 2 s.

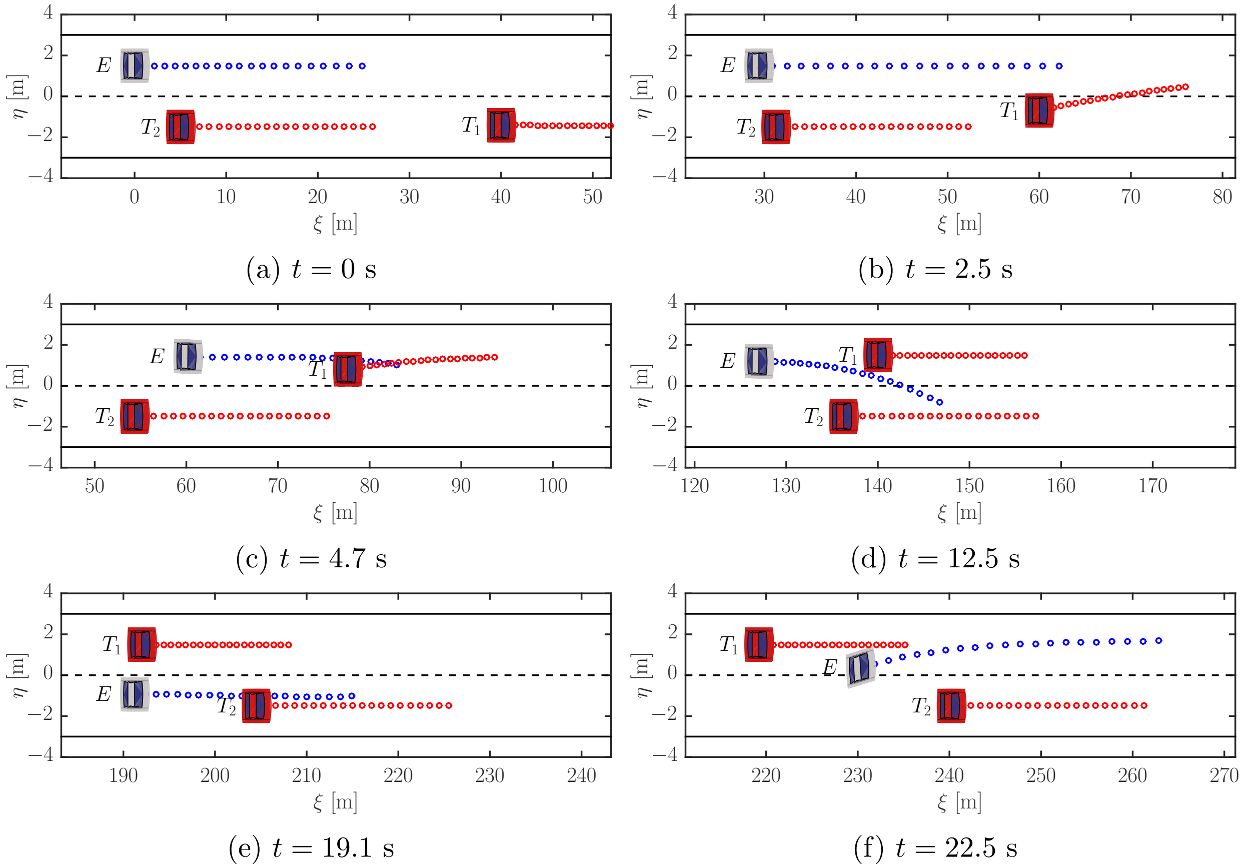
8.5.4 Results

Snapshots of the simulation at various times are shown in Figure 8.6. In each sub-figure, the blue and red circles depict the means of the open-loop (predicted) positions of the ego and target vehicles, respectively, at the corresponding time step of the simulation. The predicted positions of E are obtained from the solution of the optimization problem, and those of T_1 and T_2 are given by the environment model. The spacing between the predicted positions depicted by the circles is an indication of the predicted speed of the vehicles.

As seen in Figure 8.6b, the environment model predicts a lane change maneuver for T_1 at $t = 2.5$ s. The potential constraint violation causes the controller to command a braking action and decrease the speed of E. To attain the objective of maintaining the reference speed, the controller attempts to plan a path around T_1, but is prevented from doing so by the presence of T_2 (Figure 8.6c). When T_2 passes E and a sufficient safety distance is available, the controller plans and executes an overtaking maneuver as shown in Figures 8.6d and 8.6e. Finally, the controller plans a path in between T_1 and T_2 when feasible to cause E to return to its original lane and increase its speed (Figure 8.6f).

8.5.5 Risk vs. conservativeness

An important element of the SMPC problem is the risk parameter ϵ_k, defined as the probability of violating the probabilistic safety constraints. In order to study the effect of varying



(a) $t = 0$ s

(b) $t = 2.5$ s

(c) $t = 4.7$ s

(d) $t = 12.5$ s

(e) $t = 19.1$ s

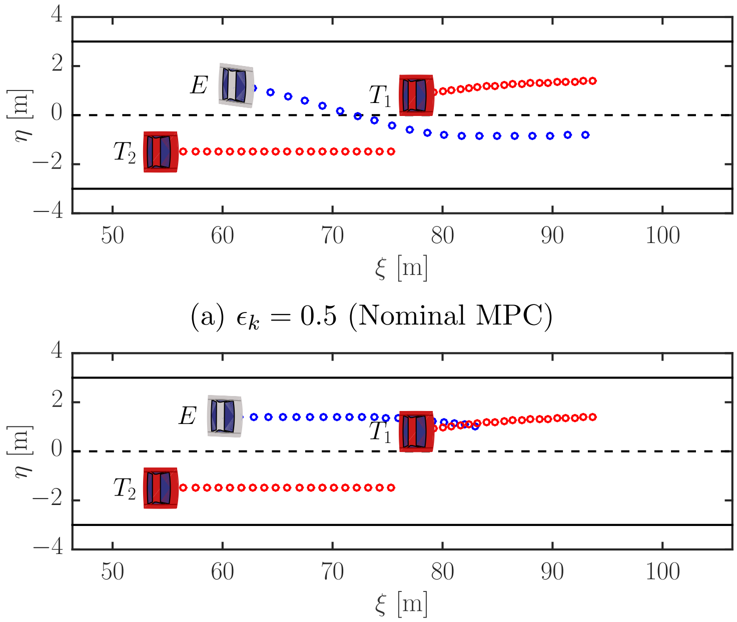
(f) $t = 22.5$ s

Figure 8.6: Snapshots of simulation with the ego vehicle E in blue and the target vehicles T_1 and T_2 in red. The blue and red circles are the predicted positions of the ego and target vehicles, respectively, over the horizon.

ϵ_k on the performance of the controller, we repeat the simulation presented above using the nominal approach. This is, all disturbances are assumed to take on their mean values and the amount of chance-constraint tightening in (8.18) is zero.

A snapshot of the simulation at $t = 4.7$ s for the nominal and stochastic MPC cases is shown in Figure 8.7. In the nominal case, the controller finds the more aggressive maneuver of changing lanes in between the two target vehicles to be feasible, as seen in 8.7a. This can be compared to the relatively conservative maneuver planned by the SMPC-based controller shown in Figure 8.7b, in which the vehicle stays behind T_1 and overtakes only when T_2 has passed.

An interesting statistic we compare in this simulation is the time taken by the ego vehicle to reach the 400 m mark. In the nominal case, the EV traverses this distance in 27.4 s. On the other hand, the SMPC strategy results in a commute time of 34.4 s for the same distance. For more complex scenarios encountered in day-to-day driving and longer distances, this difference will be much larger.



(a) $\epsilon_k = 0.5$ (Nominal MPC)

(b) $\epsilon_k = 0.998$ (Stochastic MPC)

Figure 8.7: Simulation showing the effect of varying the risk parameter ϵ_k on the conservatism of the controller. The notation is the same as that in Figure 8.6

In the context of urban driving, scenarios like the one described in our simulation study are expected to occur often. The use of conservative worst-case approaches may result in the ego vehicle stopping or slowing down unnecessarily, leading to longer commute times. Nominal approaches may lead to unsafe and aggressive behavior. The framework proposed in this work allows us to systematically trade-off between these two extremes.

8.5.6 Effect of traffic forecasts: Gaussian vs. multimodal

In the above example, predictions of the future states of the TVs is based on their *most likely mode* estimated by the multiple model filter. While this may be a reasonable assumption in most cases, the distributions of the future states tend to be multimodal as discussed in Section 8.4.1, especially in situations involving lane changes of target vehicles. Here, we show an example where a significant difference in the performance of the ego vehicle can be obtained depending on the forecast approach chosen.

The scenario consists of a single target vehicle T_1 moving erratically in the left lane at a constant speed of 8 m/s. The ego vehicle starts behind it in the right lane at a speed of 10 m/s with the objective of tracking a reference speed of 15 m/s, passing T_1 in doing so. The target vehicle T_1 is designed to be adversarial in the following sense. When the relative spacing between E and T_1 is below a certain threshold, the latter initiates a lane change into the path of E. However, T_1 aborts the lane change maneuver when it reaches the boundary of its own lane.

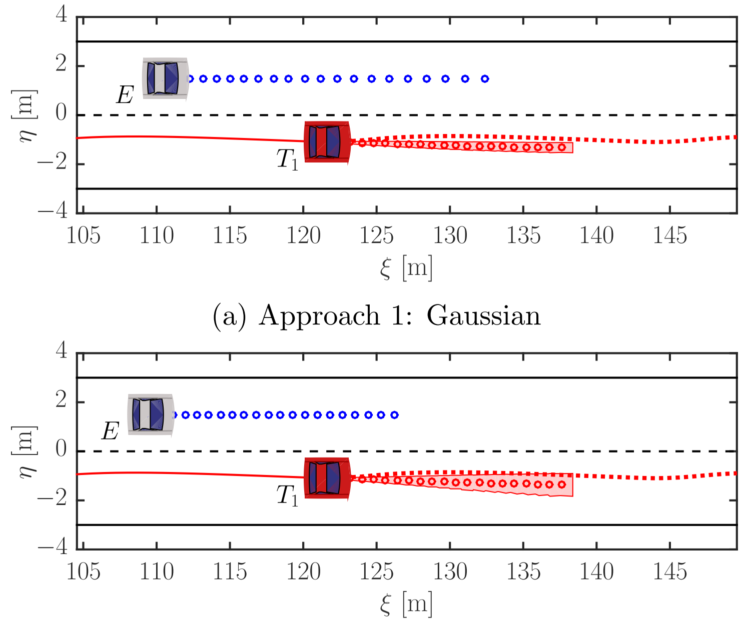
The multiple model filtering approach of Section 4.2 is used to estimate the intent and predict the motion of T_1. Two approaches for forecasting the motion of T_1 based on the output of the filter are compared. These have been presented in Section 4.2.6 and are briefly reviewed here. Recall that at each time step, the filter provides a distribution over the discrete modes and an estimate of the mean and variance of the TV state based on the history of measurements. In Approach 1, a Gaussian characterization of the uncertainty in the predictions is used. We estimate the most likely model m_t^{ML} and compute a Gaussian distribution of the future position of T_1 using the dynamics corresponding to m_t^{ML} (Equations 4.16–4.17). In Approach 2, we use a sampled representation of the uncertainty in the predictions. Here, we sample from the distribution over the discrete modes and use the corresponding dynamics to obtain a sample of the future target vehicle position (Algorithm 4.2). All samples are aggregated according to Method 2 described in Section 8.4.2. A simple scenario removal scheme is used to discard extreme scenarios corresponding to low probability maneuvers in order to reduce the conservatism of the controller.

The scene at $t = 10$ s is shown in Figure 8.8. The blue circles depict the means of the open-loop predictions of E, while the red ones show the mean predicted trajectory of T_1 over the horizon. The solid red line is the observed trajectory of T_1 so far while the dotted red line is its future trajectory which is not known to the traffic prediction model. The main difference between Approaches 1 and 2 lies in the predicted variance of the state of T_1, which is depicted by the red shaded region surrounding its predicted trajectory. When a Gaussian characterization based on the most likely model is used (Approach 1), T_1 is expected to return to the lane center. Moreover, the variance associated with the predictions is low. Hence, E plans to increase its speed and pass T_1 as is seen by the relatively larger spacing between the blue circles in Figure 8.8a. On the other hand, as Approach 2 considers the distribution over the modes, the variance associated with the predictions is relatively larger. This prevents E from planning a passing maneuver. Hence, E plans to slow down and stay behind T_1 which is observed by the relatively smaller spacing between the blue circles in Figure 8.8b.

Figure 8.9 shows the closed-loop speed profile of E over the duration of the simulation. There is a larger variation in the speed with Approach 1 due to the fact that the controller repeats the process of attempting a passing maneuver and then aborting it due to the change in the most likely model. On the other hand, Approach 2 yields a relatively cautious control policy and hence, a smoother speed profile as the controller never attempts a passing maneuver. This is closer to how a human driver would handle the given scenario.

8.6 Experimental Results

This section demonstrates the ability of the hybrid SMPC approach presented in Section 8.4 to run in real-time on our experimental vehicle and handle a typical urban driving scenario. We also show the benefit of accounting for the interaction between the ego and



(a) Approach 1: Gaussian

(b) Approach 2: Multimodal

Figure 8.8: Snapshots of the simulation at time $t = 10$ s. The blue and red circles are the open-loop predictions of E and T_1, respectively. The solid red line is the observed trajectory of T_1 so far while the dotted red line is its future trajectory which is not known to the forecast model.

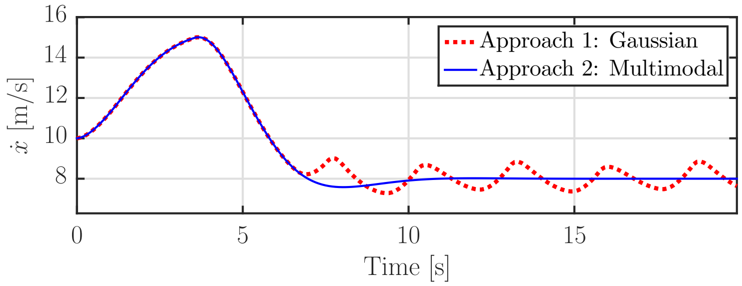


Figure 8.9: Comparison of closed-loop speed profiles using traffic forecasts based on the most likely mode (Gaussian, dotted red line) and the multimodal distribution generated by the multiple model filter (solid blue line).

target vehicles. For safety reasons, the experiments are performed at low speeds using a virtual (or simulated) target vehicle.

8.6.1 Test scenario

The scene at the start of the experiment ($t = 0$ s) is shown in Figure 8.10a. The ego vehicle E is moving in the right lane at a speed of about 6 m/s with a reference speed of 8 m/s. The virtual target vehicle T_1 is at a relative distance of about 20 m behind E in the left lane and moving at a speed of about 14 m/s. The acceleration of T_1 is computed using the Intelligent Driver Model presented in Section 4.4. We assume that T_1 is operated by a *compliant* driver. That is, when the ego vehicle indicates its desire to change lanes to the left via the turn signal, T_1 responds and slows down to create a sufficient gap for E to change lanes. However, the environment model used by E to forecast the motion of T_1 is not aware of this, and must estimate the likelihood that it is compliant.

At time $t = 2.0$ s, the operator of E turns the left turn signal on to trigger a lane change command for the ego vehicle. The lane change is achieved by changing the lateral position reference η^{ref} to the center of the left lane. The goal for the ego vehicle is to change lanes if it is safe to do so, given the predictions of the target vehicle and the corresponding safety constraints.

8.6.2 Technical approach

Vehicle model: The kinematic bicycle model described in Section 3.3.1 is used for the control design. The position and orientation of the vehicle are expressed in the road-aligned frame. Hence, $x_t^v = [\xi_t, \eta_t, \phi_t, v_t]^T$. In this application, we focus only on the lateral control and assume the existence of a low-level controller to track a constant speed reference. This is equivalent to choosing $u_t^v = \delta_{f_t}$ and setting $a_t = 0$ in the bicycle model (3.24).

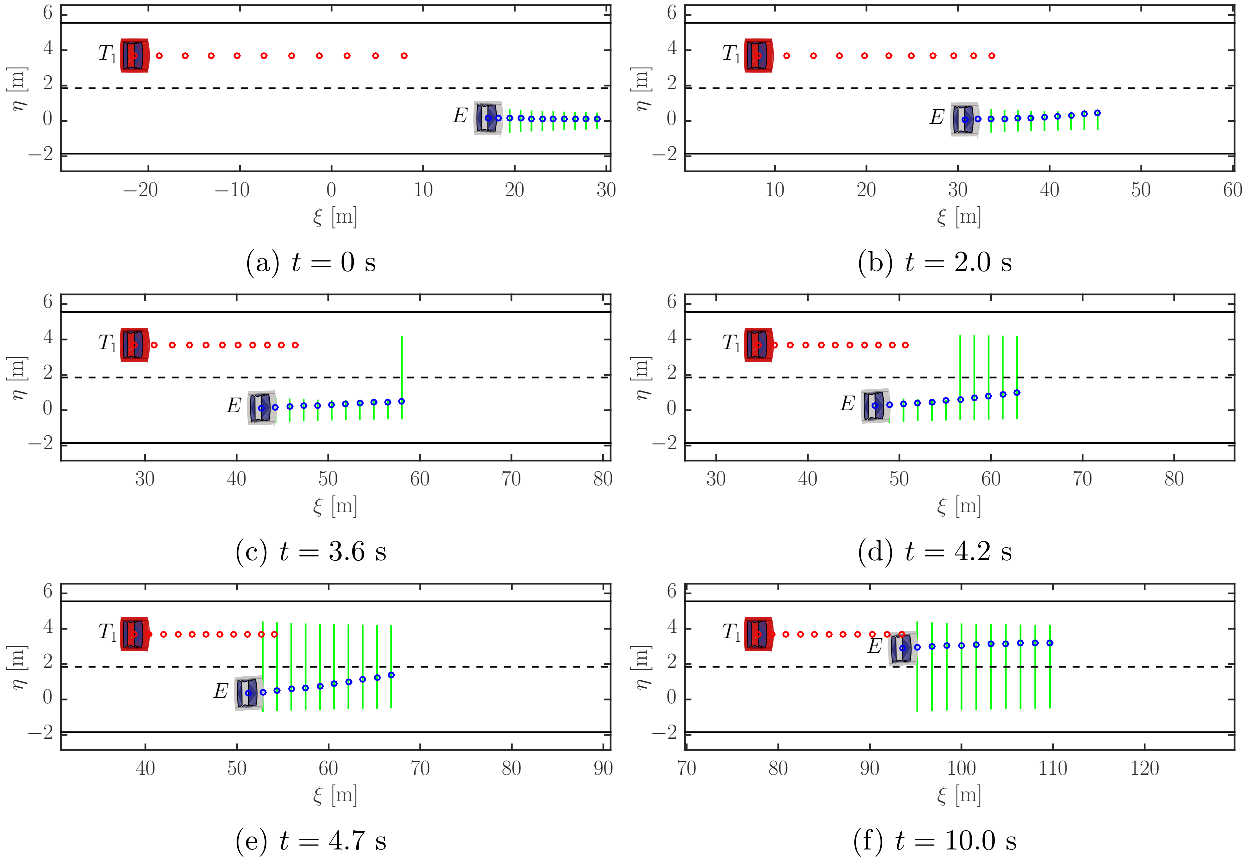
Reference generation: A simple reference generator which provides a constant lateral position set-point η^{ref} is used in this application. This constitutes the reference state sequence $x_{1:T|t}^{\text{ref}}$.

Environment model: The interaction-aware car-following model presented in Section 4.4 is used to predict the longitudinal motion of the target vehicle. Based on an estimate of the TV's acceleration, the model estimates the probability that the TV will slow down to make room for the ego vehicle to change lanes. We generate full horizon samples of the future trajectories of the TV using the method presented in Section 4.4. The reference speed of the TV in the free-flow mode is 20 m/s.

Controller: We use the hybrid SMPC approach proposed in Section 8.2. The collision avoidance constraints are formulated using the safety corridor approach of Section 4.5.1. At each time step k of the horizon, the sampled predictions from the environment model are aggregated into a single sample using Method 2 presented in Section 8.4.2. The analytical SMPC approach with full state feedback is used to propagate the variance of the state x_k^v over the horizon. The SMPC problem is solved with a constraint violation probability $\epsilon_k = 0.005$ in (8.8). The prediction horizon is $T = 10$ time steps with a discretization time of 0.2 s, resulting in a preview time of 2 s.

8.6.3 Results

Snapshots of the experiment are shown in Figure 8.10. The blue and red circles are the predicted positions of the ego and target vehicles, respectively, over the horizon. The vertical



(a) $t = 0$ s

(b) $t = 2.0$ s

(c) $t = 3.6$ s

(d) $t = 4.2$ s

(e) $t = 4.7$ s

(f) $t = 10.0$ s

Figure 8.10: Snapshots of experiment with the ego vehicle E in blue and the target vehicle T_1 in red. The blue and red circles are the predicted positions of the ego and target vehicles, respectively, over the horizon. The vertical green lines indicate the safe region for E computed by the safety corridor approach, given the sampled predictions of T_1,

green lines indicate the safe region for the center of gravity of E over the prediction horizon computed by the safety corridor approach, given the sampled predictions of T_1. At $t = 0$ s (Figure 8.10a), T_1 is relatively close to E and moving faster. Thus, the ego vehicle is restricted to move in its own lane, as seen by the vertical green lines lying completely within the right lane. When the operator of E indicates a lane change at $t = 2.0$ s (Figure 8.10b), the situation is similar. The ego vehicle plans a path to the left which is constrained to lie in it's own lane.

However, as T_1 is modeled as being compliant, it responds to the EV's intent to change lanes and starts slowing down. At $t = 3.6$ s (Figure 8.10c), we see that based on the predictions of T_1, the safe region for E extends into the left lane at the last time step of the prediction horizon. This still does not enable E to change lanes. A similar situation occurs at $t = 4.2$ s (Figure 8.10d) where the safe regions corresponding to the last 5 time steps of the prediction horizon extend into the left lane. While this does not allow E to immediately change lanes, the controller plans to initiate the lane change maneuver towards the end of the horizon. Finally, at $t = 4.7$ s (Figure 8.10e), the safety constraints allow the ego vehicle to change lanes. The lane change maneuver is completed around $t = 10.0$ s as seen in Figure 8.10f.

8.6.4 Effect of not accounting for interactions

The model used to forecast the acceleration input and the resulting longitudinal trajectory of the target vehicle T_1 estimates the probability that T_1 will yield to the ego vehicle E. When the confidence that T_1 is compliant is sufficiently high, the resulting safety constraints allow E to complete the lane change. We examine the effect of not accounting for this interaction between E and T_1 by repeating the above experiment with a modified environment model. Concretely, we assume that T_1 is always in a non-compliant mode where it does not respond to the EV's intention to change lanes. In this case, the environment model predicts the acceleration of T_1 based on its free-flow mode of operation, where T_1 is assumed to track the reference speed of 20 m/s.

Clearly, the above model is conservative as it assumes that T_1 will speed up to the reference speed starting from its current speed. However, in closed-loop, the speed of T_1 decreases to that of E. Figure 8.11 shows the scene at $t = 14.5$ during the experiment. Due to the conservative predictions, E is restricted from performing a lane change maneuver. This is seen by the safety region (depicted by the vertical green lines) lying completely within the right lane for the last 5 time steps of the horizon. On the other hand, in the previous experiment (Figure 8.10), E had completed the lane change by around $t = 10$ s. In summary, this motivates the need for considering interactions in urban driving scenarios in order to achieve realistic and non-conservative behavior by the autonomous vehicle.

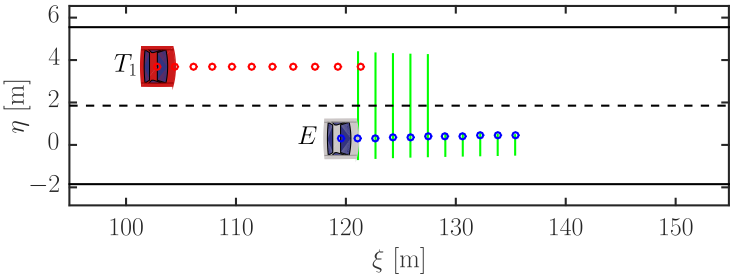


Figure 8.11: Snapshot of the experiment in which interaction between the target vehicle T_1 and the ego vehicle E is not considered. The notation is the same as that in Figure 8.10.

8.7 Conclusions

In this chapter, we presented approaches for the control design when a probabilistic view of the uncertainty in the vehicle and environment forecasts is taken. In general autonomous driving scenarios, such a view yields a less conservative control policy than a worst-case characterization of the uncertainty. Two approaches for stochastic model predictive control, analytical and sampling-based SMPC, are compared and a hybrid approach is proposed which combines the benefits of both methods from the standpoint of autonomous vehicle control. Simulation results show the ability of the controller to perform an autonomous overtaking maneuver. We also discuss the effect of tuning the bound on the constraint violation probability on the controller's performance and of using a sampled representation of environmental uncertainty. Experiments on our prototype vehicle show the real-time feasibility of the proposed approach.

Integration of previously presented control design methodologies

In the previous chapters, we presented control strategies for various applications including obstacle avoidance (Chapter 5), electronic stability control (Chapter 6) and autonomous cruise control (Chapter 7). We would like to conclude with a brief discussion of how these approaches can be unified with the stochastic MPC strategy presented in this chapter for application to general autonomous driving scenarios.

Iterative linearization approach

The iterative linearization method of Chapter 5 is a general approach for solving the nonlinear CFTOC problem and can be used with any vehicle model and safety constraint formulation. It is particularly suited for the analytical SMPC approach presented in Section 8.2 as the uncertainty propagation and constraint tightening steps are based on a linearization of the system dynamics and collision avoidance constraints, respectively. At each time step, a sequence of QPs can be solved, each refining the solution obtained at the previous iteration.

Safe set for stability control

While the lateral stability controller presented in Chapter 6 assumes the existence a driver in the loop, the analysis therein readily extends to the autonomous case. In fact, the analysis is simplified due to the absence of one of the primary sources of uncertainty, that is, the driver. The analysis in Section 6.4 is a systematic method of computing the set of robustly safe control inputs in order to prevent the vehicle from drifting out of control. This set can be augmented to the set $\mathcal{U}$ of feasible control inputs for the autonomous navigation applications in this chapter. As the robust invariant set computations are performed offline and the set to be used online is polyhedral, the increase in the computational complexity of the resulting online optimization problem is minimal.

Safe following distances

Finally, the worst-case analysis for the design of the autonomous cruise controller in Chapter 7 can also be applied to the applications involving lateral control in this chapter. It is important to note that the analysis based on the notion of the braking distance results in safety constraints which are a function not only of the relative distance between the preceding and ego vehicles but also their longitudinal speeds. This can be viewed as a systematic method of artificially increasing the dimensions of target vehicles (in the longitudinal direction) to account for safety. The consequence is that, in scenarios involving lane change and overtaking maneuvers, the ego vehicle would maintain a relative distance to surrounding vehicles based on the relative speed.

Appendix A

Experimental Vehicles

This appendix provides an overview of the hardware configurations and physical parameters of the experimental passenger vehicles used for the results in this dissertation.

A.1 Jaguar S-Type

The prototype Jaguar S-Type vehicle provided by the Ford Motor Company is equipped with an Active Front Steering (AFS) system and four wheel independent braking. The AFS allows us to command an *additive* steering angle to the vehicle which is augmented to the driver's steering angle. An Oxford Technical Solutions (OxTS) RT3002 sensing system is used to measure the position and orientation in the inertial frame, and the vehicle velocities in the body frame. The OxTS RT3002 system comprises of a differential Global Positioning System (GPS) and an Inertial Measurement Unit (IMU). A GPS base station is used to provide differential corrections to the RT3002 unit. The real-time computations are performed on a dSPACE DS1005 Autobox system which consists of a PowerPC 750GX processor running at 933 MHz. The aforementioned hardware components communicate via a CAN bus. Experiments with this vehicle were performed at the Smithers Winter Test Center in Raco, Michigan, in collaboration with the Ford Motor Company. The physical parameters and actuator limits are listed in Table A.1. The maximum value of the slip ratio σ is 0 as the engine throttle cannot be commanded in the test vehicle. This vehicle is used for the experiments performed in Chapters 5 and 6.

A.2 Hyundai Grandeur

The prototype Hyundai Grandeur vehicle provided by the Hyundai Motor Company is equipped with the sensors and actuators essential for fully autonomous driving. A production grade camera and radar provide estimates of the relative positions and velocities of vehicles in front of the ego vehicle. In addition, the camera gives an estimate of the lane geometry in terms of a 3$^{\text{rd}}$-order polynomial fitted to the lane boundaries in the body fixed

Table A.1: Parameters of the prototype Jaguar S-Type vehicle

Parameter	**Value**	**Parameter**	**Value**
m	2050 kg	I_z	3344 kg-m^2
l_f	1.43 m	l_r	1.47 m
$\delta_{f_{\min}}$	-10 deg	$\delta_{f_{\max}}$	10 deg
$\dot{\delta}_{f_{\min}}$	-60 deg/s	$\dot{\delta}_{f_{\max}}$	60 deg/s
$\sigma_{\min}$	-0.5	$\sigma_{\max}$	0

Table A.2: Parameters of the prototype Hyundai Grandeur vehicle

Parameter	**Value**	**Parameter**	**Value**
m	1830 kg	I_z	3477 kg-m^2
l_f	1.15 m	l_r	1.69 m
$\delta_{f_{\min}}$	-30 deg	$\delta_{f_{\max}}$	30 deg
$\dot{\delta}_{f_{\min}}$	-10 deg/s	$\dot{\delta}_{f_{\max}}$	10 deg/s
$a_{x_{\min}}$	-4 m/s^2	$a_{x_{\max}}$	2 m/s^2
$\tau_{s_{\min}}$	-7.5 N-m	$\tau_{s_{\max}}$	7.5 N-m

coordinate frame. The vehicle is retrofitted with six IBEO LUX laser-scanners, each with a 110 degree horizontal field of view and 4 vertical layers with a 3.2 degree vertical field of view. The point clouds from the sensors are fused by the IBEO Fusion ECU to yield a 360 degree image of the environment. In addition, the ECU performs clustering and segmentation of the point cloud to provide the relative positions, velocities and categories (e.g. car, bike, truck, pedestrian) of surrounding objects.

An Oxford Technical Solutions (OxTS) RT 2002 sensing system, comprising of a global positioning system (GPS) and an inertial measurement unit (IMU), combined with a GPS base station are used for localizing the vehicle. Two embedded computing platforms are available: (*i*) a dSPACE MicroAutoBox II (900 MHz IBM PowerPC processor), and (*ii*) a Speedgoat real-time mobile target machine (2.16 GHz Intel Core 2 Duo processor) based on Simulink Real-Time. The sensors, actuators and computing platform on the vehicle communicate via a CAN bus.

The vehicle steering is controlled by means of the overlay torque function of the built-in Electronic Power Steering (EPS) or Motor Driven Power Steering (MDPS) system. The torque is *additive* to that applied by the driver to the steering column. Longitudinal speed control is achieved by means of a desired acceleration command to the built-in ACC system on the vehicle. The physical parameters and actuator limits are listed in Table A.2. This vehicle is used for the experiments performed in Chapters 7 and 8.

Appendix B

Measurement Uncertainty in Stochastic Model Predictive Control

The analytical SMPC formulation presented in Section 8.2 assumes full state feedback as seen by the control policy parameterization in (8.9). In general, the state x_t^v cannot be measured directly. Instead, at each time step t, a measurement y_t^v is assumed to be available which is related to the state as:

$$y_t^v = h^v(x_t^v, w_t^v), \tag{B.1}$$

where $w_t^v \sim \mathcal{N}(\hat{w}_t^v, \Sigma_t^w)$. A state observer provides an estimate $\tilde{x}_t^v$ of the true state x_t^v at time t with an associated variance Σ_t^v conditioned on the history of measurements $y_{1:t}^v$.

Remark B.1 (Notation)**.** *The symbol $\hat{x}_t^v$ (or $\hat{x}_{0:T|t}^v$) refers to the mean of the true state x_t^v (or $x_{0:T|t}^v$). The symbol $\tilde{x}_t^v$ (or $\tilde{x}_{0:T|t}^v$) denotes the estimate of the true state x_t^v (or $x_{0:T|t}^v$) as computed by the state observer. In closed-loop, $\tilde{x}_t^v \equiv \hat{x}_t^v$. However, as will be shown in Section B.2, the predicted state estimate $\tilde{x}_{k|t}^v$ is itself an uncertain variable (with an associated mean and variance) due to the unknown future measurement $y_{k|t}^v$ at time t. On the other hand, the mean $\hat{x}_{k|t}^v$ is a point estimate and hence, deterministic. By definition, $\tilde{x}_t^v \sim \mathcal{N}(\hat{x}_t^v, 0)$.*

The consideration of measurement uncertainty is particularly useful in the case where the physical vehicle parameters such as tire stiffness coefficients are treated as uncertain variables. As discussed in Section 3.6.3, the physical parameters to be estimated online can be augmented to the state vector x_t^v. As these cannot be measured directly by any sensor, a state observer must be used to obtain an estimate of the mean and variance of the parameters. In this case, the measurements y_t^v correspond to the quantities such as the position, orientation, yaw rate and velocity. In the absence of parametric uncertainty, the assumption of full state feedback (Sections 8.2.2–8.2.3) is reasonable in the context of autonomous driving. The following sections discuss the characterization of the uncertainty over the horizon based on the approaches presented in [120, 124].

B.1 State Estimate Dynamics

In the presence of partial state feedback or measurement noise, the feedback policy (8.9) is modified as:

$$u_k^{\delta,v} = K_k \tilde{x}_k^{\delta,v} + c_k^v, \tag{B.2}$$

where:

$$\tilde{x}_k^{\delta,v} = \tilde{x}_k^v - \bar{x}_k^v. \tag{B.3}$$

This approach is motivated by the *certainty equivalence and separation principles* for Linear Quadratic Gaussian (LQG) systems, that is, systems with linear dynamics and additive Gaussian disturbances where the goal is to minimize a cost function quadratic in the states and control inputs [15]. However, no guarantee of optimality can be made due to the presence of constraints in the SMPC problem.

The formulation in (B.2) necessitates the consideration of the evolution of the state estimate $\tilde{x}_k^{\delta,v}$. In order to efficiently propagate the state estimates over the prediction horizon, a Kalman filter is used along with the LTV dynamics (8.5) and the following linearization of the measurement model (B.1) around the nominal trajectory:

$$y_k^{\delta,v} = H_k x_k^{\delta,v} + W_k w_k^{\delta,v}, \tag{B.4}$$

where:

$$\begin{aligned} y_k^{\delta,v} &= y_k^v - h^v(\bar{x}_k^v, \bar{w}_k^v), & \text{(B.5a)}\\ w_k^{\delta,v} &= w_k^v - \bar{w}_k^v, & \text{(B.5b)}\\ H_k &= \nabla_{x_k^v} h^v(x_k^v, w_k^v)|_{(\bar{x}_k^v, \bar{w}_k^v)}, & \text{(B.5c)}\\ W_k &= \nabla_{w_k^v} h^v(x_k^v, w_k^v)|_{(\bar{x}_k^v, \bar{w}_k^v)}. & \text{(B.5d)} \end{aligned}$$

The nominal disturbance $\bar{w}_k^v$ is identically equal to its mean $\hat{w}_k^v$. Thus, $w_k^{\delta,v} \sim \mathcal{N}(0, \Sigma_k^w)$. The state estimate $\tilde{x}_k^{\delta,v}$ evolves as:

$$\tilde{x}_{k+1}^{\delta,v} = \tilde{x}_{k+1|k}^{\delta,v} + L_{k+1}(y_{k+1}^{\delta,v} - \hat{y}_{k+1}^{\delta,v}), \tag{B.6}$$

where:

$$\begin{aligned} \tilde{x}_{k+1|k}^{\delta,v} &= A_k \tilde{x}_k^{\delta,v} + B_k u_k^{\delta,v}, & \text{(B.7a)}\\ \hat{y}_{k+1}^{\delta,v} &= H_{k+1} \tilde{x}_{k+1|k}^{\delta,v}. & \text{(B.7b)} \end{aligned}$$

The filter gains $\{L_k\}_{k=1}^T$ in (B.6) are computed by the forward recursion in Algorithm B.1.

Remark B.2. *The future measurements $y_{1:T}^{\delta,v}$ are not known a-priori. This makes the dynamics* (B.6) *of the predicted state estimate $\tilde{x}_k^{\delta,v}$ stochastic. Moreover, due to the feedback policy in* (B.2) *and the stochasticity of the state estimate $\tilde{x}_k^{\delta,v}$, the predicted inputs $u_k^{\delta,v}$ are also stochastic.*

Algorithm B.1 Kalman Filter recursion

```
1: Initialize:
   $P_0 = \Sigma_0^v$
2: for k = 0 to $T-1$:
3:     $P_{k+1|k} = A_k P_k A_k^T + D_k \Sigma_k^d D_k^T$
4:     $L_{k+1} = P_{k+1|k} H_{k+1}^T (H_{k+1} P_{k+1|k} H_{k+1}^T + W_{k+1} \Sigma_{k+1}^w W_{k+1}^T)^{-1}$
5:     $P_{k+1} = (I - L_{k+1} H_{k+1}) P_{k+1|k}$
Output: $\{L_{k+1}\}_{k=0}^{T-1}$
```

B.2 Uncertainty Propagation

As in Section 8.2.3, we characterize the uncertainty in the predicted states x_k^v and inputs u_k^v. Combining the LTV dynamics (8.5) with the control policy (B.2), we get:

$$x_{k+1}^{\delta,v} = A_k x_k^{\delta,v} + B_k K_k \tilde{x}_k^{\delta,v} + B_k c_k^v + D_k d_k^{\delta,v}. \tag{B.8}$$

The state estimate update (B.6) can be rewritten as follows (by substituting for $u_k^{\delta,v}$ and $y_{k+1}^{\delta,v}$ from (8.9) and (B.4), respectively):

$$\begin{aligned}\tilde{x}_{k+1}^{\delta,v} = (A_k + B_k K_k)\tilde{x}_k^{\delta,v} + B_k c_k^v + \\ L_{k+1}\left(H_{k+1}(A_k x_k^{\delta,v} + D_k d_k^{\delta,v}) + W_{k+1} w_k^{\delta,v} - H_{k+1} A_k \tilde{x}_k^{\delta,v}\right).\end{aligned} \tag{B.9}$$

Concisely, the joint dynamics of the true state $x_k^{\delta,v}$ and its estimate $\tilde{x}_k^{\delta,v}$ are written as:

$$z_{k+1}^v = F_k z_k^v + G_k c_k^v + Q_k q_k^v, \tag{B.10}$$

where:

$$z_k^v = \begin{bmatrix} x_k^{\delta,v} \\ \tilde{x}_k^{\delta,v} \end{bmatrix}, \qquad q_k^v = \begin{bmatrix} d_k^{\delta,v} \\ w_{k+1}^{\delta,v} \end{bmatrix} \sim \mathcal{N}\left(0, \Sigma_k^q = \begin{bmatrix} \Sigma_k^d & 0 \\ 0 & \Sigma_{k+1}^w \end{bmatrix}\right), \tag{B.11a}$$

$$F_k = \begin{bmatrix} A_k & B_k K_k \\ L_{k+1} H_{k+1} A_k & A_k + B_k K_k - L_{k+1} H_{k+1} A_k \end{bmatrix}, \tag{B.11b}$$

$$G_k = \begin{bmatrix} B_k \\ B_k \end{bmatrix}, \qquad Q_k = \begin{bmatrix} D_k & 0 \\ L_{k+1} H_{k+1} D_k & L_{k+1} W_{k+1} \end{bmatrix}. \tag{B.11c}$$

The mean $\hat{z}_k^v$ and variance Σ_k^z can now be computed as:

$$\hat{z}_{k+1}^v = F_k \hat{z}_k^v + G_k c_k^v, \tag{B.12a}$$

$$\Sigma_{k+1}^z = F_k \Sigma_k^z F_k^T + Q_k \Sigma_k^q Q_k^T, \tag{B.12b}$$

with initial conditions:

$$\hat{z}_0^v = 0, \qquad \Sigma_0^z = \begin{bmatrix} \Sigma_{0|t}^v & 0 \\ 0 & 0 \end{bmatrix}. \tag{B.13}$$

The states $x_k^{\delta,v}$ and inputs $u_k^{\delta,v}$ are obtained through the following affine transformation:

$$\begin{bmatrix} x_k^{\delta,v} \\ u_k^{\delta,v} \end{bmatrix} = S_k z_k^v + T_k, \tag{B.14}$$

with:

$$S_k = \begin{bmatrix} I & 0 \\ 0 & K_k \end{bmatrix}, \qquad T_k = \begin{bmatrix} 0 \\ c_k^v \end{bmatrix}. \tag{B.15}$$

Therefore, the joint distribution of $x_k^{\delta,v}$ and $u_k^{\delta,v}$ is given by:

$$\begin{bmatrix} x_k^{\delta,v} \\ u_k^{\delta,v} \end{bmatrix} \sim \mathcal{N}\left(S_k \hat{z}_k^v + T_k, S_k \Sigma_k^z S_k^T\right). \tag{B.16}$$

The distributions $x_k^{\delta,v} \sim \mathcal{N}(\hat{x}_k^{\delta,v}, \Sigma_k^v)$ and $u_k^{\delta,v} \sim \mathcal{N}(\hat{u}_k^{\delta,v}, \Sigma_k^u)$ can be easily obtained from (B.16) by marginalizing.

With the above characterization of the uncertainty in the state, the control synthesis proceeds in the same manner as in the full state feedback case. In particular, the constraint tightening procedure of Section 8.2.4 can be used to formulate a deterministic CFTOC problem similar to that in (8.20).

Bibliography

[1] P. Abbeel and A. Y. Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the 21st International Conference on Machine Learning*. 2004.

[2] P. Abbeel et al. "Apprenticeship learning for motion planning with application to parking lot navigation". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2008, pp. 1083–1090.

[3] J. Ackermann, D. Odenthal, and T. Bünte. "Advantages of active steering for vehicle dynamics control". In: *Proceedings of 32nd ISATA, Automotive Mechatronics Design and Engineering*. 1999, pp. 263–270.

[4] Adrian Broadhurst, Simon Baker, and Takeo Kanade. "Monte Carlo road safety reasoning". In: *2005 IEEE Intelligent Vehicles Symposium*. 2005, pp. 319–324.

[5] M. Althoff and J. M. Dolan. "Online verification of automated road vehicles using reachability analysis". In: *IEEE Transactions on Robotics* 30.4 (Aug. 2014), pp. 903–918.

[6] M. Althoff, D. Heß, and F. Gambert. "Road occupancy prediction of traffic participants". In: *16th International IEEE Conference on Intelligent Transportation Systems*. Oct. 2013, pp. 99–105.

[7] L. Alvarez and R. Horowitz. "Safe platooning in automated highway systems, Part I: Safety regions design". In: *Vehicle System Dynamics* 32.1 (1999), pp. 23–55.

[8] S. Anderson et al. "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios". In: *International Journal of Vehicle Autonomous Systems* 8.2 (2010), pp. 190–216.

[9] Andreas Eidehall and Lars Petersson. "Statistical threat assessment for general road scenes using Monte Carlo sampling". In: *IEEE Transactions on Intelligent Transportation Systems* 9(1) (2008), pp. 137–147.

[10] G. Aoude et al. "Mobile agent trajectory prediction using Bayesian nonparametric reachability trees". In: *AIAA Infotech@Aerospace 2011 Conference*. 2011, pp. 1587–1593.

[11] V. L. Bageshwar, W. L. Garrard, and R. Rajamani. "Model predictive control of transitional maneuvers for adaptive cruise control vehicles". In: *IEEE Transactions on Vehicular Technology* 53.5 (2004), pp. 1573–1585.

[12] M. Barić. "Constrained control: Computations, performance and robustness". PhD thesis. ETH, 2008.

[13] M. Behrisch et al. "SUMO–Simulation of urban mobility: An overview". In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. 2011.

[14] A. Bemporad, F. Borrelli, and M. Morari. "Min-max control of constrained uncertain discrete-time linear systems". In: *IEEE Transactions on Automatic Control* 48.9 (2003), pp. 1600–1606.

[15] D. P. Bertsekas. *Dynamic programming and optimal control*. 2nd. Athena Scientific, 2000.

[16] F. Blanchini and S. Miani. *Set-theoretic methods in control*. Birkhäuser Boston, 2007.

[17] F. Blanchini. "Survey paper: Set invariance in control". In: *Automatica* 35.11 (1999), pp. 1747–1767.

[18] P. T. Boggs and J. W. Tolle. "Sequential quadratic programming". In: *Acta Numerica* 4.1 (1995), pp. 1–51.

[19] G. Burgio and P. Zegelaar. "Integrated vehicle control using steering and brakes". In: *International Journal of Control* 79.05 (2006), pp. 534–541.

[20] G. Calafiore. "Random convex programs". In: *SIAM Journal on Optimization* 20.6 (2010), pp. 3427–3464.

[21] S. Calinon. *Robot programming by demonstration: A probabilistic approach*. EPFL/CRC Press, 2009.

[22] S. Calinon, F. Guenter, and A. Billard. "On learning, representing, and generalizing a task in a humanoid robot". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.2 (2007), pp. 286–298.

[23] S. Calinon et al. "Learning and reproduction of gestures by imitation". In: *IEEE Robotics & Automation Magazine* 17.2 (2010), pp. 44–54.

[24] M. Campi and S. Garatti. "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality". In: *Journal of Optimization Theory and Applications* 148.2 (2011), pp. 257–280.

[25] M. Cannon, P. Couchman, and B. Kouvaritakis. "MPC for stochastic systems". In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Vol. 358. Lecture Notes in Control and Information Sciences. Springer, 2007, pp. 255–268.

[26] M. Cannon, B. Kouvaritakis, and D. Ng. "Probabilistic tubes in linear stochastic model predictive control". In: *Systems & Control Letters* 58.10-11 (2009), pp. 747–753.

[27] M. Cannon, D. Ng, and B. Kouvaritakis. “Successive linearization NMPC for a class of stochastic nonlinear systems”. In: *Nonlinear Model Predictive Control*. Vol. 384. Lecture Notes in Control and Information Sciences. Springer, 2009, pp. 249–262.

[28] A. Carvalho et al. “Automated driving: The role of forecasts and uncertainty - A control perspective”. In: *European Journal of Control* 24 (2015). SI: 2015 European Control Conference, pp. 14–32.

[29] A. Carvalho et al. “Predictive control of an autonomous ground vehicle using an iterative linearization approach”. In: *16th IEEE Conference on Intelligent Transportation Systems*. 2013, pp. 2335–2340.

[30] A. Carvalho et al. “Robust vehicle stability control with an uncertain driver model”. In: *2013 European Control Conference*. 2013, pp. 440–445.

[31] A. Carvalho et al. “Stochastic predictive control of autonomous vehicles in uncertain environments”. In: *12th International Symposium on Advanced Vehicle Control*. 2014.

[32] A. Carvalho et al. “Autonomous cruise control with cut-in target vehicle detection”. In: *13th International Symposium on Advanced Vehicle Control*. 2016.

[33] G. Cesari. “Scenario model predictive control for autonomous driving on highway”. MA thesis. ETH Zurich, 2016.

[34] L. Chisci, J. Rossiter, and G. Zappa. “Systems with persistent disturbances: Predictive control with restricted constraints”. In: *Automatica* 37.7 (2001), pp. 1019–1028.

[35] Christian Laugier et al. “Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety”. In: *IEEE Intelligent Transportation Systems Magazine* 2(4) (2011), pp. 4–19.

[36] I. Dagli et al. “Cutting-in vehicle recognition for ACC systems - towards feasible situation analysis methodologies”. In: *Intelligent Vehicles Symposium, 2004 IEEE*. June 2004, pp. 925–930.

[37] S. Deb et al. “A generalized S-D assignment algorithm for multisensor-multitarget state estimation”. In: *IEEE Transactions on Aerospace and Electronic Systems* 33.2 (1997), pp. 523–538.

[38] S. Di Cairano and H. Tseng. “Driver-assist steering by active front steering and differential braking: design, implementation and experimental evaluation of a switched model predictive control approach”. In: *49th IEEE Conference on Decision and Control*. IEEE. 2010, pp. 2886–2891.

[39] M. Diehl et al. “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations”. In: *Journal of Process Control* 12.4 (2002), pp. 577–585.

[40] A. Domahidi, E. Chu, and S. Boyd. “ECOS: An SOCP solver for embedded systems”. In: *2013 European Control Conference*. 2013, pp. 3071–3076.

[41] A. Domahidi and J. Jerez. *FORCES Professional*. embotech GmbH (`http://embotech.com/FORCES-Pro`). July 2014.

[42] Z. Ercan et al. "Modeling, identification and predictive control of torque-based driver steering assistance system". submitted. 2016.

[43] P. Falcone et al. "Linear time varying model predictive control and its application to active steering systems: Stability analysis and experimental validation". In: *International Journal of Robust and Nonlinear Control* 18.8 (2008), pp. 862–875.

[44] P. Falcone et al. "Low complexity MPC schemes for integrated vehicle dynamics control problems". In: *9th International Symposium on Advanced Vehicle Control*. 2008.

[45] P. Falcone et al. "Predictive active steering control for autonomous vehicle systems". In: *Control Systems Technology, IEEE Transactions on* 15.3 (2007), pp. 566–580.

[46] P. Falcone et al. "MPC-based yaw and lateral stabilisation via active front steering and braking". In: *Vehicle System Dynamics* 46.S1 (2008), pp. 611–628.

[47] Y. Gao et al. "A tube-based robust nonlinear predictive control approach to semi-autonomous ground vehicles". In: *Vehicle System Dynamics* 52.6 (2014), pp. 802–823.

[48] Y. Gao et al. "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads". In: *3rd Annual ASME Conference on Dynamic Systems and Controls*. 2010, pp. 265–272.

[49] Y. Gao et al. "Spatial predictive control for agile semi-autonomous ground vehicles". In: *11th International Symposium on Advanced Vehicle Control*. 2012.

[50] Georg Schildbach et al. "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations". In: *Automatica* 50(12) (2014), pp. 3009–3018.

[51] P. E. Gill et al. *User's guide for NPSOL: Version 4.0*. 1986.

[52] T. Gindele, S. Brechtel, and R. Dillmann. "Learning driver behavior models from traffic observations for decision making and planning". In: *IEEE Intelligent Transportation Systems Magazine* 7.1 (Spring 2015), pp. 69–79.

[53] Government of South Australia. *The driver's handbook: Care, courtesy, common sense*. 2015.

[54] A. Gray et al. "Integrated threat assessment and control design for roadway departure avoidance". In: *15th IEEE Conference on Intelligent Transportation Systems*. 2012, pp. 1714–1719.

[55] A. Gray et al. "Semi-autonomous vehicle control for road departure and obstacle avoidance". In: *13th IFAC Symposium on Control in Transportation Systems*. 2012.

[56] A. Gray et al. “Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model”. In: *16th IEEE Conference on Intelligent Transportation Systems*. 2013.

[57] *Gurobi optimizer reference manual*. Gurobi Optimization, Inc. 2014.

[58] S. Han. “A globally convergent method for nonlinear programming”. In: *Journal of Optimization Theory and Applications* 22.3 (1977), pp. 297–309.

[59] M. Herceg et al. “Multi-Parametric Toolbox 3.0”. In: *2013 European Control Conference*. 2013, pp. 502–510.

[60] D. van Hessem and O. Bosgra. “A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints”. In: *41st IEEE Conference on Decision and Control*. Vol. 4. 2002, pp. 4643–4648.

[61] R. Hindiyeh. “Dynamics and control of drifting in automobiles”. PhD thesis. Stanford University, 2013.

[62] R. Hindiyeh and J. Gerdes. “Equilibrium analysis of drifting vehicles for control design”. In: *2nd Annual ASME Conference on Dynamic Systems and Controls*. 2009, pp. 181–188.

[63] P. Hokayem et al. “Stochastic receding horizon control with output feedback and bounded control Inputs”. In: *49th IEEE Conference on Decision and Control*. 2010, pp. 6095–6100.

[64] S. Hong et al. “A novel approach for vehicle inertial parameter identification using a dual Kalman filter”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.1 (2015), pp. 151–161.

[65] P. A. Ioannou and C. C. Chien. “Autonomous intelligent cruise control”. In: *IEEE Transactions on Vehicular Technology* 42.4 (Nov. 1993), pp. 657–672.

[66] Jürgen Wiest et al. “Probabilistic trajectory prediction using Gaussian mixture models”. In: *2012 IEEE Intelligent Vehicles Symposium*. 2012, pp. 141–146.

[67] U. Kiencke and L. Nielsen. *Automotive control systems: For engine, driveline and vehicle*. 1st. Springer-Verlag New York, Inc., 2000.

[68] J. Kong. “Model predictive control of an autonomous vehicle with a kinematic bicycle model”. MA thesis. University of California, Berkeley, 2015.

[69] J. Kong et al. “Kinematic and dynamic vehicle models for autonomous driving control design”. In: *2015 IEEE Intelligent Vehicles Symposium*. 2015, pp. 1094–1099.

[70] B. Kouvaritakis, M. Cannon, and R. Yadlin. “On the centres and scalings of robust and stochastic tubes”. In: *UKACC International Conference on Control 2010*. Sept. 2010, pp. 1–6.

[71] B. Kouvaritakis et al. “Explicit use of probabilistic distributions in linear predictive control”. In: *Automatica* 46.10 (2010), pp. 1719–1724.

[72] M. Kuderer, S. Gulati, and W. Burgard. “Learning driving styles for autonomous vehicles from demonstration”. In: *2015 IEEE International Conference on Robotics and Automation*. May 2015, pp. 2641–2646.

[73] M. Kvasnica et al. “Multi-parametric toolbox (MPT)”. In: *Hybrid Systems: Computation and Control* (2004), pp. 121–124.

[74] S. Lefèvre, A. Carvalho, and F. Borrelli. “A learning-based framework for velocity control in autonomous driving”. In: *IEEE Transactions on Automation Science and Engineering* 13.1 (Jan. 2016), pp. 32–42.

[75] S. Lefèvre, A. Carvalho, and F. Borrelli. “Autonomous car following: A learning-based approach”. In: *2015 IEEE Intelligent Vehicles Symposium*. 2015.

[76] S. Lefèvre, D. Vasquez, and C. Laugier. “A survey on motion prediction and risk assessment for intelligent vehicles”. In: *Robomech Journal* 1.1 (2014), pp. 1–14.

[77] S. Lefèvre et al. “Comparison of parametric and non-parametric approaches for vehicle speed prediction”. In: *2014 American Control Conference*. IEEE. 2014, pp. 3494–3499.

[78] S. Lefèvre et al. “Lane keeping assistance with learning-based driver model and model predictive control”. In: *12th International Symposium on Advanced Vehicle Control*. 2014.

[79] D. Lenz, T. Kessler, and A. Knoll. “Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles”. In: *2015 IEEE Intelligent Vehicles Symposium*. June 2015, pp. 292–297.

[80] S. Levine and V. Koltun. “Continuous inverse optimal control with locally optimal examples”. In: *29th International Conference on Machine Learning*. 2012.

[81] S. Li et al. “Model predictive multi-objective vehicular adaptive cruise control”. In: *IEEE Transactions on Control Systems Technology* 19.3 (2011), pp. 556–566.

[82] A. Liniger, A. Domahidi, and M. Morari. “Optimization-based autonomous racing of 1:43 scale RC cars”. In: *Optimal Control Applications and Methods* 36.5 (2015), pp. 628–647.

[83] Y. Ma, S. Vichik, and F. Borrelli. “Fast stochastic MPC with optimal risk allocation applied to building control systems”. In: *2012 IEEE 51st IEEE Conference on Decision and Control*. IEEE. 2012, pp. 7559–7564.

[84] S. Mammar and D. Koenig. “Vehicle handling improvement by active steering”. In: *Vehicle System Dynamics* 38.3 (2002), pp. 211–242.

[85] D. Q. Mayne et al. “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36.6 (2000), pp. 789–814.

[86] D. Mayne, M. Seron, and S. Raković. “Robust model predictive control of constrained linear systems with bounded disturbances”. In: *Automatica* 41.2 (2005), pp. 219–224.

[87] E. Mazor et al. "Interacting multiple model methods in target tracking: A survey". In: *IEEE Transactions on Aerospace and Electronic Systems* 34.1 (1998), pp. 103–123.

[88] A. Mesbah et al. "Stochastic nonlinear model predictive control with probabilistic constraints". In: *2014 American Control Conference*. 2014, pp. 2413–2419.

[89] W. Milliken, D. Milliken, and L. Metz. *Race car vehicle dynamics*. Sae International Warrendale, PA, 1995.

[90] S. Moon, H.-J. Kang, and K. Yi. "Multi-vehicle target selection for adaptive cruise control". In: *Vehicle System Dynamics* 48.11 (2010), pp. 1325–1343.

[91] D. Müller et al. "A generic video and radar data fusion system for improved target selection". In: *2011 IEEE Intelligent Vehicles Symposium*. 2011, pp. 679–684.

[92] M. Nagai, M. Shino, and F. Gao. "Study on integrated control of active front steer angle and direct yaw moment". In: *JSAE review* 23.3 (2002), pp. 309–315.

[93] National Center for Statistics and Analysis. *2015 motor vehicle crashes: Overview (Traffic Safety Facts Research Note. Report No. DOT HS 812 318)*. National Highway Traffic Safety Administration. Aug. 2016.

[94] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Verlag, 1999.

[95] H. P. Blom. "An efficient filter for abruptly changing systems". In: *23rd IEEE Conference on Decision and Control*. Vol. 23. 1984, pp. 656–658.

[96] H. Pacejka. *Tire and vehicle dynamics*. Elsevier, 2005.

[97] G. Palmieri et al. "Robust vehicle lateral stabilization via set-based methods for uncertain piecewise affine systems: Experimental results". In: *50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 3252–3257.

[98] S. Patil et al. "Scaling up Gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation". In: *Algorithmic Foundations of Robotics XI: Selected Contributions of the 11th International Workshop on the Algorithmic Foundations of Robotics*. Springer International Publishing, 2015, pp. 515–533.

[99] R. R. Pitre, V. P. Jilkov, and X. R. Li. "A comparative study of multiple-model algorithms for maneuvering target tracking". In: *Defense and Security*. International Society for Optics and Photonics. 2005, pp. 549–560.

[100] D. A. Pomerleau. "ALVINN: An autonomous land vehicle in a neural network". In: *Advances in Neural Information Processing Systems 1*. 1989.

[101] R. Rajamani. *Vehicle dynamics and control*. Springer, 2006.

[102] S. Rakovic, M. Baric, and M. Morari. "Max-min control problems for constrained discrete time systems". In: *47th IEEE Conference on Decision and Control*. IEEE. 2008, pp. 333–338.

[103] S. Rakovic et al. “Reachability analysis of discrete-time systems with disturbances”. In: *IEEE Transactions on Automatic Control* 51.4 (2006), pp. 546–561.

[104] S. Ross, G. J. Gordon, and J. A. Bagnell. “A reduction of imitation learning and structured prediction to no-regret online learning”. In: *Journal of Machine Learning Research* 15.AISTATS 2011 (June 2011), pp. 627–635.

[105] G. Schildbach and F. Borrelli. “Scenario model predictive control for lane change assistance on highways”. In: *2015 IEEE Intelligent Vehicles Symposium*. 2015, pp. 611–616.

[106] J. Schlechtriemen et al. “A lane change detection approach using feature ranking with maximized predictive power”. In: *2014 IEEE Intelligent Vehicles Symposium*. June 2014, pp. 108–114.

[107] J. Schlechtriemen et al. “When will it change the lane? A probabilistic regression approach for rarely occurring events”. In: *2015 IEEE Intelligent Vehicles Symposium*. June 2015, pp. 1373–1379.

[108] R. Schmied et al. “Scenario model predictive control for robust adaptive cruise control in multi-vehicle traffic situations”. In: *2016 IEEE Intelligent Vehicles Symposium*. June 2016, pp. 802–807.

[109] J. Schulman et al. “Motion planning with sequential convex optimization and convex collision checking”. In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1251–1270.

[110] S. Singh. *Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey (Traffic Safety Facts Crash Stats. Report No. DOT HS 812 115)*. National Highway Traffic Safety Administration. Feb. 2015.

[111] J. Skaf and S. Boyd. “Design of affine controllers via convex optimization”. In: *IEEE Transactions on Automatic Control* 55.11 (2010), pp. 2476–2487.

[112] State of New Jersey Motor Vehicle Comission. *The New Jersey driver manual*. 2015.

[113] S. Streif, M. Karl, and A. Mesbah. “Stochastic nonlinear model predictive control with efficient sample approximation of chance constraints”. In: *arXiv preprint arXiv:1410.4535* (2014).

[114] C. Tay. “Analysis of dynamic scenes: Application to driving assistance”. PhD thesis. Institut National Polytechnique de Grenoble, 2009.

[115] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. The MIT Press, 2005.

[116] N. E. D. Toit and J. W. Burdick. “Robotic motion planning in dynamic, cluttered, uncertain environments”. In: *2010 IEEE International Conference on Robotics and Automation*. May 2010, pp. 966–973.

[117] M. Treiber, A. Hennecke, and D. Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Phys. Rev. E* 62 (2 Aug. 2000), pp. 1805–1824.

[118] U.S. Department of Transportation. *Federal automated vehicles policy*. Sept. 2016.

[119] A. Vahidi and A. Eskandarian. "Research advances in intelligent collision avoidance and adaptive cruise control". In: *IEEE Transactions on Intelligent Transportation Systems* 4.3 (Sept. 2003), pp. 143–153.

[120] J. Van Den Berg, P. Abbeel, and K. Goldberg. "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information". In: *International Journal of Robotics Research* 30.7 (June 2011), pp. 895–913.

[121] J. Van Den Berg, S. Patil, and R. Alterovitz. "Motion planning under uncertainty using iterative local optimization in belief space". In: *International Journal of Robotics Research* 31.11 (2012), pp. 1263–1278.

[122] Vehicle Dynamics Standards Committee. "Automotive stability enhancement systems". In: *SAE J1939* (2004).

[123] M. P. Vitus and C. J. Tomlin. "A hybrid method for chance constrained control in uncertain environments". In: *51st IEEE Conference on Decision and Control*. Dec. 2012, pp. 2177–2182.

[124] M. P. Vitus and C. J. Tomlin. "Closed-loop belief space planning for linear, Gaussian systems". In: *2011 IEEE International Conference on Robotics and Automation*. May 2011, pp. 2152–2159.

[125] M. P. Vitus and C. J. Tomlin. "Belief space planning for linear, Gaussian systems In uncertain environments". In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 5902–5907.

[126] A. Williams. "A sensor fusion system for automotive environment perception". MA thesis. University of California, Berkeley, 2016.

[127] B. H. Wilson. "How soon to brake and how hard to brake: Unified analysis of the envelope of opportunity for rear-end collision warnings". In: *Proceedings of 17th International Technical Conference on the Enhanced Safety of Vehicles, Amsterdam, Holland*. Vol. 47. 2001.

[128] W. Xu et al. "Motion planning under uncertainty for on-road autonomous driving". In: *2014 IEEE International Conference on Robotics and Automation*. May 2014, pp. 2507–2512.

[129] S. Yu et al. "Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems". In: *Systems & Control Letters* 62.2 (2013), pp. 194–200.

[130] M. Zanon et al. "Model predictive control of autonomous vehicles". In: *Optimization and Optimal Control in Automotive Systems*. Ed. by H. Waschl et al. Cham: Springer International Publishing, 2014, pp. 41–57.