THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Enabling Safe Autonomous Driving in Uncertain Environments

Based on a Model Predictive Control approach

Ivo Batković

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2022

Enabling Safe Autonomous Driving in Uncertain Environments
Based on a Model Predictive Control approach

Ivo Batković
ISBN 978-91-7905-623-0

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5089
ISSN 0346-718X

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
Email: ivo.batkovic@chalmers.se; ivo.batkovic@gmail.com

Cover:
An illustration by the author of a vehicle approaching an intersection with a moving pedestrian

Printed by Chalmers Reproservice
Gothenburg, Sweden, February 2022

Till mina föräldrar ∼ To my parents ∼ Mojim roditeljima
Zvonko & Marica

Enabling Safe Autonomous Driving in Uncertain Environments

Based on a Model Predictive Control approach

Ivo Batković
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Autonomous driving technologies have been developed in the past decades with the objective of increasing safety and efficiency. However, in order to enable such systems to be deployed on a global scale, the problems and concerns regarding safety must be addressed. The difficulty in providing safety guarantees for autonomous driving applications comes from the fact that the self-driving vehicle needs to be able to handle a diverse set of environments and traffic situations. More specifically, it must be able to interact with other road users, whose intentions cannot be perfectly known.

This thesis proposes a Model Predictive Control (MPC) approach to ensure safe autonomous driving in uncertain environments. While MPC has been widely used in motion planning and control for autonomous driving applications, the standard literature cannot be directly applied to ensure safety (recursive feasibility) in the presence of other road users, i.e., pedestrians, cyclists, and other vehicles. To that end, this thesis shows how recursive feasibility can still be obtained through a slight modification of the MPC controller design.

The results of this thesis build upon the assumption that the behavior of the surrounding environment can be predicted to some extent, i.e., a future motion trajectory with some uncertainty bound can be propagated. Then, by postulating the existence of a safe set for the autonomous driving problem, and requiring that the motion prediction models have a consistent structure, safety guarantees can be derived for an MPC controller.

Finally, this thesis shows that the proposed MPC framework does not only hold in theory and simulations, but that it can also be deployed on a real vehicle test platform and operate in real-time, while still ensuring that the conditions needed for the derived safety guarantees hold.

Keywords: Autonomous driving, model predictive control, uncertain environments, robust constraint satisfaction

List of Publications

This thesis is based on the following publications:

[A] **Ivo Batković**, Nils Lübbe, Mario Zanon, and Paolo Falcone, "A Computationally Efficient Model for Pedestrian Motion Prediction". *Published in 2018 European Control Conference (ECC).*

[B] **Ivo Batković**, Mohammad Ali, Mario Zanon, and Paolo Falcone, "Real-Time Constrained Trajectory Planning and Vehicle Control for Proactive Autonomous Driving With Road Users". *Published in 2019 European Control Conference (ECC).*

[C] **Ivo Batković**, Ugo Rosolia, Mario Zanon, and Paolo Falcone, "A Robust Scenario MPC Approach for Uncertain Multi-Modal Obstacles". *Published in 2020 IEEE Control Systems Letters.*

[D] **Ivo Batković**, Mohammad Ali, Paolo Falcone, and Mario Zanon, "Safe Trajectory Tracking in Uncertain Environments". *Provisionally accepted in IEEE Transactions on Automatic Control.*

[E] **Ivo Batković**, Mohammad Ali, Paolo Falcone, and Mario Zanon, "Model Predictive Control with Infeasible Reference Trajectories". *Submitted to IEEE Transactions on Automatic Control.*

[F] **Ivo Batković**, Ankit Gupta, Paolo Falcone, and Mario Zanon, "Experimental Validation of Safe MPC for Autonomous Driving in Uncertain Environments". To be submitted to IEEE Transactions on Control Systems Technology.

Other publications by the author, not included in this thesis, are:

[G] Tommy Tram, **Ivo Batković**, and Jonas Sjöberg, "Learning When to Drive in Intersections by Combining Reinforcement Learning and Model Predictive Control". *Published in 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, Oct. 2019.

[H] ParkPredict: Motion and Intent Prediction of Vehicles in Parking Lots, "Xu Shen, **Ivo Batković**, Vijay Govindarajan, Paolo Falcone, Trevor Darrell, and Francesco Borrelli". *Published in 2020 IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, U.S.A., Nov. 2020.

[I] **Ivo Batković**, Mario Zanon, and Paolo Falcone, "Model Predictive Control for Safe Autonomous Driving Applications". *Accepted book chapter in AI-enabled Technologies for Autonomous and Connected Vehicles*, Springer.

Acknowledgments

As this journey is coming to an end, I realize that the most important memories and experiences are those that I shared with my friends and colleagues along the way. Therefore, it is only right to take a moment to acknowledge the people who have, in different ways, been part of this journey.

First, I want to thank my supervisor Professor Paolo Falcone. Thank you for always being supportive, for the motivation when times were tough, and for the possibility of pursuing a doctoral degree as your student. It has truly been an honor and a pleasure. I also want to thank my examiner Professor Jonas Sjöberg for accepting and welcoming me in the Mechatronics research group.

Professor Mario Zanon deserves recognition in particular. Mario, I had the privilege to work with you and learn from you during these years. Thank you for all the time and energy you put into this project. I have truly enjoyed all our technical, and non-technical, discussions during these years, and I will always consider you as one of my supervisors, but also as one of my friends.

I also want to express gratitude to my industrial supervisors Dr. Nils Lübbe and Dr. Mohammad Ali, who helped guide me through the academic and industrial objectives of my research. Thank you for your support and commitment during all this time. Furthermore, I am grateful that this work was financially supported by Autoliv, Zenuity, Zenseact, and the Wallenberg AI, Autonomous Systems and Software Program (WASP); and for the support from my former managers Dr. Erik Rosén, Benny Nilsson, Mats Lestander, and Dr. Mats Nordlund. I am also grateful for being part of the Advanced Graduate Program at Zenseact under the leadership of Dr. Mats Nordlund and Dr. Carl Lindberg. You have been a great support for me and all PhD students at Zenseact.

The Chalmers Revere lab deserves special recognition since the experiments in this thesis would not have been possible without it. Therefore, I would like to thank Professor Christian Berger, Dr. Fredrik Von Corswant, and Arpit Karsolia for all the help and support, but also for the great work environment throughout the years. You always turned the lab visits into a great experience.

I especially want to thank my great friends with whom I started the PhD journey with. Arian Ranjbar, you have been a great office-mate, and you always ensured that it was fun to work from “skrubben” (our Chalmers “office”). Tommy Tram, you have been a great colleague and a great friend throughout this journey. I will always be grateful for your endless support and for all

the good times we have shared. I am sure there are more to come. Ankit Gupta, it was impossible to walk past your office without starting some sort of conversation with you. I'm not sure how many hours we spent talking, instead of doing actual research. Thank you for all the good times. Special thanks go to Amrit Krishnan and Fredrik Hoxell for being great friends, and supporting me whenever times were tough. I am also deeply grateful to my friends and colleagues Giuseppe, Robert, Stefan, Krook, John, Daniel, Dapeng, Erik, Emil, Anton, Rui, Lars, Ramin, Simon, Fredrik, Elena, Masoud, Angelos, Angel, Remi, Albert, Carl-Johan, Muhammad, Sten, Maximilian, and the rest of my colleagues from Chalmers, Zenseact, and WASP.

I would like to thank WASP for all the interesting study trips, courses, and events that not only widened my skills, but also introduced me to a vast network of colleagues and friends across the world. In addition, the WASP exchange program gave me the opportunity to spend six months as a visiting PhD student at the Model Predictive Control Lab at U.C. Berkeley. To that end, I am deeply grateful that Professor Francesco Borrelli gave me the opportunity to spend time and work with his research group. Additionally, special thanks go to the MPC lab, especially to Vijay, Xu, Tommaso, Ugo, and Monimoy, for making my exchange a pleasant and memorable experience.

Na kraju bih želio posebno zahvaliti svojoj obitelji, ocu Zvonki, majci Marici, sestri Antoniji i bratu Josipu. Osobitu zahvalnost dugujem zaručnici Ivani. Hvala vam svima na beskrajnoj ljubavi i podršci tijekom ovog dijela mog života. Bez vas ovo ne bi bilo moguće.

Ivo Batković,
Göteborg, February, 2022.

Acronyms

AD:	Autonomous Driving
ADAS:	Advanced Driving Assistance Systems
DNN:	Deep Neural Networks
ISS:	Input-to-State Stability
LMI:	Linear Matrix Inequalities
LPV:	Linear Parameter-Varying
LQR:	Linear Quadratic Regulator
LTV:	Linear Time-Varying
MPC:	Model Predictive Control
MPFC:	Model Predictive path Following Control
MPFTC:	Model Predictive Flexible trajectory Tracking Control
OCP:	Optimal Control Problem
QP:	Quadratic Program
RRT:	Rapidly exploring Random Tree
RTI:	Real-Time Iteration
RTK:	Real-Time Kinematic
SQP:	Sequential Quadratic Programming

Contents

Part I

Overview

CHAPTER 1

Introduction

A vast amount of time and resources have been spent in the last decade on researching Autonomous Driving (AD) technologies with the objective of increasing safety and efficiency of passengers and goods transportation [1]. Reports have already shown that over one million people are killed every year in road traffic related accidents [2], [3]. Hence, it is envisioned that the development of safe and robust autonomous systems can reduce the number of fatal accidents by drastically decreasing the frequency of road traffic accidents [4]. Consequently, the development of such systems is also believed to be an enabler when it comes to making the transportation sector more efficient. For instance, an automated system does not need to rest like a human driver does, hence, transportation times and costs can be reduced significantly. Furthermore, if robot taxi services become cost-effective and widely available, the need for parking spaces in inner cities can be reduced. However, it must be stressed that in order for self-driving systems to be deployed on a global scale, the problems and concerns regarding *safety* must be addressed, i.e., the AD problem must be solved safely.

Most of the AD features that are commercially available today consist of Advanced Driving Assistance Systems (ADAS), and were first deployed in

structured environments such as highway driving and low-speed parking [5]. However, more general settings, such as urban driving, pose arguably a greater challenge due to the presence of other non-controllable, non-autonomous road users, e.g., pedestrians, cyclists, and other vehicles. Hence, a self-driving vehicle needs to have the capacity to reliably sense the surrounding environment but also, more importantly, to interact *safely* with it. To that end, the challenges arising from designing robust and reliable sensors (e.g., cameras, LIDARs, radars, GPS, HD-maps) to the development of robust perception and motion planning and control algorithms must be solved. Research is continuously pushing the fields of localization [6], [7], object detection and tracking [8]–[10], and planning [11]–[13], to move beyond the current state of the art. However, how to ensure *safe* autonomous driving in complex environments still remains an open problem, and is the main topic of this thesis.

1.1 Problem Formulation

The work presented in this thesis has in particular focused on the following research questions:

Q1. What requirements need to be set on the sensor-suite and prediction algorithms in order to enable safe autonomous driving?

Q2. How should a vehicle controller be designed in order for safety to be proven by design?

Q3. Can a safe AD framework be deployed on a real test platform?

The answer to Q1 does not only provide some indication on the quality of the information that the sensor system is expected to provide, but it also sets requirements on the information extracted from the collected data, i.e., the sensor data must contain enough information so that the current state-of-the-art algorithms can identify any static or dynamic obstacles present in the environment. Models of the environment need to also be developed in order for the vehicle to interpret the surrounding environment, i.e., the environment evolves differently in a dense urban traffic situation compared to a highway-like traffic scenario, hence the vehicle's behavior should adjust accordingly. We note, however, that a key problem that needs to be solved is the prediction of how the environment is going to evolve over the next time

horizon. One can therefore see that the answer to Q2 relies partly on Q1, since a self-driving vehicle needs to primarily ensure that no collisions with other road users will occur. A self-driving vehicle needs to be able to interact with the current environment, and make future plans depending on how the environment evolves. Hence, the answer to Q1 can be used to impose some structure when formulating a problem statement for Q2. Therefore, having a clear understanding of the requirements that the sensor-suite and prediction models are subject to can help in the design process of the controller. Finally, Q3 answers whether the proposed solutions to answer Q1 and Q2 are sufficient for a practical implementation of a safe autonomous driving framework.

In order to answer Q1-Q3, we propose to combine prediction models of road users with optimization-based techniques. Therefore, Sections 1.1.1 and 1.1.2 mention some available methods in the literature, and the problems they solve.

1.1.1 Environment Prediction

With the rise of interest in the autonomous driving problems, several motion prediction models have been developed for different purposes and are based on different approaches. The work in [14] considered using different linearizations of a unicycle model in order to predict pedestrian motions up to 2 s into the future, while [15] used hybrid models that considered human gaits in order to decide when to transition between different dynamical models. On top of using models based on switching dynamics, situational awareness was introduced in [16] to anticipate changes in pedestrian intent, which only proved to be efficient for short prediction horizons. Learning dynamical models using Gaussian processes have shown promising long-term prediction accuracies [17], [18], but instead often come with high computational complexities [19].

A different class of prediction models are the ones that predict the behavior of several agents and model their interactions. These so-called social force models accurately predict pedestrian movements in crowded environments [20]–[22] by using a potential function that either acts as an attractor or repellent for the modeled agents. The models in [23]–[27] on the other hand assume that each agent is moving stochastically along the shortest path to a set of predefined goals. This is done by first incorporating environmental constraints, e.g., regions that need to be avoided, and then solving for a value function that yields optimal policies for each goal. By sampling the learned policy, one can then predict how an agent would move towards a goal. As it was shown

in, e.g., [26], imposing some structure of the environment can not only help improve the short-term prediction accuracy, but also the long-term accuracy. To that end, the work in [28] introduced a model which incorporated areas that pedestrians are expected to walk on (sidewalks and zebra crossings) in the form of a graph-based network. With this graph, it was then possible to predict the average future motion with an associated uncertainty.

More recent methods, based on Deep Neural Networks (DNNs) have shown remarkable results in predicting the behavior evolution in more complex environments, which include both pedestrians and other vehicles. For instance, the work in [29] used DNNs to predict the future vehicle motion from birds-eye view images alone. By rasterizing the input space, e.g., applying specific color codes for different classes, the authors show that accurate multi-modal predictions can be generated as well as an estimation of their corresponding probabilities by only using 2D images. The work in [30] extends the results of [29] to also generate predictions that are more likely to satisfy the underlying motion kinematics. The work in [31] uses similar ideas based on [29], [30] to predict the driver behavior in parking lots, an environment which is less structured than typical road networks.

While methods based on DNNs manage to yield impressive results without the need of over-engineering feature representations, it becomes difficult to provide the guarantees that set-based prediction methods easily provide [32]. Indeed, the work from [32]–[34] rely on reachability analysis to formally express uncertainty sets which have certain properties that can be used to prove that a self-driving vehicle's planned motions can remain safe [35], [36].

1.1.2 MPC Controller Design

Vehicle motion planning and vehicle control algorithms are of particular interest for self-driving applications since they address the question of how to control the vehicle. The existing literature on motion planning in dynamic environments [12], [37] show that graph-based approaches have been rather successful in generating paths or trajectories that are suitable to be followed or tracked [38]–[40], where the Randomly Rapid Trees (RRT) and A* algorithms are commonly used together with a model of the system dynamics [41] to generate a general plan.

In structured autonomous driving settings, such as highway-driving and urban driving, the reference path can in general be obtained by the sensor

system, i.e., the center of the detected road lane, or a predefined map. In that sense, the problem of "where" the vehicle should drive is generally solved, but the problem of "how" the vehicle should drive is not, i.e., what velocity profile should the vehicle have, what distance should it keep to the lead vehicle, when should it yield to pedestrians, etc. It must be stressed, however, that the path planning problem is still relevant in such settings, as the constant presence of, e.g., road works and obstructing traffic, is not uncommon and will force vehicles to deviate from the standard road infrastructure.

Optimization-based techniques, such as Model Predictive Control (MPC), have been proven to be a favorable design tool when it comes to solving the problem of "how" self-driving vehicles should behave. Many approaches solve this problem from different point of views such as: energy consumption minimization [42]–[45], optimal coordination [46]–[48], and planning [49]–[59]. In most of these settings, a reference path is already given, and the optimization problem is designed to minimize a performance criterion while satisfying constraints that are either imposed by the internal system or by the environment. Notable examples include: (a) the maneuvering of large vehicles in highly constrained environments [41], [59], [60], (b) controlling vehicles in an intersection safely and efficiently [42], [48], and (c) controlling a vehicle in environments with uncontrollable moving obstacles [50], [58].

Indeed, the MPC framework offers well-known and well-established tools that can provide closed-loop stability w.r.t. a reference trajectory, while ensuring constraint satisfaction [61], [62]. However, a severe limitation is that these results are based on assumptions that can be challenging, or impossible, to satisfy for practical autonomous driving applications. More importantly, ensuring that the self-driving vehicle will make safe decisions at all times, i.e., ensuring *recursive feasibility* of the MPC controller, remains an open problem in environments where other road users are present. While the work in [48] proved that it is possible to formulate conditions under which the optimal coordination problem is safe for multiple vehicles, it relied on the fact that a "coordinator" is able to influence when and how each vehicle should enter the intersection. To that end, considering general autonomous driving settings, where human-driven vehicles, cyclists and pedestrians are present, the approach in [48] is not directly applicable.

The difficulty with ensuring recursive feasibility in uncertain environments is that the vehicle needs to interact with road users that may appear almost

anywhere and at any point in time within the sensor range. Hence, ensuring that the motion planner can persistently ensure collision-free trajectories at all times remains an open problem for such settings. This thesis aims to address this remaining problem by proposing ways to enable recursive collision-avoidance constraint satisfaction in the presence of pop-up or moving obstacles, under the condition that the uncertainty on their predicted motion has a consistent characterization. In other words, the uncertainty related to the predicted motion of road users is not allowed to grow as new information about the surrounding environment becomes available.

1.1.3 Scope

This thesis has focused on enforcing safety guarantees in MPC motion planning and control algorithms for AD applications. Furthermore, we assume throughout the thesis that the vehicle's sensing capabilities only come from onboard sensors, such as, e.g., cameras, LIDARs, radars, etc., and that state-of-the-art methods are able to provide the vehicle with necessary information. Therefore, sensor models and state-of-the-art methods for detecting objects from sensor data lies outside the scope of this thesis.

In addition, all experiments related to this thesis have been performed in gated controlled environments, where other road users have been simulated due to safety concerns.

1.1.4 Contributions

The main contributions of this thesis are:

1. A computationally efficient model for pedestrian motion prediction, which has been validated with an MPC framework on a real vehicle platform (Chapter 3 and Paper A, B).

2. A flexible trajectory tracking framework called Model Predictive Flexible trajectory Tracking Control (MPFTC), which relaxes the trajectory tracking requirement and enables less aggressive tracking behaviors (Chapter 2 and Paper D).

3. The derivation of conditions for persistent feasibility for a general MPC controller in uncertain environments (Chapter 3 and Paper D).

4. A control scheme which ensures robust constraint satisfaction with respect to multi-modal uncertainties that stem from other road users' behaviors (Chapter 3 and Paper C).

5. Proving under which conditions infeasible reference trajectories can be used while still ensuring tracking stability of the MPC controller (Chapter 2 and Paper E).

6. Experimental verification of the proposed safety conditions using an MPC framework on a real test setup (Chapter 4 and Paper F).

1.2 Thesis Outline

Part I of this thesis consists of six chapters, and serves as an introduction to Part II. Chapter 1 introduces the motivation and scope of this thesis, while Chapter 2 serves as a general introduction to MPC and some MPC-related technical contributions (Paper D, Paper E). Chapter 3 puts the derived work of this thesis in the setting of AD applications (Paper A, Paper B, Paper C) and introduces conditions needed to safely apply MPC to AD applications, which is instrumental to Part II. Chapter 4 provides simulation and experimental validation of a safe MPC framework (Paper F). Finally, Chapter 5 provides a summary of the included papers from Part II, while Chapter 6 concludes Part I with final remarks and outlines future research directions.

CHAPTER 2

Technical Background

In this chapter we give an overview of MPC in general and describe some of the standard properties which are used in Part II of this thesis. Section 2.1 introduces the general MPC problem formulation and provides asymptotic stability conditions w.r.t. a predefined reference trajectory. Section 2.2 extends the tracking results to also infeasible reference trajectories, and proves that a form of stability can still be ensured. Finally, Section 2.3 proposes an alternative MPC framework which enables flexible tracking of reference trajectories.

2.1 Standard Model Predictive Control

This section introduces a standard discrete-time MPC formulation. For a complete overview on the topic, we refer the interested reader to [61].

MPC is an optimization-based control technique where an Optimal Control Problem (OCP) is repeatedly solved over a receding limited time horizon, starting from the current system state. In particular, for every time instance, a mathematical model of the controlled system is used to simulate the future states over a finite horizon, while a sequence of control inputs are selected and optimized given an objective cost function. The first element in the sequence

of control inputs is then applied to the real system, and a new OCP with an updated state is solved at the next time instance.

2.1.1 Notation

In order to formulate the OCP, we assume that the system we want to control can be expressed by the following discrete-time nonlinear dynamical model

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{2.1}$$

where k is the current time instance, $\mathbf{x}_k \in \mathbb{R}^{n_\mathbf{x}}$ is state of the system, $\mathbf{u}_k \in \mathbb{R}^{n_\mathbf{u}}$ is the control input, i.e., how the system is actuated, and the system dynamics are given by the function $f(\mathbf{x}_k, \mathbf{u}_k) : \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_\mathbf{u}} \to \mathbb{R}^{n_\mathbf{x}}$. The state and input are in general subject to constraints that are known a-priori and are expressed as $h_k(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_\mathbf{u}} \to \mathbb{R}^{n_h}$, i.e., the state and input must satisfy $h(\mathbf{x}, \mathbf{u}) \leq 0$, where the inequalities are defined element-wise.

Throughout the remainder of this thesis, we will use the notation $\mathbf{x}_{n|k}$ to denote a predicted state (or a function) at time n, given the current time k. Hence, using the system (2.1), a predicted finite sequence of control inputs $\mathbb{U}_k = \{\mathbf{u}_{k|k}, \mathbf{u}_{k+1|k}, ...\mathbf{u}_{k+N-1|k}\}$ yield the corresponding sequence of predicted states $\mathbb{X}_k = \{\mathbf{x}_{k|k}, \mathbf{x}_{k+1|k}, ..., \mathbf{x}_{k+N|k}\}$, where $\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$ for all $n \geq k$. Moreover, in the case that $h(\mathbf{x}, \mathbf{u})$ is time-varying, we note that $h_{n|k}(\mathbf{x}, \mathbf{u}) := h_n(\mathbf{x}, \mathbf{u})$ holds $\forall k$ by definition. In addition, to denote a set of integers, we use $\mathbb{I}_a^b := \{a, a+1, ..., b\}$.

2.1.2 Problem Description

In standard MPC formulations, the goal is to design a controller that steers the system towards a reference which might either be a setpoint $\mathbf{r}^\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$, $\mathbf{r}^\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$, or a parameterized reference trajectory $\mathbf{r}(\tau) := (\mathbf{r}^\mathbf{x}(\tau), \mathbf{r}^\mathbf{u}(\tau))$. In the most common settings, the reference parameter τ is selected to be time, so that its natural dynamics are given by

$$\tau_{k+1} = \tau_k + t_\mathrm{s}, \tag{2.2}$$

where t_s is the sampling time for sampled-data systems and $t_\mathrm{s} = 1$ in the discrete-time framework.

The objective of tracking the given reference $\mathbf{r}(\tau)$, is formulated as the minimization of the following performance criterion

$$V(\mathbf{x}_k, \tau_k) = \sum_{n=k}^{k+N-1} q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_n) + p_{\mathbf{r}}(\mathbf{x}_{n|k}, \tau_{k+N}), \tag{2.3}$$

which is minimized for $(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) = (\mathbf{r}^{\mathbf{x}}(\tau_n), \mathbf{r}^{\mathbf{u}}(\tau_n))$, i.e., the minimum for $q_{\mathbf{r}}$ and $p_{\mathbf{r}}$ is attained at the reference $\mathbf{r}(\tau)$. Therefore, in order to select the sequence of control inputs $\mathbf{u}_{n|k}$ that give the best performance criterion, i.e., the sequence that is closest to the reference trajectory $\mathbf{r}(\tau)$, we formulate the following OCP

$$V(\mathbf{x}_k, \tau_k) := \min_{\mathbf{x},\mathbf{u}} \sum_{n=k}^{k+N-1} q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{k+n}) + p_{\mathbf{r}}(\mathbf{x}_{k+N|k}, \tau_{k+N}) \tag{2.4a}$$

$$\text{subject to (s.t.) } \mathbf{x}_{k|k} = \mathbf{x}_k, \tag{2.4b}$$

$$\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), \qquad n \in \mathbb{I}_k^{k+N-1}, \tag{2.4c}$$

$$h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \qquad n \in \mathbb{I}_k^{k+N-1}, \tag{2.4d}$$

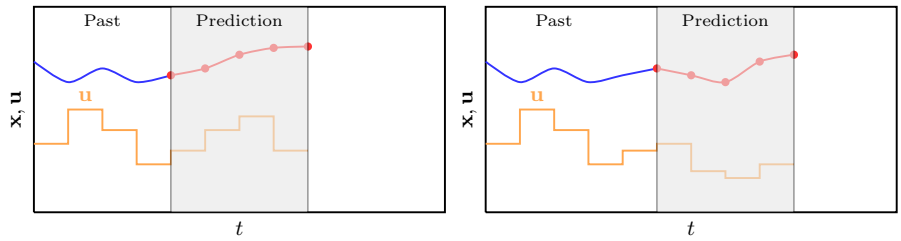
$$\mathbf{x}_{k+N|k} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(\tau_{k+N}), \tag{2.4e}$$

where N is the prediction horizon. In tracking MPC, typical choices for the costs $q_{\mathbf{r}}$ and $p_{\mathbf{r}}$ are given by

$$q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_n) := \begin{bmatrix} \mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(\tau_n) \\ \mathbf{u}_{n|k} - \mathbf{r}^{\mathbf{u}}(\tau_n) \end{bmatrix}^{\top} W \begin{bmatrix} \mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(\tau_n) \\ \mathbf{u}_{n|k} - \mathbf{r}^{\mathbf{u}}(\tau_n) \end{bmatrix}, \tag{2.5}$$

$$p_{\mathbf{r}}(\mathbf{x}_{n|k}, \tau_n) := (\mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(\tau_n))^{\top} P (\mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(\tau_n)), \tag{2.6}$$

where the matrices $W \in \mathbb{R}^{(n_{\mathbf{x}}+n_{\mathbf{u}})\times(n_{\mathbf{x}}+n_{\mathbf{u}})}$ and $P \in \mathbb{R}^{n_{\mathbf{x}}\times n_{\mathbf{x}}}$ are symmetric positive-definite. Furthermore, we note that the cost functions $q_{\mathbf{r}}$ and $p_{\mathbf{r}}$ depend on τ only through the reference trajectory. Note that in the case of reference set points, the costs $q_{\mathbf{r}}$ and $p_{\mathbf{r}}$ are time invariant. As mentioned above, the predicted state and controls are defined as $\mathbf{x}_{n|k}$ and $\mathbf{u}_{n|k}$, respectively, and are subject to constraints (2.4b)-(2.4e). Constraint (2.4b) initializes the state prediction to the current system state $\mathbf{x}_k$, (2.4c) imposes that the predicted states satisfy the system dynamics, and (2.4d) enforces constraints stemming from, e.g., physical actuator limitations and reference trajectory bounds. Finally, (2.4e) is a terminal set constraint which, differently from



(a) Time t_k. **(b)** Time t_{k+1}.

Figure 2.1: Illustration of the receding horizon strategy.

standard formulations, depends on the auxiliary state τ_{k+N} relative to the reference parameter.

Solving Problem (2.4) at time k, yields the optimal sequence of inputs $\mathbb{U}_k^\star = \{\mathbf{u}_{k|k}^\star, \mathbf{u}_{k+1|k}^\star, ..., \mathbf{u}_{k+N-1|k}^\star\}$ and states $\mathbb{X}_k^\star = \{\mathbf{x}_{k|k}^\star, \mathbf{x}_{k+1|k}^\star, ..., \mathbf{x}_{k+N|k}^\star\}$. While these sequences minimize the performance criterion (2.4a) subject to constraints (2.4b)-(2.4e), an MPC controller does not entirely apply $\mathbb{U}_k^\star$ to the real system, instead, the sequence $\mathbb{U}_k^\star$ is applied to the system in a *receding horizon* fashion which can be summarized with the following steps

1. obtain the current measurement (or estimate) of $\mathbf{x}_k$ at time instance k;
2. solve OCP (2.4) with the (estimated) state $\mathbf{x}_k$ and obtain $\mathbb{U}_k^\star$;
3. apply $\mathbf{u}_{k|k}^\star$ to the real system; and
4. wait until $t_k + t_\mathrm{s}$ and go to step 1.

This strategy allows MPC to use the system model (2.1) to form predictions over a finite time horizon, and compute the optimal actions $\mathbb{U}_k^\star$ at each time instance k. In order to obtain a state-feedback policy, only the first input $\mathbf{u}_{k|k}^\star$ is applied, and the same process is repeated at time $k+1$. This introduces feedback in the MPC scheme and makes it possible to compensate for model mismatch and disturbances that the real system is subject to. An illustration of these operating principles is given in Figure 2.1.

MPC can be computationally demanding, since it requires that an optimization problem is solved every sampling time instance. To that end, it is clear that MPC relies heavily on optimization techniques that enable sufficiently

fast solutions of the underlying OCP (2.4). Although a detailed discussion on the variety of customized optimization techniques is outside the scope of this thesis, we note the following. For convex (quadratic/linear costs, linear dynamics, convex constraint sets) MPC problems, the OCP can be formulated as a Quadratic Program (QP) and efficient solvers [63]–[65] are used to solve the underlying OCP, while for *nonlinear* (nonlinear dynamics and constraints) MPC problems, the Sequential Quadratic Programming (SQP) approach is used, which sequentially approximates the nonlinear problem into a QP. For a detailed discussion on optimization techniques specifically intended for MPC, the reader is referred to [66], [67].

In the next section we recall standard MPC results available in the literature by introducing conditions under which the MPC controller is ensured to track the predefined reference trajectory $\mathbf{r}(\tau)$, while ensuring satisfaction of constraint $h_k(\mathbf{x}_k, \mathbf{u}_k)$.

2.1.3 Reference Tracking

In order to prove that the MPC controller based on (2.4) ensures that the states $(\mathbf{x}_k, \mathbf{u}_k)$ track the user-provided *parameterized reference trajectory* $(\mathbf{r}_k^{\mathbf{x}}, \mathbf{r}_k^{\mathbf{u}}) := (\mathbf{r}^{\mathbf{x}}(\tau_k), \mathbf{r}^{\mathbf{u}}(\tau_k))$, we must first recall the following standard assumptions, see e.g., [62], [68].

Assumption 2.1 (System and cost regularity)**.** *The system model f is continuous, and the stage cost $q_{\mathbf{r}} : \mathbb{R}^{n_{\mathbf{x}}} \times \mathbb{R}^{n_{\mathbf{u}}} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$, and terminal cost $p_{\mathbf{r}} : \mathbb{R}^{n_{\mathbf{x}}} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$, are continuous at the reference $(\mathbf{r}_k^{\mathbf{x}}, \mathbf{r}_k^{\mathbf{u}})$ and satisfy $q_{\mathbf{r}}(\mathbf{r}_k^{\mathbf{x}}, \mathbf{r}_k^{\mathbf{u}}, \tau_k) = 0$, and $p_{\mathbf{r}}(\mathbf{r}_k^{\mathbf{x}}, \tau_k) = 0$. Additionally, $q_{\mathbf{r}}(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \geq \alpha_1(\|\mathbf{x}_k - \mathbf{r}_k^{\mathbf{x}}\|)$ for all feasible $\mathbf{x}_k$, $\mathbf{u}_k$, and $p_{\mathbf{r}}(\mathbf{x}_k, \tau_k) \leq \alpha_2(\|\mathbf{x}_k - \mathbf{r}_k^{\mathbf{x}}\|)$, where α_1 and α_2 are $\mathcal{K}_\infty$-functions.*

This assumption is commonly seen in MPC [62], [68], where the MPC problem is instead formulated as a setpoint stabilization problem. Furthermore, it can be relaxed in case one wants to adopt "economic" costs, see, e.g., [69]–[76] for a generic theory and [42], [76] for applications to autonomous driving.

Assumption 2.2 (Reference feasibility)**.** *The reference is feasible for the system dynamics, i.e., $\mathbf{r}^{\mathbf{x}}(t + t_{\mathrm{s}}) = f(\mathbf{r}^{\mathbf{x}}(t), \mathbf{r}^{\mathbf{u}}(t))$, and the reference satisfies the known constraints* (2.4d)*, i.e., $h_n(\mathbf{r}^{\mathbf{x}}(t_n), \mathbf{r}^{\mathbf{u}}(t_n)) \leq 0$, for all $n \in \mathbb{I}_0^\infty$.*

Assumption 2.3 (Stabilizing Terminal Conditions)**.** *There exists a parametric stabilizing terminal set* $\mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t)$ *and a terminal control law* $\kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)$ *yielding:*

$$\mathbf{x}_{+}^{\kappa} = f(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)), \qquad t_{+} = t + t_{\mathrm{s}},$$

such that $p_{\mathbf{r}}(\mathbf{x}_{+}^{\kappa}, t_{+}) - p_{\mathbf{r}}(\mathbf{x}, t) \leq -q_{\mathbf{r}}(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t), t)$*, and* $\mathbf{x} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t) \Rightarrow \mathbf{x}_{+}^{\kappa} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t_{+})$*, and* $h_n(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)) \leq 0$*, for all* $n, k \in \mathbb{I}_0^{\infty}$*.*

Assumptions 2.1-2.3 make it possible to derive the following standard stability result.

Proposition 2.1 (Nominal Asymptotic Stability)**.** *Suppose that Assumptions 2.1, 2.2, and 2.3 hold, and that the initial state* $(\mathbf{x}_k, \tau_k)$ *at time* k *belongs to the feasible set of Problem* (2.4)*. Then the system* (2.1) *in closed loop with the solution of* (2.4) *applied in receding horizon is an asymptotically stable system.*

Proof. The proof follows from standard arguments, see, e.g., [61], [62]. □

Proposition 2.1 recalls well-known stability results from the existing literature which apply to tracking MPC schemes. We emphasize that the design procedure resulting from Proposition 2.1 requires one to precompute a feasible reference trajectory $(\mathbf{r}_k^{\mathbf{x}}, \mathbf{r}_k^{\mathbf{u}})$ that satisfies Assumption 2.2. However, in practice, it may be convenient to use a reference trajectory that is infeasible w.r.t. the system dynamics, but simpler to define. For example, when designing a motion planner and controller for an autonomous vehicle, it would be convenient to use just use the available lane centerline as a reference trajectory, even if it would not in general be feasible for kinematic or dynamic vehicle models.

In the next section, we will show that Assumption 2.2 can be relaxed by resorting to Input-to-State Stability (ISS) analysis.

2.2 Infeasible References

In [77] stability with respect to an unreachable set point was studied, however, the approach therein only applies to time-invariant infeasible references. In order to overcome such a limitation and extend the results, we next consider a setting where the reference can be time-varying, but does not need to satisfy Assumption 2.2, and the terminal conditions (2.4e) do not need to hold for the

reference trajectory $\mathbf{r}(\tau)$, but in a neighborhood around it. To state the main result of this section (Theorem 2.2), we first must introduce a few preliminary results.

2.2.1 Economic MPC Formulation

Consider the optimal state and input trajectories obtained from the infinite-horizon OCP

$$(\mathbf{x}^{\mathrm{r}}, \mathbf{u}^{\mathrm{r}}) := \lim_{M\to\infty} \arg\min_{\boldsymbol{\xi},\boldsymbol{\nu}} \sum_{n=0}^{M-1} q_{\mathbf{r}}(\boldsymbol{\xi}_n, \boldsymbol{\nu}_n, \tau_n) + p_{\mathbf{r}}(\boldsymbol{\xi}_M, \tau_n) \tag{2.7a}$$

$$\text{s.t. } \boldsymbol{\xi}_0 = \mathbf{x}_0, \tag{2.7b}$$

$$\boldsymbol{\xi}_{n+1} = f(\boldsymbol{\xi}_n, \boldsymbol{\nu}_n), \qquad n \in \mathbb{I}_0^{M-1}, \tag{2.7c}$$

$$h(\boldsymbol{\xi}_n, \boldsymbol{\nu}_n) \leq 0, \qquad n \in \mathbb{I}_0^{M-1}, \tag{2.7d}$$

and let $\mathbf{y}^{\mathrm{r}} := (\mathbf{x}^{\mathrm{r}}, \mathbf{u}^{\mathrm{r}})$ denote the solution of (2.7), and its multipliers as $\boldsymbol{\lambda}^{\mathrm{r}}$ and $\boldsymbol{\mu}^{\mathrm{r}}$. Hereafter we will refer to $\mathbf{y}^{\mathrm{r}}$ as the *feasible* reference, as it now is ensured to satisfy Assumption 2.2.

The result in Theorem 2.2 builds upon the stability theory for *economic* MPC schemes, where the cost is not of *tracking* type. In this section we only recall that the main difference between economic and tracking MPC schemes is in the cost function. For tracking schemes, the cost function satisfies

$$q_{\mathbf{r}}(\mathbf{x}_k^{\mathrm{r}}, \mathbf{u}_k^{\mathrm{r}}, \tau_k) = 0,\ q_{\mathbf{r}}(\mathbf{x}_k, \mathbf{u}_k, \tau_k) > 0,\ \forall\ \mathbf{x}_k \neq \mathbf{x}_k^{\mathrm{r}},\ \mathbf{u}_k \neq \mathbf{u}_k^{\mathrm{r}}, \tag{2.8}$$

but not in economic ones. For an in-depth review of the stability analysis tools for economic MPC, we refer the reader to [69], [73], [78], [79]. An MPC scheme that satisfies Assumption 2.2 is by construction of tracking type. However, if the reference trajectory is *infeasible* with respect to the system dynamics, an economic cost is instead obtained, i.e., $q_{\mathbf{r}}(\mathbf{x}_k^{\mathrm{r}}, \mathbf{u}_k^{\mathrm{r}}, \tau_k) \neq 0$ and $p_{\mathbf{r}}(\mathbf{x}_k^{\mathrm{r}}, \tau_k) \neq 0$.

In order to retrieve a tracking cost from the economic one, we introduce the following *rotated* costs

$$\begin{aligned}\bar{q}_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_n) &:= q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_n) - q_{\mathbf{r}}(\mathbf{x}_n^{\mathrm{r}}, \mathbf{u}_n^{\mathrm{r}}, \tau_n) \\ &\quad + \boldsymbol{\lambda}_n^{\mathrm{r}\top}(\mathbf{x}_{n|k} - \mathbf{x}_n^{\mathrm{r}}) - \boldsymbol{\lambda}_{n+1}^{\mathrm{r}\top}(f_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) - f_n(\mathbf{x}_n^{\mathrm{r}}, \mathbf{u}_n^{\mathrm{r}})),\end{aligned} \tag{2.9}$$

$$\bar{p}_{\mathbf{r}}(\mathbf{x}_{n|k}, \tau_n) := p_{\mathbf{r}}(\mathbf{x}_{n|k}, \tau_n) - p_{\mathbf{r}}(\mathbf{x}_n^{\mathrm{r}}, \tau_n) + \boldsymbol{\lambda}_n^{\mathrm{r}\top}(\mathbf{x}_{n|k} - \mathbf{x}_n^{\mathrm{r}}), \tag{2.10}$$

that are commonly used in economic MPC. The idea behind the rotated cost is to adjust the stage and terminal costs $q_{\mathbf{r}}$ and $p_{\mathbf{r}}$ such that the minimum is attained at the *feasible* reference $(\mathbf{x}^{\mathrm{r}}, \mathbf{u}^{\mathrm{r}})$, i.e., $\bar{q}_{\mathbf{r}}(\mathbf{x}^{\mathrm{r}}_k, \mathbf{u}^{\mathrm{r}}_k, \tau_k) = 0$, and $\bar{p}_{\mathbf{r}}(\mathbf{x}^{\mathrm{r}}_k, \tau^{\mathrm{r}}_k) = 0$. To ensure this, a requirement is set on the rotated costs to remain positive definite. Hence, we must assume that the dynamics are linear time-varying (LTV) as in Assumption 2.4.

Assumption 2.4. *The system dynamics f are linear time-varying, i.e.,*

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k) = A_k\mathbf{x}_k + B_k\mathbf{u}_k. \tag{2.11}$$

This assumption is rather technical and in practice it is expected that the results can be extended to the fully nonlinear case. Next, we show that if suitable terminal conditions are applied, an MPC scheme using an *infeasible* reference can still be asymptotically stabilized towards a neighborhood of the optimal trajectory $\mathbf{x}^{\mathrm{r}}$.

2.2.2 Ideal Formulation

In order to track the *feasible reference* obtained by solving the infinite-horizon (2.7), we formulate the following *ideal* problem

$$V^{\mathrm{i}}(\mathbf{x}_k, \tau_k) = \min_{\mathbf{x},\mathbf{u}} \sum_{n=k}^{k+N-1} q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_n) + p_{\tilde{\mathbf{y}}^{\mathrm{r}}}(\mathbf{x}_{k+N|k}, \tau_{k+N}) \tag{2.12a}$$

$$\text{s.t. } (2.4\text{b}) - (2.4\text{d}),\ \mathbf{x}_{k+N|k} \in \mathcal{X}^{\mathrm{f}}_{\mathbf{y}^{\mathrm{r}}}(\tau_{k+N}), \tag{2.12b}$$

where

$$\tilde{\mathbf{y}}^{\mathrm{r}}_k := \arg\min_{\mathbf{x}} p_{\mathbf{y}^{\mathrm{r}}}(\mathbf{x}, \tau_k) - \boldsymbol{\lambda}^{\mathrm{r}\top}_k(\mathbf{x} - \mathbf{x}^{\mathrm{r}}_k). \tag{2.13}$$

We refer to this formulation as *ideal* since it uses terminal conditions that cannot in general be known, unless one solves the OCP (2.7).

Theorem 2.1. *Suppose that*

1. *Assumption 2.1 holds,*
2. *Problem* (2.7) *is feasible,*
3. *Assumption 2.3 holds for $\bar{q}_{\mathbf{r}}$ and $\bar{p}_{\tilde{\mathbf{y}}^{\mathrm{r}}}$, with terminal set $\mathcal{X}^{\mathrm{f}}_{\mathbf{y}^{\mathrm{r}}}$, and*

4. Assumption 2.4 holds for the system dynamics.

Then, the system (2.1) *in closed-loop with the ideal MPC* (2.12) *is asymptotically stabilized to the optimal trajectory* $\mathbf{x}^{\mathrm{r}}$.

Proof. See proof in [80] (Paper E). □

Theorem 2.1 establishes the first steps towards proving Theorem 2.2, by proving that an MPC problem, which uses an infeasible reference $(\mathbf{r}_k^{\mathbf{x}}, \mathbf{r}_k^{\mathbf{u}})$, can stabilize the system (2.1) to the *feasible* reference obtained by solving (2.7). However, we note that in order to be able to formulate Problem (2.12), one must express the terminal set $\mathcal{X}_{\mathbf{y}^{\mathbf{r}}}^{\mathrm{f}}(\tau_{k+N})$ as a positive invariant set (point 3 in Theorem 2.1) containing $\mathbf{x}^{\mathrm{r}}$, and a terminal control law that stabilizes system (2.1) to $\mathbf{x}^{\mathrm{r}}$. In other words, one has to solve Problem (2.7).

On the other hand, if we express the terminal conditions on the infeasible reference $\mathbf{r}^{\mathbf{x}}$, asymptotic stability (in the sense of Proposition 2.1) cannot be proven anymore. We therefore resort to Input-to-State Stability (ISS) analysis for the closed-loop system, where the considered input will be a terminal reference $\mathbf{y}^{\mathrm{f}} = (\mathbf{x}^{\mathrm{f}}, \mathbf{u}^{\mathrm{f}})$ satisfying the following assumption.

Assumption 2.5 (Approximate feasibility of the reference)**.** *The reference* $\mathbf{y}^{\mathrm{f}}$ *satisfies the constraints* (2.4d)*, i.e.,* $h(\mathbf{x}_n^{\mathrm{f}}, \mathbf{u}_n^{\mathrm{f}}) \leq 0$*,* $n \in \mathbb{I}_k^{k+N-1}$*, for all* $k \in \mathcal{N}^+$*. Additionally, recursive feasibility holds for both Problem* (2.4) *and* (2.12) *when the system is controlled in closed-loop using the feedback from Problem* (2.4)*.*

Assumption 2.5 sets a rather mild requirement from a practical standpoint. Using an infeasible reference, yet simple to obtain, or approximating the system dynamics to capture the most relevant dynamics of the system ($\|\mathbf{x}_{n+1}^{\mathrm{f}} - f_n(\mathbf{x}_n^{\mathrm{f}}, \mathbf{u}_n^{\mathrm{f}})\| \leq \epsilon$, for some small ϵ) is not uncommon in practice. In particular, in a practical setting we can select $\mathbf{y}^{\mathrm{f}} = \mathbf{r}(t_{k+N})$, or in an ideal setting $\mathbf{y}^{\mathrm{f}} = \mathbf{y}^{\mathbf{r}}(t_{k+N})$. To that end, we define the following closed-loop dynamics

$$\mathbf{x}_{k+1}(\mathbf{y}^{\mathrm{f}}) = f_k(\mathbf{x}_k, \mathbf{u}_{\mathrm{MPC}}(\mathbf{x}_k, \mathbf{y}^{\mathrm{f}})) = \bar{f}_k(\mathbf{x}_k, \mathbf{y}^{\mathrm{f}}), \tag{2.14}$$

where $\mathbf{u}_{\mathrm{MPC}}$ is obtained as $\mathbf{u}_{k|k}^{\star}$ solving Problem (2.4) in case $\mathbf{y}^{\mathrm{f}} = \mathbf{r}$; and as $\mathbf{u}_{k|k}^{\mathrm{i}}$ solving the *ideal* Problem (2.12) in case $\mathbf{y}^{\mathrm{f}} = \mathbf{y}^{\mathrm{r}}$.

We are now ready to state the main results of this section.

Theorem 2.2. *Suppose that*

1. *Problem* (2.7) *is feasible,*
2. *Assumptions 2.1 and 2.3 hold for the reference* $\mathbf{y}^{\mathrm{r}}$ *with costs* $\bar{q}_{\mathbf{r}}$ *and* $\bar{p}_{\tilde{\mathbf{y}}^{\mathrm{r}}}$ *and terminal set* $\mathcal{X}_{\mathbf{y}^{\mathrm{r}}}$*,*
3. *Problem* (2.4) *and Problem* (2.12) *are feasible at time* k *with initial state* $(\mathbf{x}_k, t_k)$*, and*
4. *the reference* $\mathbf{y}^{\mathrm{f}}$*, with terminal set* $\mathcal{X}^{\mathrm{f}}_{\mathbf{y}^{\mathrm{f}}}$*, satisfies Assumption 2.5.*

Then, system (2.14) *obtained from* (2.1) *in closed-loop with MPC formulation* (2.4) *is ISS.*

Proof. See the proof in [80](Paper E). □

This theorem proves that if an infeasible reference is used, system (2.1) does not converge exactly to the (unknown) optimal trajectory from OCP (2.7), but to a neighborhood around it which depends on how inaccurate the terminal reference is. However, as was proven in [73], [79], the effect that the terminal conditions have on the closed-loop system decreases as the prediction horizon increases.

To illustrate the derived results, we consider next an illustrative example using a robotic arm manipulator.

2.2.3 Results

To illustrate the results of Theorems 2.1 and 2.2, we consider the planar robot with two degrees of freedom presented in [81] with dynamics

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ B^{-1}(x_1)(u - C(x_1, x_2)x_2 - g(x_1)) \end{bmatrix}, \tag{2.15}$$

where $x_1 = (q_1, q_2)$ are the joint angles, and $x_2 = (\dot{q}_1, \dot{q}_2)$ are the joint velocities. The matrices and vectors B, C, and g are given by

$$B(x_1) := \begin{bmatrix} 200 + 50\cos(q_2) & 23.5 + 25\cos(q_2) \\ 23.5 + 25\cos(q_2) & 122.5 \end{bmatrix}, \tag{2.16a}$$

$$C(x_1, x_2) := 25\sin(q_2) \begin{bmatrix} \dot{q}_1 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix}, \tag{2.16b}$$

$$g(x_1) := \begin{bmatrix} 784.8\cos(q_1) + 245.3\cos(q_1 + q_2) \\ 245.3\cos(q_1 + q_2) \end{bmatrix}, \tag{2.16c}$$

and the system is assumed to be subject to the following constraints

$$\|x_2\|_\infty \leq 3/2\pi, \qquad \|u\|_\infty \leq 4000. \tag{2.17}$$

Applying the following transformation

$$u = C(x_1, x_2)x_2 + g(x_1) + B(x_1)v, \tag{2.18}$$

we can rewrite system (2.15) into the linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ v \end{bmatrix}, \tag{2.19}$$

subject to the non-linear input constraint

$$\|C(x_1, x_2)x_2 + g(x_1) + B(x_1)v\|_\infty \leq 4000. \tag{2.20}$$

Defining the reference path

$$p(\theta) = \left(\theta - \frac{\pi}{3},\, 5\sin\left(0.6\left(\theta - \frac{\pi}{3}\right)\right)\right), \tag{2.21}$$

and timing law

$$\theta(t_0) = -5.3,\ \dot{\theta}(t) = \frac{v_{\text{ref}}(t)}{\|\nabla_\theta \rho(\theta(t))\|_2},\ v_{\text{ref}}(t) = \begin{cases} 1 & \text{if } \theta < 0 \\ 0 & \text{otherwise} \end{cases},$$

with $t_0 = 0$ s, we obtain the following reference trajectory

$$\mathbf{r}^{\mathbf{x}}(t) = \begin{bmatrix} p(\theta(t)) & \frac{\partial p}{\partial \theta}\dot{\theta}(t) \end{bmatrix}^\top,\ \mathbf{r}^{\mathbf{u}}(t) = \begin{bmatrix} \frac{\partial^2 p}{\partial \theta^2}\dot{\theta}^2 + \frac{\partial p}{\partial \theta}\ddot{\theta} \end{bmatrix}^\top, \tag{2.22}$$

which has a discontinuity at $\theta = 0$.

For the stage cost we use $W = \text{blockdiag}(Q, R)$ with

$$Q = \text{diag}(10, 10, 1, 1),\ R = \text{diag}(1, 1),$$

and the terminal cost matrix is computed using an LQR controller with the cost defined by Q and R and is given by

$$P = \begin{bmatrix} 290.34 & 0 & 105.42 & 0 \\ 0 & 290.34 & 0 & 105.42 \\ 105.42 & 0 & 90.74 & 0 \\ 0 & 105.42 & 0 & 90.74 \end{bmatrix}.$$

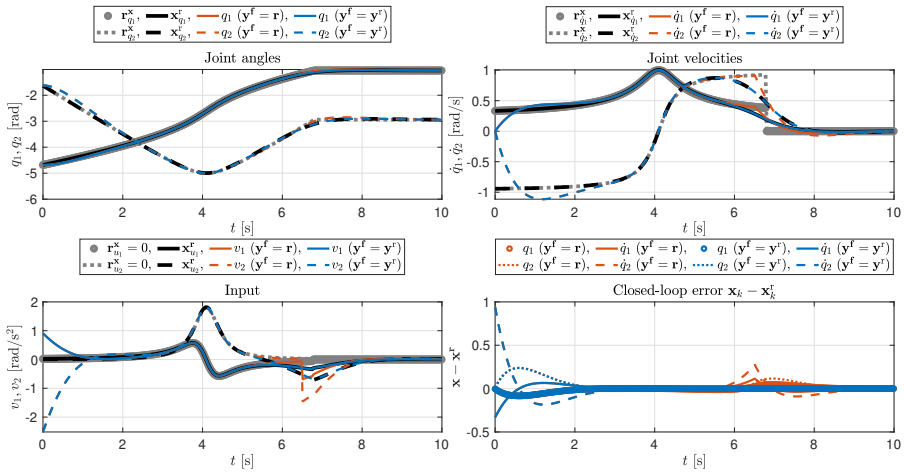


Figure 2.2: Closed-loop simulation for the initial time $k = 167$ and initial condition $(x_1, x_2) = (-4.69, -1.62, 0, 0)$. The reference trajectory $\mathbf{r}(\tau) = (\mathbf{r}^{\mathbf{x}}(\tau), \mathbf{r}^{\mathbf{u}}(\tau))$ is represented by the gray lines, while the feasible reference trajectory $\mathbf{y}^{\mathrm{r}}$ is represented by the black lines. The orange lines show the closed-loop trajectories for the *practical* formulation (2.4) while the blue trajectories show the closed-loop behavior of the *ideal* formulation (2.12).

The corresponding terminal set is given by

$$\mathcal{X}^{\mathrm{f}}_{\mathbf{r}}(t_n) = \{\mathbf{x} \,|\, (\mathbf{x} - \mathbf{r}^{\mathbf{x}}_n)^\top P(\mathbf{x} - \mathbf{r}^{\mathbf{x}}_n) \leq 61.39\}.$$

For detailed derivations of the terminal cost and terminal set, we refer the reader to the Appendix in [36], [81] (Paper D).

Since the reference in (2.22) is clearly infeasible, we obtain the feasible reference $\mathbf{y}^{\mathrm{r}} = (\mathbf{x}^{\mathrm{r}}, \mathbf{u}^{\mathrm{r}})$, by approximating the infinite horizon OCP (2.7) with a prediction horizon of $M = 1200$, and use a sampling time of $t_{\mathrm{s}} = 0.03$ s to discretize the problem. For the closed-loop simulations, we use the control input obtained from formulations (2.4) and (2.12) with horizon $N = 10$. Note that both problems were formulated using the linear system (2.19).

Figure 2.2 shows the closed-loop trajectories for the initial time $k = 167$ and initial condition $(x_1, x_2) = (-4.69, -1.62, 0, 0)$. The infeasible reference $\mathbf{r} = (\mathbf{r}^{\mathbf{x}}, \mathbf{r}^{\mathbf{u}})$ is denoted by the gray lines, while the black lines denote the

optimal reference $\mathbf{y}^{\mathrm{r}} = (\mathbf{x}^{\mathrm{r}}, \mathbf{u}^{\mathrm{r}})$ from (2.7). The closed-loop trajectories from the practical MPC formulation (2.4) are shown with the orange lines, while the closed-loop trajectories from the *ideal* formulation (2.12) are shown with the black lines. For times $t < 5$ s both the practical MPC (orange lines) and the *ideal* MPC (blue lines) stabilize towards the reference $\mathbf{r}$. However, between times $5 \text{ s} \leq t \leq 9$ s, the discontinuity (infeasibility) of the reference trajectory affects how the two formulations behave. The *ideal* formulation successfully tracks the optimal reference (black line), while the practical MPC struggles. After the discontinuity, the rest of the reference trajectory is feasible, enabling both formulations to track the reference.

2.3 Model Predictive Flexible Trajectory Tracking Control

Thus far, we have considered a standard tracking MPC scheme, in which, the reference trajectory evolves through its natural time dynamics. However, in many practical settings, the presence of constraints might force the system to deviate from the reference trajectory by either forcing the system to slow down or completely stop, i.e., Assumption 2.2 does not hold due to the constraints $h(\mathbf{x}, \mathbf{u})$. Therefore, if the reference trajectory is not adjusted in such settings, one typically faces aggressive tracking behaviors if the reference manages to get far away from the system state.

The established Model Predictive Path-Following Control (MPFC) proposed in [81]–[86] alleviates such issues by penalizing deviations from a path, rather than a reference trajectory. By introducing additional variables to the problem formulation, the MPFC framework is able to control the position along the reference path, instead of blindly tracking a reference trajectory. However, one difficulty with this approach is the need to select an appropriate output function and to define a path, which typically is a dimension lower than the system state space $n_{\mathbf{x}}$. While MPFC is a valid technique for tackling this problem, an alternative approach called Model Predictive Flexible trajectory Tracking Control (MPFTC) has been proposed in [36], where based on ideas similar to MPFC, auxiliary variables are introduced to artificially modify the time derivative of the reference trajectory in a time-warping fashion. As noted earlier, while the main difficulty in MPFC is to establish a suitable output and define a path, the main difficulty in MPFTC is to precompute a parameterized

feasible reference trajectory.

To avoid aggressive behaviors which can be caused by constraints $h(\mathbf{x}, \mathbf{u}) \leq 0$, we adapt the dynamics of the reference trajectory by the means of the parameter τ, which can be seen as a fictitious time with dynamics given by

$$\tau_{k+1} = \tau_k + t_\mathrm{s} + v_k, \tag{2.23}$$

where v_k is an auxiliary control input, and τ_k becomes an auxiliary state. It is important to stress that the dynamics in (2.23) do not affect the system dynamics (2.1), but only how the reference dynamics evolve in time.

The MPFTC problem is then formulated as follows

$$\begin{aligned}
V(\mathbf{x}_k, \tau_k) := \min_{\substack{\mathbf{x}\,\mathbf{u} \\ \tau, v}} \; & \sum_{n=k}^{k+N-1} q_\mathbf{r}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) + w v_{n|k}^2 \\
& + p_\mathbf{r}(\mathbf{x}_{k+N|k}, \tau_{k+N|k}) && \text{(2.24a)} \\
\text{s.t. } & \mathbf{x}_{k|k} = \mathbf{x}_k, \; \tau_{k|k} = \tau_k, && \text{(2.24b)} \\
& \mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), && n \in \mathbb{I}_k^{k+N-1}, \quad \text{(2.24c)} \\
& \tau_{n+1|k} = \tau_{n|k} + t_\mathrm{s} + v_{n|k}, && n \in \mathbb{I}_k^{k+N-1}, \quad \text{(2.24d)} \\
& h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, && n \in \mathbb{I}_k^{k+N-1}, \quad \text{(2.24e)} \\
& \mathbf{x}_{k+N|k} \in \mathcal{X}_\mathbf{r}^\mathrm{f}(\tau_{k+N|k}), && \text{(2.24f)}
\end{aligned}$$

where, differently to Problem (2.4), constraint (2.24d) enforces the evolution of the fictitious time dynamics, and $w > 0$ is the weight defining the cost associated with the auxiliary input $v_{n|k}$, which can take any real value. We define $q_\mathbf{r}$ and $p_\mathbf{r}$ according to (2.5) for simplicity, but note that the proposed framework can accommodate more general cost definitions. Furthermore, this formulation faces an increased computational complexity, since an additional state and control variable have been introduced, however, such an increase is typically small since these variables have decoupled linear dynamics.

Remark 2.1. *If the constraint $v_{n|k} = 0$ is added, a standard MPC formulation is obtained. The terminal set $\mathcal{X}_\mathbf{r}^\mathrm{f}$ can therefore be designed as in standard MPC, where one assumes that the reference trajectory evolves according to its natural dynamics* (2.2)*.*

Using the same assumptions as in Section 2.1, and that $w > 0$, we can also prove that asymptotic stability also holds for MPFTC.

Theorem 2.3 (MPFTC Asymptotic Stability)**.** *Suppose that Assumptions 2.1, 2.2, and 2.3 hold, and that the initial state* $(\mathbf{x}_k, \tau_k)$ *at time* k *belongs to the feasible set of Problem* (2.24)*. Then the system* (2.1),(2.23) *in closed loop with the solution of* (2.24) *applied in receding horizon is an asymptotically stable system.*

Proof. The first part of the proof follows standard arguments used to prove stability for MPC *for some* τ_k, while the second part of the proof, which is non-standard, shows that $\lim_{k\to\infty} v_{k|k} = 0$. See [36] (Paper D) for the full derivation. □

Remark 2.2. *Assumption 2.2 can be relaxed to only require the reference to be feasible for an unspecified* $\tau_0 = t_0$*. In this case, the system will be stabilized to the reference with a time shift which is an integer multiple of the sampling time* t_s*. If, instead, feasibility holds for all initial times, then the time shift can be any real number. As opposed to MPFTC, in standard MPC the time shift is* 0 *by construction.*

Remark 2.3. *Note that the initial constraint* $\tau_{k|k} = \tau_k$ *is not necessary, and Theorem 2.3 holds also in case the initial auxiliary state* $\tau_{k|k}$ *is free to be selected by the optimizer.*

Next, we consider a simple simulation example, which highlights the differences between normal MPC and MPFTC when constraints $h_k(\mathbf{x}, \mathbf{u}) \leq 0$ force the system to temporarily deviate from the reference trajectory.

2.3.1 Results

We consider the following double integrator dynamics to illustrate in the simplest fashion the main difference between MPC and MPFTC. The state and control are defined as $\mathbf{x} = [p, \dot{p}]^\top$, $\dot{p} \geq 0$ and $\mathbf{u} = a \in [-5, 5]$, respectively, with reference trajectories $\mathbf{r}^\mathbf{x}(\tau) = [4\tau, 4]^\top$, $\mathbf{r}^\mathbf{u}(\tau) = 0$. For the stage cost we use $W = \mathrm{blockdiag}(Q, R)$ with

$$Q = \mathrm{diag}(10, 10), \qquad R = 1, \qquad w = 1,$$

while the terminal cost matrix P is obtained from the LQR cost corresponding to $Q_\mathrm{LQR} = \mathrm{diag}(1, 1)$, $R_\mathrm{LQR} = 10$, with the corresponding terminal set $\mathcal{X}_\mathbf{r}^\mathrm{f}(t) = \{\mathbf{x} \mid -K(\mathbf{x} - \mathbf{r}^\mathbf{x}(t)) \in [-5, 5]\}$.

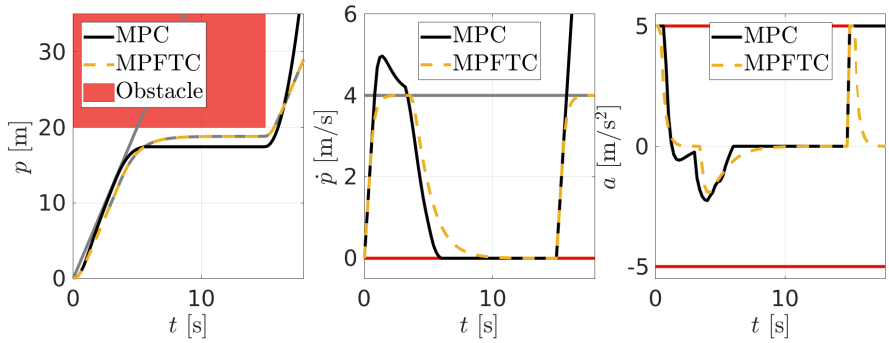


Figure 2.3: Closed-loop trajectories of the double integrator example in Section 2.3.1. The red box denotes the constraint $h(\mathbf{x}, \mathbf{u}) \leq 0$, while the gray lines denote the reference trajectories.

In order to simulate a constraint that does not satisfy Assumption 2.2, we introduce a static obstacle at position

$$p_n^{\text{obs}} := \begin{cases} 20 & \text{for } n \leq 75 \\ \infty & \text{otherwise} \end{cases} \tag{2.25}$$

and formulate the constraint

$$h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := p_{n|k} - p_n^{\text{obs}} \leq 0, \tag{2.26}$$

which upper bounds the position for a finite amount of time.

Figure 2.3 shows the difference in tracking between a standard MPC controller based on (2.4) and the MPFTC controller based on (2.24) which both start from the initial states $\mathbf{x}_k = [0, 0]^\top$ and $\tau_k = 0$. Both controllers were formulated with a sampling time $t_\text{s} = 0.2$ s and used a prediction horizon of $N = 10$. Since the MPC controller starts from standstill, we can see that it is aggressive and tries to "catch up" with the reference trajectory, i.e., it overshoots the velocity reference since the position is also being tracked. The MPFTC controller instead adapts the reference parameter τ and therefore does not cause an overshoot. The need for a reference trajectory adaptation is seen more clearly after $t = 15$ s. Here, MPFTC tracks the reference nicely, while MPC on the other hand tries to catch up with all the "lost time" and reach the reference position that moved far away. Figure 2.4 shows the closed loop

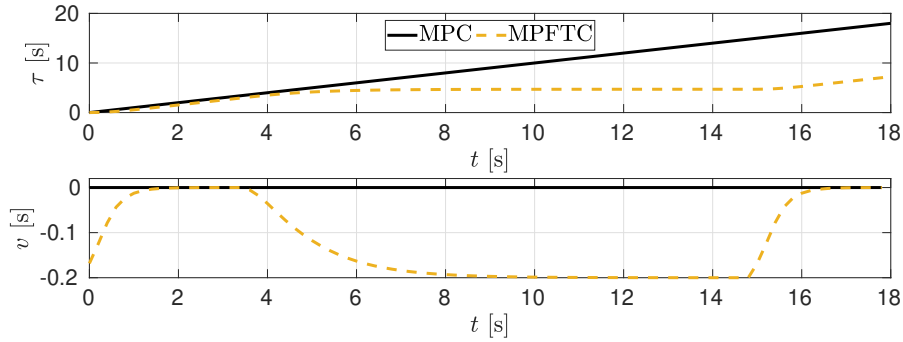


Figure 2.4: The standard MPC (black lines) does not adjust its reference time parameter, and therefore introduces the aggressive tracking shown in Figure 2.3. MPFTC (yellow line) on the other hand adjusts the reference time τ and manages to avoid aggressive tracking behaviors.

trajectories of τ_k and v_k, where it is visible that MPFTC halts the evolution of τ between times $t \in [10,\ 15]$ s.

2.4 Discussion

This chapter has provided a general overview of MPC while recalling a few commonly used technical results. Indeed, we have shown that for standard MPC settings, where the constraints are assumed to be known a-priori and a reference trajectory that satisfies the system dynamics is available, a controller based on MPC can guarantee robust constraint satisfaction, while converging to a reference trajectory. However, requiring that the reference trajectory satisfies the system dynamics can in practice be a tedious task. To that end, using *infeasible reference trajectories*, in the sense that they do not satisfy the system dynamics, is of great interest in MPC-based motion planning and control algorithms, due to the convenience and simplicity they offer. We have discussed how the use of such reference trajectories affect the closed-loop behavior of the system, and proposed in Section 2.2 sufficient conditions for stability to hold. Additionally, we introduced the so-called MPFTC framework, which relaxes the trajectory tracking requirement, while still inheriting the standard MPC stability guarantees. In particular, the main idea behind the

MPFTC framework is to reduce potentially aggressive tracking behaviors when the reference trajectory cannot be freely tracked due to active constraints.

To conclude this chapter we note that while the results presented here hold in theory, one must question the validity of the necessary assumptions. The assumption that a-priori constraints always satisfy the reference dynamics may very well hold true in confined controllable settings where no other agents, or sudden changes in the environment can occur. However, in settings where one needs to control a system which is subject to an environment that is uncontrollable, e.g., typical AD situations, the required assumptions in this chapter become difficult to verify.

In the next chapter, we consider an MPC formulation specifically intended for such uncertain situations, and formulate mild conditions which ensure safety of the controlled system, i.e., the self-driving vehicle.

CHAPTER 3

Model Predictive Control for Autonomous Driving

Chapter 2 provided an overview of MPC, where the reference trajectory is assumed to be feasible at all times with respect to constraints $h(\mathbf{x}, \mathbf{u})$ that are assumed to be known at all times. However, in practical scenarios, especially those typical in AD applications, not all constraints can be known beforehand. In particular, a self-driving vehicle must identify online, e.g., pedestrians that want to cross the road and other vehicles that need to perform lane changes. To that end, ensuring recursive feasibility, and stability, cannot be done by naively applying the results presented in Chapter 2.

In order to clearly distinguish between the class of constraints that are known a-priori, versus constraints that model, e.g., moving obstacles present in the environment, we introduce a new group of constraints we refer to as: a-priori *unknown* constraints. More formally, we denote by $g(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_\mathbf{u}} \to \mathbb{R}^{n_g}$ the unknown constraint, and use the notation $g_{n|k}(\mathbf{x}, \mathbf{u}) \leq 0$ to denote g at time n given the current available information at time k. The real constraint is defined as $g_n(\mathbf{x}, \mathbf{u}) := g_{n|\infty}(\mathbf{x}, \mathbf{u})$ since $g_n \neq g_{n|k}$ in general. On the other hand, we note that for the a-priori known constraints $h_{n|k}(\mathbf{x}, \mathbf{u}) := h_n(\mathbf{x}, \mathbf{u})$ holds by construction.

In order to use MPC for AD settings, our first and most essential objective

is to ensure safety of the controlled system (2.1), which is formally defined as

Definition 3.1 (Safety)**.** *A controller is said to be safe in a given set* $\mathcal{S} \subseteq \mathbb{R}^{n_\mathbf{x}}$ *if* $\forall \mathbf{x} \in \mathcal{S}$ *it generates control inputs* $\mathbb{U} = \{\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_\infty\}$ *and corresponding state trajectories* $\mathbb{X} = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_\infty\}$ *such that* $h_k(\mathbf{x}_k, \mathbf{u}_k) \leq 0$ *and* $g_k(\mathbf{x}_k, \mathbf{u}_k) \leq 0$, $\forall k \geq 0$.

The secondary objective is to control the system (2.1) such that it tracks a predefined reference trajectory $\mathbf{r}(\tau)$, as closely as *safety* allows. To that end, the underlying OCP one needs to solve can be written as

$$
\begin{aligned}
V(\mathbf{x}_k, \tau_k) := \min_{\substack{\mathbf{x}\ \mathbf{u} \\ \tau, v}} \ & \sum_{n=k}^{k+N-1} q_\mathbf{r}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) + w v_{n|k}^2 \\
& + p_\mathbf{r}(\mathbf{x}_{k+N|k}, \tau_{k+N|k}) && \text{(3.1a)} \\
\text{s.t. } & \mathbf{x}_{k|k} = \mathbf{x}_k, \ \tau_{k|k} = \tau_k, && \text{(3.1b)} \\
& \mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), \qquad n \in \mathbb{I}_k^{k+N-1}, && \text{(3.1c)} \\
& \tau_{n+1|k} = \tau_{n|k} + t_\mathrm{s} + v_{n|k}, \qquad n \in \mathbb{I}_k^{k+N-1}, && \text{(3.1d)} \\
& h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \qquad n \in \mathbb{I}_k^{k+N-1}, && \text{(3.1e)} \\
& g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \qquad n \in \mathbb{I}_k^{k+N-1}, && \text{(3.1f)} \\
& \mathbf{x}_{k+N|k} \in \mathcal{X}_\mathbf{r}^\mathrm{f}(\tau_{k+N|k}). && \text{(3.1g)}
\end{aligned}
$$

Note that the only difference between Problem (3.1) and (2.24) lies in the fact that Problem (3.1) also considers the a-priori *unknown* constraints $g_{n|k}$.

Stability and safety (recursive feasibility) guarantees can also be proven for this problem, however, it requires the introduction of the following (restrictive) assumptions

Assumption 3.1 (Reference feasibility)**.** *Under its natural time dynamics, the reference is feasible for the system dynamics, i.e.,* $\mathbf{r}^\mathbf{x}(t + t_\mathrm{s}) = f(\mathbf{r}^\mathbf{x}(t), \mathbf{r}^\mathbf{u}(t))$, *and:*

a) *the reference satisfies the known constraints* (3.1e)*, i.e.,* $h_n(\mathbf{r}^\mathbf{x}(t_n), \mathbf{r}^\mathbf{u}(t_n)) \leq 0$, *for all* $n \in \mathbb{I}_0^\infty$*;*

b) *the reference satisfies the unknown constraints* (3.1f)*, i.e.,* $g_{n|k}(\mathbf{r}^\mathbf{x}(t_n), \mathbf{r}^\mathbf{u}(t_n)) \leq 0$, *for all* $n, k \in \mathbb{I}_0^\infty$.

Assumption 3.1b is indeed a strong assumption since it assumes that the predefined reference trajectory $\mathbf{r}(\tau)$ should always be feasible with respect to the unknown constraints. Therefore, Assumption 3.1a, which coincides with Assumption 2.2, serves as a relaxed version that is more realistic and will be used later on to drop Assumption 3.1b.

Assumption 3.2 (Stabilizing Terminal Conditions)**.** *There exists a parametric stabilizing terminal set $\mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t)$ and a terminal control law $\kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)$ yielding:*

$$\mathbf{x}_+^{\kappa} = f(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)), \qquad\qquad t_+ = t + t_{\mathrm{s}},$$

such that $p_{\mathbf{r}}(\mathbf{x}_+^{\kappa}, t_+) - p_{\mathbf{r}}(\mathbf{x}, t) \leq -q_{\mathbf{r}}(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t), t)$, and

a) *$\mathbf{x} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t) \Rightarrow \mathbf{x}_+^{\kappa} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t_+)$, and $h_n(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)) \leq 0$, for all $n, k \in \mathbb{I}_0^{\infty}$;*

b) *$\mathbf{x} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t) \Rightarrow g_{n|k}(\mathbf{x}, \kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, t)) \leq 0$, for all $n, k \in \mathbb{I}_0^{\infty}$.*

Similarly to Assumption 3.1b, Assumption 3.2b is also difficult to verify due to the unknown constraints. Hence, the relaxed version Assumption 3.2a, which is standard in MPC settings (see, e.g., Assumption 2.3), will be used later on to drop Assumption 3.2b.

Finally, we introduce the following assumption, imposing some structure on $g_{n|k}$ which is needed in order to ensure that the feasibility of a solution does not become jeopardized between consecutive time instances.

Assumption 3.3 (Unknown constraint dynamics)**.** *The a-priori unknown constraint functions satisfy $g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$, for all $n \geq k$.*

This assumption essentially requires the availability of a consistent characterization of the a priori unknown constraints, as we will further detail in the next section.

Proposition 3.1. *Suppose that Assumptions 2.1, 3.1, 3.2, and 3.3 hold, and that the initial state $(\mathbf{x}_k, \tau_k)$ at time k belongs to the feasible set of Problem* (3.1)*. Then the system* (2.1) *and* (2.23) *in closed loop with the solution of* (3.1) *applied in receding horizon is an asymptotically stable system.*

Proof. The proof follows the same steps as in Theorem 2.3 since Assumptions 3.1b and 3.3 are assumed to hold for the reference trajectory $\mathbf{r}(\tau)$. For further details, the reader is referred to the proof in [36] (Paper D). □

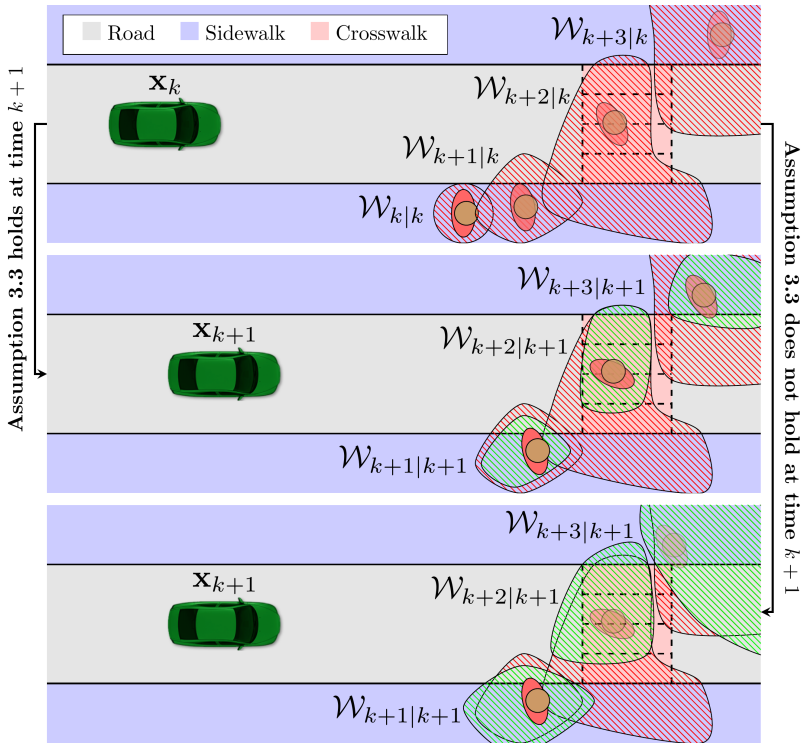


Figure 3.1: The top panel shows the sets $\mathcal{W}_{n|k}$ for $n \in \mathbb{I}_k^{k+3}$ representing the predicted positions of the pedestrian at time k, while the middle and bottom panel show predictions made at time $k+1$. A model that satisfies Assumption 3.3 is expected to provide sets that are captured by the middle panel, while a model that does not satisfy Assumption 3.3 can instead predict sets that are not contained at the next time instance.

Proposition 3.1 solves *in theory* the problem of applying MPC to settings in uncertain environments. However, we note that Assumptions 3.1b, 3.2b, and 3.3 are difficult to enforce in practice, since they require feasibility with respect to constraints that are unknown.

The next sections will therefore show how Assumptions 3.1b and 3.2b can be replaced with more realistic ones, and how Assumption 3.3 can be verified.

3.1 Predictive Collision Avoidance

This section aims to provide a discussion on how constraint $g_{n|k}$ can be constructed so that Assumption 3.3 is satisfied. While Figure 3.1 provides a visual interpretation of Assumption 3.3, it is natural to wonder how g should be constructed. In order to answer this question, we first provide a formal description of how g can be constructed, and then provide an example of how a pedestrian model can be derived and used together with a collision-avoiding MPC scheme.

We consider a function $\gamma(\mathbf{x}, \mathbf{u}, \mathbf{w}) : \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_\mathbf{u}} \times \mathbb{R}^{n_\mathbf{w}} \to \mathbb{R}^{n_g}$ together with the uncertain variable $\mathbf{w}_{n|k}$ which models the predicted states of, e.g., other road users. We assume that this variable belongs to a set $\mathbf{w}_{n|k} \in \mathcal{W}_{n|k} \subseteq \mathbb{R}^{n_\mathbf{w}}$ that gathers all uncertainties related to the a-priori unknown constraints as depicted in Figure 3.1. With the knowledge of such a set, it is possible to formulate the constraint $g_{n|k}$ in the following way

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}), \tag{3.2}$$

which by construction also implies robust constraint satisfaction, i.e.,

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0 \qquad \Leftrightarrow \qquad \begin{cases} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}) \leq 0, \\ \forall\ \mathbf{w}_{n|k} \in \mathcal{W}_{n|k}. \end{cases} \tag{3.3}$$

However, in order to construct $g_{n|k}$ in accordance to (3.2), we rely on the availability of sets $\mathcal{W}_{n|k}$ that express the uncertainty of an obstacle. In a general setting, we can select $\mathbf{w}_k$ to be a state of the dynamical system

$$\mathbf{w}_{k+1} = \omega(\mathbf{w}_k, \boldsymbol{\eta}_k, \mathbf{x}_k, \mathbf{u}_k), \tag{3.4}$$

where the function ω describes the dynamics, and $\boldsymbol{\eta}_{n|k} \in \mathcal{E} \subseteq \mathbb{R}^{n_\boldsymbol{\eta}}$ is an associated control input which can either model process noise or some underlying policy/intent that controls the state $\mathbf{w}_k$, while the inclusion of $\mathbf{x}_k$, and $\mathbf{u}_k$ allow the possibility to model interactions between the state $\mathbf{w}_k$ and system (2.1). For instance, a road user will most likely approach an intersection cautiously if a vehicle drives with a high velocity and shows no indication of slowing down.

The remaining question to answer is, however, how to derive a dynamical system like the one proposed in (3.4). In the next section we provide an example of how a pedestrian prediction model can be constructed and then show how it can be used in a collision-avoiding MPC scheme in Section 3.1.2.

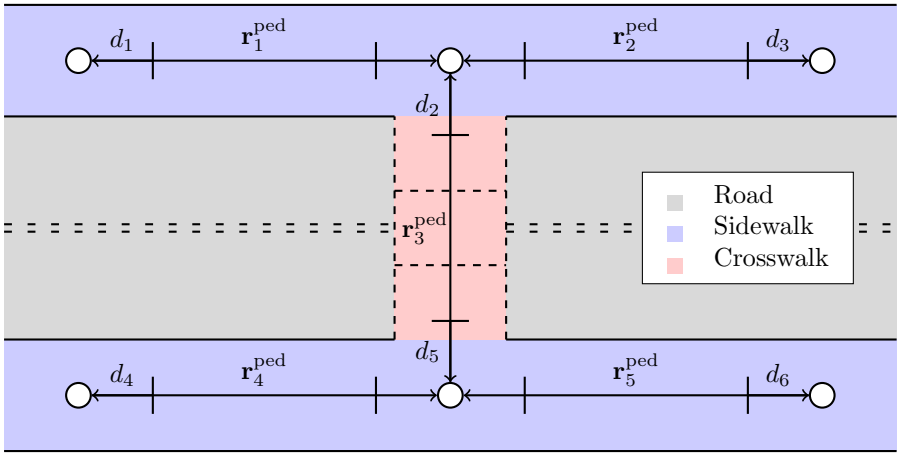


Figure 3.2: An illustration of a graph-based map which connects the different walkable segments together. Each edge i is associated with a walking reference $\mathbf{r}_i^{\text{ped}}$.

3.1.1 Pedestrian Prediction Model

A self-driving vehicle needs to be able to consistently detect the surrounding environment and update its belief about how the environment can change. To that end, modeling how road users behave in traffic, e.g., predicting their most likely behavior, is of great importance and needs to be computationally efficient, but also sufficiently accurate without being overly conservative. In this section, we present a pedestrian model [28] (Paper A), and discuss some of its benefits and limitations.

To make qualified predictions of the future movements of pedestrians, one can leverage the information that is provided by a map of the road configuration. In particular, we consider all areas where pedestrians are expected to walk, e.g., sidewalks and crosswalks, and construct a graph of connected edges which cover all walkable pedestrian areas. Then, by assigning a reference $\mathbf{r}_i^{\text{ped}} = (\mathbf{r}_i^{\mathbf{x},\text{ped}}, \mathbf{r}_i^{\mathbf{u},\text{ped}})$ for each edge i, a map like the one shown in Figure 3.2 can be obtained.

The pedestrian dynamics are modeled using the state and input vectors

$$\mathbf{w} = [x^{\text{ped}}, y^{\text{ped}}, v^{\text{ped}}, \theta^{\text{ped}}]^\top, \qquad \boldsymbol{\eta} = [a^{\text{ped}}, \alpha^{\text{ped}}]^\top, \tag{3.5}$$

where $(x^{\text{ped}}, y^{\text{ped}})$ denote the position coordinates, v^{ped} the walking velocity, and θ^{ped} the orientation. The inputs a^{ped} and α^{ped} model the pedestrian acceleration and angular velocity, respectively. With this state representation, the pedestrian dynamics can be modeled with the following unicycle equations

$$\dot{\mathbf{w}} = \begin{bmatrix} \dot{x}^{\text{ped}} \\ \dot{y}^{\text{ped}} \\ \dot{v}^{\text{ped}} \\ \dot{\theta}^{\text{ped}} \end{bmatrix} = \begin{bmatrix} v^{\text{ped}} \cos\theta^{\text{ped}} \\ v^{\text{ped}} \sin\theta^{\text{ped}} \\ a^{\text{ped}} \\ \alpha^{\text{ped}} \end{bmatrix} + \boldsymbol{\zeta} := \omega(\mathbf{w}, \boldsymbol{\eta}) + \boldsymbol{\zeta}, \tag{3.6}$$

where $\boldsymbol{\zeta} := [\zeta^x, \zeta^y, \zeta^v, \zeta^\theta]^\top$ represents some Gaussian process noise with zero mean. For simplicity, the pedestrian model is linearized and discretized along the reference edges $(\mathbf{r}_i^{\mathbf{x},\text{ped}}, \mathbf{r}_i^{\mathbf{u},\text{ped}})$ to obtain the linear discrete-time model

$$\Delta\mathbf{w}_{k+1} = A^{\text{ped}}\Delta\mathbf{w}_k + B^{\text{ped}}\Delta\boldsymbol{\eta} + E^{\text{ped}}\boldsymbol{\zeta}_k. \tag{3.7}$$

While (3.6) can in most cases model the pedestrian dynamics adequately, we note that the difficulty lies in figuring out the underlying intent that drives the dynamics, i.e., one must understand the structure of the control input $\boldsymbol{\eta}$. It is quickly realized that it becomes intractable to compute different intent policies for different pedestrians, since all pedestrians follow their own preferences and would most likely make different choices to reach the same goal. To that end, by using model (3.6) we aim to predict the average behavior of the pedestrian, and use some estimate on the covariance to address the uncertainty of the predicted motion.

To capture the average motion behavior, we apply the following structure on the control input

$$\boldsymbol{\eta} = -K^{\text{ped}}(\mathbf{w}_k - \mathbf{r}_i^{\mathbf{x},\text{ped}}), \tag{3.8}$$

where the feedback matrix K^{ped} can easily be implemented using an LQR formulation. Then, using the closed-loop model $A_K^{\text{ped}} = A^{\text{ped}} - B^{\text{ped}}K^{\text{ped}}$, one can easily compute the average error $\Delta\bar{\mathbf{w}}_k := \mathbb{E}[\Delta\mathbf{w}_k]$ and covariance $\Sigma := \mathbb{E}[\Delta\mathbf{w}_k\Delta\mathbf{w}_k^\top]$ as

$$\Delta\bar{\mathbf{w}}_{n+1|k} = A_K^{\text{ped}}\Delta\mathbf{w}_{n|k}, \qquad \Sigma_{n+1|k} = A_K\Sigma_{n|k}A_k^\top + Z, \tag{3.9}$$

where Z is the associated covariance matrix for the process noise $\boldsymbol{\zeta}$.

This structure allows for easy propagation of the uncertainty and average behavior of the pedestrian motion along a reference edge. Since the edges are

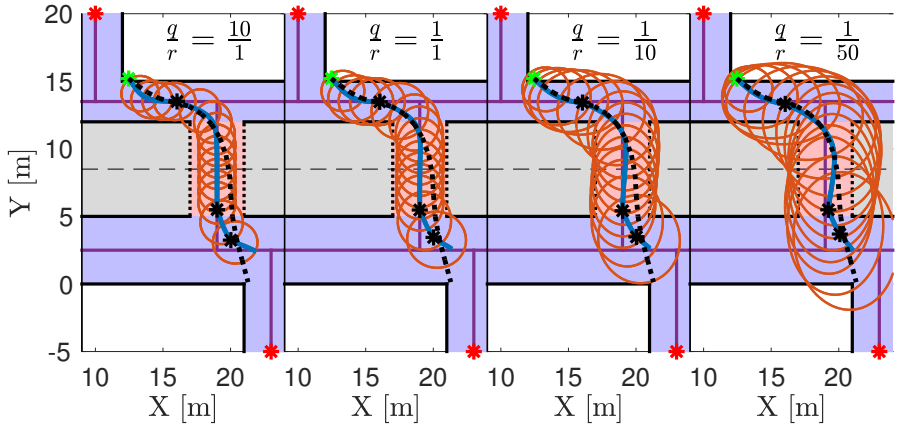


Figure 3.3: The predicted average positions are represented by the blue lines while the corresponding 99-percentile ellipses are displayed in red. The black dashed line represents the ground truth of actual pedestrian measurements.

connected in a graph, we allow for the prediction to switch to the next edge whenever it comes close to the neighboring node by using a simple distance-based rule, e.g., when the prediction reaches the distance threshold d_i for node i (see Figure 3.2). Furthermore, if more than one neighboring edge exists, i.e., there exists a bifurcation in the graph, the prediction can be propagated along all the neighboring edges, such that a multi-modal prediction is obtained.

A clarifying example is shown in Figure 3.3, where the blue lines show the average predictions, the red lines show the 99-percentile of the covariance matrix Σ projected onto the xy-plane, and the black lines show actual ground truth measurements of a pedestrian taken from [28]. The green asterisk denotes the start of the prediction, while the black asterisks denote the place where a switch in the reference edge occurred, i.e., the model switched from tracking edge i to tracking edge j. Since K^{ped} was designed using an LQR control with tuning $Q^{\text{ped}} = q \cdot \text{diag}(1,1,1,1)$ and $R^{\text{ped}} = r \cdot \text{diag}(1,1)$, it is evident that the ratio q/r affect the predicted covariance and average position. Figure 3.3 shows that the overall average behavior manages to capture the actual motion of the pedestrian well, and that the uncertainty can be tuned so that it always captures the underlying ground truth trajectory.

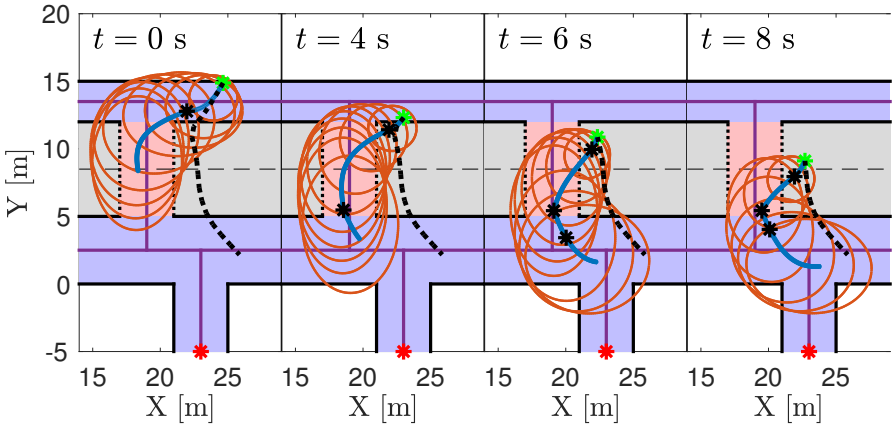


Figure 3.4: An example of prediction mismatch when a pedestrian does not fully adhere to the assumptions of the prediction model. The color scheme matches the one of Figure 3.3.

It is clear that this model has its limitations, since it is designed to predict average behaviors only. Consider the situation presented in Figure 3.4 where a pedestrian does not follow the underlying assumption. While the method is able to accurately predict on a higher level that the pedestrian can cross the road, the detailed prediction does not capture the true behavior well enough.

Next, we show how such a model can be incorporated in an MPC framework, and applied to a real test vehicle.

3.1.2 Combining Pedestrian Prediction Models with MPC

In this section we provide an overview of the framework presented in [50], with the main focus of illustrating how the constraint $g_{n|k}$ can be constructed based on pedestrian predictions obtained from the previous section. For full implementation details, the reader is referred to [50] (Paper B).

Consider the setting illustrated in Figure 3.5, where a self-driving vehicle needs to perform a left turn in an intersection with crosswalks. In this setting, the vehicle must be able to clear the intersection in a comfortable manner, while avoiding collisions with any potential road user. In order to achieve this, we formulate an MPC problem, similar to the formulations mentioned

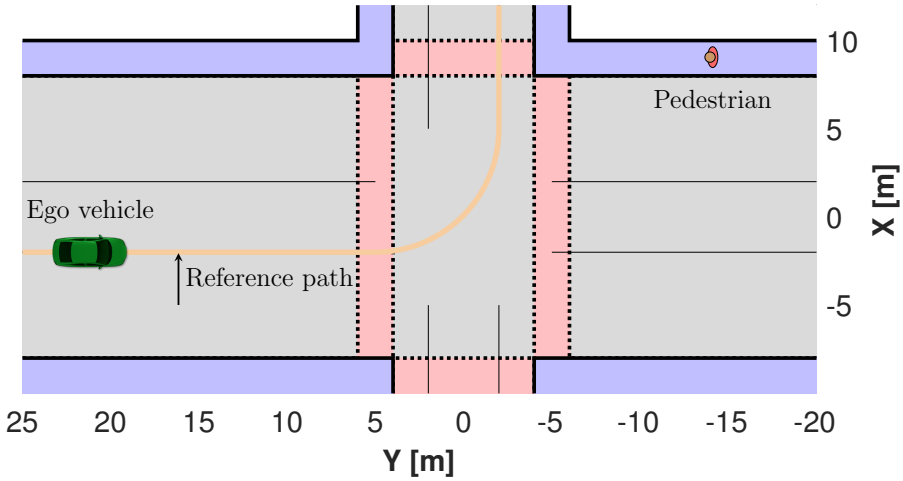


Figure 3.5: The vehicle needs to drive along its given path, while considering the future motion of the pedestrian.

in Chapter 2, which tracks a predefined reference trajectory (orange line in Figure 3.5), and satisfies actuation constraints $h_k(\mathbf{x}, \mathbf{u}) \leq 0$ and the a-priori unknown constraints $g_{n|k}(\mathbf{x}, \mathbf{u}) \leq 0$ coming from the predicted pedestrian positions. To construct $g_{n|k}$ we first use the information about the predicted average position $(x^{\text{ped}}_{n|k}, y^{\text{ped}}_{n|k})$ and the uncertainty $\Sigma_{n|k}$. Selecting a suitable confidence level σ of the covariance matrix $\Sigma_{n|k}$, we can formulate the following bounded region which, according to the prediction model, will most likely be occupied by the pedestrian

$$\mathcal{W}_{n|k} := \{\mathbf{w} \mid \begin{bmatrix} x^{\text{ped}} - x^{\text{ped}}_{n|k} \\ y^{\text{ped}} - y^{\text{ped}}_{n|k} \end{bmatrix}^\top \bar{\Sigma}^{-1}_{n|k} \begin{bmatrix} x^{\text{ped}} - x^{\text{ped}}_{n|k} \\ y^{\text{ped}} - y^{\text{ped}}_{n|k} \end{bmatrix} \leq \sigma^2\}, \tag{3.10}$$

where $\bar{\Sigma}_{n|k}$ is the uncertainty related to $(x^{\text{ped}}_{n|k}, y^{\text{ped}}_{n|k})$, i.e.,

$$\bar{\Sigma}_{n|k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \Sigma_{n|k} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^\top,$$

and $\mathbf{w} := [x^{\text{ped}}, y^{\text{ped}}, v^{\text{ped}}, \theta^{\text{ped}}]^\top$ denotes the pedestrian state from Section 3.1.1.

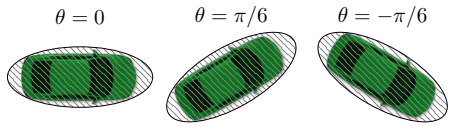


Figure 3.6: Approximation of the self-driving vehicle bounding box from (3.11).

From a collision-avoidance point of view, it becomes natural that the self-driving vehicle should avoid the regions that might be occupied by the pedestrian. More formally, the position component (x, y) of the self-driving vehicle should not intersect with any position component $(x^{\text{ped}}_{n|k}, y^{\text{ped}}_{n|k}) \in \mathcal{W}_{n|k}$. Since the dimensions of the vehicle must also be considered for collision avoidance, we use the following ellipse to over-approximate the self-driving vehicle bounding box

$$\begin{bmatrix} x - x^{\text{ego}} \\ y - y^{\text{ego}} \end{bmatrix}^\top R(\theta) \begin{bmatrix} a^{-2} & 0 \\ 0 & b^{-2} \end{bmatrix} R(\theta)^\top \begin{bmatrix} x - x^{\text{ego}} \\ y - y^{\text{ego}} \end{bmatrix} = 1, \tag{3.11}$$

where $R(\theta) : \mathbb{R} \to \mathbb{R}^2 \times \mathbb{R}^2$ denotes a two-dimensional rotation matrix, $(x^{\text{ego}}, y^{\text{ego}})$ is the vehicle center, and a and b denote the ellipse axes. See Figure 3.6 for a clarifying illustration. Using the ellipse (3.11) and the uncertainty set (3.10), we can form the collision avoidance function γ as

$$\begin{aligned} &\gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}) := \\ &\quad 1 - \begin{bmatrix} x_{n|k} - x^{\text{ped}}_{n|k} \\ y_{n|k} - y^{\text{ped}}_{n|k} \end{bmatrix}^\top R(\theta)^\top \begin{bmatrix} a^{-2} & 0 \\ 0 & b^{-2} \end{bmatrix} R(\theta) \begin{bmatrix} x_{n|k} - x^{\text{ped}}_{n|k} \\ y_{n|k} - y^{\text{ped}}_{n|k} \end{bmatrix}, \end{aligned} \tag{3.12}$$

which allows us to formulate $g_{n|k}$ according to (3.2), i.e.,

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}). \tag{3.13}$$

Results and Discussion

An MPC controller which uses the predictions generated from Section 3.1.1 to form $g_{n|k}$ was implemented and validated at the Astazero[1] test track

[1] `https://astazero.com`

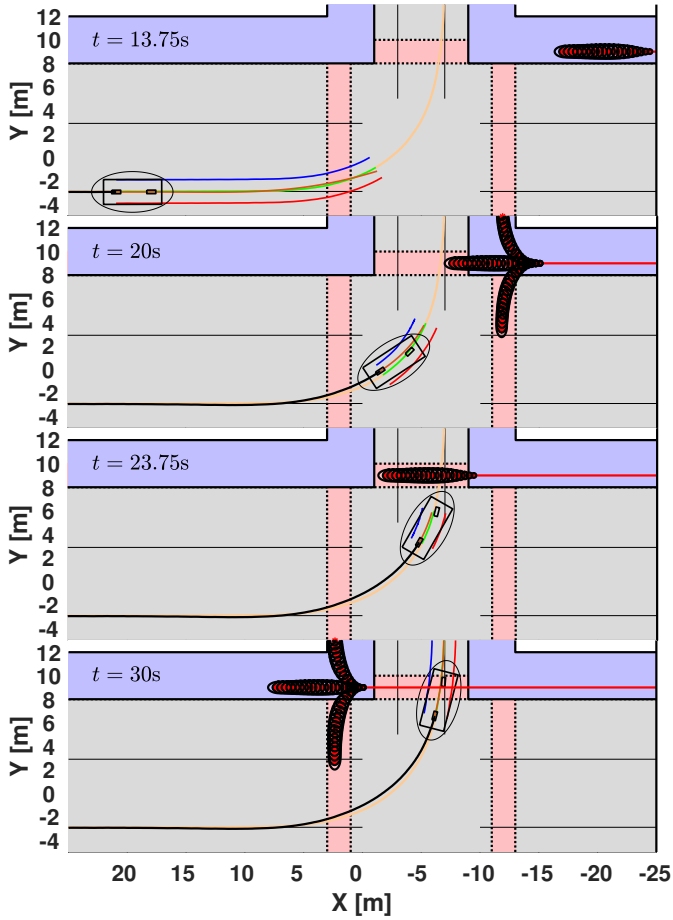


Figure 3.7: The blue and red lines denote the road boundary constraints, while the black ellipse surrounding the vehicle box denotes the ellipsoidal constraint (3.11). Projections of the reference $\mathbf{r}_k^{\mathbf{x}}$ and predicted open-loop state $\mathbf{x}_{n|k}$ are illustrated by the green and black lines, respectively, while the predicted pedestrian states are depicted as red points, and the uncertainties as black ellipses.

outside Gothenburg, Sweden. The proposed framework used ACADO [87] and HPMPC [64] to formulate and solve the MPC problem using the SQP

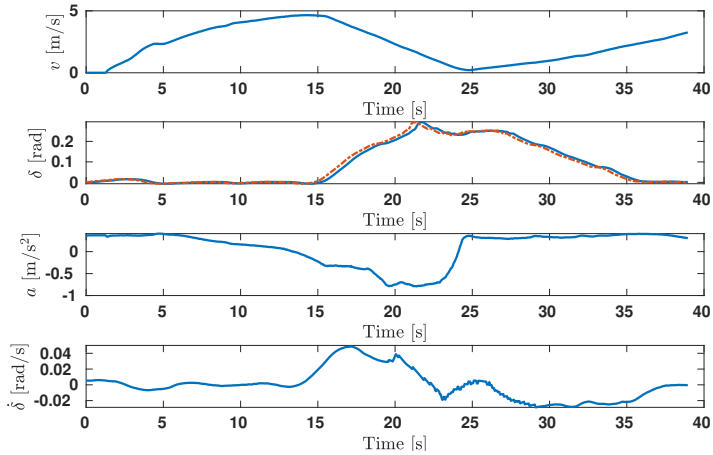


Figure 3.8: Closed-loop states from Figure 3.7. The first plot shows the velocity profile and the second plot shows the steering angle (blue line), and the steering angle setpoint (dashed brown line). The last two plots show the acceleration request a and steering angle rate $\dot{\delta}$.

Real-Time Iteration (RTI) [88] scheme, and was applied to a real Volvo XC90 T6 petrol-turbo SUV. More details regarding the vehicle platform will be provided in Chapter 4.

Figure 3.7 shows the open-loop predictions made by the MPC controller and by the pedestrian predictor across four different time instances. Here, it is visible how the vehicle approaches the intersection by slowing down and letting the predicted pedestrian pass, while keeping a distance between itself (the ellipse surrounding the vehicle bounding box) and the predicted pedestrian positions (black ellipses centered around the red points). Figure 3.8 shows the closed-loop evolution for some vehicle states. Here it is clear how the vehicle starts from zero velocity, accelerates up to the intersection, and almost comes to a complete stop while waiting for the pedestrian to pass. Only after the pedestrian is predicted to have cleared the intersection, the vehicle is allowed to accelerate again.

Evidently, it is possible to formulate a controller based on MPC that avoids collisions with moving obstacles by predicting their future motion. However, we note that this framework does not necessarily ensure recursive feasibility

since the prediction model based on Section 3.1.1 does not necessarily satisfy Assumption 3.3, and obstacles may "suddenly appear" due to limitations in the sensor system, e.g., a pedestrian or a cyclist may appear right behind a corner. To that end, it is of great importance to have a consistent characterization on $g_{n|k}$, in order to avoid any "surprises" from the environment.

In the next section we discuss how Assumption 3.3 can be satisfied by properly constructing the uncertainty sets $\mathcal{W}_{n|k}$.

3.2 Consistent Characterization

In order to construct $g_{n|k}$ robustly, one needs to propagate the uncertainty state $\mathbf{w}_{n|k}$. In other words, if one has access to the model (3.4) it is possible to resort to reachability analysis tools to generate possible outer approximations of the set $\mathcal{W}_{n|k}$ as

$$\begin{aligned}\mathcal{W}_{n+1|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) :\supseteq \{\, \omega(\mathbf{w}_{n|k}, \boldsymbol{\eta}_n, \mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \,| \\ \mathbf{w}_{n|k} \in \mathcal{W}_{n|k},\ \forall\, \boldsymbol{\zeta}_n \in \mathcal{E} \,\}\end{aligned}, \tag{3.14}$$

for some initial $\mathcal{W}_{k|k} = \mathbf{w}_{k|k}$. As it was shown in Section 3.1.1, the propagation of the uncertainty sets $\mathcal{W}_{n|k}$ allows one to model road users that (a) are detectable by the sensors; and (b) are either beyond the onboard sensor range or hidden by other obstacles as illustrated in Figure 3.9. For the case (a), we must assume that the prediction model does not underestimate the set of future states that can be reached by the road users, while for type (b), the uncertainty model must predict that a road user might appear behind the boundary of the sensor range at all times.

We now state the following result related to the structure of the constraint $g_{n|k}$, and Assumption 3.3.

Lemma 3.1. *Suppose that $g_{n|k}$ is defined according to* (3.2) *with $\mathcal{W}_{n|k}$ satisfying* (3.14)*. Then, Assumption 3.3 holds.*

Lemma 3.1 essentially assumes that the uncertainty in $g_{n|k}$ cannot increase as additional information becomes available. In addition, a direct consequence of Lemma 3.1 is that if $g_{n|k}(\mathbf{r}^{\mathbf{x}}(t_n), \mathbf{r}^{\mathbf{u}}(t_n)) \leq 0 \Rightarrow g_{n|k+1}(\mathbf{r}^{\mathbf{x}}(t_n), \mathbf{r}^{\mathbf{u}}(t_n)) \leq 0$, which partly satisfies Assumption 3.1b. Since Lemma 3.1 and Assumption 3.3 play a central part in the deriving the safety guarantees, we provide next clarifying examples to provide more intuition around them.

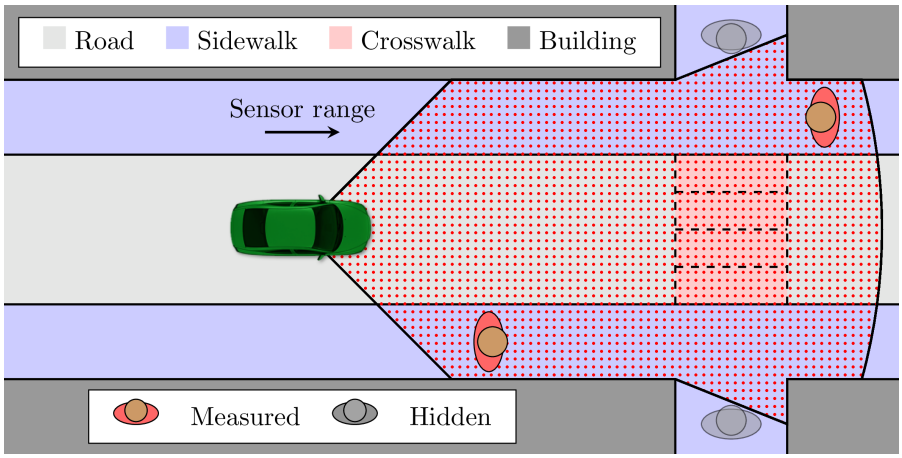


Figure 3.9: A sensor system can only measure objects that are directly visible within the sensing range.

Figure 3.1 illustrates the difference in the propagated sets $\mathcal{W}_{n|k}$ when Lemma 3.1 is satisfied, versus when it is not. The middle panel shows that predictions made at time $k+1$ are a subset of the predictions made at time k, while the bottom panel on the other hand does not have the same consistent characterization and does not satisfy Lemma 3.1. We note that by ensuring that the uncertainty sets have the property $\mathcal{W}_{n|k+1} \subseteq \mathcal{W}_{n|k}$ one can conclude that if $(\mathbf{x}^\star_{n|k}, \mathbf{u}^\star_{n|k})$ satisfy $g_{n|k}(\mathbf{x}^\star_{n|k}, \mathbf{u}^\star_{n|k}) \leq 0$ at time k, then $g_{n|k+1}(\mathbf{x}^\star_{n|k}, \mathbf{u}^\star_{n|k}) \leq 0$ also holds at time $k+1$. In other words, the solution obtained at time k is also ensured to hold at time $k+1$.

Figure 3.9 illustrates a setting where all road users cannot be detected due to occlusions caused from the environment. Hence, in order to satisfy Assumption 3.3 at all times, it is a necessity to model the possibility that a road user might appear at the boundary of the sensor range. In combination with a limited sensing range, that also make it impossible to detect obstacles which are too far away, one can adopt a worst-case approach which ensures that the planned trajectory $\mathbf{x}_{n|k}$ never leaves the sensor range. A visual example of this is shown in Figure 3.10.

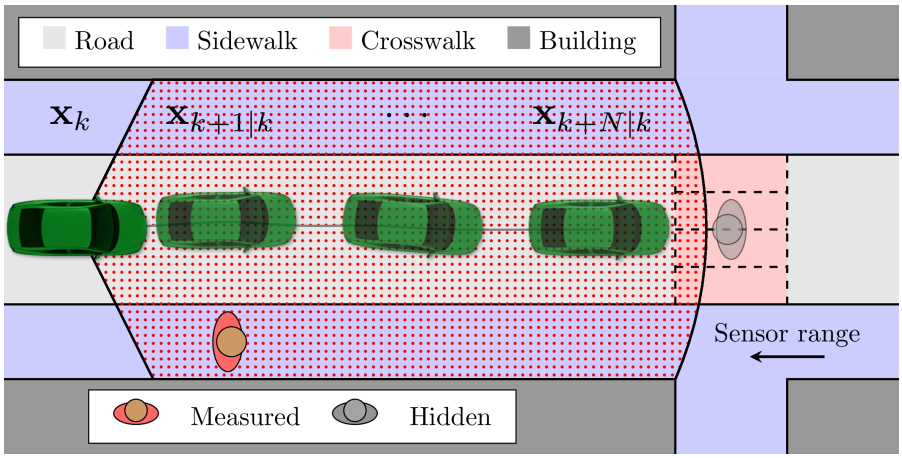


Figure 3.10: The planned trajectory $\{\mathbf{x}_{k|k}, \mathbf{x}_{k+1|k}, ..., \mathbf{x}_{k+N|k}\}$ is forced to remain within the sensor range due to limited sensing capabilities.

3.3 Multi-Modal Uncertainties

In practical settings the predicted set $\mathcal{W}_{n|k}$ can represent a collection of disjoint sets, i.e., $\mathcal{W}_{n|k} = \mathcal{O}^i_{n|k} \bigcap \mathcal{O}^j_{n|k}$, that model some underlying intention. Consider, e.g., the situation presented in Figure 3.11 where a pedestrian is approaching an intersection. In this situation it becomes natural for the prediction model to consider two possibilities: either the pedestrian crosses the road, or it does not; based on each choice, the resulting predicted behavior changes drastically.

Assuming that the number of modes are finite, i.e., $i \in \mathbb{I}_1^L$, and that the probability β_i for each mode could be estimated, it would be natural to consider this in the construction of $g_{n|k}$, but also in the MPC Problem (3.1). For instance, a human-driver in the situation presented in Figure 3.11 would most likely drive with a higher speed if it was believed that mode i had a low probability, compared to if mode i had a high probability. In that sense, using a controller based on (3.1), with constraints $g_{n|k}$ modeled as (3.13), can yield conservative behaviors since this amounts to a robust tube formulation, i.e., all uncertainties in $\mathcal{W}_{n|k}$ are considered at the same time.

An alternative to such a robust tube formulation, is to instead consider a formulation where the feedback policy is a function of the predicted obstacle

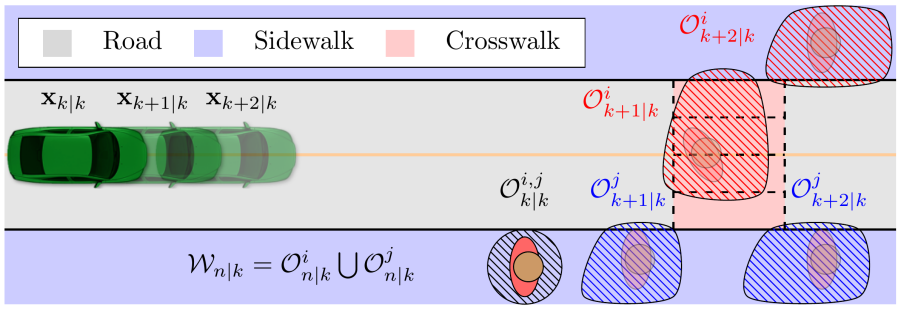


Figure 3.11: Example of a setting where the predicted set $\mathcal{W}_{n|k}$ can be split up into two distinct prediction modes $\mathcal{O}^i_{n|k}$ and $\mathcal{O}^j_{n|k}$.

mode i, which is rather similar to the principles presented in [89]. This would allow the controller to pick different control actions $\mathbf{u}^i_{n|k}$ for each mode i if it can be distinguished from the other modes, i.e., $\mathbf{u}^i_{n|k} = \mathbf{u}^j_{n|k}$ for all $n < \bar{n}_{ij}$ where

$$\bar{n}_{ij} := \min_n n, \text{ s.t. } \mathcal{O}^i_{n|k} \bigcap \mathcal{O}^j_{n|k} = \emptyset, \ \forall\ n \geq k. \tag{3.15}$$

The resulting MPC controller would then be able to be formulated as

$$\begin{aligned}
V(\mathbf{x}_k, \tau_k) := \min_{\substack{\mathbf{x}^i, \mathbf{u}^i, \tau^i, v^i \\ i \in \mathbb{I}_1^L}} & \sum_{i=1}^{L} \beta_i \Big(\sum_{n=k}^{k+N-1} q_{\mathbf{r}}(\mathbf{x}^i_{n|k}, \mathbf{u}^i_{n|k}, \tau^i_{n|k}) + w(v^i_{n|k})^2 \\
& + p_{\mathbf{r}}(\mathbf{x}^i_{k+N|k}, \tau^i_{k+N|k}) \Big) && \text{(3.16a)} \\
\text{s.t. } & \mathbf{x}^i_{k|k} = \mathbf{x}_k, \ \tau^i_{k|k} = \tau_k, && i \in \mathbb{I}_1^L, && \text{(3.16b)} \\
& \mathbf{x}^i_{n+1|k} = f(\mathbf{x}^i_{n|k}, \mathbf{u}^i_{n|k}), && i \in \mathbb{I}_1^L, n \in \mathbb{I}_k^{k+N-1}, && \text{(3.16c)} \\
& \tau^i_{n+1|k} = \tau^i_{n|k} + t_{\mathrm{s}} + v^i_{n|k}, && i \in \mathbb{I}_1^L, n \in \mathbb{I}_k^{k+N-1}, && \text{(3.16d)} \\
& h_n(\mathbf{x}^i_{n|k}, \mathbf{u}^i_{n|k}) \leq 0, && i \in \mathbb{I}_1^L, n \in \mathbb{I}_k^{k+N-1}, && \text{(3.16e)} \\
& g^i_{n|k}(\mathbf{x}^i_{n|k}, \mathbf{u}^i_{n|k}) \leq 0, && i \in \mathbb{I}_1^L, n \in \mathbb{I}_k^{k+N-1}, && \text{(3.16f)} \\
& \mathbf{u}^i_{n|k} = \mathbf{u}^j_{n|k}, && n < \bar{n}_{ij},\ i, j \in \mathbb{I}_1^L, && \text{(3.16g)} \\
& \mathbf{x}^i_{k+N|k} \in \mathcal{X}^{\mathrm{f}}_{\mathbf{r}}(\tau^i_{k+N|k}), && i \in \mathbb{I}_1^L, && \text{(3.16h)}
\end{aligned}$$

where the constraint $g^i_{n|k}$ is now instead defined w.r.t. to each mode i, i.e.,

$$g^i_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := \max_{\mathbf{w}_{n|k} \in \mathcal{O}^i_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}). \tag{3.17}$$

Note that formulation (3.16) does not optimize over L separate trajectories for L different scenarios. On the contrary, it groups L scenarios together in the same problem, where the control input ties together the scenarios and trajectories through constraint (3.16g). Furthermore, since β_i denotes the probability for each mode i, i.e., $\sum_{i=1}^{L} \beta_i = 1$, the cost function (3.16a) is an approximation of the expected cost. For a more detailed discussion on Problem (3.16) the reader is referred to [49] (Paper C).

3.3.1 Results

The difference in performance for a controller based on (3.1) and (3.16) can be evaluated for the scenario illustrated in Figure 3.11. By assuming a simple pedestrian walking model that upon reaching the intersection can cross with probability γ or keep on walking with probability $\gamma - 1$, one can obtain through simulations the expected cost (average performance) for the two controllers. Figure 3.12 shows the expected cost for this simple setting, where a prescient MPC, i.e., an MPC controller which knows all uncertain constraints a-priori, is included to show the lower bound on the expected cost. Here it is visible that the scenario-based MPC (3.16) on average performs better than a robust MPC based on (3.1). Clearly, the idea to leverage the probability information for each mode i pays-off on average, especially for situations where the crossing probability is low. However, while formulation (3.16) seems to outperform (3.1), we note that it comes with a computational complexity that increases with the number of total modes L.

3.4 Safety Guarantees

So far, we have shown how $g_{n|k}$ can be constructed so that Assumption 3.3 holds for practical settings. However, ensuring that *safety* (according to Definition 3.1) holds for system (2.1) requires the controller based on the MPC problem (3.1) to be *recursively feasible*. This places requirements on the terminal constraint (3.1g), which needs to be designed in a way that guarantees

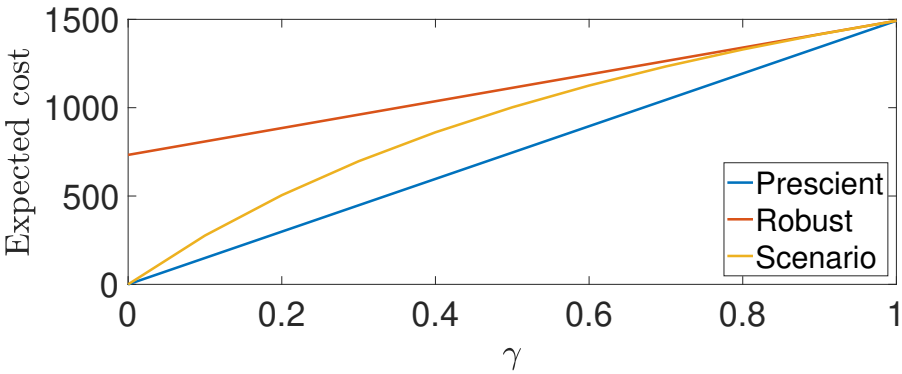


Figure 3.12: Expected cost for a prescient MPC, robust tube MPC, and scenario MPC (3.16) for different crossing probabilities γ.

satisfaction of constraints h_n and $g_{n|k}$ for all future times. In other words, the terminal region must ensure that the vehicle will never violate any constraints. The problem with this is that constraints (3.1f), i.e., $g_{n|k}$ can grow unbounded in time, such that no state after a certain amount of time remains safe, see e.g., Figure 3.13. To solve this issue, one alternative is to design the terminal set (3.1g) by assuming that there exists a *safe* set where the constraints h_n and $g_{n|k}$ are guaranteed to be satisfied *regardless*. With this approach, one can rely on the standard ideas found in MPC [61], [90], [91] which are based on the existence of robust invariant sets.

More formally, we assume that the *safe* set has the following properties.

Assumption 3.4 (Safe set)**.** *There exists a robust invariant set denoted* $\mathcal{X}_{\text{safe}}(\tau_{n|k}) \subseteq \mathbb{R}^{n_x}$ *such that for all* $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$ *there exists a safe control set* $\mathcal{U}_{\text{safe}}(\mathbf{x}_{n|k}, \tau_{n|k}) \subseteq \mathbb{R}^{n_u+1}$ *entailing that* $f(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}) \in \mathcal{X}_{\text{safe}}(\tau_{n|k} + t_{\text{s}} + v_{\text{safe}})$*, and* $h_n(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}) \leq 0$*, for all* $(\mathbf{u}_{\text{safe}},\ v_{\text{safe}}) \in \mathcal{U}_{\text{safe}}(\mathbf{x}_{n|k}, \tau_{n|k})$ *and for all* $n \geq k$*. Moreover, for all* $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$ *the a-priori unknown constraints can never be violated, i.e., by construction* $g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}) \leq 0$ *for all* $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$ *and* $(\mathbf{u}_{\text{safe}},\ v_{\text{safe}}) \in \mathcal{U}_{\text{safe}}(\mathbf{x}_{n|k}, \tau_{n|k})$*.*

At first glance, this assumption might seem strong, however, it only postulates the existence of *known* safe configurations for system (2.1). In particular, we stress that if no such configurations exist, then a controller based on the MPC Problem (3.1) is intrinsically unsafe. On the contrary, if such a set does

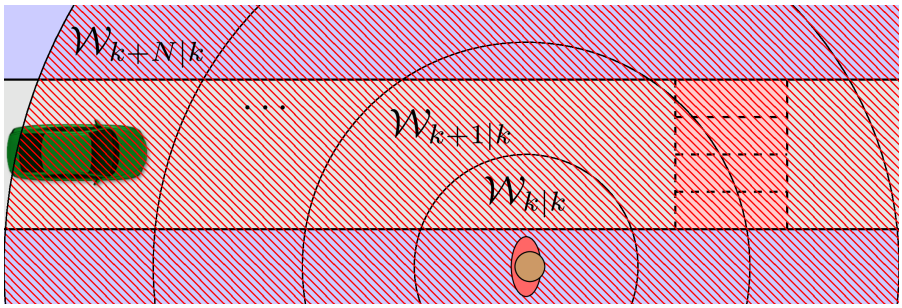


Figure 3.13: Given enough time, prediction models can predict that a pedestrian can be anywhere, such that no state is safe.

exist, then it is non-empty and must, by construction, be invariant. We provide next an example of a safe set for practical settings.

Example 3.1. *Many practical settings where safety is emphasized consider a system to be safe at steady-state, in which case the safety set $\mathcal{X}_{\text{safe}}$ can be formulated as*

$$\mathcal{X}_{\text{safe}}(\tau_k) := \{\, \mathbf{x} \,|\, \mathbf{x} = f(\mathbf{x}, \mathbf{u}),\, h_k(\mathbf{x}, \mathbf{u}) \leq 0,\, m_k(\mathbf{x}, \mathbf{u}) \leq 0 \,\}, \tag{3.18}$$

where function m_k defines additional constraints which might be needed in the set definition. A notable example for automotive settings is that a vehicle parked in a safe configuration, e.g., a parking lot, emergency lane or any other safe environment that can be modeled by m_k, is not responsible for collisions with other road users. This reasoning can be applied to the setting illustrated in Fig. 3.13, where the uncertainty grows such that the only safe thing to do is to force the vehicle to a complete stop.

A safe set based on Assumption 3.4 makes it possible to drop the restrictive Assumption 3.2b, and enables us to build an approach based on standard strategies in MPC [61], [90], [91], which relies on stabilizing terminal control laws $\kappa_{\mathbf{r}}^{\text{s}}(\mathbf{x}, t)$ and sets $\mathcal{X}_{\mathbf{r}}^{\text{s}}(t)$ that satisfy Assumption 3.2a. With this in mind,

the following terminal set is proposed

$$\begin{aligned}
\mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(\tau_{k+N|k}) := \{\mathbf{x}_{k+N|k} \mid \exists\ \mathbf{u}_{n|k}, v_{n|k}, & && (3.19a)\\
\tau_{n+1|k} = \tau_{n|k} + t_{\mathrm{s}} + v_{n|k}, & && (3.19b)\\
\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), & && (3.19c)\\
h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, & && (3.19d)\\
g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, & && (3.19e)\\
\mathbf{x}_{n|k} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{s}}(\tau_{n|k}), & && (3.19f)\\
\mathbf{x}_{k+M|k} \in \mathcal{X}_{\mathrm{safe}}(\tau_{k+M|k}) \subseteq \mathcal{X}_{\mathbf{r}}^{\mathrm{s}}(\tau_{k+M|k}), & && (3.19g)\\
(3.19\mathrm{a}) - (3.19\mathrm{f}),\ \forall n \in \mathbb{I}_{k+N}^{k+M-1}\}, & && (3.19h)
\end{aligned}$$

where $M \geq N$ serves as a degree of freedom, and by construction implies that $\mathcal{X}_{\mathrm{safe}}(\tau) \subseteq \mathcal{X}_{\mathbf{r}}^{\mathrm{s}}(\tau), \forall \tau > 0$. We note that if (3.19) is non-empty, then there exist control inputs $\mathbf{u}_k$ and v_k for all $\mathbf{x}_k \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(\tau_k)$ that steer $\mathbf{x}_k$ to the safe set.

The control law associated to set (3.19) can in practice be designed by first designing a stabilizing control law $\kappa_{\mathbf{r}}^{\mathrm{s}}$ as one would do in standard MPC formulations. This is done by, i.e., ignoring the a-priori unknown constraints $g_{n|k}$ and by forcing the evolution of τ to evolve according to its natural dynamics. Then, using the standard stabilizing control law $\kappa_{\mathbf{r}}^{\mathrm{s}}$, one can define the terminal control law $(\kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}_{k+N|k}, \tau_{k+N|k}), \nu_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}_{k+N|k}, \tau_{k+N|k}))$ for (3.19) as the solution of

$$\begin{aligned}
\min_{\mathbf{u},\nu}\ & \|\mathbf{u} - \kappa_{\mathrm{r}}^{\mathrm{s}}(\mathbf{x}_{k+N|k}, \tau_{k+N|k})\|_2 + \nu^2 && (3.20a)\\
\text{s.t.}\ & f(\mathbf{x}_{k+N|k}, \mathbf{u}) \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(\tau_{k+N|k} + t_{\mathrm{s}} + \nu), && (3.20b)\\
& h_{k+N}(\mathbf{x}_{k+N|k}, \mathbf{u}) \leq 0, && (3.20c)\\
& g_{k+N|k}(\mathbf{x}_{k+N|k}, \mathbf{u}) \leq 0. && (3.20d)
\end{aligned}$$

Terminal set (3.19) is essentially a backwards reachable set from $\mathcal{X}_{\mathrm{safe}}(t)$, since it ensures that for any $\mathbf{x} \in \mathcal{X}_{\mathbf{r}}^{\mathrm{f}}(t)$, the state $\mathbf{x}$ can reach the safe set $\mathcal{X}_{\mathrm{safe}}$ in the finite amount of time $M - N$, while remaining inside the standard stabilizing set $\mathcal{X}_{\mathbf{r}}^{\mathrm{f}}$ and satisfying constraints h_n and $g_{n|k}$. In that sense M is a parameter that can be used to tune $\mathcal{X}_{\mathbf{r}}^{\mathrm{f}}$. In particular, if $M = N$ the terminal set coincides with the safe set, and possibly limits the capabilities of the terminal control law, i.e., $\kappa_{\mathbf{r}}^{\mathrm{f}}(\mathbf{x}, \tau) \neq \kappa_{\mathbf{r}}^{\mathrm{s}}$ and $v_{\mathbf{r}}^{\mathrm{f}} \neq 0$. Consequently, if $M >> N$, then the computational complexity of $\mathcal{X}_{\mathbf{r}}^{\mathrm{f}}$ can become excessive.

We are now ready to state the main result of this section.

Theorem 3.1 (Recursive Feasibility)**.** *Suppose that Assumptions 2.1, 3.1a, 3.2a, 3.3, and 3.4 hold, and that Problem* (2.24) *is feasible for the initial state* $(\mathbf{x}_k, \tau_k)$*, with terminal set and terminal controllers given by* (3.19) *and* (3.20)*, respectively. Then, system* (2.1),(2.23) *in closed loop with the solution of* (3.1) *applied in receding horizon is safe (recursively feasible) at all times.*

Proof. The full proof is presented in [36] (Paper D). □

We note that the presence of obstacles make it difficult to discuss closed-loop stability, as the reference trajectory no longer is assumed to always be feasible w.r.t. $g_{n|k}$. However, whenever the a-priori unknown constraints become inactive, the proposed formulation yields nominal asymptotic stability in the sense of Theorem 2.3.

The next chapter uses Theorem 3.1 to implement and verify a safe MPC controller based on the results presented in this chapter.

CHAPTER 4

Experimental Validation of Safe MPC

Chapter 2 laid out an overview of the MPC theory, while Chapter 3 proposed ways of implementing it *safely* in autonomous driving settings. This chapter will present a practical implementation of an MPC formulation that satisfies Theorem 3.1 for a real vehicle. In particular, we will show in practice how a safe MPC framework can be deployed in a real vehicle approaching an intersection with moving pedestrians.

In order to present the experimental results in this chapter, we must first introduce the following preliminaries: the considered vehicle model, how consistent constraints can be generated for an AD setting, the computation of the terminal set in practice, and the vehicle platform.

4.1 Vehicle Model

We consider a single-track vehicle kinematics model, which well describes the vehicle motion of typical speeds in urban driving scenarios (i.e., less than

60 kph). We use the following model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\delta} \\ \dot{\alpha} \\ \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ vL^{-1}\tan\delta \\ \alpha \\ w_0^2(\delta^{\mathrm{sp}} - \delta) - 2w_0w_1\alpha \\ a \\ t_{\mathrm{d}}(a^{\mathrm{req}} - a) \end{bmatrix}, \tag{4.1}$$

where x, y denote the position coordinates in a global frame, v is the velocity, ψ is the orientation angle, L is the vehicle wheelbase length, δ is the steering angle, α is the steering rate, and a is the acceleration. The control inputs to the system are defined as δ^{sp} and a^{req} which denote a steering angle setpoint and an acceleration request. As mentioned in the previous chapters, the objective of the self-driving vehicle is to track a reference trajectory as safely as possible. Hence, assuming a predefined reference trajectory $\mathbf{r}(\tau)$, it is possible to derive the following vehicle kinematics in the frame of the reference path [54]

$$\begin{bmatrix} \dot{s} \\ \dot{e}_y \\ \dot{e}_\psi \\ \dot{\delta} \\ \dot{\alpha} \\ \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} v\cos(e_\psi)(1 - \kappa^{\mathbf{r}}(s)e_y)^{-1} \\ v\sin(e_\psi) \\ vL^{-1}\tan(\delta) - \dot{s}L^{-1}\tan(\delta^{\mathbf{r}}(s)) \\ \alpha \\ w_0^2(\delta^{\mathrm{sp}} - \delta) - 2w_0w_1\alpha \\ a \\ t_{\mathrm{d}}(a^{\mathrm{req}} - a) \end{bmatrix}, \tag{4.2}$$

$$\mathbf{x} = \begin{bmatrix} e_y & e_\psi & \delta & \alpha & v & a \end{bmatrix}^\top, \ \mathbf{u} = \begin{bmatrix} a^{\mathrm{req}} & \delta^{\mathrm{sp}} \end{bmatrix}^\top.$$

In this setting, s represents the longitudinal position along the path, $\kappa^{\mathbf{r}}$ is the path curvature, e_y and e_ψ are the lateral and yaw errors w.r.t. the reference $\mathbf{r}$, and $\delta^{\mathbf{r}}$ is the reference steering angle. The parameter values for (4.2) are: $L = 2.9$, $w_0 = 20$, $w_1 = 0.9$, and $t_{\mathrm{d}} = 1.8$. Since the dynamics depend on the reference curvature $\kappa^{\mathbf{r}}$ and steering reference $\delta^{\mathbf{r}}$, which in turn depend on the position s, we parameterize the reference trajectory in the longitudinal position, i.e., $\mathbf{r}(s) = (\mathbf{r}^{\mathbf{x}}(s), \mathbf{r}^{\mathbf{u}}(s))$. Hence, in this setting, we consider s to be an auxiliary state and only track the velocity instead of the longitudinal position (as opposed to τ in the MPFTC framework).

We consider that the system (4.1) is subject to the following a-priori known state constraints

$$\|e_y\| \leq 0.4 \text{ m}, \ \|e_\psi\| \leq 0.61 \text{ rad}, \ \|\delta\| \leq 0.53 \text{ rad}, \tag{4.3}$$

$$0 \leq v \leq 55/3.6 \text{ m/s}, \ -5 \leq a \leq 2 \text{ m/s}^2, \ \|\alpha\| \leq 0.35 \text{ rad/s}, \tag{4.4}$$

while the constraints $g_{n|k}$ must be formed by, e.g., considering the future predicted states of a road user. Since the pedestrian model in Section 3.1.1 does not necessarily satisfy Assumption 3.3, we propose a different model by slightly modifying the one in Section 3.1.1.

4.2 Consistent Pedestrian Prediction Model

The work in [28] (Paper A) showed that by using a map of the road configuration, e.g., Figure 3.2, it is possible to capture the average walking behavior at intersections. To that end, we propose to model the pedestrian dynamics along each edge i of the graph network. However, instead of considering the unicycle kinematics, we consider a simpler two-dimensional single integrator model that is expressed in the longitudinal and lateral directions of the reference edge. More formally, we assume that the pedestrian states $\mathbf{w}_k := [w_k^{\text{lon}}, \ w_k^{\text{lat}}]^\top \in \mathbb{R}^2$ model the longitudinal and lateral position along a reference edge $\mathbf{r}^{\text{ped}}$, and that $\boldsymbol{\eta}$ is the associated control input that relates to the walking speed in the longitudinal and lateral direction. Similarly to Section 3.1.1, one can then apply a feedback law

$$\boldsymbol{\eta}_k = [\mathbf{r}^{\text{ped},v} \ \ - K^{\text{ped}} w_k^{\text{lat}}]^\top, \tag{4.5}$$

where $\mathbf{r}^{\text{ped},v}$ denotes the longitudinal walking speed reference along an edge. Using the feedback law above, we can then write the closed-loop model as

$$\mathbf{w}_{k+1} = \omega(\mathbf{w}_k, \boldsymbol{\zeta}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 - t_{\text{s}} K^{\text{ped}} \end{bmatrix} \mathbf{w}_k + \begin{bmatrix} t_{\text{s}} \\ 0 \end{bmatrix} \mathbf{r}^{\text{ped},v} + \begin{bmatrix} t_{\text{s}} & 0 \\ 0 & t_{\text{s}} \end{bmatrix} \boldsymbol{\zeta}, \tag{4.6}$$

where $\boldsymbol{\zeta} \in \mathcal{Z} \subseteq \mathbb{R}^2$ is assumed to be some bounded noise. Similarly to Section 3.1.1, the nominal predicted motion can be propagated using

$$\mathbf{w}_{n+1|k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 - t_{\text{s}} K^{\text{ped}} \end{bmatrix} \mathbf{w}_{n|k} + \begin{bmatrix} t_{\text{s}} \\ 0 \end{bmatrix} \mathbf{r}^{\text{ped},v}, \tag{4.7}$$

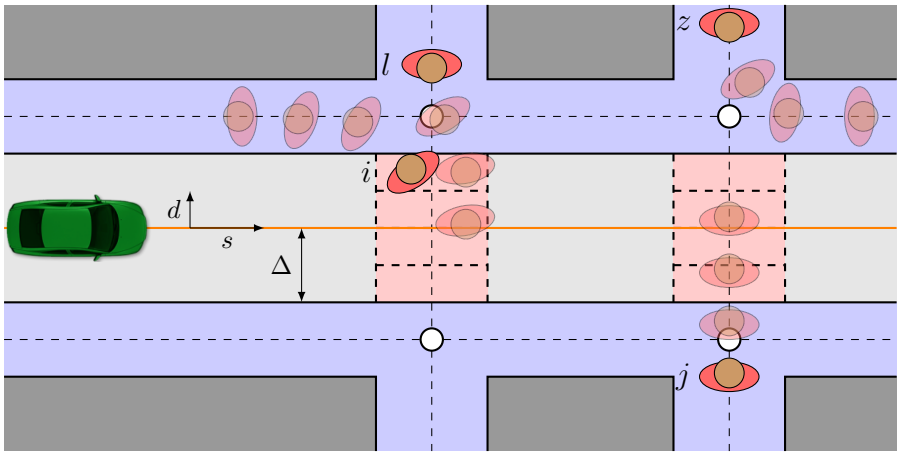


Figure 4.1: An example setting where the future motion of four pedestrians is predicted. The future motion can be translated into collision-avoidance sets $\mathcal{C}_{n|k}$ which need to be avoided, see e.g., Figure 4.2.

for an initial $\mathbf{w}_{k|k} = \mathbf{w}_k$. By resorting to reachability analysis tools [92], [93], the uncertainty sets $\mathcal{W}_{n|k}$ can easily be propagated for this system along each reference edge $\mathbf{r}_i^{\text{ped}}$. In the case that a prediction $\mathbf{w}_{n|k}$ reaches the final node of edge i, the future predictions can be propagated along all new neighboring edges j, similarly to [28], by using some distance-based rule.

According to Lemma 3.1, constructing the sets $\mathcal{W}_{n|k}$ based on reachability analysis makes it is possible to construct a collision-avoidance function γ that satisfies Assumption 3.3. We next provide an example how γ can be designed in an AD setting.

4.2.1 Constraint Generation

Consider a setting where there are O road users in the environment that the sensor system can detect, and where the future states for each road user $i \in \mathbb{I}_1^O$ are predicted in a set $\mathcal{W}_{n|k}^i$ by a prediction model that satisfies $\mathcal{W}_{n|k+1}^i \subseteq \mathcal{W}_{n|k}^i$. In order to formulate a collision-avoidance constraint, we only need to consider road users that may interact with the planned trajectory that the vehicle will track, i.e., the reference trajectory $\mathbf{r}(s)$. This can be done by computing the

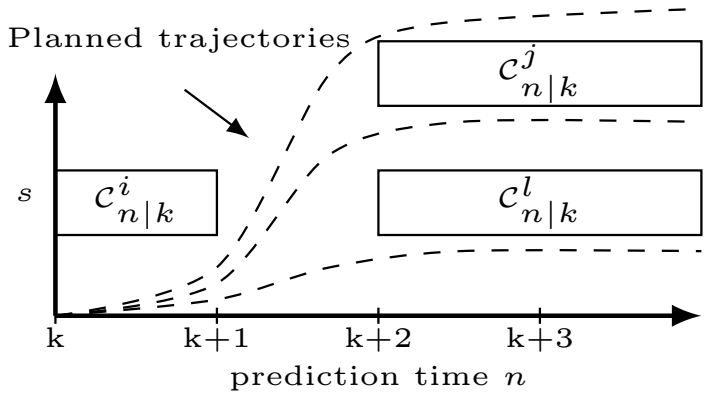


Figure 4.2: Illustration of the collision avoidance sets that are obtained from the setting shown in Figure 4.1.

closest distance of the uncertainty $\mathcal{W}^i_{n|k}$ w.r.t. the reference, i.e.,

$$d^i_{n|k}(\mathbf{w}^i_{n|k}) := \min_s \|T^{\mathbf{x}}(\mathbf{r}^{\mathbf{x}}(s)) - T^{\mathbf{w}}(\mathbf{w}^i_{n|k})\|_2, \tag{4.8}$$

where $T^{\mathbf{x}} : \mathbb{R}^{n_{\mathbf{x}}} \to \mathbb{R}^2$ and $T^{\mathbf{w}} : \mathbb{R}^{n_{\mathbf{w}}} \to \mathbb{R}^2$ map the position components of the reference and the uncertainty set into a common global frame. The solution to (4.8) is denoted as $\sigma^i_{n|k}(\mathbf{w}_{n|k})$, which represents the longitudinal position along the reference.

By computing the metrics $d_{n|k}$ and $\sigma_{n|k}$, it becomes convenient to only consider settings where the uncertainty $\mathcal{W}^i_{n|k}$ enters a driving corridor Δ around the reference $\mathbf{r}$, see Figure 4.1. Hence, to consider collision-avoidance with road user i at prediction time n, we collect the lateral and longitudinal distances $d^i_{n|k}(\mathbf{w}^i_{n|k})$ and $\sigma^i_{n|k}(\mathbf{w}^i_{n|k})$ in the following sets

$$S^i_{n|k} := \{\sigma^i_{n|k}(\mathbf{w}^i_{n|k}) \mid \forall \mathbf{w}^i_{n|k} \in \mathcal{W}^i_{n|k}\}, \tag{4.9}$$

$$D^i_{n|k} := \{d^i_{n|k}(\mathbf{w}^i_{n|k}) \mid \forall \mathbf{w}^i_{n|k} \in \mathcal{W}^i_{n|k}\}, \tag{4.10}$$

and then formulate the following avoidance set

$$\mathcal{C}^i_{n|k} := \begin{cases} S^i_{n|k} & \text{if } |d_{n|k}(\mathbf{w}^i_{n|k})| \leq \Delta \text{ for any } \mathbf{w}^i_{n|k} \in \mathcal{W}^i_{n|k}, \\ \emptyset & \text{otherwise,} \end{cases} \tag{4.11}$$

which corresponds to all longitudinal positions that the vehicle should avoid at prediction time n. Figure 4.2 shows an illustration of the set (4.11) for the situation presented in Figure 4.1. Here, four pedestrians are present in the scene, but only three of them are predicted to be intersecting with the vehicle path (orange line). For instance, pedestrian i is only on the crosswalk at the initial prediction time k, while pedestrians l and j enter the intersection at time $k+2$. In this particular situation the collision avoidance constraint can be constructed in multiple ways, i.e., the vehicle can either: (a) stop for all pedestrians, (b) yield to pedestrian j but not to pedestrian l, or (c) yield only to pedestrian i. The dashed line in Figure 4.2 illustrates these options.

Indeed, for each road user i, the vehicle needs to make a decision whether it should yield to it or drive ahead. Hence, the collision avoidance function γ^i for road user i can be constructed as

$$\gamma^i_{n|k}(s_{n|k},\mathbf{u}_{n|k},\mathbf{w}^i_{n|k}) := \begin{cases} s_{n|k}-\sigma^i_{n|k}(\mathbf{w}^i_{n|k}) & \text{if } |d_{n|k}(\mathbf{w}^i_{n|k})|\leq\Delta \text{ and yielding} \\ \sigma^i_{n|k}(\mathbf{w}^i_{n|k})-s_{n|k} & \text{if } |d_{n|k}(\mathbf{w}^i_{n|k})|\leq\Delta \text{ and not yielding} \\ \infty & \text{otherwise} \end{cases}, \tag{4.12}$$

and the resulting a-priori unknown constraint can be written as

$$g^i_{n|k}(s_{n|k},\mathbf{u}_{n|k}) := \max_{\mathbf{w}^i_{n|k}\in\mathcal{W}^i_{n|k}} \gamma^i_{n|k}(s_{n|k},\mathbf{u}_{n|k},\mathbf{w}^i_{n|k}). \tag{4.13}$$

Remark 4.1. *The combinatorics of deciding for whom the vehicle should yield to increases significantly with the number of road users that are predicted to interact with the vehicle. In order to alleviate this issue, the prediction layer could instead use clustering techniques to simplify the decision in the controller.*

As mentioned in Remark 4.1, in order to reduce the combinatorics arising from (4.13), we use a simple strategy that only considers the combinatorics of the closest intersection and yields to road users further away. For the example presented in Figure 4.1, this would translate to always yield for pedestrian j and try to: (i) drive ahead of pedestrian i; (ii) yield for pedestrian i but drive ahead of road user l; or (iii) yield for road user i and l. In other words, the introduction of constraint (4.13) to Problem (3.1) yields a mixed-integer programming problem that needs to be solved online.

Furthermore, as it was mentioned in Section 3.2, in order for constraint $g_{n|k}$ to satisfy Assumption 3.3, it needs to also consider that not all road users

can be directly measured as it was illustrated in Figure 3.9. To address this issue, we deploy a similar strategy as in [33], where *virtual* road users are placed wherever the sensor system detects that a potential occlusion may occur. For each such virtual road user, the same prediction model is applied, and constraints (4.12)-(4.13) can be formed.

4.3 Terminal Set Computation

Much of the derived theory in Chapters 2 and 3 require certain properties on the terminal conditions. In this section we will provide a brief overview of how such a set can be derived for system (4.2) and Problem (3.1), while referring the reader to Paper F for the full details.

We consider the quadratic stage and terminal costs from (2.5), i.e.,

$$q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, s_{n|k}) = \begin{bmatrix} \mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(s_{n|k}) \\ \mathbf{u}_{n|k} - \mathbf{r}^{\mathbf{u}}(s_{n|k}) \end{bmatrix}^{\top} W \begin{bmatrix} \mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(s_{n|k}) \\ \mathbf{u}_{n|k} - \mathbf{r}^{\mathbf{u}}(s_{n|k}) \end{bmatrix}, \quad (4.14)$$

$$p_{\mathbf{r}}(\mathbf{x}_{n|k}, s_{n|k}) = \begin{bmatrix} \mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(s_{n|k}) \end{bmatrix}^{\top} P \begin{bmatrix} \mathbf{x}_{n|k} - \mathbf{r}^{\mathbf{x}}(s_{n|k}) \end{bmatrix}, \quad (4.15)$$

where $W = \text{blockdiag}(Q, R)$, $Q = \text{diag}(1, 1, 10, 1, 1, 1)$, $R = \text{diag}(4, 10)$, and $P = \text{blockdiag}(P_{\text{lat}}, P_{\text{lon}})$. In order to express the terminal cost P, we must first construct the stabilizing terminal set $\mathcal{X}_{\mathbf{r}}^{\text{s}}(s)$ for system (4.2), which can be done by decoupling the longitudinal and lateral kinematics.

4.3.1 Longitudinal Kinematics

Since the longitudinal kinematics are given by

$$\begin{bmatrix} \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -t_{\text{d}} \end{bmatrix} \begin{bmatrix} v \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ t_{\text{d}} \end{bmatrix} a^{\text{req}}, \quad (4.16)$$

and the reference trajectory is assumed to satisfy the system dynamics, we can express the longitudinal kinematics in the following error frame

$$\begin{bmatrix} \dot{e}_v \\ \dot{e}_a \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -t_{\text{d}} \end{bmatrix} \begin{bmatrix} e_v \\ e_a \end{bmatrix} + \begin{bmatrix} 0 \\ t_{\text{d}} \end{bmatrix} e_{a^{\text{req}}}, \quad (4.17)$$

where the error states and inputs are defined as

$$e_v := v - v^{\mathbf{r}}(s), \; e_a := \dot{e}_v = a - a^{\mathbf{r}}, \; e_{a^{\text{req}}} := a^{\text{req}} - a^{\mathbf{r},\text{req}}. \quad (4.18)$$

Then, by designing an LQR controller with costs $Q_{\text{lon}}^{\text{LQR}} = \text{diag}(5 \times 10^{-3}, 1)$ and $R_{\text{lon}}^{\text{LQR}} = 1$, we obtain a feedback gain K_{lon} that stabilizes (4.17). Using the Multi-Parametric Toolbox (MPT) [94] we can finally compute a positive invariant set for the longitudinal dynamics

$$\mathcal{X}_{\mathbf{r}}^{\text{lon}}(s) := \{v,\ a \mid H_{\text{lon}} \begin{bmatrix} v - v^{\mathbf{r}}(s) \\ a - a^{\mathbf{r}}(s) \end{bmatrix} \leq b_{\text{lon}}\}, \tag{4.19}$$

where $H_{\text{lon}} \in \mathbb{R}^{6\times 2}$ and $b_{\text{lon}} \in \mathbb{R}^6$, and $v^{\mathbf{r}}(s)$ and $a^{\mathbf{r}}(s)$ denote the velocity and acceleration components of the reference. The corresponding terminal cost can be obtained by solving the following Linear Matrix Inequalities (LMIs)

$$\begin{aligned} P_{\text{lon}} := \min_{P}\ & \text{trace}(P) \\ \text{s.t. } & P \succ 0, \\ & (A_{\text{lon}} - B_{\text{lon}}K_{\text{lon}})^\top P (A_{\text{lon}} - B_{\text{lon}}K_{\text{lon}}) - P \prec \\ & \qquad - Q_{\text{lon}} - K_{\text{lon}}^\top R_{\text{lon}} K_{\text{lon}}, \end{aligned} \tag{4.20}$$

where A_{lon} and B_{lon} are obtained through zero-order hold discretization of (4.17), and Q_{lon} and R_{lon} correspond to the longitudinal contributions of Q and R in (4.14), i.e., $Q_{\text{lon}} = \text{diag}(1, 1)$ and $R_{\text{lon}} = 4$. Using YALMIP [95] and the SDPT solver [96], we solve (4.20) and obtain

$$P_{\text{lon}} = \begin{bmatrix} 210.78 & 80.19 \\ 80.19 & 38.29 \end{bmatrix}. \tag{4.21}$$

4.3.2 Lateral kinematics

The decoupled lateral kinematics of (4.2) can be written as

$$\begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{\delta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} v \sin e_\psi \\ vL^{-1}(\tan\delta - \tan\beta) \\ \alpha \\ w_0^2(\delta^{\text{sp}} - \delta) - 2w_0 w_1 \alpha \end{bmatrix}, \tag{4.22}$$

where we define $\beta := \arctan(\dot{s}/v \tan(\delta^{\mathbf{r}}(s)))$ and treat the velocity v as an uncertain parameter in the range of $v \in [1, 55/3.6]$ m/s. To account for the nonlinear terms in (4.22), we introduce parameters

$$\varepsilon_\psi := \frac{\sin e_\psi}{e_\psi},\ \varepsilon_\delta := \frac{\tan\delta - \tan\beta}{\delta - \beta}, \tag{4.23}$$

and rewrite (4.22) to the following form

$$\begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{\delta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} v\varepsilon_\psi e_\psi \\ vL^{-1}\varepsilon_\delta(\delta - \beta) \\ \alpha \\ w_0^2(\delta^{\text{sp}} - \delta) - 2w_0 w_1 \alpha \end{bmatrix}. \tag{4.24}$$

Then, by denoting the steering error and steering error rate as

$$e_\delta := \delta - \beta, \qquad e_\alpha = \dot{e}_\delta = \dot{\delta} - \dot{\beta} = \alpha - \dot{\beta}, \tag{4.25}$$

we rewrite the first three lines of (4.24) as

$$\begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{e}_\delta \end{bmatrix} = \begin{bmatrix} v\varepsilon_\psi e_\psi \\ vL^{-1}\varepsilon_\delta e_\delta \\ e_\alpha \end{bmatrix}. \tag{4.26}$$

Finally, in order to deal with the last equation we change the control input from δ^{sp} to

$$\rho := \delta^{\text{sp}} - \beta - 2w_0^{-1} w_1 \dot{\beta} - w_0^{-2} \ddot{\beta}, \tag{4.27}$$

and define $\nu_\psi := v\varepsilon_\psi$, $\nu_\delta := vL^{-1}\varepsilon_\delta$, to obtain the Linear Parameter Varying (LPV) system

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{e}_\delta \\ \dot{e}_\alpha \end{bmatrix} = \begin{bmatrix} \nu_\psi e_\psi \\ \nu_\delta e_\delta \\ e_\alpha \\ w_0^2(\rho - e_\delta) - 2w_0 w_1 e_\alpha \end{bmatrix}, \ \mathbf{e} = \begin{bmatrix} e_y \\ e_\psi \\ e_\delta \\ e_\alpha \end{bmatrix} \tag{4.28}$$

with state $\mathbf{e}$, input ρ, and parameters $\boldsymbol{\nu} := [\nu_\psi, \nu_\delta]^\top$, which implicitly depend on v, ε_ψ, and ε_δ. Since the bounds on the longitudinal velocity v are known, and the reference trajectory is known beforehand, we can compute worst-case realizations of $\boldsymbol{\nu}$ and contain all such combinations in the polyhedron $\mathcal{P}$ with corresponding vertices $\mathcal{V} = \{\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, ..., \boldsymbol{\nu}_6\}$.

Now, in order to compute an invariant set for the LPV system (4.28) we consider the following polytopic approximation of (4.28)

$$\Gamma := \{(A_{\text{lat}}, B_{\text{lat}}) \mid A_{\text{lat}} = A(\boldsymbol{\nu}), B_{\text{lat}} = B(\boldsymbol{\nu}),\ \forall \boldsymbol{\nu} \in \mathcal{V}\}, \tag{4.29}$$

where $A(\boldsymbol{\nu})$ and $B(\boldsymbol{\nu})$ are obtained through zero-order hold discretization of (4.28). Using the feedback gain K_{lat} from an LQR controller for system $(A(\boldsymbol{\nu}_{\text{nom}}), B(\boldsymbol{\nu}_{\text{nom}}))$ with $\boldsymbol{\nu}_{\text{nom}} = [13.89,\ 4.79]^\top$ and tuning $Q_{\text{lat}}^{\text{LQR}} =$

$\text{diag}([1, 500, 1, 0.1])$ and $R_{\text{lat}}^{\text{LQR}} = 10^{-4}$, we can stabilize the polytopic system Γ for all vertices $\boldsymbol{\nu} \in \mathcal{V}$. Then, similarly to Section 4.3.1, we use the Multi-Parameteric Toolbox [94] to obtain the following invariant set for the lateral kinematics

$$\mathcal{X}_{\mathbf{r}}^{\text{lat}}(s) = \{e_y, e_\psi, \delta, \alpha \mid H_{\text{lat}}[e_y, e_\psi, e_\delta, e_\alpha]^\top \leq b_{\text{lat}}\}, \tag{4.30}$$

$$\tag{4.31}$$

where $\mathcal{X}_{\mathbf{r}}^{\text{lat}}$ consists of 16 hyper planes, i.e., $H_{\text{lat}} \in \mathbb{R}^{16\times 4}$, $b_{\text{lat}} \in \mathbb{R}^{16}$.

Finally, in order to compute a suitable terminal cost for the lateral kinematics, we can use Finsler's lemma to write the LMI

$$\begin{aligned} P_{\text{lat}} := \min_{P}\ & \text{trace}(P) \\ \text{s.t. } & P \succ 0, \\ & \begin{bmatrix} P - Q_{\text{lat}} - K_{\text{lat}}^\top R_{\text{lat}} K_{\text{lat}} & A_{\text{lat,cl}}(\boldsymbol{\nu})^\top P^\top \\ P A_{\text{lat,cl}}(\boldsymbol{\nu}) & P \end{bmatrix} \succ 0,\ \forall \boldsymbol{\nu} \in \mathcal{V}, \end{aligned} \tag{4.32}$$

where $A_{\text{lat,cl}}(\boldsymbol{\nu}) := A(\boldsymbol{\nu}) - B(\boldsymbol{\nu})K_{\text{lat}}$, and Q_{lat} and R_{lat} correspond to the lateral contributions of Q and R in (4.14), i.e., $Q_{\text{lat}} = \text{diag}(1, 1, 10, 1)$ and $R_{\text{lat}} = 10$. Then, using YALMIP and the SDPT3 solver we obtain the terminal cost

$$P_{\text{lat}} = \begin{bmatrix} 325.51 & 593.13 & 97.32 & 1.46 \\ 593.13 & 6091.11 & 1979.43 & 29.75 \\ 97.32 & 1979.43 & 1159.47 & 17.15 \\ 1.46 & 29.75 & 17.15 & 1.28 \end{bmatrix}. \tag{4.33}$$

4.3.3 Terminal set

Having computed $\mathcal{X}_{\mathbf{r}}^{\text{lon}}(s)$ and $\mathcal{X}_{\mathbf{r}}^{\text{lat}}(s)$, we can now express the stabilizing set

$$\mathcal{X}_{\mathbf{r}}^{\text{s}}(s) := \{\mathbf{x} \mid [v,\ a]^\top \in \mathcal{X}_{\mathbf{r}}^{\text{lon}}(s),\ [e_y,\ e_\psi,\ \delta,\ \alpha]^\top \in \mathcal{X}_{\mathbf{r}}^{\text{lat}}(s)\}, \tag{4.34}$$

and the terminal cost $P = \text{blockdiag}(P_{\text{lat}}, P_{\text{lon}})$.

In order to construct the terminal set $\mathcal{X}_{\mathbf{r}}^{\text{f}}$ presented in (3.19), we must also specify a *safe* set. Similar to Example 3.1, we consider the safe set to be given when the vehicle is fully stopped, i.e.,

$$\mathcal{X}_{\text{safe}}(s) = \{\mathbf{x} \mid v = 0\}. \tag{4.35}$$

Figure 4.3: Test vehicle from the Chalmers Resource for Vehicle Research (Revere) Lab.

We note that the set description (4.35) may not always be a suitable safe set for general autonomous driving and that it may be beneficial to formulate it differently in other settings. However, for the urban driving experiment that is presented in this chapter, we consider the set (4.35) to be sufficient. Selecting the prediction horizon to be $N = 65$ and $M = 100$, we use (4.34) and (4.35) to construct the final terminal set (3.19) implicitly. This is done by extending the prediction horizon in Problem (3.1) to M, but only keeping an associated cost up to prediction time N. The reader is referred to [36] (Paper D) for full details on this practical approach.

4.4 Implementation Details

We implement the safe MPC controller in the full-scale Volvo XC90 T6 petrol-turbo SUV seen in Figure 4.3, which offers an actuation interface that accepts longitudinal acceleration and steering angle setpoint requests. The vehicle CAN bus and sensor system are interfaced through the open source software

OpenDLV[1], and the middleware library Cluon[2], which enable the reading of sensor data and the sending of actuation requests through the vehicle CAN bus.

4.4.1 Platform Limitations

For each time instance k, we read measurements from the onboard Real-Time Kinematic (RTK) GPS unit and the built-in vehicle IMU sensors to form an estimate of the initial state $\mathbf{x}_k$. No real-time corrections were available at the time of testing, which resulted in a lowered accuracy in the positioning. Furthermore, we noticed that the interface between OpenDLV and the vehicle exhibited a delay of 150 ms when sending signals to the steering actuator, but also when reading from it. Since the steering dynamics were fast, and already considered in the system model (4.1), we used dead-reckoning to estimate both the steering wheel angle δ and the steering wheel angle rate α. Furthermore, to account for the delay, the state space vector in (4.2) was augmented with additional time-delayed states for the steering angle to model the input delay, similarly to [50].

The vehicle interface was equipped with a safety feature that limited the steering wheel actuator for different velocities. To model this limitation, an additional constraint of the following form was added

$$|\delta_k| \leq (10^4(1 + e^{0.5v})^{-1} + 40)\frac{16.8\pi}{180} \text{ rad.} \tag{4.36}$$

Finally, while the vehicle was equipped with cameras, it did not have access to a computer vision software stack at the time of testing, i.e., any detection of the environment, the drivable road, and other road users was not possible. We therefore relied solely on GPS to localize the vehicle w.r.t to the reference path, and simulated a surrounding environment including moving pedestrians. While this resulted in a somewhat limited experimental scope, we note that the experiments on the other hand could be conducted safely, while still testing the controller performance in a real vehicle.

[1] See `https://opendlv.org/` for more information.
[2] See `https://github.com/chrberger/libcluon` for more information.

Figure 4.4: The vehicle is being controlled by solving an OCP on a laptop computer, and sending the actuation requests through a network connection through OpenDLV.

4.4.2 Controller Details

We formulate the MPC OCP (3.1) in the multiple shooting framework using a sampling time of $t_s = 0.05$ s, prediction horizons $N = 65$ and $M = 100$, and obtain the discretized dynamics of (4.1) using a fourth order Runge-Kutta integrator with 5 steps per control interval. The OCP was then solved using Acados [97] together with the HPIPM solver [65] in a Real Time Iteration (RTI) SQP scheme [88]. A C++ interface was written to connect Acados and HPIPM with the OpenDLV framework, and the resulting controller was deployed on a Linux laptop (i9 2.4 GHz, 32 GB RAM) as seen in Figure 4.4.

The mixed-integer problem introduced by the constraints presented in Section 4.2.1 was addressed by formulating and solving an OCP for each constraint configuration, e.g., one OCP is solved where the vehicle considers to yield for a pedestrian, while another OCP is solved where the vehicle considers to drive ahead of a pedestrian. Then, the control action corresponding to the OCP with best cost value was selected and applied to the vehicle. To make this setup real-time feasible, the written C++ wrapper therefore included parallel execution of all OCPs for each time instance.

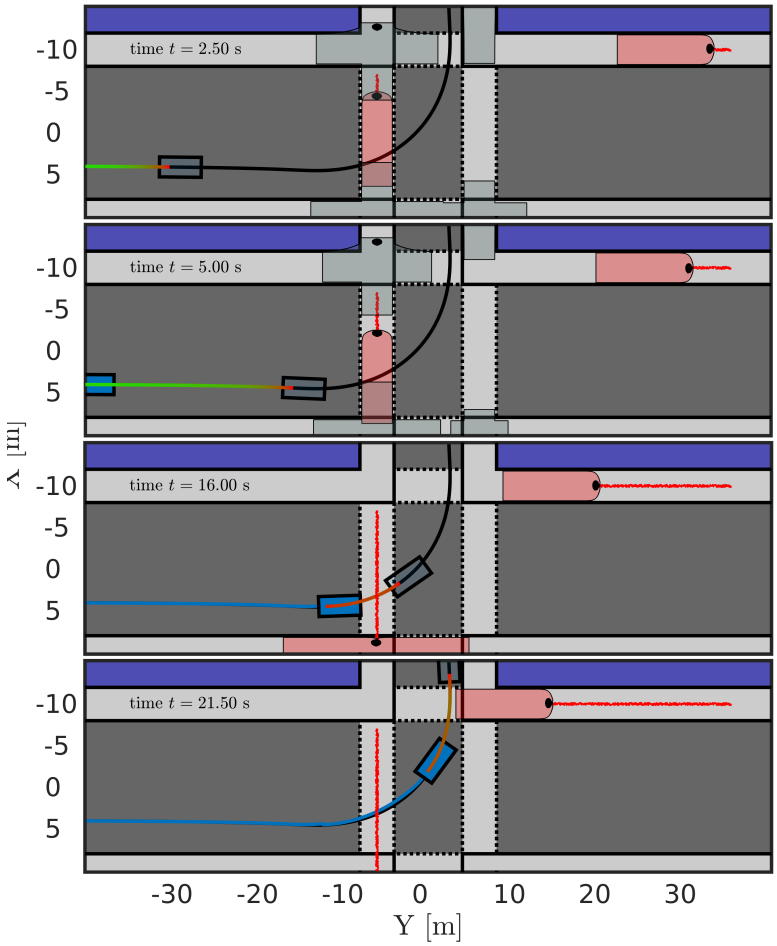


Figure 4.5: The vehicle bounding box is illustrated by the blue box, while the predicted pedestrian motion is illustrated by the red regions for the real pedestrians and gray regions for the virtual pedestrians. The opaque box represents the terminal state $\mathbf{x}_{k+M|k}$, and the line connecting the vehicle bounding box and the opaque box illustrate the predicted open-loop solution. Finally, the blue and red lines denote the travelled history of the vehicle and measured pedestrians, respectively.

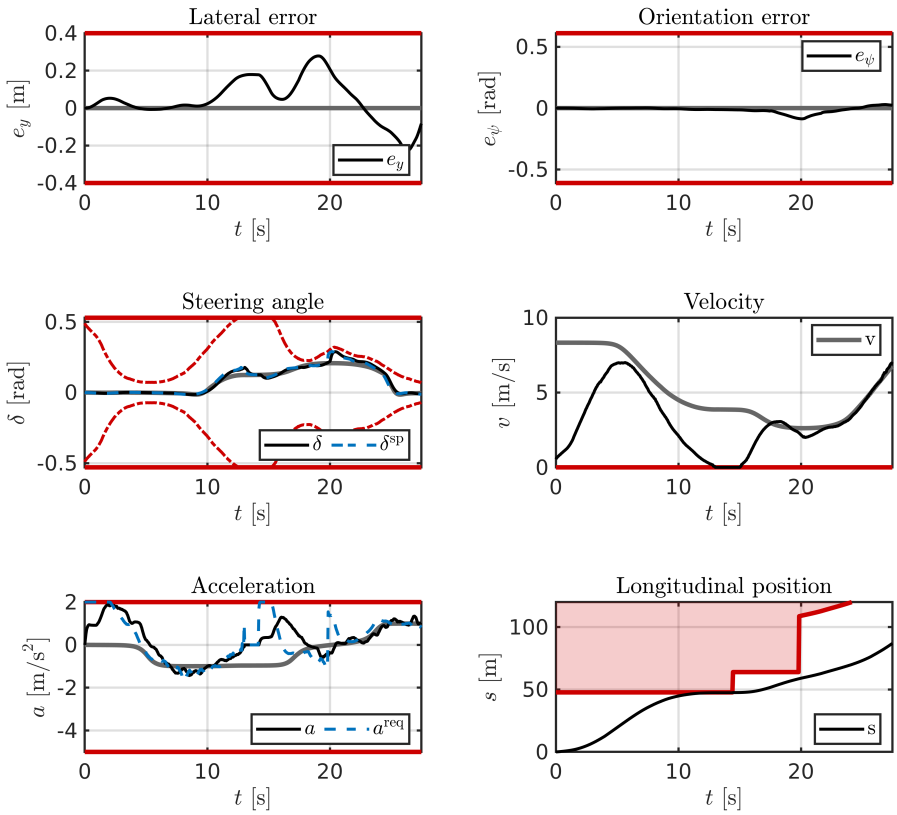


Figure 4.6: Closed-loop states from Figure 4.5. The gray lines represent the reference trajectory, while the black and blue lines show the states and inputs. The red lines denote the state and input constraints.

4.5 Results

We evaluated the safe MPC framework in a four-way intersection, with moving pedestrians. Indeed, while a more general setting would include other road users, we consider only pedestrians for simplicity and to be able to clearly illustrate the results.

Figure 4.5 shows the open-loop solutions across different time instants for the controller. Here, the grayed out region denotes the uncertainty from the

virtual pedestrians that are expected to be behind the corner, while the red region denotes the predicted uncertainty from the two measured pedestrians. The blue box represents the vehicle's position at the current time instance, while the opaque box denotes the final predicted state, i.e., $\mathbf{x}_{k+M|k} \in \mathcal{X}_{\text{safe}}$, which is at standstill. The line connecting the blue box with the opaque box, represents the open-loop solution projected onto the xy-plane and the color coding represents the predicted velocity, i.e., higher velocities are green while lower are red. The blue line behind the vehicle shows the position history, and the red lines behind the pedestrians show the history of their measured position.

It is visible from Figure 4.5 that the vehicle cautiously approaches the intersection and yields to first the pedestrian. Only when the pedestrian is predicted to be far enough from the intersection, the vehicle accelerates up to the next intersection. Upon reaching close enough, it realizes that it can clear the intersection without colliding with the oncoming pedestrian and therefore drives past the intersection.

Figure 4.6 shows the corresponding closed-loop trajectories of the vehicle for the intersection shown in Figure 4.5. It is visible that the vehicle accelerates from the start and tries to reach the reference velocity. Around time $t = 7$ s, it starts deviating from the reference trajectory since it needs to avoid a collision with the first crossing pedestrian. After $t = 15$ s the vehicle is allowed to drive again, and starts accelerating up to the next intersection, where after $t = 20$ s it realizes that it can cross the final intersection safely, and continues to track the reference freely.

4.6 Conclusions

In this chapter we have shown how a real-time capable controller can be designed for urban AD settings using the results presented in this thesis. In particular, we have shown how the MPC formulation can be designed such that the safety guarantees from Theorem 3.1 can be used, and implemented it in a real vehicle platform. While only (simulated) pedestrians were considered in this chapter, the framework can generalize to any other road user as long as the prediction model can be predicted consistently in the sense of Assumption 3.3. Future work will therefore aim to further verify the framework by considering situations with non-simulated road users.

Indeed, part of the safety argumentation relies on models that generate consistent predictions in the sense of Assumption 3.3. Therefore, future research needs to consider deriving models that are acceptable for general AD settings, i.e., they cannot rely on the road users always being rational and abide by the road code. We note that while conservative methods will be able to capture all worst-case behaviors, they may also result in more conservative driving behaviors.

CHAPTER 5

Summary of included papers

This chapter provides a summary of the included papers.

5.1 Paper A

Ivo Batković, Nils Lübbe, Mario Zanon, and Paolo Falcone
A Computationally Efficient Model for Pedestrian Motion Prediction
Published in 2018 European Control Conference (ECC),
pp. 374–379, Jun. 2018.
 DOI: 10.23919/ECC.2018.8550300.

This paper proposes a computationally efficient pedestrian prediction model that can propagate an estimate on the average position together with the uncertainty in the form of a covariance matrix. The method relies on a graph-based map of the road topology, i.e., it considers how sidewalks and zebra crossings are connected, which is then used to predict the future movement. With this information, an LQR controller is used to guide the predictions, while making the assumption of rational pedestrian behavior. This prediction strategy therefore allows for efficient computations of the future motion through

simple Kalman predict steps. The model was compared to the state-of-the-art at that time, and discussed its accuracy and limitations both in simulations but also in comparison to real data.

The thesis author contributed to the problem formulation, data preparation, implementation and evaluation of the models, and writing the paper.

5.2 Paper B

Ivo Batković, Mohammad Ali, Mario Zanon, and Paolo Falcone
Real-Time Constrained Trajectory Planning and Vehicle Control for Proactive Autonomous Driving With Road Users
Published in 2019 European Control Conference (ECC),
pp. 256–262, Jun. 2019.
 DOI: 10.23919/ECC.2019.8796099.

Motion planning and control algorithms for autonomous vehicles need to consider the future actions of the road users in efforts to remain safe and proactive. This paper presents a vehicle motion planning and control framework based on MPC that accounts for moving obstacles in the environment. It assumes that all measured pedestrian states are fed into a prediction layer, which generates future pedestrians motions that the MPC problem can consider. Simulations and experimental validation were performed with simulated crossing pedestrians to show the performance of the framework.

The thesis author contributed with the problem formulation, simulation and experimental implementation, and the writing of the paper.

5.3 Paper C

Ivo Batković, Ugo Rosolia, Mario Zanon, and Paolo Falcone
A Robust Scenario MPC Approach for Uncertain Multi-Modal Obstacles
Published in IEEE Control Systems Letters,
vol. 5, no. 3, pp. 3263–3268, Jul. 2020.
 DOI: 10.1109/LCSYS.2020.3006819.

This paper presents a control scheme based on MPC which ensures robust constraint satisfaction when the uncertainty is multi-modal and stems from

other road users' behaviors. By combining ideas from tube-based and scenario-based MPC strategies, we show that it is possible to guarantee robust state and input constraint satisfaction, while taking less conservative actions. In particular, we design a feedback policy that is a function of the disturbance mode, and approximate the expected cost by leveraging the associated mode probability.

The thesis author contributed to the problem formulation, implementation of the framework, and writing of the paper.

5.4 Paper D

> **Ivo Batković**, Mohammad Ali, Paolo Falcone, and Mario Zanon
> Safe Trajectory Tracking in Uncertain Environments
> *Provisionally accepted in IEEE Transactions on Automatic Control.*

This paper considers settings where reference trajectories can become infeasible due to a-priori unknown constraints. Such settings may cause the loss of recursive feasibility guarantees, and aggressive tracking whenever the state is far away from the reference trajectory. To alleviate these issues, we first propose a new framework called Model Predictive Flexible trajectory Tracking Control (MPFTC) which relaxes the trajectory tracking requirement, i.e., it reduces aggressive tracking behaviors. Additionally, we discuss and prove sufficient conditions which guarantee constraint satisfaction at all times under the presence of a-priori unknown constraints which might render the reference trajectory infeasible.

The thesis author contributed to the problem formulation, framework implementation and writing of the paper.

5.5 Paper E

> **Ivo Batković**, Mohammad Ali, Paolo Falcone, and Mario Zanon
> Model Predictive Control with Infeasible Reference Trajectories
> *Submitted to IEEE Transactions on Automatic Control.*

Asymptotic stability cannot be guaranteed in the standard sense when a reference trajectory is not feasible with respect to the system dynamics. Through

the use of economic MPC tools we show that asymptotic stability can be obtained with respect to an optimal reference trajectory if proper terminal conditions are used. However, since obtaining such terminal conditions are not practical, we show that Input-to-State (ISS) holds when the terminal conditions are instead based on suboptimal ones

The thesis author contributed to the problem formulation, theoretical derivation, implementation and paper writing.

5.6 Paper F

Ivo Batković, Ankit Gupta, Paolo Falcone, and Mario Zanon
Experimental Validation of Safe MPC for Autonomous Driving in Uncertain Environments
To be submitted to IEEE Transactions on Control Systems Technology.

In this paper we used the derived theory from Paper D and applied it to the setting of autonomous vehicles. We show that a vehicle motion controller based on MPC can be implemented in real-time for practical autonomous driving settings, while satisfying conditions needed to ensure safety. Through simulations, and experiments with a real test vehicle, we show that the MPC framework is able to safely and comfortably control a vehicle through an intersection with moving pedestrians.

The thesis author contributed with the problem formulation, simulation and experimental implementation, and the writing of the paper.

CHAPTER 6

Concluding Remarks and Future Work

While trying to answer the three research questions presented in Section 1, this thesis introduced a set of different MPC formulations intended for AD applications. In particular, we introduced MPC in a general sense in Chapter 2, and discussed why the standard results available in the literature cannot be directly applied to ensure safety in AD applications.

Indeed, in order for a self-driving vehicle to become safe, it must be able to sense and predict its surrounding environment. Chapter 3 therefore introduced an efficient pedestrian prediction model (Paper A) and specified a requirement on prediction properties (Assumption 3.3) that proves to be fundamental in ensuring safety (research question Q1).

Assuming that the environmental prediction models have a specific structure, we then show that safety guarantees could be enforced by the introduction of a safe set (Paper D). In other words, designing an MPC controller that is safe by design is possible (research question Q2). However, since MPC is an optimization-based technique, and can in general be computationally demanding, we show in Chapter 4 that it is still possible to deploy a safe MPC framework in a real vehicle platform (Paper B, Paper F), while satisfying the safety guarantees in Theorem 3.1 (research question Q3).

Even though this thesis presented an approach to ensure safety for AD applications, it is far from being a "silver bullet" that solves all problems related to autonomous driving. Therefore, we will next comment on some possible extensions and also outline future research directions.

6.1 Future Work

While the results in Theorem 3.1 guarantee safety, i.e., recursive feasibility, of the controller, it does not prove asymptotic stability when the a-priori unknown constraints are active. A promising future research direction, can therefore be to first expand the ISS results from Paper E to general nonlinear systems, and then show some sort of ISS results also for Theorem 3.1 by considering the road users as an input. Furthermore, while Paper E showed that using infeasible references can still yield some stability results, it may be quite limiting to always use a pre-defined reference trajectory. To that end, it would also be interesting to combine the proposed framework with some form of re-planning or online learning, e.g., updating the reference velocity in the presence of road users that make the pre-defined reference infeasible.

Paper C showed that multi-modal prediction models could be used to improve the overall performance of the system. However, the results relied partly on the availability of such models, but also on a probability estimate for each predicted mode. Indeed, to be able to use these results more widely, a future direction would be to use some learning-based methods to estimate such probabilities from real data. In addition, since the predictability of the environment plays a large part in enabling safe autonomous driving applications, it is necessary to also direct research attention towards deriving models that can accurately represent the surrounding environment.

Finally, in order to further strengthen the results presented in this thesis, future work aims to implement the framework from Paper F to a more general autonomous driving setting, where road users are not simulated, but measured and predicted in real-time.

References

[1] J. Ziegler *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[2] World Health Organization, "Global status report on road safety 2018: Summary," World Health Organization, Tech. Rep., 2018.

[3] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," National Highway Traffic Safety Administration, Tech. Rep., 2015.

[4] N. Lubbe, H. Jeppsson, A. Ranjbar, J. Fredriksson, J. Bärgman, and M. Östling, "Predicted road traffic fatalities in germany: The potential and limitations of vehicle safety technologies from passive safety to highly automated driving," in *Proceedings of IRCOBI conference. Athena, Greece*, 2018.

[5] J. Becker, M.-B. A. Colas, S. Nordbruch, and M. Fausten, "Bosch's vision and roadmap toward fully autonomous driving," in *Road Vehicle Automation*, G. Meyer and S. Beiker, Eds. Cham: Springer International Publishing, 2014, pp. 49–59.

[6] E. Stenborg, "Long-term localization for self-driving cars," PhD thesis, Chalmers University of Technology, 2020.

[7] E. Stenborg and L. Hammarstrand, "Using a single band gnss receiver to improve relative positioning in autonomous cars," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 921–926.

[8] G. Braso and L. Leal-Taixe, "Learning a neural solver for multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[9] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[10] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[11] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[12] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[13] S. Shalev-Shwartz, S. Shammah, and A. Shashua, *On a formal model of safe and scalable self-driving cars*, arXiv:1708.06374, 2018.

[14] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive bayesian filters: A comparative study," in *Pattern Recognition*, J. Weickert, M. Hein, and B. Schiele, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 174–183.

[15] A. Bissacco and S. Soatto, "Hybrid dynamical models of human motion for the recognition of human gaits," *International journal of computer vision*, vol. 85, no. 1, pp. 101–114, 2009.

[16] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 618–633.

[17] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectory patterns with gaussian processes," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, pp. 1229–1234.

[18] Y. F. Chen, M. Liu, S.-Y. Liu, J. Miller, and J. P. How, "Predictive modeling of pedestrian motion patterns with bayesian nonparametrics," in *AIAA Guidance, Navigation, and Control Conference*, 2016, pp. 1861–1875.

[19] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[20] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[21] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 261–268.

[22] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1345–1352.

[23] A. F. Foka and P. E. Trahanias, "Predictive autonomous robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 490–495.

[24] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 201–214.

[25] D. Xie, S. Todorovic, and S.-C. Zhu, "Inferring "dark matter" and "dark energy" from videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2013, pp. 2224–2231.

[26] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2543–2549.

[27] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, P. Langley, Ed., Morgan Kaufmann, 2000, pp. 663–670.

[28] I. Batkovic, M. Zanon, N. Lubbe, and P. Falcone, "A computationally efficient model for pedestrian motion prediction," in *2018 European Control Conference (ECC)*, 2018, pp. 374–379.

[29] H. Cui *et al.*, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2090–2096.

[30] H. Cui *et al.*, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," pp. 10 563–10 569, May 2020.

[31] X. Shen, I. Batkovic, V. Govindarajan, P. Falcone, T. Darrell, and F. Borrelli, "Parkpredict: Motion and intent prediction of vehicles in parking lots," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1170–1175.

[32] M. Koschi, C. Pek, M. Beikirch, and M. Althoff, "Set-based prediction of pedestrians in urban environments considering formalized traffic rules," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2704–2711.

[33] M. Koschi and M. Althoff, "Set-based prediction of traffic participants considering occlusions and traffic rules," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 249–265, 2021.

[34] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 518–528, 2020.

[35] C. Pek and M. Althoff, "Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1447–1454.

[36] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, "Safe trajectory tracking in uncertain environments," *Provisionally accepted to IEEE Transactions on Automatic Control. Preprint https://arxiv.org/abs/2001.11602*, 2020.

[37] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.

[38] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[39] J. van den Berg and M. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, 2005.

[40] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier, "Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1056–1062.

[41] N. Evestedt, O. Ljungqvist, and D. Axehill, "Motion planning for a reversing general 2-trailer configuration using closed-loop rrt," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3690–3697.

[42] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Energy-optimal coordination of autonomous vehicles at intersections," in *2018 European Control Conference (ECC)*, 2018, pp. 602–607.

[43] S. Uebel, N. Murgovski, B. Bäker, and J. Sjöberg, "A two-level mpc for energy management including velocity control of hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5494–5505, 2019.

[44] M. Zanon, "A gauss–newton-like hessian approximation for economic nmpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 4206–4213, 2021.

[45] S. Uebel, N. Murgovski, C. Tempelhahn, and B. Bäker, "Optimal energy management and velocity control of hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 327–337, 2018.

[46] G. Rodrigues de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, 2017.

[47] G. R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative receding horizon conflict resolution at traffic intersections," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 2932–2937.

[48] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2510–2525, 2019.

[49] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario mpc approach for uncertain multi-modal obstacles," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 947–952, 2021.

[50] I. Batkovic, M. Zanon, M. Ali, and P. Falcone, "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users," in *2019 18th European Control Conference (ECC)*, 2019, pp. 256–262.

[51] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 23–35, 2017.

[52] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, *Interactive multi-modal motion planning with branch model predictive control*, arXiv:2109.05128, 2021.

[53] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear mpc: Bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.

[54] P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Experimental validation of model predictive control stability for autonomous driving," *Control Engineering Practice*, vol. 81, pp. 244–255, 2018.

[55] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, *Stochastic mpc with multi-modal predictions for traffic intersections*, arXiv:2109.09792, 2021.

[56] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, 2018.

[57] O. Ljungqvist, D. Axehill, and J. Löfberg, "On stability for state-lattice trajectory tracking control," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 5868–5875.

[58] O. Andersson, O. Ljungqvist, M. Tiger, D. Axehill, and F. Heintz, "Receding-horizon lattice-based motion planning with dynamic obstacle avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 4467–4474.

[59] R. Oliveira, P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Path planning for autonomous bus driving in highly constrained environments," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2743–2749.

[60] O. Ljungqvist, N. Evestedt, M. Cirillo, D. Axehill, and O. Holmer, "Lattice-based motion planning for a general 2-trailer system," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 819–824.

[61] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[62] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub. Madison, Wisconsin, 2009.

[63] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[64] G. Frison, H. H. B. Sørensen, B. Dammann, and J. B. Jørgensen, "High-performance small-scale solvers for linear model predictive control," in *2014 European Control Conference (ECC)*, 2014, pp. 128–133.

[65] G. Frison and M. Diehl, "Hpipm: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[66] E. Klintberg, "Structure exploiting optimization methods for model predictive control," PhD thesis, Chalmers University of Technology, 2017.

[67] R. Hult, "Optimization-based coordination strategies for connected and autonomous vehicles," PhD thesis, Chalmers University of Technology, 2019.

[68] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Cham: Springer International Publishing, 2017, pp. 45–69.

[69] R. Amrit, J. B. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annual Reviews in Control*, vol. 35, no. 2, pp. 178–186, 2011.

[70] T. Faulwasser, L. Grüne, and M. A. Müller, "Economic nonlinear model predictive control," *Foundations and Trends in Systems and Control*, vol. 5, no. 1, pp. 1–98, 2018.

[71] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, no. 3, pp. 725–734, 2013.

[72] M. A. Müller, D. Angeli, and F. Allgöwer, "On necessity and robustness of dissipativity in economic model predictive control," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1671–1676, 2015.

[73] M. Zanon and T. Faulwasser, "Economic mpc without terminal constraints: Gradient-correcting end penalties enforce asymptotic stability," *Journal of Process Control*, vol. 63, pp. 1–14, 2018.

[74] M. Zanon, S. Gros, and M. Diehl, "A Lyapunov function for periodic economic optimizing model predictive control," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 5107–5112.

[75] M. Zanon, L. Grüne, and M. Diehl, "Periodic optimal control, dissipativity and mpc," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2943–2949, 2017.

[76] M. Zanon, "A gauss–newton-like hessian approximation for economic nmpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 4206–4213, 2021.

[77] J. B. Rawlings, D. Bonne, J. B. Jorgensen, A. N. Venkat, and S. B. Jorgensen, "Unreachable setpoints in model predictive control," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2209–2215, 2008.

[78] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov function for economic optimizing model predictive control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, 2011.

[79] T. Faulwasser and M. Zanon, "Asymptotic stability of economic nmpc: The importance of adjoints," 20, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018, vol. 51, 2018, pp. 157–168.

[80] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, "Model predictive control with infeasible reference trajectories," *Submited to IEEE Transactions on Automatic Control. Preprint https://arxiv.org/abs/2109.04846*, 2021.

[81] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1026–1039, 2016.

[82] A. Aguiar, J. Hespanha, and P. Kokotovic, "Path-following for nonminimum phase systems removes performance limitations," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 234–239, 2005.

[83] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 8642–8647.

[84] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3341–3346.

[85] A. Alessandretti, A. P. Aguiar, and C. N. Jones, "Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control," in *2013 European Control Conference (ECC)*, 2013, pp. 1371–1376.

[86] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1505–1511, 2017.

[87] R. Quirynen, S. Gros, B. Houska, and M. Diehl, "Lifted collocation integrators for direct optimal control in acado toolkit," *Mathematical Programming Computation*, vol. 9, no. 4, pp. 527–571, 2017.

[88] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[89] P. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[90] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," PhD thesis, University of Cambridge, 2001.

[91] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Systems & Control Letters*, vol. 62, no. 2, pp. 194–200, 2013.

[92] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," PhD thesis, Technische Universität München, 2010.

[93] S. K. Mulagaleti, A. Bemporad, and M. Zanon, *Computation of input disturbance sets for constrained output reachability*, arXiv:2111.03009, 2021.

[94] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European Control Conference (ECC)*, 2013, pp. 502–510.

[95] J. Lofberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation*, 2004, pp. 284–289.

[96] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "Sdpt3 — a matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[97] R. Verschueren *et al.*, "Acados: A modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, 2021.