Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Safety and Efficiency in Model Predictive Control for Systems with Uncertainty

Tim Brüdigam

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Klaus Diepold

Prüfer*innen der Dissertation:

1. Priv.-Doz. Dr.-Ing. habil. Marion Leibold

2. Prof. Dr.-Ing. Boris Lohmann

Die Dissertation wurde am 12.01.2022 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 11.04.2022 angenommen.

Preface

This thesis summarizes my research conducted at the Chair of Automatic Control Engineering at the Technical University of Munich. This work would not have looked the same without the support of numerous people who I would like to thank in the following.

I sincerely thank my advisor Dr. Marion Leibold, who provided countless support. I enjoyed scientific freedom but always got important input and feedback whenever necessary. I am especially grateful for highly detailed last-minute feedback whenever publication deadlines approached quickly. I would also like to thank Dr. Dirk Wollherr for his scientific support and suggestions for new research directions throughout my thesis. I appreciate all the feedback and discussions. Furthermore, I want to thank Prof. Martin Buss for his scientific input and for running a highly productive and enjoyable laboratory environment.

I am grateful that Prof. Francesco Borrelli gave me the opportunity to spend half a year in 2021 at UC Berkeley, finalizing my thesis research and experiencing a new scientific environment. My stay has truly been a defining experience.

I would like to thank Prof. Boris Lohmann for being part of my thesis committee and for his encouraging feedback, as well as Prof. Klaus Diepold for heading the committee.

Throughout my time at TUM, I have had numerous great colleagues and friends. A big thank you to Stefan Friedrich for not only being a great long-time office mate, but also for his feedback, advice, and our long discussions; his recommendations made my journey at LSR so much easier. I also enjoyed the time with all my other, short-time office mates Yongxu He, Samuel Tesfazgi, and Annalena Daniels. I also thank all other PhD students at LSR, specifically Volker Gabler, Sebastian Kerz, Gerold Huber, Tommaso Benciolini, Yongchao Wang, and Markus Schill, for contributing to an always enjoyable time at the lab as well as countless support and exchange. I want to mention Volker Gabler again for endless IT support, which allowed my students and me to work efficiently. Thank you also to ITR, specifically Alexandre Capone, Jakob Fröhner, Jonas Umlauft, Armin Lederer, Vedad Čaušević, and Petar Bevanda, for fruitful collaboration, especially during Kicker breaks. Furthermore, I want to thank the MPC lab, specifically Edward Zhu, Charlott Vallon, and Tony Zheng, for allowing me to make lasting experiences during my stay abroad.

This thesis majorly profited from the support by Larissa Schmid, Miruna Werkmeister, Wolfgang Jaschik, Brigitta Renner, Ulrike Scholze, and all other non-scientific staff at LSR, who contribute majorly to helping us PhD students succeed - a big thanks to all of you.

I also want to thank Daniel Althoff, Moritz Werling, Georg Tanzmeister, Sascha Steyer, and Christian Pek for their scientific support during my project work with BMW. I am also grateful for discussions with Prof. Matthias Althoff, boosting my research on safe planning.

During my PhD, I had the pleasure of working with over 50 intelligent and hard-working students, who significantly contributed to my research. I will not be able to mention everyone, but I highly appreciate the effort and contribution of every single student. I specifically want to thank Michael Olbrich, Johannes Teutsch, Michael Fink, Victor Gaßmann, Kenan Ahmic, Fulvio di Luzio, and Jie Zhan for their excellent contribution to our research.

Finally, I want to express my deepest gratitude to my family and friends. Thank you to my parents for always having put me in a position to succeed, and for their encouraging advice but also critical feedback whenever necessary. Thank you Jan, who always challenges me and is a brother I can look up to. And thank you, Alysha, for always supporting me, cheering me up, inspiring me to make new experiences, as well as being a role model.

Acknowledgments

This thesis was supported by the BMW Group, a fellowship within the IFI program of the German Academic Exchange Service (DAAD), and the Bavaria California Technology Center (BaCaTeC) grant 1-[2020-2].

Abstract

A defining feature of the past decade has been the increased emergence of automation in various fields. Early automation was mostly employed in safe, often restricted areas such as the isolated work space of a factory. However, recent research has attempted to introduce automation in safety-critical applications, e.g., automated vehicles sharing a road with human drivers and pedestrians, or robots collaborating with humans. What makes this challenging is uncertainty and, particularly, predicting uncertain future behavior of other agents.

Taking into account this uncertainty is, therefore, a major challenge when designing controllers for safety-critical applications. Model Predictive Control (MPC) has been shown to be a suitable method for such tasks, as MPC is able to satisfy input and state constraints, potentially subject to uncertainty. There are two main MPC approaches when uncertainty is present, robust Model Predictive Control (RMPC) and stochastic Model Predictive Control (SMPC). RMPC uses worst-case uncertainty realizations to provide robust, but conservative control behavior, which satisfies all constraints under any possible uncertainty realization. Alternatively, SMPC reduces this conservatism by neglecting the more unlikely uncertainty realizations. Instead of treating constraints robustly, SMPC employs chance constraints, which only need to be satisfied with a chosen probability. However, these chance constraints allow a small probability of constraint violation, which is unsuitable for safety-critical applications. Hence, new MPC approaches are required that allow for non-conservative control without sacrificing constraint satisfaction.

This thesis focuses on facilitating the use of MPC for safety-critical applications subject to uncertainty. On the one hand, this is achieved by ensuring safe control behavior, i.e., satisfying all constraints. On the other hand, existing MPC methods are advanced to increase efficiency, i.e., maximizing the desired control objectives.

We provide two new contributions to safety. We propose a safe SMPC algorithm, which is particularly suitable for automated vehicles. This safety algorithm allows us to safely apply SMPC inputs. This is achieved by using a backup safety planner, which overwrites the SMPC input in cases where applying the SMPC input would lead to a state that is not safe anymore. This procedure yields similar control behavior to SMPC for most uncertainty realizations, but provides a safety guarantee if unlikely uncertainty realizations occur. This safe SMPC algorithm is based on assumptions on the constraints and uncertainty bound. For situations where these assumptions are violated, we propose a novel MPC method that minimizes the constraint violation probability. For both of these safety algorithms, we prove recursive feasibility of the MPC optimal control problems, and we address stability for the latter approach.

Besides focusing on safety, we address further aspects of MPC. We propose a combination of increasing the sampling time along the prediction horizon and using simplified models for long-term predictions to enable an extended MPC prediction horizon without increased computational complexity. This method can handle system uncertainty, where we propose to use robust constraints for the short-term horizon and chance constraints for long-term predictions to decrease conservatism.

Furthermore, we propose two advances to SMPC for application-specific uncertainty. First, we design an SMPC method specifically suited for applications with a two-fold uncertainty structure. This method efficiently considers task uncertainty and task execution uncertainty of other agents, e.g., a vehicle choosing between multiple possible maneuvers where each maneuver may be executed in various ways. Second, we propose a grid-based SMPC method for two-dimensional applications, which allows for a simple chance constraint reformulation, based on a grid-based representation of the environment and a risk threshold parameter.

We evaluate all methods in simulation studies, specifically focusing on automated vehicle examples. The simulation results show that the proposed methods enable increased efficiency with respect to comparable MPC methods and that safety is enhanced, even in complex situations.

Zusammenfassung

Im letzten Jahrzehnt hat die Automatisierung in verschiedenen Bereichen stark zugenommen. Erste Einsätze von Automatisierung fanden meist in sicheren, oft eingeschränkten Bereichen statt, z. B. in einer abgegrenzten Fläche einer Fabrik. Forschung der letzten Jahre fokussierte sich darauf, die Automatisierung in sicherheitskritischen Anwendungen zu erleichtern, z. B. bei automatisierten Fahrzeugen, die die Straße mit menschlichen Fahrern und Fußgängern teilen, oder bei Robotern, die mit Menschen zusammenarbeiten. Eine Herausforderung bei diesen Forschungsvorhaben ist die Unsicherheit und insbesondere die Vorhersage unsicheren Verhaltens.

Die Berücksichtigung dieser Unsicherheit ist daher eine große Herausforderung beim Entwurf von Regelungen, insbesondere für sicherheitskritische Anwendungen. Model Predictive Control (MPC) hat sich als geeignetes Verfahren für solche Aufgaben erwiesen, da MPC mit Eingangs- und Zustandsnebenbedingungen umgehen, aber auch Unsicherheit berücksichtigen kann. Robust Model Predictive Control (RMPC) berücksichtigt Worst-Case-Unsicherheitsrealisierungen, um ein robustes, aber konservatives Steuerungsverhalten zu erreichen. Alternativ dazu reduziert stochastic Model Predictive Control (SMPC) dieses konservative Verhalten durch Vernachlässigung unwahrscheinlicher Unsicherheitsrealisierungen. Anstatt Einschränkungen robust zu behandeln, verwendet SMPC daher probabilistische Nebenbedingungen (chance constraints), die nur bis zu einem vordefinierten Risikoparameter erfüllt werden müssen. Diese probabilistischen Nebenbedingungen erlauben jedoch eine geringe Wahrscheinlichkeit der Verletzung einer Nebenbedingung, was für sicherheitskritische Anwendungen ungeeignet ist.

Diese Dissertation konzentriert sich darauf, den Einsatz von MPC für sicherheitskritische Anwendungen zu erleichtern. Einerseits wird dies durch den Fokus auf Sicherheitsgarantien (Erfüllung aller Nebenbedingungen) erreicht. Andererseits werden bestehende MPC-Methoden weiterentwickelt, um die Effizienz zu erhöhen (Maximierung der gewünschten Regelziele).

Wir liefern zwei Beiträge zu Sicherheit. Wir schlagen einen sicheren SMPC-Algorithmus vor, der es ermöglicht, SMPC-Stellgrößen sicher anzuwenden, was besonders für autonome Fahrzeuge wichtig ist. Dies wird durch die Verwendung eines Backup-Sicherheitsplaners erreicht, der die SMPC-Stellgröße ersetzt, wenn die SMPC-Stellgröße zu einem Zustand führen würde, für den im Anschluss keine sichere Trajektorie mehr bestimmt werden kann. Dieses Verfahren ergibt ein ähnliches Regelverhalten wie SMPC für die meisten Unsicherheitsrealisierungen, bietet aber eine Sicherheitsgarantie, wenn unwahrscheinliche Unsicherheitsrealisierungen auftreten. Dieser sichere SMPC-Algorithmus basiert auf Annahmen für die Nebenbedingungen und Unsicherheitsgrenzen. Für den Fall, dass diese Annahmen verletzt werden, schlagen wir eine neuartige MPC-Methode vor, die die Wahrscheinlichkeit der Verletzung von Nebenbedingungen minimiert. Für beide Sicherheitsalgorithmen beweisen wir die rekursive Machbarkeit der MPC-Optimierungsprobleme und untersuchen die Konvergenz für den letztgenannten Ansatz.

Neben dem Fokus auf Sicherheit befassen wir uns mit weiteren Aspekten von MPC. Wir schlagen eine Kombination aus zunehmenden Abtastzeiten über den Horizont und der Verwendung vereinfachter Modelle für langfristige Prädiktionen vor, um einen erweiterten MPC-Prädiktionshorizont ohne erhöhte Rechenkomplexität zu ermöglichen. Diese Methode kann mit Systemunsicherheiten umgehen. Wir schlagen vor, aus Sicherheitsgründen für kurzfristige Prädiktionen robuste Nebenbedingungen und für langfristige Prädiktionen probabilistische Nebenbedingungen zu verwenden, um die Effizienz hochzuhalten.

Darüber hinaus schlagen wir zwei Verbesserungen für SMPC vor. Erstens entwerfen wir eine SMPC-Methode, die speziell für Anwendungen mit einer zweifältigen Unsicherheitsstruktur geeignet ist. Diese Methode berücksichtigt effizient Aufgabenunsicherheit und die Unsicherheit der Aufgabenausführung durch andere Agenten, z. B. ein Fahrzeug, das zwischen mehreren möglichen Manövern wählt, wobei jedes Manöver auf verschiedene Weise ausgeführt werden kann. Zweitens schlagen wir eine Grid-basierte SMPC-Methode für zweidimensionale Anwendungen vor, die eine einfache Umformulierung der probabilistische Nebenbedingungen auf der Grundlage eines Grids und eines Risikoschwellenparameters ermöglicht.

Wir evaluieren alle Methoden in Simulationsstudien, wobei wir uns speziell auf Beispiele für automatisierte Fahrzeuge konzentrieren, da das automatisierte Fahren zusammen mit der Robotik wohl die bedeutendsten sicherheitskritischen Anwendungen der jüngsten Forschung sind. Die Simulationsergebnisse zeigen, dass die vorgeschlagenen Methoden eine Effizienzsteigerung gegenüber vergleichbaren MPC-Methoden ermöglichen und die Sicherheit auch in komplexen Situationen erhöhen.

Contents

1 Introduction

Increased demand for automation in industry applications has led to a variety of research directions, focusing on solving challenges arising from new application requirements. In previous decades, safety in automation had often been ensured by preventing any interactions between automated machines and dynamic objects in the environment, e.g., only allowing industrial robots to operate within an isolated work space. However, requiring automated machines to work in separated work spaces majorly limits the potential benefits of automation. The recent demand for interactive automation has resulted in various new open problems, especially in sensing, perception, prediction, planning, and control. A major challenge is dealing with uncertainty that is present in dynamic environments. Whereas uncertainty of the automated agent, e.g., due to state estimation errors, can often be limited and quantified, uncertainty of other agents in the environment is usually more difficult to quantify, e.g., predicting future behavior of other vehicles on the road.

In general, sensing, perception, and prediction modules provide a stochastic representation of the environment and its potential future evolution. For planning and control, the challenge is then to provide suitable behavior of the automated agent. Since many of these automated agents act in safety-critical environments, the main tasks are guaranteeing safety (satisfying all constraints) and enabling efficient behavior (maximizing control objectives). However, these objectives are often conflicting.

This conflict becomes apparent when considering the example of a self-driving vehicle. In dense traffic, moving slowly or stopping is safe, but not efficient. For self-driving vehicles, uncertainty arises from multiple sources, e.g., localization errors regarding the self-driving vehicle as well as prediction uncertainty regarding the future behavior of other traffic participants. These predictions are especially challenging: predicting a sudden change in pedestrian motion is tricky, other vehicles may perform one of multiple possible maneuvers, and maneuver execution may also vary, e.g., a fast or slow lane change. In order to enable efficient behavior of the self-driving vehicle, these uncertainties must be taken into account during the motion planning phase. However, efficient planning does not necessarily result in safe vehicle behavior. Efficiently maneuvering through dense traffic may require overly optimistic vehicle behavior, assuming that other vehicles follow the most likely paths in the future. As not all possible actions by other vehicles and traffic participant are taken into account this way, collision avoidance cannot be guaranteed. While planning vehicle motion that accounts for all possible future actions of other vehicles leads to safe behavior of the self-driving vehicle, this robust way of motion planning is often highly conservative and, in the worst case, requires the vehicle to stop and wait until traffic is less dense.

In general, motion planning is comprised of multiple tasks, such as path or trajectory planning and specifically maneuver planning for cars. Over the past decades, a variety of motion planning approaches have been proposed, especially for automated vehicles and mobile robots. Among these approaches are graph-based planners based on the Dijkstra algorithm [6,60], A* [20,82], or rapidly-exploring random trees (RRTs) [109,120], as well as methods focusing on artificial forces such as artificial potential fields [100,101,200]. Furthermore, learning-based methods for motion planning have emerged, using neural networks [18,161] or reinforcement learning methods [113,143,196]. Extensive surveys on path planning may be found in [93,204], whereas [54] specifically focuses on motion planning for autonomous driving on highways.

Motion planning often requires considering constraints in dynamic environments, e.g., road limits or collision avoidance with obstacles. Furthermore, for many applications it is possible to obtain prediction models, allowing for approximated forecasting of future behavior based on the current system state and planned system inputs. Model Predictive Control (MPC) is a motion planning approach that enables using these prediction models and has the capability to consider constraints on the system state and input. In MPC, a finite horizon optimal control problem is repeatedly solved. At each time step, only the first input element of the optimized input sequence is applied to the system. Then, in the subsequent time step, the MPC optimal control problem is solved again with a shifted horizon.

Early work on MPC began in the 1970s [56,162,163]. Key requirements for each MPC algorithm are stability and recursive feasibility. Recursive feasibility ensures that if the MPC optimal control problem is feasible at one time step, it must be guaranteed that the MPC optimal control problem remains feasible at the subsequent time step. Major contributions to stability and recursive feasibility began in the late 1980s and 1990s [96,131,136,141]. Standard MPC, however, does not consider uncertainties. This led to the developments of robust Model Predictive Control (RMPC) [9] and more recently stochastic Model Predictive Control (SMPC) [68,134]. RMPC treats uncertainties robustly by considering worst-case uncertainty realizations. However, the robust constraints of RMPC may lead to overly conservative and inefficient control behavior. SMPC relaxes the conservative robust constraints of RMPC with probabilistic chance constraints. Unlike a robust constraint, a chance constraint is only required to hold up to a predefined probability value, also known as the risk parameter. The downside of reducing conservatism with SMPC is that constraint violations are allowed based on the risk parameter. Both RMPC and SMPC have been successfully developed for applications subject to uncertainty [71,107,121,144,198].

This thesis focuses on MPC for systems with uncertainty. We focus on general MPC algorithms, applicable to a broad system class, and specifically apply MPC to automated vehicle scenarios subject to environment uncertainty. A major aim of this thesis is to combine efficient and safe motion planning. In order to achieve this aim, on the one hand, we extend current SMPC and RMPC methods to increase efficiency, and on the other hand, we provide safety guarantees for MPC methods considering system uncertainty. The novel MPC algorithms are evaluated in simulations, exhibiting beneficial behavior. In summary, this thesis advances MPC for safe and efficient control of safety-critical systems.

1.1 Challenges in Model Predictive Control for Systems with Uncertainty

Various challenges arise when designing control algorithms for systems with uncertainty. Here, we focus specifically on MPC designed for systems subject to uncertainty. We believe that the following challenges are fundamental in order to successfully design MPC for safety-critical applications.

Challenge 1. *How may application-specific environment uncertainties be handled efficiently within SMPC?*

Chance constraints in SMPC cannot be handled directly by a solver, requiring a deterministic, and potentially approximating, reformulation. If the underlying uncertainty is Gaussian, an analytic chance constraint reformulation is possible. For other, potentially unknown probability distributions, sampling-based SMPC methods may be applied. However, these sampling-based methods are often overly conservative, scale poorly with a larger state space, and providing theoretic guarantees is challenging. In applications, we often find probability distributions that are either difficult to model or are too complex to be handled suitably with current SMPC approaches. These challenges require new approaches to formulate SMPC chance constraints for application uncertainty and subsequently determine tractable chance constraint expressions.

Challenge 2. *How may detailed prediction models be combined with long prediction horizons in MPC while keeping computational complexity manageable?*

MPC requires a prediction model, which is used on a finite horizon. When designing MPC for an application, model complexity and horizon length must be chosen. In general, detailed prediction models and long prediction horizons are beneficial for performance; however, computational complexity significantly increases with more detailed models and larger horizons. This trade-off can be improved by using models of different granularity with approximated long-term models [7], move-blocking techniques [39,75,177,180], and non-uniform prediction horizons [188,191] to allow for longer horizons. However, safely and efficiently incorporating uncertainty into these methods remains a challenge. Especially when dealing with simplified models for the long-term prediction horizon, it is important to consider uncertainty that arises from using a more coarse model.

Challenge 3. *How may safety be guaranteed within SMPC?*

The benefit of SMPC, i.e., relaxing robust constraints to reduce conservatism, is at the same time also its disadvantage: a small probability of constraint violation is allowed. Whereas this is acceptable for certain applications, potential constraint violation is intolerable for safety-critical applications. It is therefore of fundamental interest to develop SMPC methods that are able to reduce conservatism, compared to robust approaches, but still allow for a safety guarantee.

Challenge 4. *How may constraint violation probability be minimized?*

Addressing safety in control usually relies on assumptions, e.g., knowing the probability distribution and bounds of the uncertainty a priori and having time-invariant constraints. Once the application violates these assumptions, proven theoretic guarantees are lost. Consider an automated driving example where another vehicle suddenly behaves in an unpredictable manner, e.g., the other vehicle performs an illegal lane change. In this case, it may not be possible to guarantee collision avoidance anymore, as it is a regular assumption for collision avoidance algorithms that other vehicle adhere to traffic rules. In these situations,

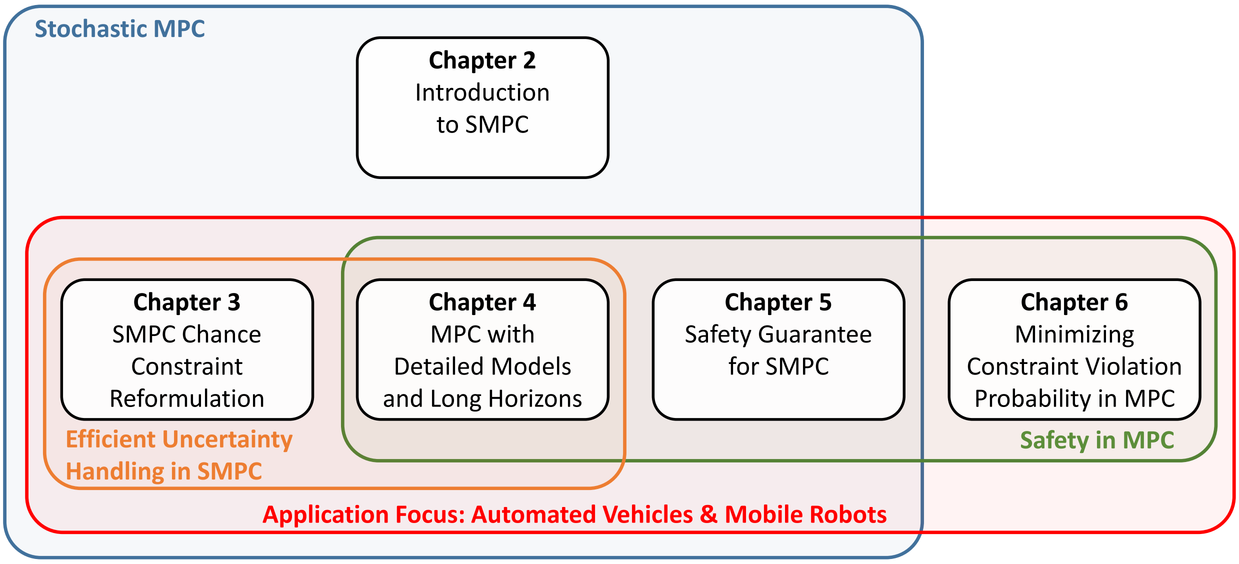


Figure 1.1: Areas of contribution within this dissertation.

the focus changes from guaranteeing collision avoidance to minimizing the collision probability. How to deal with this issue is still an open challenge in MPC. It is therefore fundamental to address minimizing constraint violation probability in MPC.

The following section summarizes how these four challenges are addressed within this thesis. As each chapter may cover more than one of the challenges, we describe the contribution of each chapter in the following.

1.2 Main Contributions and Outline

This thesis proposes MPC approaches for systems with uncertainty, addressing the previously introduced challenges. We specifically focus on ensuring safety while still enabling efficient system behavior. The majority of this work centers around SMPC, but RMPC is also addressed. The main application examples of this thesis are automated vehicles and mobile robots, as these applications are a prime example for safety-critical applications, requiring safety guarantees in addition to efficient behavior.

First, Chapter 2 introduces SMPC theory and the state of the art. The following main part consists of four chapters, proposing novel MPC algorithms for systems with uncertainty. In Chapter 3, we present two approaches to exploit application-specific uncertainty to reformulate chance constraints in SMPC. Chapter 4 presents our approaches to enabling extended MPC prediction horizons and combining RMPC as well as SMPC to handle uncertainty. A safety guarantee for SMPC, with application focus on automated vehicles, is developed in Chapter 5, whereas Chapter 6 provides an approach to minimize constraint violation probability in MPC in case of unforeseen events. Chapter 7 provides an overview of our additional work not covered in detail within this thesis. Conclusive remarks and an outlook to future research directions are given in Chapter 8. The areas of contribution are summarized in Figure 1.1.

Chapter 2: Introduction to SMPC. Before presenting the contributions of this thesis, we briefly introduce SMPC. First, the state of the art in SMPC is presented. Then, we explain the theory behind SMPC, specifically focusing on reformulating SMPC chance constraints into tractable expressions for the optimal control problem. Eventually, we illustrate the idea of SMPC in a simple simulation example.

Chapter 3: SMPC chance constraint reformulation. Challenge 1 is addressed within this chapter by providing novel approaches to consider and reformulate chance constraints efficiently. Here, efficiency refers to reducing conservative control behavior, which often results from uncertainty approximations when obtaining a tractable chance constraint expression.

First, we propose an SMPC approach specifically suited for applications with a task uncertainty and a task execution uncertainty. This includes robotic applications, or automated vehicles with maneuver and maneuver execution uncertainty. The proposed method exploits these different uncertainty types and distributions, which then allows us to formulate a less conservative SMPC optimal control problem compared to standard SMPC approaches.

The second presented SMPC method proposes a novel chance constraint reformulation specifically suited for vehicle applications. A two-dimensional space is divided into a grid, and each cell of this grid is assigned a probability of being inadmissible. Then, based on a probability threshold parameter, a convex set of admissible cells is generated, which then yields a convex SMPC optimal control problem. This grid-based SMPC approach is especially useful for vehicle application where (dynamic) occupancy grids are used for perception and prediction. The grid-based SMPC allows for the direct use of the data from occupancy grids. The results presented in this chapter have been published in [28, 30, 36].

Chapter 4: Detailed prediction models and long prediction horizons. This chapter majorly focuses on Challenge 2, but also addresses efficient uncertainty handling (Challenge 1) and safe planning (Challenge 3). The proposed approaches depend on two ideas: MPC with models of different granularity [7] and MPC with a non-uniformly spaced optimization horizon, i.e., larger sampling time steps for long-term predictions [188, 191]. First, we propose an MPC approach that combines these two ideas and ensures recursive feasibility of the MPC optimal control problem. This approach allows us to use an extended MPC prediction horizon as, compared to standard MPC, the model complexity is reduced by using a coarse prediction model for long-term predictions and less optimization variables are used due to the non-uniform horizon. Second, we consider system uncertainty, extending the previously proposed approach. We use robust constraints for the short-term horizon to ensure immediate constraint satisfaction and employ chance constraints for the long-term horizon, with a coarse model and larger sampling time steps, allowing for less conservative planning. The results presented in this chapter have been published in [33–35].

Chapter 5: Safety in SMPC. This chapter focuses on safety for SMPC, addressing Challenge 3. While chance constraints in SMPC allow for optimistic control behavior by planning less conservatively than RMPC, constraint violations are possible. The trade-off between efficiency and safety is problematic in safety-critical applications such as automated vehicles. We therefore propose a safe SMPC algorithm, particularly suitable for vehicles, which exploits the benefits of planning with SMPC while still guaranteeing collision avoidance. This safety guarantee is ensured by using a failsafe backup planner, which may overwrite the SMPC input. The resulting control behavior is similar to SMPC. However, in case of unlikely events, e.g., emergency scenarios, the proposed failsafe SMPC algorithm ensures safe vehicle behavior, unlike SMPC. The results presented in this chapter have partly been published in [32].

Chapter 6: Minimizing constraint violation probability in MPC. The major focus of this chapter is providing a solution to Challenge 4 by considering unmodeled or unforeseen system and uncertainty behavior. However, the proposed approach is also relevant for Challenge 3. Previously, safety was guaranteed by ensuring that all constraints are satisfied at all times, based on system assumptions. In applications, assumptions may fail, which causes issues for RMPC and SMPC approaches. In cases where constraint satisfaction may not be guaranteed anymore, the probability of collision should still be minimized. We therefore propose a novel MPC approach, which minimizes the probability of constraint violations. This proposed method is suited for safety-critical applications, as control objectives are maximized whenever safety can be guaranteed but the probability of future constraint violation is minimized whenever a safety guarantee is not possible. The proposed MPC method complements SMPC and RMPC. The results presented in this chapter have partly been published in [29].

1.3 List of Publications

The contributions presented in this thesis are based on the following works by the author.

- Chapter 3 is based on:
  - T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr, “Combining stochastic and scenario model predictive control to handle target vehicle uncertainty in autonomous driving,” in *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 2018, pp. 1317-1324.
  - T. Brüdigam, J. Zhan, D. Wollherr, and M. Leibold, “Collision avoidance with stochastic model predictive control for systems with a twofold uncertainty structure,” in *2021 IEEE 24th International Conference on Intelligent Transportation Systems (ITSC)*, Indianapolis, IN, USA, 2021, pp. 432-438.
  - T. Brüdigam, F. di Luzio, L. Pallottino, D. Wollherr, and M. Leibold, “Grid-based stochastic model predictive control for trajectory planning in uncertain environments,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1-8.
- Chapter 4 is based on:
  - T. Brüdigam, D. Prader, D. Wollherr, and M. Leibold, “Model predictive control with models of different granularity and a non-uniformly spaced prediction horizon,” in *2021 American Control Conference (ACC)*, New Orleans, LA, USA, 2021, pp. 3876-3881.
  - T. Brüdigam, J. Teutsch, D. Wollherr, and M. Leibold, “Combined robust and stochastic model predictive control for models of different granularity,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7123-7129, 2020, *21st IFAC World Congress.*

  - T. Brüdigam, J. Teutsch, D. Wollherr, M. Leibold, and M. Buss, "Probabilistic model predictive control for extended prediction horizons," *at - Automatisierungstechnik*, vol. 69, no. 9, pp. 759-770, 2021.

- Chapter 5 is based on:
  - T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IEEE Transactions on Intelligent Vehicles*, pp. 1-1, 2021.

- Chapter 6 is based on:
  - T. Brüdigam, V. Gaßmann, D. Wollherr, and M. Leibold, "Minimization of constraint violation probability in model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 14, pp. 6740-6772, 2021.

- Chapter 7 is based on:
  - T. Brüdigam, K. Ahmic, M. Leibold, and D. Wollherr, "Legible model predictive control for autonomous driving on highways," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 215-221, 2018, *6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018*.
  - T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian process-based stochastic model predictive control for overtaking in autonomous racing," in *2021 IEEE International Conference on Robotics and Automation (ICRA) Workshop on Opportunities and Challenges with Autonomous Racing*, Xian, China, 2021.

Introduction to Stochastic MPC 2

SMPC is a major focus of this thesis. Therefore, in the following, we provide a brief overview of recent SMPC literature that is relevant throughout this thesis. Then, each of the following chapters includes a related work section specifically suited for the content of the respective chapter. In this chapter, in addition to the SMPC literature overview, we briefly introduce theory of SMPC, state fundamental definitions for MPC properties, and give a simple SMPC simulation example, which illustrates the idea and implementation of SMPC.

2.1 State of the Art in Stochastic MPC

Standard MPC does not consider uncertainty. However, most applications are either subject to uncertainty or the prediction model used for MPC is only an approximation and, therefore, does not allow for a perfect prediction. Hence, uncertainty is introduced into the MPC formulation. For bounded uncertainties it is possible to define an RMPC optimal control problem, which guarantees constraint satisfaction given any assumed uncertainty realization [9, 160]. However, these RMPC solutions are often conservative and potentially computationally complex.

SMPC reduces this conservatism and, depending on the problem and approach, provides computational complexity similar to standard MPC. Instead of considering worst-case uncertainty realizations, relaxed formulations are used in SMPC. On the one hand, this affects the computation of the objective function, for example using the expected value and not a robust formulation. On the other hand, SMPC employs chance constraints instead of robust constraints. Depending on a predefined risk parameter, the chance constraint only needs to be satisfied to a certain degree, allowing to neglect unlikely uncertainty realizations. Extensive surveys on SMPC are found in [68, 134]. A performance analysis comparing MPC and SMPC is presented in [178]. The majority of SMPC approaches consider linear systems; however, there are SMPC approaches suitable for nonlinear systems, e.g., early work [122, 192] and subsequent research [37, 65, 135]. In the following, we discuss SMPC approaches relevant for this thesis, focusing on three general approaches to reformulate chance constraints: SMPC based on analytic reformulation, affine-disturbance feedback SMPC, and sampling-based SMPC.

If the system is subject to additive uncertainty that is normally distributed, the chance constraint within the optimal control problem can be reformulated exactly into a deterministic expression [150, 176]. This reformulation is also known as constraint tightening. For general distributions with known mean and covariance matrix, Chebyshev's inequality can be used [66, 67] and is also applicable to output-feedback MPC [67], however, the resulting constraint tightening is conservative [67, 68]. A similar approach is used in stochastic tube

MPC where state and input are split into a deterministic and probabilistic part [44, 105]. Then, a predefined stabilizing state feedback matrix is used to control the nominal system and error dynamics, whereas the decision variables of the optimal control problem account for the uncertainty.

Instead of using a predefined state feedback law, affine-disturbance feedback SMPC approaches consider a feedback law that depends on an affine function of past disturbances [104, 147]. An equivalent exists between state feedback control policies and disturbance feedback control policies [76, 134].

Sampling-based SMPC approaches are suitable for arbitrary uncertainty distributions. In particle SMPC, the risk parameter defines the amount of uncertainty samples drawn for which the constraints may be violated [15]. In scenario Model Predictive Control (SCMPC), based on the scenario approach [41, 43], the risk parameter is used to define a number of samples for which the constraints must hold [42, 171]. Whereas these sampling-based SMPC approaches are useful to deal with arbitrary uncertainties, large state spaces may be computationally challenging and it is, in general, difficult to provide guarantees.

Besides these main three SMPC ideas, other SMPC approaches exist that solve specific problems. Whereas the previously presented SMPC approaches consider open-loop chance constraint satisfaction, [86] guarantees closed-loop chance constraint satisfaction. In [149], an adaptive SMPC approach is proposed, where the constraint tightening depends on the observed probability of constraint satisfaction. Covariance-steering SMPC steers the covariance of the probabilistic system towards a target value while an expectation value-based cost function is minimized. SMPC has recently also been used in data-driven control, enabling less conservative handling of noisy data and system disturbances [98, 151].

Two main properties for MPC algorithms are recursive feasibility and stability. Recursive feasibility ensures that if the MPC optimal control problem is feasible at a time step, a solution to the MPC optimal control problem also exists at the next time step. For stability, various definitions exist, where SMPC requires accounting for uncertainty in the stability proofs. As previously mentioned, sampling-based approaches, in general, do not provide a recursive feasibility guarantee. In contrast, stochastic tube MPC is designed to guarantee recursive feasibility. For disturbance-feedback SMPC, it is also possible to provide a recursive feasibility guarantee, e.g., for bounded disturbances [104]. In [116] an SMPC approach is proposed, based on [104, 105], with an additional first step constraint, a terminal constraint, and a tuning parameter enabling to shift priority between performance and a larger feasible region. These measures ensure recursive feasibility, and stability is proved in probability. Mean square stability of SMPC is proved, for example, in [45, 67], and stability based on the expected value is ensured in [13]. Stability of SMPC without a terminal constraint is addressed in [118]. An extensive literature overview of recursive feasibility and stability in SMPC is given in [68, Table 3]

Due to its reduced conservatism, SMPC has been applied to various applications; however, the majority of work only focuses on simulations. The probability of constraint violation, due to the chance constraint, makes SMPC suitable for applications where rare constraint violations are tolerable. Examples are energy control [148], power systems [90, 107], battery control [164] and general automotive applications [14, 15], process control [91, 176], and finance [154]. But even for safety-critical applications, SMPC solutions have been proposed. Most commonly, these are automotive applications, specifically automated driving. Examples are [50, 51, 187] for road vehicles using an analytic chance constraint reformulation, SCMPC [51,53,170] for automated highway driving, [179] for general path planning, and the related method [210] for racing, affine-disturbance feedback SMPC for racing [114] and road vehicles [142], or even nonlinear SMPC for adapting to changing conditions with respect to tire-road interaction [193]. Other safety-critical SMPC applications include, for example, autonomous space maneuvers [125] or microair vehicles [212].

The problem of employing SMPC with chance constraints to safety-critical applications such as automated driving is the non-zero constraint violation probability. Whereas this problem has hardly been addressed for SMPC, recent MPC research has increasingly focused on safety. Combining MPC and control barrier functions has been proposed in [77,208], where control barrier functions handle safety similarly to how Lyapunov functions are used for stability. Whereas these proposed ideas are promising, challenges remain such as guaranteeing recursive feasibility. In [194, 195] an MPC safety filter is proposed for learning-based controllers, which is also suitable for other controllers. Contingency MPC provides an alternate trajectory to potentially problematic events that might occur [2,3]. However, exploiting the advantages of SMPC (planning optimistically with chance constraints) while still guaranteeing safety (satisfying all hard constraints) remains an unaddressed problem.

2.2 Fundamentals on Stochastic MPC

In the following, based on [25], we introduce fundamentals of MPC and SMPC, followed by a simple SMPC simulation example in the next section. We first introduce a system with constraints in Section 2.2.1 and a problem formulation in Section 2.2.2. Then, we introduce the MPC optimal control problem for the undisturbed system in Section 2.2.3, followed by the SMPC formulation for the system subject to uncertainty in Section 2.2.4 including approaches to reformulate chance constraints.

2.2.1 System and Constraints

We consider the discrete-time, linear system

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{E}\boldsymbol{w}_t \tag{2.1}$$

with time step t, state $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$ with the initial state $\boldsymbol{x}_{t_0}$ at $t = t_0$, input $\boldsymbol{u}_t \in \mathbb{R}^{n_{\boldsymbol{u}}}$, and uncertainty $\boldsymbol{w}_t \in \mathcal{W}_t \in \mathbb{R}^{n_{\boldsymbol{w}}}$. The input is bounded by $\boldsymbol{u}_t \in \mathcal{U}_t$ and the state is subject to the constraint $\boldsymbol{x}_t \in \mathcal{X}_t$. The constraint sets may also be time-invariant, i.e., $\mathcal{U}_t = \mathcal{U}$ and $\mathcal{X}_t = \mathcal{X}$.

In this thesis, we consider discrete-time systems since MPC uses discrete-time prediction models. We only specifically state the continuous-time system if necessary from an application perspective. Furthermore, t generally represents time steps since we focus on discrete-time systems; in cases where t denotes time, this is specifically mentioned. This section focuses on linear systems; however, nonlinear systems $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t)$ can also be handled by MPC.

2.2.2 Problem Formulation

We want to minimize an infinite horizon objective function

$$J_\infty = \min \sum_{t=t_0}^{\infty} l\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) \tag{2.2}$$

with stage cost $l\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right)$. This allows us to formulate the following objective.

Objective 2.1. *Based on an initial state $\boldsymbol{x}_{t_0}$ and system* (2.1)*, we aim to maximize a desired control objective, described by minimizing the infinite horizon objective function J_∞, while accounting for input constraints $\mathcal{U}_t$ and state constraints $\mathcal{X}_t$.*

In the following, we briefly discuss how achieving Objective 2.1 is approached by MPC.

2.2.3 Model Predictive Control

First, we look at the undisturbed system with $\boldsymbol{w}_t = \boldsymbol{0}$. We use MPC to tackle the previously stated Objective 2.1. The MPC optimal control problem with finite horizon N and sampling time Δt, solved at each time step t, is given by

$$
\begin{aligned}
&\min_{\boldsymbol{U}_t} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}\right)\right) + V_\mathrm{f}\left(\boldsymbol{x}_{t+N|t}\right) && \text{(2.3a)}\\
&\text{s.t. } \boldsymbol{x}_{t+k+1|t} = \boldsymbol{A}\boldsymbol{x}_{t+k|t} + \boldsymbol{B}\boldsymbol{u}_{t+k|t} && \text{(2.3b)}\\
&\quad \boldsymbol{u}_{t+k|t} \in \mathcal{U}_{t+k|t}, \qquad k \in \mathbb{I}_{0,N-1} && \text{(2.3c)}\\
&\quad \boldsymbol{x}_{t+k|t} \in \mathcal{X}_{t+k|t}, \qquad k \in \mathbb{I}_{1,N} && \text{(2.3d)}\\
&\quad \boldsymbol{x}_{t|t} = \boldsymbol{x}_t && \text{(2.3e)}
\end{aligned}
$$

with input sequence $\boldsymbol{U}_t = \left(\boldsymbol{u}_{t|t}, ..., \boldsymbol{u}_{t+N-1|t}\right)$ and the state sequence $\boldsymbol{X}_t = \left(\boldsymbol{x}_{t+1|t}, ..., \boldsymbol{x}_{t+N|t}\right)$ resulting from (2.3b) where $\boldsymbol{x}_{t+k|t}$ denotes the state prediction after k steps starting at $\boldsymbol{x}_t$. We approximate with $V_\mathrm{f}\left(\boldsymbol{x}_N\right)$ the cost-to-go after N prediction steps. Note that we omit denoting a terminal constraint $\boldsymbol{x}_{t+N|t} \in \mathcal{X}_{\mathrm{f},t+N|t}$ in (2.3) for clarity. The prediction within an MPC optimal control problem is open-loop, whereas repetitively applying the optimized first input $\boldsymbol{u}_t = \boldsymbol{u}^*_{t|t}$ and resolving the optimal control problem at every time step t yields the closed-loop behavior of MPC. The constraint sets may not only change at time steps t, but $\mathcal{U}_{t+k|t}$ and $\mathcal{X}_{t+k|t}$ may also vary between prediction steps k within the MPC optimal control problem. Depending on the stage cost $l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$, potentially including a time-variant reference value $\boldsymbol{x}_{\mathrm{ref},k}$, as well as the constraints $\mathcal{U}_{t+k|t}$ and $\mathcal{X}_{t+k|t}$, the MPC optimal control problem may be time-variant or time-invariant.

Within this thesis, we make notation simplifications to improve readability if suitable. If the time step t, at which the MPC optimal control problem is solved, is irrelevant, we assume $t = 0$. This simplifies the optimal control problem (2.3) and the subscript notation, yielding

$$
\begin{aligned}
&\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right) + V_\mathrm{f}\left(\boldsymbol{x}_N\right) && \text{(2.4a)}\\
&\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k && \text{(2.4b)}\\
&\quad \boldsymbol{u}_k \in \mathcal{U}_k, \qquad k \in \mathbb{I}_{0,N-1} && \text{(2.4c)}\\
&\quad \boldsymbol{x}_k \in \mathcal{X}_k, \qquad k \in \mathbb{I}_{1,N} && \text{(2.4d)}
\end{aligned}
$$

with input sequence $\boldsymbol{U} = \left(\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1}\right)$ and the resulting state sequence $\boldsymbol{X} = \left(\boldsymbol{x}_1, ..., \boldsymbol{x}_N\right)$, where we neglect specifically denoting $\boldsymbol{x}_0 = \boldsymbol{x}_t$. With this simplification, it follows from context whether a subscript integer of an input or state indicates a prediction step k or a time step t. In general, the elements of an input sequence are clear from context; however, we add a subscript if necessary for clarification, e.g., $\boldsymbol{U}_{0:N-1} = \left(\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1}\right)$.

Remark 2.1. *If the MPC optimal control problem is solved using a simultaneous approach instead of a sequential approach, the state sequence is an additional decision variable, i.e.,* (2.4a) *changes to*

$$\min_{\boldsymbol{U},\boldsymbol{X}} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right) + V_{\mathrm{f}}\left(\boldsymbol{x}_N\right). \tag{2.5}$$

In this thesis, both simultaneous and sequential approaches are used to solve MPC optimal control problems. Therefore, we simplify notation by omitting to specifically denote the state sequence $\boldsymbol{X}$ *as a decision variable since this does not affect the underlying theory.*

2.2.4 Stochastic MPC for Systems with Uncertainty

We now consider system (2.1) subject to uncertainty. In SMPC, uncertainty affecting constraints is considered by replacing the state constraint (2.4d) with a chance constraint

$$\Pr(\boldsymbol{x}_k \in \mathcal{X}_k) \geq \beta \tag{2.6}$$

where β is a risk parameter. The constraint (2.4d) is not required to hold always, but only up to a level specified by the predefined risk parameter β. The higher the risk parameter is chosen, the lower the risk allowed.

Remark 2.2. *The risk parameter may also be defined as* $\tilde{\beta} = 1 - \beta$*, where* β *in* (2.6) *is replaced with* $1 - \tilde{\beta}$*. It then holds that increasing the risk parameter* $\tilde{\beta}$ *also increases risk.*

This chance constraint is required to be reformulated in order to be used within the optimal control problem. In the following, we briefly discuss how (2.6) may be reformulated analytically and with the scenario approach, representing two major SMPC directions.

Remark 2.3. *In this thesis, we mostly consider environment uncertainty, affecting the constraint set* $\mathcal{X}_k$*. Therefore, in this section, we focus on chance constraints and chance constraint reformulation. We refer to [68, Section 3.4] for details on how uncertainty may be considered within SMPC objective functions.*

Analytic Chance Constraint Reformulation

An analytic chance constraint reformulation requires an assumption about the uncertainty and the constraint.

Assumption 2.1. *The uncertainty* $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$ *is a zero-mean, normally distributed uncertainty with covariance matrix* $\boldsymbol{\Sigma}_w$*.*

Assumption 2.2. *We consider polytopic constraints* (2.4d)*, i.e.,* $\boldsymbol{x}_k \in \mathcal{X}_k = \left\{ \boldsymbol{x} \,\middle|\, \boldsymbol{g}_j^\top \boldsymbol{x} \leq h_j \right\}$ *with* $\boldsymbol{g}_j \in \mathbb{R}^{n_x}$ *and* $j \in \mathbb{I}_{1,n_{\mathrm{ic}}}$ *with a total of* n_{ic} *inequality constraints.*

In the following, we only consider one inequality constraint for simplicity, i.e., $n_{\mathrm{ic}} = 1$. For normally distributed uncertainties according to Assumptions 2.1 and state constraints according to Assumption 2.2, an error propagation is possible for the prediction horizon. This is achieved by first splitting the system dynamics (2.1) into a deterministic and a probabilistic part

$$\boldsymbol{x}_k = \boldsymbol{z}_k + \boldsymbol{e}_k. \tag{2.7}$$

Furthermore, the input is also split into two parts, a feedback law stabilizing the deterministic system, as well as a new input $\boldsymbol{c}_k$ accounting for uncertainty, yielding

$$\boldsymbol{u}_k = \boldsymbol{K}\boldsymbol{x}_k + \boldsymbol{c}_k \tag{2.8}$$

where $\boldsymbol{K}$ is a stabilizing state feedback matrix.

Splitting the system state results in the new system models

$$\boldsymbol{z}_{k+1} = \boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{z}_k + \boldsymbol{B}\boldsymbol{c}_k \tag{2.9a}$$

$$\boldsymbol{e}_{k+1} = \boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{e}_k + \boldsymbol{E}\boldsymbol{w}_k \tag{2.9b}$$

with $\boldsymbol{A}_{\boldsymbol{K}} = \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$.

Assumption 2.3. *Within an MPC optimal control problem, the initial error $\boldsymbol{e}_0$ and the initial error covariance matrix $\boldsymbol{\Sigma}_0^{\mathrm{e}}$ are known.*

It is often assumed that $\boldsymbol{e}_0 = \boldsymbol{0}$ and $\boldsymbol{\Sigma}_0^{\mathrm{e}} = \boldsymbol{0}$. Given Assumptions 2.1, 2.3, and let $\boldsymbol{e}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_k^{\mathrm{e}}\right)$ with error covariance matrix $\boldsymbol{\Sigma}_k^{\mathrm{e}}$, then it follows with (2.9b) that

$$\boldsymbol{\Sigma}_{k+1}^{\mathrm{e}} = \operatorname{cov}\left(\boldsymbol{e}_{k+1}\right) = \operatorname{cov}\left(\boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{e}_k + \boldsymbol{E}\boldsymbol{w}_k\right) \tag{2.10a}$$

$$= \operatorname{cov}\left(\boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{e}_k\right) + \operatorname{cov}\left(\boldsymbol{E}\boldsymbol{w}_k\right) \tag{2.10b}$$

$$= \boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{\Sigma}_k^{\mathrm{e}}\left(\boldsymbol{A}_{\boldsymbol{K}}\right)^\top + \boldsymbol{E}\boldsymbol{\Sigma}_w\boldsymbol{E}^\top \tag{2.10c}$$

for every prediction step. The error covariance matrix depends on the previous error propagated through the system, as well as the covariance of the uncertainty additionally added at each step. Note the difference between the uncertainty covariance matrix $\boldsymbol{\Sigma}_w$ and the error covariance matrix $\boldsymbol{\Sigma}_k^{\mathrm{e}}$, which is computed for every prediction step k, based on the prediction model and on $\boldsymbol{\Sigma}_w$.

Then, the analytic chance constraint reformulation of (2.6) yields

$$\boldsymbol{g}_k^\top \boldsymbol{z}_k \leq h_k - \gamma_k \tag{2.11a}$$

$$\gamma_k = \sqrt{2\boldsymbol{g}_k^\top \boldsymbol{\Sigma}_k^{\mathrm{e}} \boldsymbol{g}_k}\ \operatorname{erf}^{-1}(2\beta - 1) \tag{2.11b}$$

where $\operatorname{erf}^{-1}(\cdot)$ denotes the inverse error function. In other words, the state constraint is tightened by the tightening parameter γ_k, which itself depends on the risk parameter β and the error covariance matrix $\boldsymbol{\Sigma}_k^{\mathrm{e}}$. Based on Appendix A, details on how to obtain (2.11) are given in Appendix B.

The full SMPC optimal control problem with an analytically reformulated chance constraint is

$$\min_{(\boldsymbol{c}_0,\dots,\boldsymbol{c}_{N-1})} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{z}_k, \boldsymbol{c}_k - \boldsymbol{K}\boldsymbol{x}_k\right)\right) + V_{\mathrm{f}}\left(\boldsymbol{z}_N\right) \tag{2.12a}$$

$$\text{s.t. } \boldsymbol{z}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k + \boldsymbol{B}(\boldsymbol{c}_k - \boldsymbol{K}\boldsymbol{z}_k) \tag{2.12b}$$

$$\boldsymbol{c}_k - \boldsymbol{K}\boldsymbol{z}_k \in \mathcal{U}_k, \qquad k \in \mathbb{I}_{0,N-1} \tag{2.12c}$$

$$\boldsymbol{g}_k^\top \boldsymbol{z}_k \leq h_k - \gamma_k, \qquad k \in \mathbb{I}_{1,N} \tag{2.12d}$$

$$\gamma_k = \sqrt{2\boldsymbol{g}_k^\top \boldsymbol{\Sigma}_k^{\mathrm{e}} \boldsymbol{g}_k}\ \operatorname{erf}^{-1}(2\beta - 1), \ k \in \mathbb{I}_{1,N} \tag{2.12e}$$

where $(\boldsymbol{c}_0, ..., \boldsymbol{c}_{N-1})$ replace the decision variables $(\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1})$.

In general, the risk parameter is bounded by $0.5 \leq \beta < 1$. If the state is exactly on the constraint, given the normal distribution, the probability of violating the constraint in the next step without tightening the constraint is exactly 50 %. Therefore, no constraint tightening corresponds to a risk parameter of $\beta = 0.5$. A risk parameter $\beta = 1$ would guarantee constraint satisfaction. Given the unbounded uncertainty due to the normal distribution, it would follow that $\gamma_k = \infty$ (as $\operatorname{erf}^{-1}(1) = \infty$). Obviously, this is not practical. While it is mathematically possible to choose $\beta < 0.5$, this is not reasonable, as this equals to softening the original hard constraint (constraint softening instead of constraint tightening). The results of this analytic chance constraint reformulation may be generalized using Cantelli's inequality, where we provide details in Appendix B.2.

Scenario Approach for Stochastic MPC

Often, uncertainties do not follow Assumption 2.1 or the uncertainty distribution may not even be known. For these uncertainties, SCMPC is suitable. In the following, based on [171], we briefly summarize how the scenario approach is used in SMPC to reformulate chance constraints.

Assumption 2.4. *Uncertainty realizations $\boldsymbol{w}_t$ are independent and identically distributed (i.i.d.). It is possible to obtain a sufficient number of i.i.d. samples of $\boldsymbol{w}_t$.*

Instead of considering (2.6), here we use

$$\Pr(\boldsymbol{x}_k \notin \mathcal{X}_k) \leq \tilde{\beta} \tag{2.13}$$

where $\tilde{\beta}$ is the permissible upper bound of the constraint violation probability, as mentioned in Remark 2.2.

The idea of the scenario approach is to first draw K samples of the uncertainty and then ensure that the predicted states $\boldsymbol{x}_k$ within the MPC optimal control problem satisfy the constraint $\boldsymbol{x}_k \in \mathcal{X}_k$ for all uncertainty samples. Therefore, a larger sample size K decreases the risk. Hence, the risk parameter $\tilde{\beta}$ is bounded by

$$\Pr(\boldsymbol{x}_k \notin \mathcal{X}_k) \leq \tilde{\beta} \leq \frac{1}{K+1}, \tag{2.14}$$

which allows us to compute the minimum sample size for which (2.13) is ensured, i.e.,

$$K \geq \frac{1}{\tilde{\beta}} - 1. \tag{2.15}$$

Then, the SCMPC optimal control problem is given by

$$\begin{aligned}
\min_{\boldsymbol{U}} & \sum_{\omega=1}^{K} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k^{(\omega)}, \boldsymbol{u}_k\right)\right) + V_\mathrm{f}\left(\boldsymbol{x}_N^{(\omega)}\right) && \text{(2.16a)}\\
\text{s.t. } & \boldsymbol{x}_{k+1}^{(\omega)} = \boldsymbol{A}\boldsymbol{x}_k^{(\omega)} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{E}\boldsymbol{w}_k^{(\omega)} && \text{(2.16b)}\\
& \boldsymbol{u}_k \in \mathcal{U}_k, \qquad k \in \mathbb{I}_{0,N-1} && \text{(2.16c)}\\
& \boldsymbol{x}_k^{(\omega)} \in \mathcal{X}_k, \qquad k \in \mathbb{I}_{1,N},\ \omega \in \mathbb{I}_{1,K} && \text{(2.16d)}
\end{aligned}$$

with sampled scenarios $\boldsymbol{w}_k^{(\omega)}$ affecting the states $\boldsymbol{x}_{k+1}^{(\omega)}$.

This brief presentation of SCMPC only covers the basic idea of the chance constraint reformulation and excludes extensions such as a sample-and-remove strategy [171].

2.3 Fundamental Definitions

In the following, we introduce basic definitions that are important for MPC and are used throughout this thesis. Instead of the linear system (2.1), here, we consider the general nonlinear system

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) \tag{2.17}$$

with $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$, $\boldsymbol{u}_t \in \mathcal{U}$, and $\boldsymbol{w}_t \in \mathcal{W}$.

Invariant Sets

Often, proofs in MPC are based on positively invariant sets. A positively invariant set ensures that once the set is reached by a state, the state remains within this set. The following definitions are based on [17].

Definition 2.1 (Positively Invariant Set)**.** *A set $\mathcal{X}_{\text{pis}}$ is positively invariant for the autonomous, undisturbed system $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{0}, \boldsymbol{0})$ if $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{0}, \boldsymbol{0}) \in \mathcal{X}_{\text{pis}}$ for all $\boldsymbol{x}_t \in \mathcal{X}_{\text{pis}}$.*

Definition 2.2 (Robustly Positively Invariant Set)**.** *A set $\mathcal{X}_{\text{ris}}$ is robustly positively invariant for the autonomous system $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{0}, \boldsymbol{w}_t)$ if $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{0}, \boldsymbol{w}_t) \in \mathcal{X}_{\text{ris}}$ for all $\boldsymbol{x}_t \in \mathcal{X}_{\text{ris}}$ and for all $\boldsymbol{w}_t \in \mathcal{W}$.*

Definition 2.3 (Controlled Invariant Set)**.** *A set $\mathcal{X}_{\text{cis}}$ is controlled invariant for the undisturbed system $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{0})$ if there exists $\boldsymbol{u}_t \in \mathcal{U}$ such that $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{0}) \in \mathcal{X}_{\text{cis}}$ for all $\boldsymbol{x}_t \in \mathcal{X}_{\text{cis}}$.*

Definition 2.4 (Robustly Controlled Invariant Set)**.** *A set $\mathcal{X}_{\text{rcis}}$ is robustly controlled invariant for the system $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t)$ if there exists $\boldsymbol{u}_t \in \mathcal{U}$ such that $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) \in \mathcal{X}_{\text{rcis}}$ for all $\boldsymbol{x}_t \in \mathcal{X}_{\text{rcis}}$ and for all $\boldsymbol{w}_t \in \mathcal{W}$.*

Recursive Feasibility

A fundamental requirement for MPC optimal control problems is recursive feasibility, which is addressed in multiple following chapters.

Definition 2.5 (Recursive Feasibility)**.** *An MPC optimal control problem with input sequence $\boldsymbol{U}_t$ is recursively feasible for system* (2.17) *if the existence of an admissible input sequence $\boldsymbol{U}_t$ implies the existence of an admissible input sequence $\boldsymbol{U}_{t+1}$ for all $t \geq 0$.*

We consider the set of feasible input sequences

$$\mathcal{D}(\boldsymbol{x}_t) = \{\boldsymbol{U}_t \mid \boldsymbol{u}_k \in \mathcal{U}_{\boldsymbol{x},k} \ \forall k \in \mathbb{I}_{0,N-1}\} \tag{2.18}$$

for time step t, where $\mathcal{U}_{\boldsymbol{x},k}$ represents the set of admissible input elements at prediction step k that satisfy all constraints of an MPC optimal control problem. This allows us to summarize the definition of recursive feasibility with

$$\mathcal{D}(\boldsymbol{x}_t) \neq \emptyset \Longrightarrow \mathcal{D}(\boldsymbol{x}_{t+1}) \neq \emptyset \ \ \forall t \geq t_0. \tag{2.19}$$

Note that proving recursive feasibility requires considering the system uncertainty $\boldsymbol{w}_t$. It is not always possible to ensure recursive feasibility, especially in real-world applications due to unexpected uncertainty. Often, recovery strategies are then employed until the MPC optimal control problem becomes feasible again.

Input-to-State Stability

There are various definitions for stability. Here, we use the idea of input-to-state stability, based on [76]. If a system is input-to-state stable (ISS), it is ensured that the system state remains bounded for a bounded disturbance (or system input). In the following, we consider the autonomous system (2.17) with $\boldsymbol{u}_t = \boldsymbol{0}$, which we denote by $\boldsymbol{x}_{t+1} = \boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{w}_t)$.

Definition 2.6 (Input-to-State Stability)**.** *The origin of a system $\boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{w}_t)$ is ISS, with region of attraction $\mathcal{X}_{\text{ris}} \subseteq \mathbb{R}^{n_{\boldsymbol{x}}}$ that contains the origin, if $\mathcal{X}_{\text{ris}}$ is robustly positively invariant and if there exist a continuous function $V : \mathcal{X}_{\text{ris}} \to \mathbb{R}_{\geq 0}$ and functions $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$, $\gamma \in \mathcal{K}$ such that for all $\boldsymbol{x}_t \in \mathcal{X}_{\text{ris}}$ and $\boldsymbol{w}_t \in \mathcal{W}$ it holds that*

$$\alpha_1(||\boldsymbol{x}_t||) \leq V(\boldsymbol{x}_t) \leq \alpha_2(||\boldsymbol{x}_t||), \tag{2.20a}$$

$$V\left(\boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{w}_t)\right) - V(\boldsymbol{x}_t) \leq -\alpha_3(||\boldsymbol{x}_t||) + \gamma(||\boldsymbol{w}_t||). \tag{2.20b}$$

For ISS, function $V(\cdot)$ is called an ISS Lyapunov function. If the origin of a system is ISS, it is guaranteed that the change in V is bounded as long as the uncertainty is bounded. If the uncertainty is zero, the origin of an ISS system is asymptotically stable with region of attraction $\mathcal{X}_{\text{ris}}$.

It is possible to formulate the descent property (2.21) of Definition 2.6 in an alternative way.

Lemma 2.1 (Lipschitz ISS Lyapunov Function [76])**.** *Let $\boldsymbol{f}' : \mathcal{X}_{\text{ris}} \times \mathcal{W} \to \mathbb{R}^{n_{\boldsymbol{x}}}$ be Lipschitz continuous on $\mathcal{X}_{\text{ris}} \times \mathcal{W}$. Let $\mathcal{X}_{\text{ris}} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ contain the origin and be a robustly positively invariant set for the function $\boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{w}_t)$. Let there exist a Lipschitz continuous function $V : \mathcal{X}_{\text{ris}} \to \mathbb{R}_{\geq 0}$ such that for all $\boldsymbol{x}_t \in \mathcal{X}_{\text{ris}}$ it holds that*

$$\alpha_1(||\boldsymbol{x}_t||) \leq V(\boldsymbol{x}_t) \leq \alpha_2(||\boldsymbol{x}_t||), \tag{2.21a}$$

$$V\left(\boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{0})\right) - V(\boldsymbol{x}_t) \leq -\alpha_3(||\boldsymbol{x}_t||) \tag{2.21b}$$

with functions $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$. Then, $V(\cdot)$ is an ISS Lyapunov function and the origin is ISS for system $\boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{w}_t)$ with region of attraction $\mathcal{X}_{\text{ris}}$.

Lemma 2.1 ensures that the origin of a system subject to uncertainty is ISS if the undisturbed system is asymptotically stable and the system is Lipschitz continuous with respect to state $\boldsymbol{x}_t$ and uncertainty $\boldsymbol{w}_t$.

Remark 2.4. *Instead of analyzing input-to-state stability with respect to a disturbance $\boldsymbol{w}_t$, it is also possible to investigate if an origin is ISS with respect to a system input $\boldsymbol{u}_t$.*

2.4 Stochastic MPC - a Simulation Example

After having introduced the basic theoretic concept of SMPC and chance constraint reformulation, we now provide a simple simulation example. We consider the system example described in [116] and use the MPC toolbox of [79]. While this example system is considered here without any physical context, the system represents a linearization of a Buck-Boost DC–DC converter [44, 110].

2.4.1 Simulation Setup

We consider system (2.1) with states $\boldsymbol{x}_t = (x_{1,t},\ x_{2,t})^\top$, input u_t, uncertainty $\boldsymbol{w}_t$, and

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.0075 \\ -0.143 & 0.996 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 4.798 \\ 0.115 \end{bmatrix}, \quad \boldsymbol{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.22}$$

We consider a normally distributed uncertainty $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$ with covariance matrix

$$\boldsymbol{\Sigma}_w = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix}. \tag{2.23}$$

The initial state $\boldsymbol{x}_{t_0} = (-1.3,\ 3.5)^\top$ is given. In addition, we consider input constraints

$$-0.2 \leq u_t \leq 0.2 \quad \forall t \geq t_0 \tag{2.24}$$

and a state constraint bounding the state x_1 by

$$x_{1,t} \leq x_{1,\text{lim}} \quad \forall t \geq t_0 \tag{2.25}$$

where $x_{1,\text{lim}} = 2.8$ is the limit.

For the MPC optimal control problem, we choose

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \quad \boldsymbol{R} = 1, \tag{2.26}$$

a prediction horizon $N = 11$, and a sampling time $\Delta t = 0.1$.

2.4.2 Simulation Results

In the following, we first analyze the autonomous, unconstrained system behavior. Then, we apply MPC to the undisturbed, constrained system. Eventually, we investigate the results when employing SMPC with the disturbed, constrained system.

Autonomous, undisturbed system

First, we look at the autonomous, unconstrained and undisturbed system with $\boldsymbol{u}_t = \boldsymbol{0}$ and $\boldsymbol{w}_t = \boldsymbol{0}$. The system behavior increases the x_1-value drastically before moving towards the origin, as shown in Figure 2.1 (black line).

MPC for the undisturbed system

We again consider the undisturbed system with $\boldsymbol{w}_t = \boldsymbol{0}$. We analyze the system behavior when using the MPC optimal control problem (2.4).

Applying MPC (2.4) with no state constraints, i.e., without (2.4d), results in a curved motion towards the origin where x_1 first increases (beyond the value 2.8), Figure 2.1 (dashed blue line). When considering the state constraint (2.4d), the MPC causes x_1 to increase until the value $x_1 = 2.8$ is reached, Figure 2.1 (blue line). Then, the states move towards the origin along the constraint.

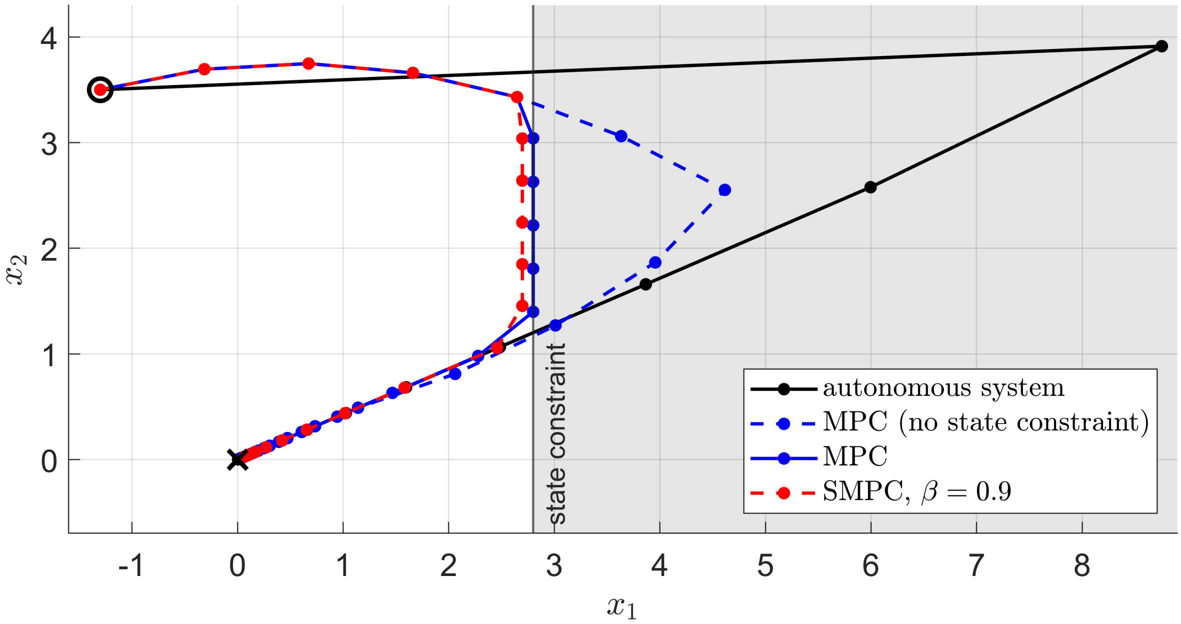


Figure 2.1: Undisturbed system with different controllers.

SMPC for the disturbed system

We now consider system (2.1) subject to uncertainty. Assuming we again used the MPC optimal control problem (2.4), the MPC would not account for uncertainty and the state constraint may be violated frequently. In order to cope with this, the SMPC optimal control problem (2.12) is employed with $\beta = 0.9$, accounting for the uncertainty $\boldsymbol{w}_t$ with the chance constraint

$$\Pr(x_{1,k} \leq 2.8) \geq \beta \tag{2.27}$$

If (2.12) is applied to control the undisturbed system, the states do not converge fully towards the state constraint, but a safety margin is established, providing space to account for uncertainty (the constraint is tightened), as seen in Figure 2.1 (dashed red line). Increasing the risk parameter β increases the margin between the state constraint and the state values, again resulting in a lower probability of constraint violation.

Figure 2.2 shows five resulting runs of applying SMPC to the disturbed system. The tightened constraint allows for satisfaction of the state constraint under most uncertainty realizations (filled red circles); however, as the chance constraint and risk parameter allow for constraint violation, the state constraint is occasionally violated (blank red circles).

In general, SMPC chance constraints affect the open loop constraint satisfaction probability, as the chance constraints are employed within the open loop optimal control problem. If an analytic reformulation of chance constraints is not possible, the closed loop constraint satisfaction probability may differ, often yielding more conservative behavior than specified by the SMPC risk parameter.

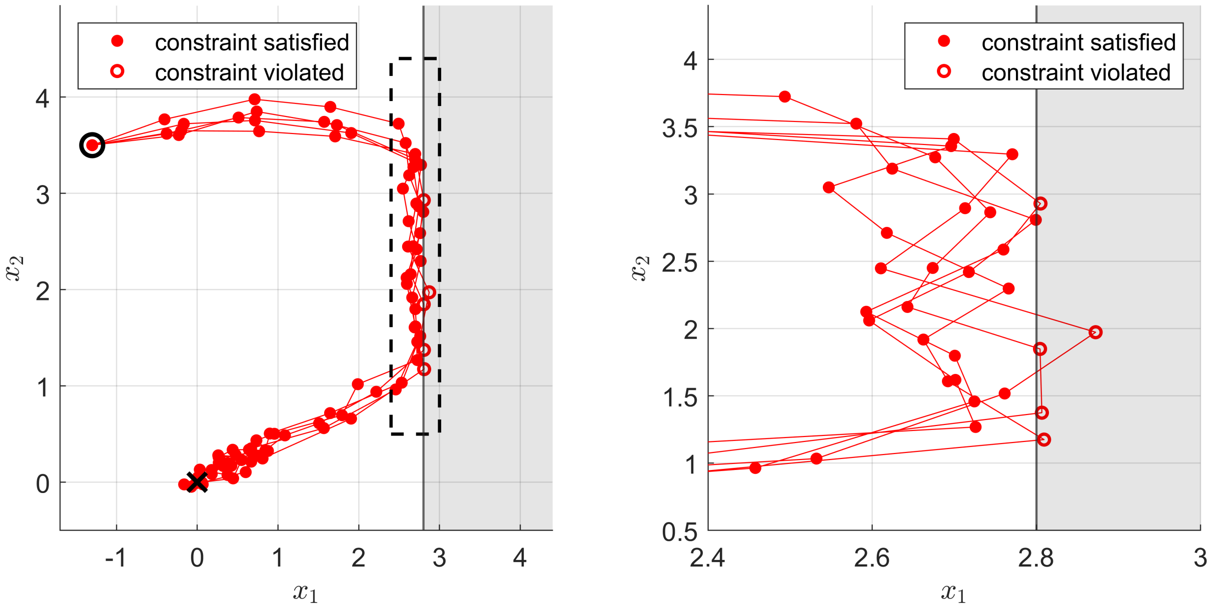


Figure 2.2: Five simulation runs with SMPC applied to the disturbed system.

Advanced Stochastic MPC Approaches to Application-specific Uncertainty

3

This chapter proposes advances to SMPC. On the one hand, we focus on application-specific uncertainty distributions. On the other hand, we propose a novel way of reformulating SMPC chance constraints for applications with a two-dimensional output space, e.g., vehicles in a planar environment. Both SMPC methods are evaluated in automated driving simulations. The content of this chapter was published in [28, 30, 36].

3.1 Introduction

Advances in research on automated systems are facilitating the use of controllers for complex applications, which is especially evident when looking into recent developments for automated vehicles. In many of these applications, there is one controlled agent, e.g., a vehicle or mobile robot, which is required to act and move among other agents. In order to move efficiently and avoid collisions, it is necessary for the controlled agent to anticipate the future behavior of the surrounding agents.

The challenge here is that future behavior of other agents is subject to uncertainty. In many applications, this uncertainty is either a combination of multiple different probability distributions or difficult to describe mathematically in general. As an example, consider automated vehicles where the future motion of other surrounding vehicles is first subject to specific maneuvers, such as lane keeping or lane changing. Second, the execution of these maneuvers may vary again. A lane change may be executed quickly and aggressively, or slowly over a longer period of time. In this case, maneuver and maneuver execution uncertainty must be considered individually such that collision constraints in an optimal control problem are not overly conservative.

Such application uncertainties are challenging to handle for current SMPC approaches, which often either assume normally distributed system uncertainties or do not scale well with a larger system state and when considering an increased number of dynamic obstacles.

In this chapter, we propose two novel SMPC approaches. First, we present a general stochastic + scenario Model Predictive Control (S+SC MPC) algorithm, applicable to a variety of automated systems. We specifically focus on collision avoidance, which requires considering multiple other agents that may perform multiple different tasks. The proposed S+SC MPC approach utilizes an SCMPC approach for task uncertainty and an analytic SMPC approach for task execution uncertainty. Combining these two approaches into a single MPC optimal control problem allows us to efficiently consider the twofold uncertainty structure of many practical applications with task and task execution uncertainty, e.g., automated vehicles [28, 50, 53, 139, 170]. An automated vehicle simulation study illustrates the applicability of the proposed S+SC MPC algorithm.

Second, we present a grid-based SMPC method for trajectory planning in uncertain environments. While it is possible to apply the proposed method to various autonomous systems, here we focus on automated vehicles. The proposed method provides a simple strategy to handle arbitrary uncertainty of future vehicle motion. The computational effort of the SMPC optimal control problem is manageable and the approach scales well with an increased number of surrounding vehicles or other obstacles. We consider a grid for the environment, i.e., the road. For every predicted step, each cell then is assigned a probability value, representing its occupancy probability by an obstacle. All cells with an occupancy probability value larger than a predefined SMPC risk parameter are inadmissible, where the risk parameter works as a threshold. Given the admissible cells, convex admissible state constraints are defined for the optimal control problem. The two main benefits of using a grid-based SMPC approach for autonomous driving, compared to other SMPC approaches, are the following. First, it is not required to generate and consider an individual chance constraint for each vehicle or obstacle considered. The probability grid is generated given all obstacles and then the risk parameter threshold is applied to all cells, yielding a deterministic reformulation of the chance-constrained optimal control problem. This results in an optimal control problem with convex state constraints, which is computationally efficient to solve. However, the stochastic nature of the problem is still accounted for as a probabilistic grid is initially generated. Second, it is not necessary to decide on a most likely behavior of other vehicles, as multiple predicted behavior options with arbitrary probability distribution can be considered. These properties facilitate the application to autonomous driving. The effectiveness of the presented approach is demonstrated in a highway simulation.

In summary, we present the following contributions.

- An SMPC approach for applications with a two-fold uncertainty structure by combining analytic and sampling-based chance constraint reformulations.
- A novel grid-based SMPC approach for chance constraint reformulation.
- Adaptation of the proposed SMPC methods for suitable application in automated driving.

Whereas both proposed methods are evaluated in automated driving simulations, a general application is possible.

3.1.1 Related Work

SMPC approaches [68, 134] provide more efficient solutions compared to RMPC by utilizing probabilistic chance constraints instead of hard constraints. These chance constraints enable increased efficiency by allowing for a small probability of constraint violation, limited by a predefined acceptable risk. Various SMPC methods exist, approximating the chance constraint to obtain a tractable representation that may be solved in an optimal control problem. In general, each SMPC method considers one type of uncertainty within the prediction model.

Analytic SMPC approaches [50, 105, 176] yield an analytic approximation of the chance constraint, but these approaches are mostly restricted to Gaussian uncertainties. In particle-based SMPC [15] and SCMPC [171], samples of the uncertainty are drawn that are then used to approximate the chance constraint. While arbitrary uncertainty distributions are possible, large numbers of samples are required to provide sufficient approximations for some uncertainty distributions, which increases computational complexity. If complex uncertainty structures best describe the system behavior, the chance constraint approximations of these SMPC approaches are not necessarily suitable.

Robots and automated vehicles often use a grid-based representation of the environment, e.g., an occupancy grid (OG). An OG [190] is a mapping grid of the environment where each grid cell is assigned a probability that a certain area is occupied. In [55] and [189] first approaches of OGs for autonomous vehicles have been proposed. In OGs, concepts like objects, pedestrians, and vehicles do not exist and data fusion from multiple sensors is efficient. As summarized in [168], OGs have been developed in many ways with focus on autonomous driving for both highway and urban traffic scenarios, e.g., in [185,186]. Research has been mainly carried out on how to treat data in order to determine a correspondence between sensors and grid cells, and how occupancy probability is assigned and updated with the accumulation of new data. However, the direct use of OGs to account for uncertainty in planning algorithms has received little attention.

3.1.2 Chapter Overview

Section 3.2 presents an SMPC method for systems with a two-fold uncertainty structure. First, we formulate the problem in Section 3.2.1. Then, the method is introduced in Section 3.2.2, followed by a simulation study in Section 3.2.3, and a discussion in Section 3.2.4. Second, we present a novel grid-based SMPC method in Section 3.3. The structure of this section is similar to before. We first state the problem in Section 3.3.1. Afterwards, we develop the method in Section 3.3.2, then show simulation results in Section 3.3.3, and discuss the method in Section 3.3.4. Section 3.4 provides a summary and conclusion of the proposed SMPC advances.

3.2 Exploiting Uncertainty Structures by Combining Stochastic and Scenario MPC

In the following, we present an SMPC approach for systems with a two-fold uncertainty structure: task uncertainty and task execution uncertainty. This is achieved by combining ideas from analytic chance constraint reformulations and sampling-based SMPC. This section is based on the work published in [30,36].

3.2.1 Problem Formulation

MPC for collision avoidance with multiple agents requires two prediction models, one for the controlled agent (CA) and one for the dynamic obstacles (DOs) to be avoided.

We consider the CA dynamics

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) \tag{3.1}$$

depending on the nonlinear function $\boldsymbol{f}$ with state $\boldsymbol{x}_t$ and input $\boldsymbol{u}_t$ at time step t.

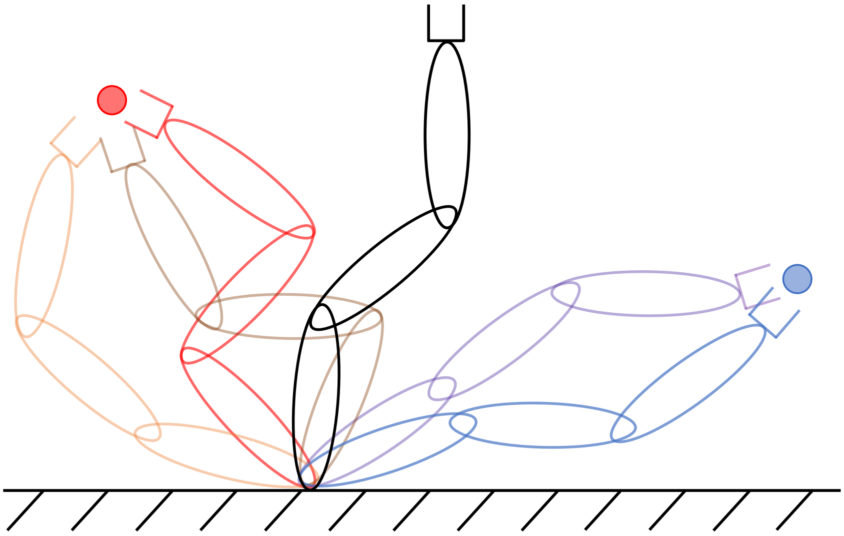


Figure 3.1: Example illustration of a robot arm with two possible task goals (red and blue circles) and different task execution possibilities. Black shows the initial configuration, colors indicate task execution possibilities.

Two types of uncertainties are considered for the DOs: task uncertainty and task execution uncertainty. This distinction reflects the situation of many applications, where the motion of surrounding agents is divided into discrete tasks with multiple task execution possibilities. An example is shown in Figure 3.1, displaying a robotic arm with two possible task goals and different execution strategies.

Definition 3.1 (Tasks)**.** *At each time step, a DO decides to execute exactly one task T_i defined by the task set $\mathcal{T} = \{T_i \mid i = 1, ..., n_\mathcal{T}\}$. Each task T_i is assigned a probability p_i, subject to the probability distribution $P_\mathcal{T}$, where $\sum_{i=1}^{n_\mathcal{T}} p_i = 1$ and $0 < p_1 \leq ... \leq p_{n_\mathcal{T}} \leq 1$. A DO input corresponding to task T_i is denoted by $\boldsymbol{u}^{\mathrm{DO}}(T_i)$.*

Definition 3.2 (Task Execution)**.** *Each task T_i is subject to a nominal motion governed by the DO dynamics, a reference state, and an additive Gaussian uncertainty $\boldsymbol{w}_t^{\mathrm{ex}} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_t^{\mathrm{ex}}\right)$ with covariance matrix $\boldsymbol{\Sigma}_t^{\mathrm{ex}}$, representing uncertainty while executing task T_i.*

We consider multiple DOs. The dynamics for a single DO is then given by

$$\boldsymbol{x}_{t+1}^{\mathrm{DO}} = \boldsymbol{A}^{\mathrm{DO}}\boldsymbol{x}_t^{\mathrm{DO}} + \boldsymbol{B}^{\mathrm{DO}}\boldsymbol{u}_t^{\mathrm{DO}}\left(T_i\right) + \boldsymbol{E}^{\mathrm{DO}}\boldsymbol{w}_t^{\mathrm{ex}} \tag{3.2}$$

with the DO state $\boldsymbol{x}_t^{\mathrm{DO}}$, the input $\boldsymbol{u}_t^{\mathrm{DO}}$ as well as the state and input matrices $\boldsymbol{A}^{\mathrm{DO}}$, $\boldsymbol{B}^{\mathrm{DO}}$, $\boldsymbol{E}^{\mathrm{DO}}$. A DO stabilizing feedback controller is assumed of the form

$$\boldsymbol{u}_t^{\mathrm{DO}}\left(T_i\right) = \boldsymbol{K}^{\mathrm{DO}}\left(\boldsymbol{x}_t^{\mathrm{DO}} - \boldsymbol{x}_{t,\mathrm{ref}}^{\mathrm{DO}}\left(T_i\right)\right) \tag{3.3}$$

with feedback matrix $\boldsymbol{K}^{\mathrm{DO}}$ and a reference state $\boldsymbol{x}_{t,\mathrm{ref}}^{\mathrm{DO}}$ depending on task T_i. The nominal state, assuming zero uncertainty and task T_i, follows

$$\overline{\boldsymbol{x}}_{t+1}^{\mathrm{DO}} = \boldsymbol{A}^{\mathrm{DO}}\overline{\boldsymbol{x}}_t^{\mathrm{DO}} + \boldsymbol{B}^{\mathrm{DO}}\boldsymbol{u}_t^{\mathrm{DO}}\left(T_i\right). \tag{3.4}$$

Collisions with DOs are avoided by determining a set of safe states for the CA.

Definition 3.3 (Safe Set)**.** *The safe set $\mathcal{X}_t^{\text{safe}}$ for time step t ensures that all CA states $\boldsymbol{x}_t \in \mathcal{X}_t^{\text{safe}}$ guarantee collision avoidance at time step t.*

We now formulate the optimal control problem to be solved within this section. Without loss of generality, the SMPC optimal control problem starts at time step 0 where prediction steps are denoted by k. The SMPC optimal control problem is given by

$$\min_{\boldsymbol{U}} J\left(\boldsymbol{x}_0, \boldsymbol{U}\right) \tag{3.5a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{3.5b}$$

$$\boldsymbol{x}_{k+1}^{\text{DO}} = \boldsymbol{A}^{\text{DO}}\boldsymbol{x}_k^{\text{DO}} + \boldsymbol{B}^{\text{DO}}\boldsymbol{u}_k^{\text{DO}}\left(T_i\right) + \boldsymbol{E}^{\text{DO}}\boldsymbol{w}_k^{\text{ex}} \tag{3.5c}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{3.5d}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N} \tag{3.5e}$$

$$\Pr\left(\boldsymbol{x}_k \in \mathcal{X}_k^{\text{safe}}\right) \geq \beta, \qquad k \in \mathbb{I}_{1,N} \tag{3.5f}$$

with $\boldsymbol{U} = (\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1})$, cost function J, horizon N, actuator constraints $\mathcal{U}$, and deterministic state constraints $\mathcal{X}$. As the DO dynamics (3.5c) are subject to uncertainty, the chance constraint (3.5f) is employed for collision avoidance. At each prediction step t, the probability of the CA state $\boldsymbol{x}_k$ lying within the safe set $\mathcal{X}_k^{\text{safe}}$ must be larger than the risk parameter $\beta := \beta\left(\beta^{\text{ta}}, \beta^{\text{ex}}\right)$, $0 \leq \beta \leq 1$. The function $\beta\left(\beta^{\text{ta}}, \beta^{\text{ex}}\right)$ indicates that overall risk parameter β depends on a task uncertainty risk parameter β^{ta} and a task execution uncertainty risk parameter β^{ex}.

It is not possible to directly solve the chance-constrained optimal control problem, which yields the following objective.

Objective 3.1. *In order to allow for efficient trajectory planning, the probabilistic safety constraint* (3.5f) *must be reformulated into a tractable expression, considering the task uncertainty (Definition 3.1) and the task execution uncertainty (Definition 3.2) of multiple dynamic obstacles.*

In the following, a method is derived that approximates the chance constraint (3.5f) to obtain a tractable optimal control problem. We first focus on task uncertainty, followed by task execution uncertainty, which then allows us to consider both uncertainties simultaneously.

3.2.2 Method

In the following, the S+SC MPC framework is derived, using chance constraint reformulation ideas from SCMPC and analytic SMPC approaches. Whereas both of these methods have been introduced before, we provide a novel way to use a combination for application-specific uncertainty. We first individually consider how chance constraints are reformulated into tractable expressions for task uncertainty and task execution uncertainty. Then, the overall MPC optimal control problem is stated, considering both uncertainty types.

Chance Constraint Reformulation based on Scenario MPC for Task Uncertainty

We first focus on task uncertainty. At each time step, one task is performed. The control action, corresponding to different tasks, may significantly vary between different tasks. Therefore, describing task uncertainty with Gaussian noise is impractical, rendering analytic

SMPC approaches inapplicable. Considering every possible task may lead to highly conservative control behavior. However, applying SCMPC-based chance constraint reformulation is a suitable approach to handle task uncertainty. Within SCMPC, task uncertainty is approximated by a small number of samples, as the number of possible tasks is usually small. In this section, no task execution uncertainty is considered, i.e., $\boldsymbol{w}_k^{\text{ex}} = \mathbf{0}$.

Here, a similar SCMPC chance constraint reformulation approach is used as in Section 2.2.4, which is based on [171]. By drawing K samples from the probability distribution $P_{\mathcal{T}}$, the task uncertainty is approximated. A task T_i is assigned to each of the K samples $\omega \in \boldsymbol{\Omega}$. An agent may execute the same task for multiple time steps. However, the agent task may change at every time step.

Assumption 3.1. *Within each MPC optimal control problem, each sampled task is assumed to be executed for the entire prediction horizon.*

In other words, within the prediction, a sampled task is assumed to continue. This assumption is reasonable, as a new optimal control problem with new samples is initiated at each time step, making it possible to account for different agent tasks at consecutive time steps.

If Assumption 3.1 holds, a DO input sequence is obtained for each sample in $\boldsymbol{\Omega}$. The resulting input sequence $\boldsymbol{U}^{\text{DO}}(\omega) = \left(\boldsymbol{u}_0^{\text{DO}}(\omega), ..., \boldsymbol{u}_{N-1}^{\text{DO}}(\omega)\right)$ depends on the individual inputs $\boldsymbol{u}_k^{\text{DO}}(T_i)$, performing task T_i corresponding to sample ω. Based on $\boldsymbol{U}^{\text{DO}}(\omega)$, the predicted DO states for each sample are obtained according to the DO dynamics (3.2) with $\boldsymbol{w}_k^{\text{ex}} = \mathbf{0}$, resulting in the predicted states $\boldsymbol{x}_k^{\text{DO}}(\omega_i)$ for $k = 1, ..., N$.

Depending on the predicted DO states, a safe set $\mathcal{X}_k^{\text{safe}}(\omega_i)$ may be computed for each drawn sample ω. Each safe set, corresponding to a sample ω, delivers an individual constraint in the optimal control problem. Therefore, for the SCMPC approach, the chance constraint (3.5f) is adapted to

$$\Pr\left(\boldsymbol{x}_k \in \mathcal{X}_k^{\text{safe}}(\omega)\right) \geq \beta^{\text{ta}} \quad \forall\, \omega \in \boldsymbol{\Omega}, \quad k \in \mathbb{I}_{1,N}. \tag{3.6}$$

Multiple methods exist to generate safe sets, e.g., signed distance [174] or grid-based methods [28].

The sample size K depends on the chosen risk parameter. We propose a strategy to obtain K that focuses on the least likely task T_1 in $\mathcal{T}$, with probability p_1.

Theorem 3.1. *The sample size K with*

$$K > \log_{1-p_1}\left(\frac{1-\beta^{\text{ta}}}{p_1}\right) \tag{3.7}$$

ensures that the probability of not having sampled the least probable task T_1, if it later occurs, is lower than the allowed risk $1-\beta^{\text{ta}}$, i.e., (3.6) *is satisfied.*

Proof. The proof is based on [30]. Given i.i.d. samples, the worst-case probability of not sampling task T_1, if it later occurs, is given by $p_1(1-p_1)^K$. The sample size K in (3.7) then follows from solving for K with $1-\beta^{\text{ta}} > p_1(1-p_1)^K$, i.e., bounding the worst-case probability given the risk parameter β^{ta}. □

If the least likely task T_1 is actually performed by the DO, this worst-case probability of not having sampled task T_1 is lower than the acceptable risk, defined by the SCMPC risk parameter β^{ta}.

After having introduced an SCMPC approach to handle task uncertainty, the following section introduces an analytic SMPC approximation for task execution uncertainty.

Analytic Chance Constraint reformulation for Task Execution Uncertainty

We now focus on task execution uncertainty, assuming only one task is possible. In the DO dynamics (3.2), task execution uncertainty is described by the additive Gaussian uncertainty, representing uncertainty considering the nominal trajectory of a task. Approximating a Gaussian distribution potentially requires a large number of samples, therefore, an analytic SMPC-based chance constraint reformulation approach is more suitable than an SCMPC-based approach.

Assumption 3.2. *The constraint $\boldsymbol{x}_k \in \mathcal{X}_k^{\mathrm{safe}}$ may be described by a set of functions*

$$\boldsymbol{d}_k\left(\boldsymbol{x}_k, \boldsymbol{x}_k^{\mathrm{DO}}\right) \geq \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{x}_k \in \mathcal{X}_k^{\mathrm{safe}} \tag{3.8}$$

with $\boldsymbol{d}_k = (d_{k,1}, ..., d_{k,n_\mathrm{d}})^\top$, where n_d denotes the number of constraint functions.

In order to find an analytic approximation for the chance constraint (3.5f) with only one task, a linearized description of the chance constraint is required. Therefore, the nonlinear constraint (3.8) is linearized around the nominal states with $\boldsymbol{x}_k^{\mathrm{DO}} = \overline{\boldsymbol{x}}_k^{\mathrm{DO}} + \boldsymbol{e}_k^{\mathrm{DO}}$ and the prediction error $\boldsymbol{e}_k^{\mathrm{DO}}$. Based on $\boldsymbol{w}_k^{\mathrm{ex}}$, the prediction error follows $\boldsymbol{e}_k^{\mathrm{DO}} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_k^{\mathrm{e}}\right)$ where, similar to (2.10),

$$\boldsymbol{\Sigma}_{k+1}^{\mathrm{e}} = \boldsymbol{A}_{\boldsymbol{K}}^{\mathrm{DO}} \boldsymbol{\Sigma}_k^{\mathrm{e}} \boldsymbol{A}_{\boldsymbol{K}}^{\mathrm{DO}\top} + \boldsymbol{E}^{\mathrm{DO}} \boldsymbol{\Sigma}_k^{\mathrm{ex}} \boldsymbol{E}^{\mathrm{DO}\top} \tag{3.9}$$

with $\boldsymbol{A}_{\boldsymbol{K}}^{\mathrm{DO}} = \boldsymbol{A}^{\mathrm{DO}} + \boldsymbol{B}^{\mathrm{DO}} \boldsymbol{K}^{\mathrm{DO}}$.

The resulting linearized description of (3.8) is

$$\boldsymbol{d}_k\left(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k^{\mathrm{DO}}\right) + \nabla \boldsymbol{d}_k^{\mathrm{DO}} \boldsymbol{e}_k^{\mathrm{DO}} \geq \boldsymbol{0} \tag{3.10}$$

with

$$\nabla \boldsymbol{d}_k^{\mathrm{DO}} = \left.\frac{\partial \boldsymbol{d}_k}{\partial \boldsymbol{x}_k^{\mathrm{DO}}}\right|_{\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k^{\mathrm{DO}}}. \tag{3.11}$$

The linearized chance constraint is then given by

$$\Pr\left(\nabla \boldsymbol{d}_k^{\mathrm{DO}} \boldsymbol{e}_k^{\mathrm{DO}} \geq -\boldsymbol{d}_k\left(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k^{\mathrm{DO}}\right)\right) \geq \beta^{\mathrm{ex}}, \tag{3.12}$$

which is still a probabilistic expression. However, (3.12) may be approximated into an analytic expression similar to [30] and as outlined in Section 2.2.4.

Theorem 3.2. *The probabilistic chance constraint* (3.12) *may be approximated by the analytic expression*

$$d_{k,i}\left(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_k^{\mathrm{DO}}\right) \geq \gamma_{k,i} \tag{3.13a}$$

$$\gamma_{k,i} = \sqrt{2 \nabla d_{k,i}^{\mathrm{DO}} \boldsymbol{\Sigma}_k^{\mathrm{e}} \nabla d_{k,i}^{\mathrm{DO}\top}}\, \operatorname{erf}^{-1}\left(1 - 2\beta^{\mathrm{ex}}\right) \tag{3.13b}$$

with $\boldsymbol{\gamma}_k = (\gamma_{k,1}, ..., \gamma_{k,n_\mathrm{d}})^\top$ and $0.5 \leq \beta^{\mathrm{ex}} \leq 1$.

Proof. The proof follows [30, 50]. Due to (3.9) it holds that

$$\nabla \boldsymbol{d}_k^{\mathrm{DO}} \boldsymbol{e}_k^{\mathrm{DO}} \sim \mathcal{N}\left(\mathbf{0}, \nabla d_{k,i}^{\mathrm{DO}} \boldsymbol{\Sigma}_k^{\mathrm{e}} \nabla d_{k,i}^{\mathrm{DO}\top}\right) \tag{3.14}$$

in (3.12). The quantile function for univariate normal distributions allows us to reformulate (3.12) into (3.13). □

Note that $\nabla d_{k,i}^{\mathrm{DO}}$ is defined similar to (3.11). The individual approaches for handling task uncertainty and task execution uncertainty are combined in the following section.

S+SC MPC Optimal Control Problem

The previous results are now combined in order to obtain the S+SC MPC framework, which is able to efficiently handle the mixed uncertainty structure. In addition, multiple DOs are considered with the DO dynamics

$$\boldsymbol{x}_{k+1}^{\mathrm{DO},j} = \boldsymbol{A}^{\mathrm{DO},j} \boldsymbol{x}_k^{\mathrm{DO},j} + \boldsymbol{B}^{\mathrm{DO},j} \boldsymbol{u}_k^{\mathrm{DO},j}\left(T_i^j\right) + \boldsymbol{E}^{\mathrm{DO},j} \boldsymbol{w}_k^{\mathrm{ex},j} \tag{3.15}$$

with stabilizing feedback matrix $\boldsymbol{K}^{\mathrm{DO},j}$ for the DOs $j = 1, \dots, n_{\mathrm{DO}}$.

The tractable S+SC MPC optimal control problem for multiple DOs is then given by

$$\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right) + V_{\mathrm{f}}\left(\boldsymbol{x}_N\right) \tag{3.16a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{3.16b}$$

$$\overline{\boldsymbol{x}}_{k+1,i}^{\mathrm{DO},j} = \boldsymbol{A}^{\mathrm{DO},j} \overline{\boldsymbol{x}}_{k,i}^{\mathrm{DO},j} + \boldsymbol{B}^{\mathrm{DO},j} \boldsymbol{u}_k^{\mathrm{DO},j}\left(\omega^j\right) \tag{3.16c}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{3.16d}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N} \tag{3.16e}$$

$$d_{k,i}^j\left(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_{k,i}^{\mathrm{DO},j}\right) \geq \boldsymbol{\gamma}_{k,i}^j, \qquad k \in \mathbb{I}_{1,N} \tag{3.16f}$$

$$\gamma_{k,i}^j = \sqrt{2 \nabla d_{k,i}^{\mathrm{DO},j} \boldsymbol{\Sigma}_k^{\mathrm{e},j} \nabla d_{k,i}^{\mathrm{DO},j\top}} \operatorname{erf}^{-1}\left(1 - 2\beta^{\mathrm{ex}}\right) \tag{3.16g}$$

with $\omega^j \in \boldsymbol{\Omega}^j$ where the sample size K_j of $\boldsymbol{\Omega}^j$ is determined according to (3.7) for each DO, given the DOs $j = 1, \dots, n_{\mathrm{DO}}$, as well as $i = 1, \dots, n_{\mathrm{d}}$.

In (3.16f), an individual approximated chance constraint is generated for each sample ω, depending on K_j. While this approach is reasonable for a small number of samples, it becomes computationally expensive for larger K_j. A possible alternative for application is to combine similar individual task in order to reduce the number of total constraints. This approach is illustrated in the simulation example in Section 3.2.3.

If it is required to guarantee safety or recursive feasibility, the proposed S+SC MPC method may be extended by the safety framework for SMPC approaches that is explained in detail in Chapter 5.

3.2.3 Simulation Study

To evaluate the effectiveness of the S+SC MPC algorithm presented in Section 3.2.2, a highway scenario involving five target vehicles (TVs) is simulated, using the Control Toolbox [73].

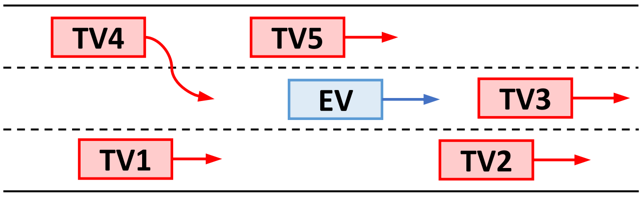


Figure 3.2: S+SC MPC initial scenario configuration. © 2021 IEEE.

Here, the CA and DOs become ego vehicle (EV) and TVs, respectively. The initial vehicle configuration is depicted in Figure 3.2.

We first present the results of the simulation study with the proposed S+SC MPC algorithm, and then, for comparison, we investigate the stand-alone algorithms SMPC and SCMPC. Eventually, we investigate applying S+SC MPC to varying scenario configurations.

Simulation Setup

All simulations are run on an Intel i5-2500K CPU @ 3.30GHz with 15.6GB RAM. Each simulation consists of $N_{\text{sim}} = 100$ MPC iterations, which is equivalent to a scenario duration of 20 s with $\Delta t = 0.2$ s. In the following, SI units are assumed for variables and parameters expressed without units.

As a special case of (3.1), the EV dynamics are represented using the linear, discrete-time point mass model

$$\boldsymbol{x}_{t+1}^{\text{EV}} = \boldsymbol{A}\boldsymbol{x}_t^{\text{EV}} + \boldsymbol{B}\boldsymbol{u}_t^{\text{EV}} \tag{3.17}$$

with the EV states $\boldsymbol{x}_k = (x_k, v_{x,k}, y_k, v_{y,k})^\top$ and inputs $\boldsymbol{u}_k = (u_{x,k}, u_{y,k})^\top$ where

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0.5\Delta t^2 & 0 \\ \Delta t & 0 \\ 0 & 0.5\Delta t^2 \\ 0 & \Delta t \end{bmatrix}. \tag{3.18}$$

The TV dynamics are assumed to be subject to uncertainties. In the case of vehicles, tasks are maneuvers. Therefore, we consider maneuver uncertainty and maneuver execution uncertainty. The TV dynamics are in the form of (3.2) with $\boldsymbol{A}^{\text{DO}}$, $\boldsymbol{B}^{\text{DO}}$, states, and inputs similar to (3.18) as well as $\boldsymbol{E}^{\text{DO}} = \text{diag}\,(0.05,\, 0.067,\, 0.013,\, 0.03)$ accounting for diverse TV uncertainty in longitudinal and lateral direction. The covariance matrix of the normally distributed TV maneuver execution uncertainty $\boldsymbol{w}_t^{\text{ex}} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_t^{\text{ex}}\right)$ is an identity matrix $\boldsymbol{\Sigma}_t^{\text{ex}} = \text{diag}(1,\, 1,\, 1,\, 1)$. Furthermore, additive measurement noise $\boldsymbol{w}_t^{\text{meas}} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_t^{w^{\text{meas}}}\right)$ is considered for x_t^{TV} and y_t^{TV} with $\boldsymbol{\Sigma}_k^{\nu} = \text{diag}\,(0.16,\, 0.01)$. The TVs have multiple maneuver options with associated maneuver probabilities. The possible maneuvers consist of lane changes to left (LCL) and right (LCR), lane keeping (LK), accelerating (AC), braking (BR), and insignificant acceleration (IA), as well as a combination of the lateral and longitudinal maneuvers, resulting in a total of nine possible maneuvers.

The road consists of three lanes with lane width $l_{\text{lane}} = 3.5$ m, where the center of the left lane represents $y = 0$. All vehicles are $l_{\text{veh}} = 6$ m in length and $w_{\text{veh}} = 2$ m in width. The initial lateral position of all vehicles coincides with the lateral center of the vehicles' respective lanes with zero lateral velocity. The initial longitudinal positions and velocities

	EV	TV1	TV2	TV3	TV4	TV5
x-**position** (m)	0	-25	25	40	-30	-10
x-**velocity** ($\mathrm{m\,s^{-1}}$)	27	17	27	27	27	22

Table 3.1: Initial Vehicle Configuration.

of all vehicles are summarized in Table 3.1. The TV reference state is chosen as $\boldsymbol{x}^{\mathrm{TV}}_{\mathrm{ref},k} = \left(0,\ v^{\mathrm{TV}}_{x,\mathrm{ref},k},\ y^{\mathrm{TV}}_{\mathrm{ref},k},\ 0\right)^{\top}$, where $v^{\mathrm{TV}}_{x,\mathrm{ref},k}$ and $y^{\mathrm{TV}}_{\mathrm{ref},k}$ may vary over time depending on the scenario. The feedback controller for the TVs is

$$\boldsymbol{K}^{\mathrm{DO}} = \begin{bmatrix} 0 & -1.0 & 0 & 0 \\ 0 & 0 & -0.8 & -2.2 \end{bmatrix}. \tag{3.19}$$

To prevent collisions, a region around the TV is inadmissible for the EV. This is referred to as the safety constraint, where the admissible area is the safe set $\mathcal{X}^{\mathrm{safe}}_k$. We impose a safety constraint modeled as an ellipse. Its definition adheres to

$$d_k = \frac{(\Delta x_k)^2}{a_{\mathrm{r}}^2} + \frac{(\Delta y_k)^2}{b_{\mathrm{r}}^2} - 1 \geq 0, \tag{3.20}$$

where we decompose the distance between the EV and TV into a longitudinal and a lateral component $\Delta x_k = x^{\mathrm{EV}}_k - \overline{x}^{\mathrm{TV}}_k$ and $\Delta y_k = y^{\mathrm{EV}}_k - \overline{y}^{\mathrm{TV}}_k$.

The ellipse center coincides with the TV center. Therefore, (3.20) is fulfilled if the EV center lies outside the inner space or on the edge of the ellipse, i.e., $d_k \geq 0$. The parameters $a_{\mathrm{r}} = 30$ and $b_{\mathrm{r}} = 2$ represent the semi-major and semi-minor axis of the ellipse, respectively. The values of a_{r} and b_{r} are chosen conservatively, i.e., the area covered by the safety ellipse is larger than the vehicle shape.

To reduce the number of constraints for sampled TV maneuvers, we first introduce a method to adapt the safety constraint ellipse (3.20), generating one ellipse that covers multiple individual safety constraint ellipses. As an example, we assume that all possible maneuvers are sampled. Then, as mentioned as a possibility in Section 3.2.2, we combine the individual constraint ellipses of all sampled maneuvers at each prediction step, as shown in Figure 3.3. If less maneuvers are sampled, the aggregated ellipse only covers the sampled maneuvers.

The result is the aggregated ellipse

$$\tilde{d}_k = \frac{(\Delta \tilde{x}_k)^2}{\tilde{a}_{\mathrm{r}_k}^2} + \frac{(\Delta \tilde{y}_k)^2}{\tilde{b}_{\mathrm{r}_k}^2} - 1 \geq 0, \tag{3.21a}$$

$$\Delta \tilde{x}_k = x_k - \tilde{x}^{\mathrm{TV}}_k, \tag{3.21b}$$

$$\Delta \tilde{y}_k = y_k - \tilde{y}^{\mathrm{TV}}_k, \tag{3.21c}$$

$$\tilde{x}^{\mathrm{TV}}_k = \frac{x^{\mathrm{TV,IA}}_k + x^{\mathrm{TV,BR}}_k + x^{\mathrm{TV,AC}}_k}{3}, \tag{3.21d}$$

$$\tilde{y}^{\mathrm{TV}}_k = \frac{y^{\mathrm{TV,LK}}_k + y^{\mathrm{TV,LCL}}_k + y^{\mathrm{TV,LCR}}_k}{3} \tag{3.21e}$$

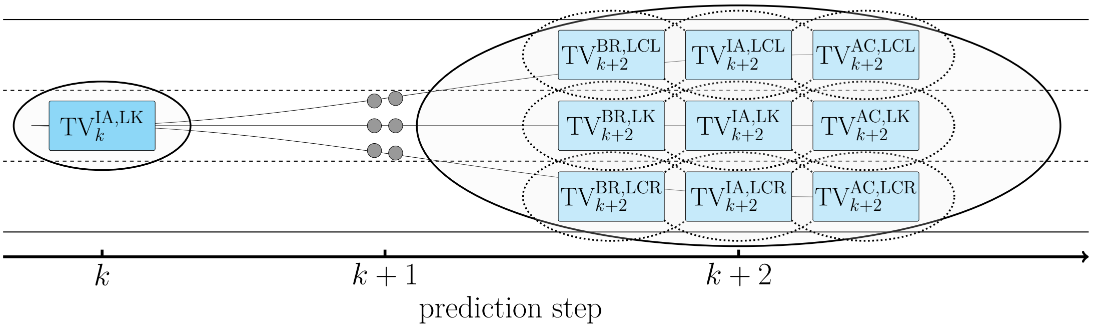

$\mathrm{TV}_k^{\mathrm{IA,LK}}$
$\mathrm{TV}_{k+2}^{\mathrm{BR,LCL}}$
$\mathrm{TV}_{k+2}^{\mathrm{IA,LCL}}$
$\mathrm{TV}_{k+2}^{\mathrm{AC,LCL}}$
$\mathrm{TV}_{k+2}^{\mathrm{BR,LK}}$
$\mathrm{TV}_{k+2}^{\mathrm{IA,LK}}$
$\mathrm{TV}_{k+2}^{\mathrm{AC,LK}}$
$\mathrm{TV}_{k+2}^{\mathrm{BR,LCR}}$
$\mathrm{TV}_{k+2}^{\mathrm{IA,LCR}}$
$\mathrm{TV}_{k+2}^{\mathrm{AC,LCR}}$
k
$k+1$
$k+2$
prediction step

Figure 3.3: Qualitative depiction of the aggregated safety constraint ellipse. Safety ellipses for step $k+1$ omitted. © 2021 IEEE.

with center $\left(\tilde{x}_k^{\mathrm{TV}}, \tilde{y}_k^{\mathrm{TV}}\right)$. The longitudinal and lateral position of the TV corresponding to the respective maneuvers are indicated by the variables $x_k^{\mathrm{TV},M}$ and $y_k^{\mathrm{TV},M}$ where

$$M \in \{\mathrm{IA}, \mathrm{BR}, \mathrm{AC}, \mathrm{LK}, \mathrm{LCL}, \mathrm{LCR}\}\,. \tag{3.22}$$

The aggregated ellipse has the adjusted semi-major and semi-minor axes

$$\tilde{a}_{\mathrm{r},k} = a_\mathrm{r} + 0.5\left|x_k^{\mathrm{TV,\,AC}} - x_k^{\mathrm{TV,\,BR}}\right| + \frac{2}{l_{\mathrm{lane}}}\left(\tilde{b}_{\mathrm{r},k} - b_\mathrm{r}\right), \tag{3.23a}$$

$$\tilde{b}_{\mathrm{r},k} = b_\mathrm{r} + 0.5\left|y_k^{\mathrm{TV,\,LCL}} - y_k^{\mathrm{TV,\,LCR}}\right|. \tag{3.23b}$$

By generating the aggregated safety ellipse, the number of necessary constraints is reduced. As seen in Figure 3.3, the aggregated ellipse does not necessarily cover all individual safety ellipses perfectly, which is still reasonable as the individual safety ellipses are designed larger than necessary.

For the MPC optimal control problem (3.16), a prediction horizon $N = 12$ is selected. The cost function terms are set to $l = \|\Delta \boldsymbol{x}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2$, $V_\mathrm{f} = \|\Delta \boldsymbol{x}_N\|_{\boldsymbol{S}}^2$, with the cost function weights $\boldsymbol{Q}, \boldsymbol{S} \in \mathbb{R}^{4\times 4}$, and $\boldsymbol{R} \in \mathbb{R}^{2\times 2}$, as well as $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{\mathrm{ref},k}$ with reference $\boldsymbol{x}_{\mathrm{ref},k}$. For positional reference tracking in y-direction, the EV reference is set to its current lane center, while $v_{y,\mathrm{ref},k} = 0$ and $v_{x,\mathrm{ref},k} = 27\,\mathrm{m\,s^{-1}}$. For the cost function, the first element of $\Delta \boldsymbol{x}_k$ is neglected, since no reference for x_k is imposed. Here, $\boldsymbol{Q} = \boldsymbol{S} = \mathrm{diag}(0,\ 3,\ 0.5,\ 0.1)$, $\boldsymbol{R} = \mathrm{diag}(1,\ 0.1)$ are selected.

While the maneuver probabilities are scenario specific and different task uncertainty risk parameters β^{ta} are evaluated, the task execution risk parameter is chosen to be $\beta^{\mathrm{ex}} = 0.8$. Apart from the safety constraint, the EV plans its motion subject to the constraints

$$-1.75 \leq \quad y_k \quad \leq 8.75 \tag{3.24}$$

$$-5 \leq \ u_{x,k} \ \leq 5 \tag{3.25}$$

$$-0.5 \leq \ u_{y,k} \ \leq 0.5 \tag{3.26}$$

$$-1 \leq \Delta u_{x,k} \leq 1 \tag{3.27}$$

$$-0.2 \leq \Delta u_{y,k} \leq 0.2 \tag{3.28}$$

with $\Delta u_{x,k} = u_{x,k} - u_{x,k-1}$, $\Delta u_{y,k} = u_{y,k} - u_{y,k-1}$.

Recovery SMPC Optimal Control Problem

As feasibility can become an issue for the MPC problem (3.16), we implement a recovery strategy that relies on an alternative MPC problem. The MPC problem (3.16) is altered by introducing a slack variable σ to soften the chance-constraint. This results in a recovery strategy MPC problem that is solved if the original MPC problem (3.16) becomes infeasible. For the next time step $t+1$, the original MPC (3.16) problem is solved again. The recovery problem is given by

$$\min_{\boldsymbol{U},\sigma} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \lambda \sigma_\lambda \right) + V_\mathrm{f}\left(\boldsymbol{x}_N\right) \tag{3.29a}$$

$$\text{s.t.}\quad \boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{3.29b}$$

$$\overline{\boldsymbol{x}}_{k+1,i}^{\mathrm{DO},j} = \boldsymbol{A}^{\mathrm{DO},j} \overline{\boldsymbol{x}}_{k,i}^{\mathrm{DO},j} + \boldsymbol{B}^{\mathrm{DO},j} \boldsymbol{u}_k^{\mathrm{DO},j}\left(\omega^j\right) \tag{3.29c}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{3.29d}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N} \tag{3.29e}$$

$$d_{k,i}^{j}\left(\boldsymbol{x}_k, \overline{\boldsymbol{x}}_{k,i}^{\mathrm{DO},j}\right) \geq \boldsymbol{\gamma}_{k,i}^{j} - \sigma_\lambda, \quad \sigma_\lambda \geq 0, \qquad k \in \mathbb{I}_{1,N} \tag{3.29f}$$

$$\gamma_{k,i}^{j} = \sqrt{2 \nabla d_{k,i}^{\mathrm{DO},j} \boldsymbol{\Sigma}_k^{\mathrm{e},j} \nabla d_{k,i}^{\mathrm{DO},j^\top}} \operatorname{erf}^{-1}\left(1 - 2\beta_\lambda^{\mathrm{ex}}\right) \tag{3.29g}$$

with the recovery strategy trajectory risk parameter $\beta_\lambda^{\mathrm{ex}} = 0.995$ and the positive slack variable σ_λ that transforms the deterministic chance-constraint (3.16f) into a soft constraint. The scalar slack variable weight $\lambda = 50$ may be chosen depending on the selected ratio between performance and conservatism.

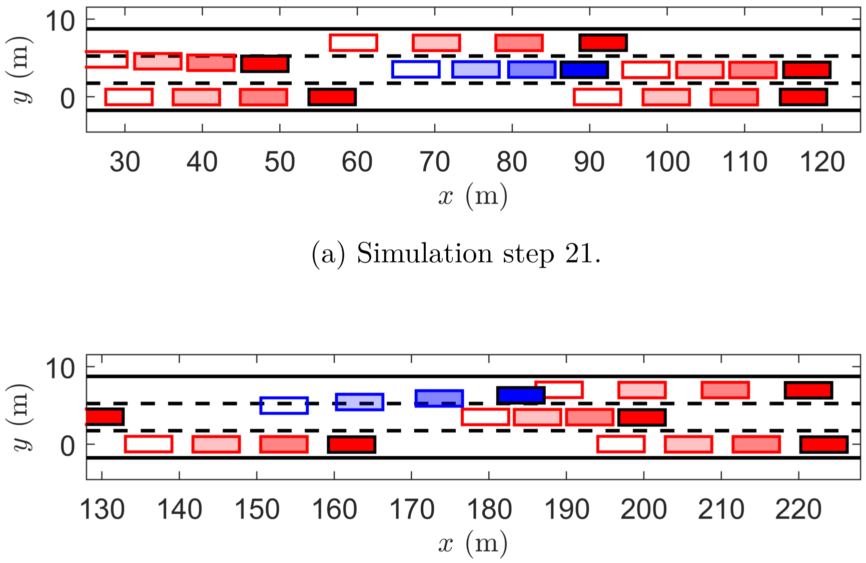
Remark 3.1. *Within the recovery strategy optimal control problem, a different risk parameter $\beta_\lambda^{\mathrm{ex}}$ can be chosen than the risk parameter β^{ex} in the original optimal control problem* (3.16) *to put increased focus on constraint satisfaction.*

In case the recovery problem fails, the solver selects the last feasible point as the solution to the optimal control problem.

Highway Driving with Multiple Target Vehicles

In the following, the S+SC MPC algorithm is evaluated in the previously displayed scenario. As mentioned, the maneuver risk parameter β^{ta} is varied, resulting in a varying sample size K. Monte Carlo simulations are conducted 150 times for each risk parameter value.

Each simulation consists of two parts. For the first 20 steps, the EV follows a conservative behavior with $\beta^{\mathrm{ta}} = 0.999$, representing a behavior prediction initialization phase. The EV assumes that the TV probabilities for lane changes or changes in acceleration are $p^{\mathrm{LC}} = 0.80$ and $p^{\mathrm{AC}} = p^{\mathrm{BR}} = 0.40$. In case a lane change is possible to the left or right, p^{LC} is assigned equally. In the second part from step 21 to step 100, it is assumed that the EV has adapted its behavior prediction. Therefore, the probabilities of TV maneuvers change to $p^{\mathrm{LC}} = 0.20$ and $p^{\mathrm{AC}} = p^{\mathrm{BR}} = 0.10$. For the second part of the simulation, different risk parameters $\beta^{\mathrm{ta}} \in \{0.99, 0.95, 0.89, 0.83\}$ are evaluated. Within the actual simulation, all TVs maintain their respective lanes, except TV4, which moves to the center lane. The reference velocities in x-direction are $v_{x,\mathrm{ref}}^{\mathrm{TV1}} = 22\,\mathrm{m\,s^{-1}}$, $v_{x,\mathrm{ref}}^{\mathrm{TV2}} = 22\,\mathrm{m\,s^{-1}}$, $v_{x,\mathrm{ref}}^{\mathrm{TV3}} = 17\,\mathrm{m\,s^{-1}}$, $v_{x,\mathrm{ref}}^{\mathrm{TV4}} = 17\,\mathrm{m\,s^{-1}}$, $v_{x,\mathrm{ref}}^{\mathrm{TV5}} = 27\,\mathrm{m\,s^{-1}}$.



(a) Simulation step 21.

(b) Simulation step 45.

Figure 3.4: S+SC MPC vehicle motion. The EV is shown in red, TVs in blue. Fading boxes represent past states. © 2021 IEEE.

risk parameter β^{ta}	0.99	0.95	0.89	0.83
closed-loop collisions	0	0	0	0
cost J_{sim}	$3.64 \cdot 10^4$	$3.40 \cdot 10^4$	$3.59 \cdot 10^4$	$3.76 \cdot 10^4$
infeasible OCP steps	26.3	25.2	24.2	26.6
infeasible recovery OCP steps	2.2	3.2	5.2	7.4

Table 3.2: S+SC MPC simulation results.

The result of an individual example with $\beta^{\text{ta}} = 0.95$ is illustrated in Figure 3.4. While there initially is a gap between TV3 and TV5, the EV does not plan to overtake, as a potential lane change of either TV3 or TV5 would result in an inevitable collision. Therefore, the EV slows down such that TV5 passes TV3 first. Subsequently, the EV safely moves to the left lane to overtake TV3 (Figure 3.4b).

Even though SMPC, in general, allows for a small probability of constraint violation, in regular scenarios collisions are avoided as the repetitively updated SMPC inputs allows for constant adjustment. For example, it may not be possible to satisfy the chance constraint for a late prediction step within the SMPC horizon, due to an unexpected uncertainty realization. The optimal control problem is therefore infeasible, i.e., the solver is unable to obtain an admissible input sequence. However, a collision may still be prevented in the next steps, depending on the future uncertainty realizations. Here, we designed a challenging situation for the EV, as lane changes are considered to be probable for all TVs and must be accounted for. The results of the Monte Carlo simulations are shown in Table 3.2.

	SMPC	SCMPC			
risk parameter	0.8	0.99	0.95	0.89	0.83
closed-loop collisions	79	49	43	45	41
cost J_{sim}	$3.22 \cdot 10^4$	$6.77 \cdot 10^4$	$6.27 \cdot 10^4$	$6.88 \cdot 10^4$	$7.08 \cdot 10^4$
infeasible OCP steps	31.2	54.1	53.6	55.7	54.7
infeasible rec. OCP step	21.8	33.8	33.6	34.1	34.3

Table 3.3: SMPC and SCMPC simulation results.

Summarizing the simulation results, the first important observation is that no collisions occurred. While the safety ellipse is slightly violated in some simulation runs, the safety ellipse is chosen large enough that no collisions followed.

The performance is evaluated by computing the cost at each time step, based on the actual states and inputs, with

$$J_{\text{sim}} = \sum_{t=0}^{99} \left(\|\Delta \boldsymbol{x}_{t+1}\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_t\|_{\boldsymbol{R}}^2 \right). \tag{3.30}$$

The cost remains on a similar level for all risk parameters, where the best choice in this scenario is $\beta^{\text{ta}} = 0.95$. Lower risk increases conservatism, while high risk results in less smooth control inputs, again increasing the cost.

As mentioned before, the potential lane changes of all TVs pose a challenging situation for the EV, resulting in steps where the optimal control problem becomes infeasible. However, the steps with successfully solved recovery optimal control problems are significantly more likely, especially for a low accepted level of risk. The average computation time per optimal control problem is 214 ms.

Comparison to SMPC and SCMPC. We now compare the results of S+SC MPC to treating the two-fold uncertainty structure with an analytic SMPC chance constraint reformulation (similar to (2.12)) or an SCMPC-based chance constraint reformulation (similar to (2.16)). The results are shown in Table 3.3. First, an analytic SMPC algorithm, inspired by [50], is analyzed with $\beta^{\text{ex}} = 0.8$. The advantage of S+SC MPC is that the mixed uncertainty structure is exploited. Applying only analytic SMPC, in order to account for maneuver and execution uncertainty, multiple possible maneuvers would need to be approximated by a Gaussian uncertainty. However, this would result in a major increase of the safety ellipse, covering the entire road width, rendering overtaking other TVs impossible. Therefore, in the analytic SMPC simulation, the SMPC chance constraint only accounts for maneuver execution uncertainty for the most likely maneuver.

A total of 79 collisions occurred. While the cost is slightly lower compared to S+SC MPC, significantly more steps with infeasible optimal control problems occur, especially for the recovery problem.

In the SCMPC simulation, inspired by [170], the maneuver execution uncertainty is approximated by samples. To compare a similar situation as in the SMPC simulation, no task uncertainty is considered here. Again, a significant number of simulation runs result in collisions, while the cost also increases compared to S+SC MPC. The steps with infeasible optimal control problems (OCPs) appear more often than in the S+SC MPC simulation runs. The computation times for SMPC and SCMPC are similar to S+SC MPC.

Vehicle settings. So far, only one vehicle setting is considered with fixed initial positions. Therefore, we additionally ran 150 simulations with randomly chosen TV settings for each simulation run (similar initial EV state as before). The TVs get assigned random initial positions $x_0^{\mathrm{TV}} \in [-150, 150]$ and are placed on one of the three lanes, i.e., $y_0 \in \{0, 3.5, 7\}$. The constant longitudinal velocity for each TV is randomly chosen according to $v_x^{\mathrm{TV}} \in [17, 27]$ with $v_y^{\mathrm{TV}} = 0$. It is ensured that all vehicles positioned on similar lanes have a longitudinal distance of at least 50, and velocities are selected such that TV collisions are avoided. The proposed S+SC MPC method successfully handled all 150 simulation runs and no collisions occurred.

Overall, S+SC MPC allows us to exploit the uncertainty structure of the simulation setting, achieving adequate performance and avoiding collisions. While the results presented here are promising, it is to note that the benefits of the proposed method depend on the application setting and to which degree the uncertainty structure may be exploited.

3.2.4 Discussion

In the presented approach there are two risk parameters to be chosen, β^{ta} and β^{ex}. By adjusting the task risk parameter β^{ta}, a trade-off is possible between performance and risk in the presence of multiple agent tasks. A lower value for β^{ta} leads to more samples considered, increasing the probability of having a larger restricted area. As handling uncertainty within the task execution by only using SCMPC would require extensive modeling and sampling to cover all possible cases, a different approach, similar to SMPC with chance-constraints, is taken to ensure a specified level of constraint violation is not exceeded. This leads to the task execution risk parameter β^{ex}, which influences how much risk is accepted for the controlled agent in the presence of dynamic obstacle task execution uncertainty.

As shown in the simulation, it may be useful not to consider each possible task individually. Depending on the application, tasks that are similar can be combined, which reduces the complexity of the optimal control problem. However, this approach comes with two challenges. The first one is that how to combine tasks, and how many tasks should reasonable be combined, is highly application-specific. Second, combining multiple tasks yields an additional trade-off: either the combined constraint over-approximates the individual constraints resulting in more conservative control behavior; or the combined constraint does not fully cover the individual constraints, increasing the probability of constraint violation.

3.3 Grid-based Stochastic MPC

In the previous section, traditional SMPC chance constraint reformulation approaches were used to consider application-specific uncertainty. Here, based on occupancy grids, we provide a new approach to consider prediction uncertainty of surrounding vehicles within an SMPC optimal control problem. Occupancy probabilities are assigned to each grid cell. Then, a threshold parameter (similar to a risk parameter) provides the admissible cells, which are then used for trajectory planning. This section is based on the work published in [28].

3.3.1 Problem Formulation

MPC requires system models for the prediction of futures states within the prediction horizon. We specifically consider trajectory planning for vehicles, where the controlled vehicle is known as the EV and surrounding vehicles as TVs.

We consider a nonlinear EV model

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) \tag{3.31}$$

with EV state $\boldsymbol{x}_t$ and EV input $\boldsymbol{u}_t$. Control constraints are imposed on both the steering angle and the acceleration, i.e., $\boldsymbol{u}_{\text{min}} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\text{max}}$, summarized as $\boldsymbol{u}_t \in \mathcal{U}$. The EV is subject to state constraints $\boldsymbol{x}_t \in \mathcal{X}_t$, such as road restrictions, and specifically safety constraints $\boldsymbol{x}_t \in \mathcal{X}_t^{\text{safe}}$, which ensure collision avoidance with other vehicles.

It is necessary for the EV to predict the future TV motion. It is assumed that the future TV motion is described by a linear discrete-time point-mass model subject to prediction uncertainty

$$\boldsymbol{x}_{t+1}^{\text{TV}} = \boldsymbol{A}\boldsymbol{x}_t^{\text{TV}} + \boldsymbol{B}\boldsymbol{u}_t^{\text{TV}} + \boldsymbol{E}\boldsymbol{w}_t^{\text{TV}} \tag{3.32}$$

where $\boldsymbol{x}_t^{\text{TV}} = \left(x_t^{\text{TV}}, v_{x,t}^{\text{TV}}, y_t^{\text{TV}}, v_{y,t}^{\text{TV}}\right)^\top$ is the TV state at time step t, represented by longitudinal and lateral positions and velocities, and $\boldsymbol{u}_t^{\text{TV}} = \left(u_{x,t}^{\text{TV}}, u_{y,t}^{\text{TV}}\right)^\top$ is the control input consisting of longitudinal and lateral acceleration with the assumed to be known TV reference trajectory $\boldsymbol{x}_{\text{ref},t}^{\text{TV}}$ and feedback law

$$\boldsymbol{u}_t^{\text{TV}} = \boldsymbol{K}\left(\boldsymbol{x}_t^{\text{TV}} - \boldsymbol{x}_{\text{ref},t}^{\text{TV}}\right), \quad \boldsymbol{K} = \begin{bmatrix} 0 & k_{12} & 0 & 0 \\ 0 & 0 & k_{21} & k_{22} \end{bmatrix}. \tag{3.33}$$

The uncertainty in the prediction is taken into account by the random variable $\boldsymbol{w}_t^{\text{TV}}$ and the uncertainty input matrix $\boldsymbol{E}$.

The presented vehicle models are then usable in an MPC optimal control problem to predict the future EV and TV states.

Objective 3.2. *The goal is to develop a grid-based SMPC method, applicable to automated vehicles, where the chance constraint reformulation is based on a grid-based description of the environment, e.g., an occupancy grid for vehicles on a road.*

Focusing on vehicles allows us to develop a grid-based SMPC method more comprehensively. However, TVs can be interpreted as dynamic obstacles in non-vehicle related trajectory planning tasks.

3.3.2 Method

The following presents the main contribution of this section. Surrounding TVs have uncertain behavior. EV safety constraints must therefore consider the stochastic nature of the future TV motion to avoid collisions. Assuming a grid representation of the environment, we derive a grid-based SMPC approach, which enables a simple approach to obtain a reformulated SMPC optimal control problem with tractable constraints.

The idea of the method is the following. First, for each time step of the SMPC prediction horizon, a probabilistic occupancy grid is computed. A probability is assigned to each cell of this grid, indicating the probability that the cell is occupied by a TV. Then, we formulate a probabilistic constraint expression to avoid collisions between the EV and TV. A tractable expression of the probabilistic constraint expression is found by deriving a binary grid in order to clearly identify the admissible road grid cells. The admissible road grid cells are then used to obtain a convex hull in which the EV can operate. Finally, we solve the optimal control problem of the SMPC.

In the following sections, we describe the method in detail, starting with the probabilistic occupancy grid.

Probabilistic Grid

The environment is represented by a grid $\mathcal{C}^{\mathrm{G}}$, which is an evenly spaced field of cells $c_{ij}^{\mathrm{G}} \in \mathcal{C}^{\mathrm{G}}$. Each cell has dimensions l_x and l_y accounting for length and width, respectively, and is identified in a two-dimensional space by two indices i and j. Due to clarity, the cell indices are omitted when irrelevant in the following. The level of approximation depends on the size of the cell.

Probabilistic grid setup. For every prediction step k, an individual probabilistic grid (PG) is generated in order to represent occupancy probabilities of each grid cell. The PG, represented by a matrix $\mathcal{P}^{\mathrm{G}}$, consists of elements p_{ij} that describe the occupancy probability. Note that the PG used here is defined differently compared to standard OG literature.

Remark 3.2. *The probability p_{ij} does not necessarily correspond to the exact probability that cell c_{ij}^{G} is occupied by a TV. This is necessary as the proposed method later considers the vehicle shape and combines grids of multiple individual TVs.*

In the following, we provide a brief example of how cell probabilities may be assigned, based on a Gaussian distribution.

Example 3.1. *In the following, we assume a bivariate Gaussian probability distribution for the TV motion prediction in order to demonstrate the method. The probability density function (PDF) is given by*

$$f_{c^{\mathrm{G,TV}}}\left(c^{\mathrm{G}}\right) = \frac{\exp\left(-\frac{1}{2}\left(c^{\mathrm{G}} - c_k^{\mathrm{G,TV}}\right)^\top \boldsymbol{\Sigma}_k^{-1}\left(c^{\mathrm{G}} - c_k^{\mathrm{G,TV}}\right)\right)}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_k|}} \tag{3.34}$$

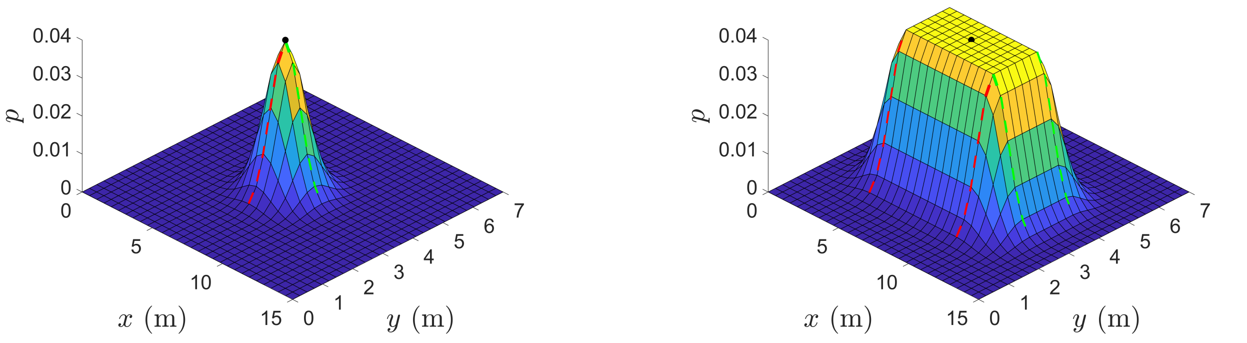
where $c_k^{\mathrm{G,TV}}$ is the cell corresponding to the estimated TV center at prediction step k and $\boldsymbol{\Sigma}_k$ is a covariance matrix.

The covariance matrix $\boldsymbol{\Sigma}_k$ in (3.34) *is obtained by using a recursive technique similar to* (2.10)*, yielding*

$$\boldsymbol{\Sigma}_{k+1} = \left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)\boldsymbol{\Sigma}_k\left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)^\top + \boldsymbol{E}\boldsymbol{\Sigma}_w\boldsymbol{E}^\top \tag{3.35}$$

given the assumed TV prediction model (3.32)*, the initial condition $\boldsymbol{\Sigma}_0 = \boldsymbol{0}$, and the uncertainty covariance matrix $\boldsymbol{\Sigma}_w$.*

Note that it is possible to apply arbitrary probability distributions within the proposed method.



(a) Probability occupancy grid given TV CoG.

(b) Probability occupancy grid taking into account the TV dimensions.

Figure 3.5: Illustration for building the TV probability occupancy grid. The black point represents $c_k^{\mathrm{G,TV}}$. © 2020 IEEE.

Vehicle dimensions. We need to consider the TV dimensions, which we illustrate by continuing Example 3.1. The initially determined probabilities for each cell, according to (3.34), are displayed in Figure 3.5a. The black point represents $c_k^{\mathrm{G,TV}}$, i.e., the center of gravity (CoG) of the TV, the green dashed line represents the normal distribution on the x-p-plane passing through $c_k^{\mathrm{G,TV}}$, while the red dashed line represents the normal distribution on the y-p-plane. Then, we start to expand the maximum value computed with (3.34) along both dimensions x and y in order to cover the vehicle area. The final result is displayed in Figure 3.5b, where the distance between the two green dashed lines equals the width of the vehicle, and the distance between the red dashed lines equals the length. As mentioned in Remark 3.2, Figure 3.5b does not show a PDF, as the grid probabilities were artificially expanded. In the following, the PG is extended for the case of multiple vehicles.

Multiple target vehicles. The presented PG can easily be extended to scenarios with more than one TV by computing a PG for each TV and then adding up the probabilities of the different grids for each cell. If a certain area on the road can potentially be occupied by more than one vehicle in the future, its probability to be occupied increases. This yields

$$p_{ij} = \sum_{n_\mathrm{v}=1}^{n_\mathrm{DO}} p_{ij,n_\mathrm{v}} \tag{3.36}$$

where n_DO is the number of TVs (or dynamic obstacles) localized in the detection range of the EV.

We additionally consider n_M possible TV maneuvers. This is achieved by computing a PG for each maneuver and weigh it with the probability that this maneuver is actually performed, resulting in

$$p_{ij} = \sum_{n_\mathrm{v}=1}^{n_\mathrm{DO}} \sum_{n_\mathrm{m}=1}^{n_\mathrm{M}} p^{\mathrm{m}}_{n_\mathrm{v},n_\mathrm{m}} p_{ij,n_\mathrm{v},n_\mathrm{m}} \tag{3.37}$$

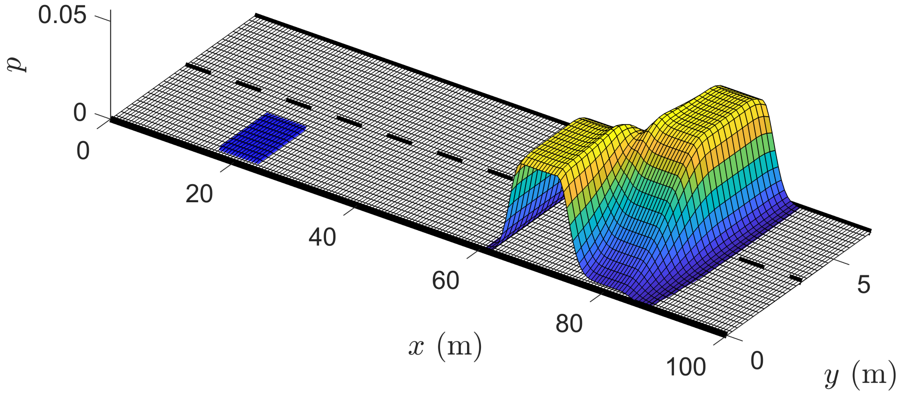


Figure 3.6: Probabilistic Grid of a prediction step. The EV is on the left in blue and two TVs are displayed on the right. © 2020 IEEE.

where $p^{\mathrm{M}}_{n_{\mathrm{v}},n_{\mathrm{m}}}$ is the probability that the maneuver n_{m} is performed by TV n_{v}. A resulting example grid with two TVs is shown in Figure 3.6.

In the following, the PG, considering all surrounding TVs, is adapted to be included in an SMPC optimal control problem.

Binary Grid

SMPC requires deterministic expressions of probabilistic constraints. We achieve this by transforming the probabilistic grid $\mathcal{P}^{\mathrm{G}}$ into a binary grid (BG), represented by a matrix $\mathcal{B}^{\mathrm{G}}$ with elements b_{ij}, by imposing a probability threshold p_{th},

$$b_{ij}= \begin{cases} 1, & \text{if } p_{ij} \geq p_{\mathrm{th}}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.38}$$

The value 1 indicates that a certain cell c^{G}_{ij} is considered to be occupied and therefore inadmissible. The parameter p_{th} is a trade-off between risk and conservatism: the lower the threshold, the more conservative the controller. By transforming the probabilistic grid $\mathcal{P}^{\mathrm{G}}$ into the binary grid $\mathcal{B}^{\mathrm{G}}$, we make a clear distinction between admissible and inadmissible space for the EV.

Remark 3.3. *The threshold parameter p_{th} in the proposed grid-based SMPC replaces the SMPC risk parameter β.*

This makes it possible to determine a hard constraint

$$\boldsymbol{x} \in \mathcal{X}^{\mathrm{adm}}_{k}, \tag{3.39}$$

where $\mathcal{X}^{\mathrm{adm}}_{k}$ includes all admissible cells in the BG. The constraint (3.39) can generally be handled in an optimal control problem.

This approach resembles the use of chance constraints in other SMPC approaches, as cells with low occupancy probability are considered as safe, depending on the threshold p_{th}. We then obtain constraints (3.39) that can be handled by a solver. In the next section, we show how to derive a linear inequality description of the constraint in (3.39).

Convex Hull of the Admissible Cells

In order to obtain a fast SMPC framework, it is not sufficient to define a deterministic constraint reformulation, as seen above, it is also beneficial to find a linear inequality description of the safety constraint. This is achieved by applying Bresenham's line algorithm [22]. Given two points in a grid, which are connected by a straight line, Bresenham's line algorithm yields all cells which are touched by the connecting line.

Remark 3.4. *Even if it was developed in the field of computer graphics to select the bitmaps of an image, Bresenham's algorithm is applicable here - instead of bitmaps, we have grid cells.*

The goal here is to find a convex hull for all admissible cells of the BG, applying Bresenham's algorithm. The convex hull, consisting of valid cells, can be described by linear constraints.

The basic steps of obtaining the convex hull of all admissible cells are shown in Figure 3.7, where the blue cells represent the EV vehicle and the gray cells represent the inadmissible space according to (3.38). The scenario is a straight road with direction of motion along the x-axis. First, valid cells in the EV detection range are determined (Figure 3.7a). Then, using Bresenham's algorithm, it is checked whether the cells on the detection range boundary allow for straight connecting lines to the EV without intersecting occupied cells. Second, the area of valid cells around the EV is enlarged (Figure 3.7b), resulting in a convex hull (Figure 3.7c). The detailed approach is described in Algorithm 1.

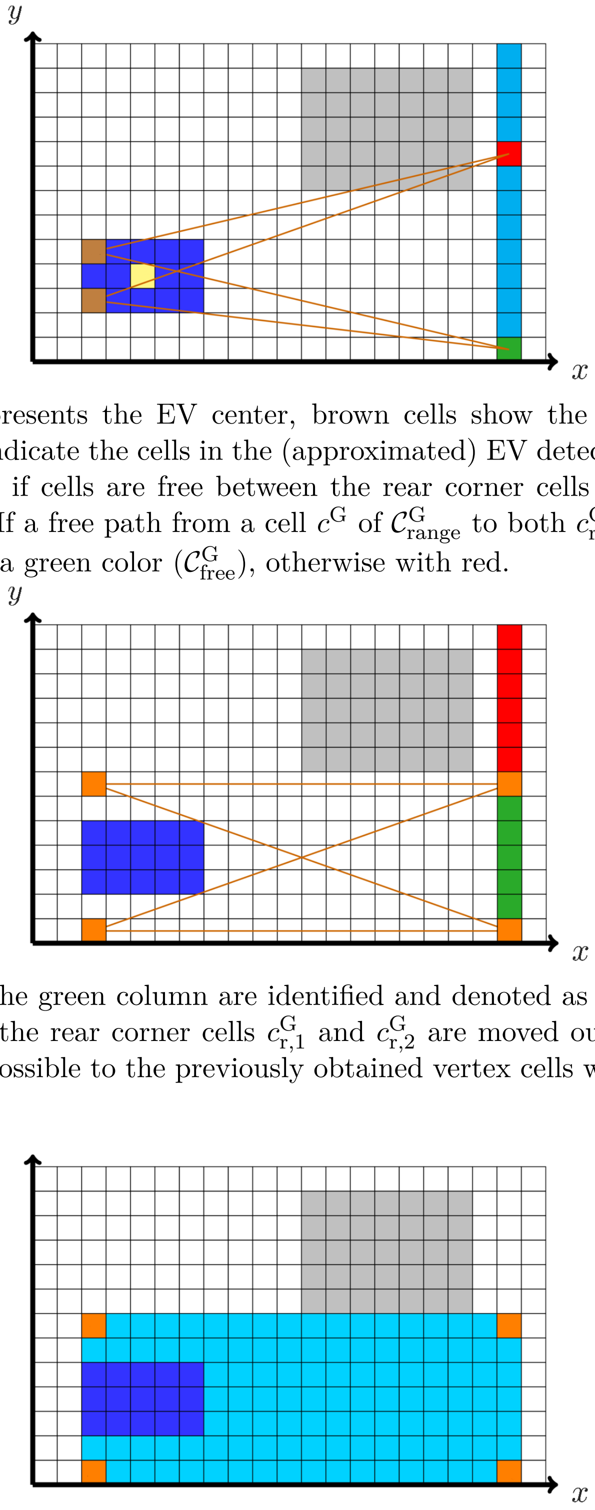
Algorithm 1 Admissible Safe State Space search

```
Input: $\boldsymbol{x}_k$, $\mathcal{B}_k^{\mathrm{G}}$
Identify EV center (Figure 3.7a, yellow) and select rear corners $c_{\mathrm{r},1}^{\mathrm{G}}$ and $c_{\mathrm{r},2}^{\mathrm{G}}$ (Figure 3.7a, brown)
Select admissible cells in the detection range of the EV and store them in a matrix $\mathcal{C}_{\mathrm{range}}^{\mathrm{G}}$ (Figure 3.7a, cyan)
for all cells $c^{\mathrm{G}}$ in $\mathcal{C}_{\mathrm{range}}^{\mathrm{G}}$ do
    Verify free path from $c^{\mathrm{G}}$ to $c_{\mathrm{r},1}^{\mathrm{G}}$, and from $c^{\mathrm{G}}$ to $c_{\mathrm{r},2}^{\mathrm{G}}$
    if free paths are confirmed then
        Store $c^{\mathrm{G}}$ in matrix $\mathcal{C}_{\mathrm{free}}^{\mathrm{G}}$
    end if
end for
Select vertex cells of $\mathcal{C}_{\mathrm{free}}^{\mathrm{G}}$ (Figure 3.7b, orange)
Extend rear corners $c_{\mathrm{r},1}^{\mathrm{G}}$ and $c_{\mathrm{r},2}^{\mathrm{G}}$ if connections are possible to the previously obtained vertex cells without intersecting inadmissible cells (Figure 3.7b, orange)
Obtain convex hull of admissible cells (Figure 3.7c, light blue)
Output: Linear inequality representation (3.40) of convex hull
```

Algorithm 1 is used for each prediction step k of the prediction horizon N. This results in a linear inequality description of a convex hull, i.e.,

$$\boldsymbol{H}_k^{\mathrm{safe}} \boldsymbol{x}_k \leq \boldsymbol{h}_k^{\mathrm{safe}}. \tag{3.40}$$

This linear safety constraint is now included in an MPC optimal control problem. Note that there is no guarantee that an admissible convex hull is found for each prediction step k



(a) The yellow cell represents the EV center, brown cells show the rear corners of the EV, cyan cells ($\mathcal{C}^{\mathrm{G}}_{\mathrm{range}}$) indicate the cells in the (approximated) EV detection range. Bresenham's algorithm evaluates if cells are free between the rear corner cells and the detection range cells (orange line). If a free path from a cell c^{G} of $\mathcal{C}^{\mathrm{G}}_{\mathrm{range}}$ to both $c^{\mathrm{G}}_{\mathrm{r},1}$ and $c^{\mathrm{G}}_{\mathrm{r},2}$ is verified, the cell is marked with a green color ($\mathcal{C}^{\mathrm{G}}_{\mathrm{free}}$), otherwise with red.

(b) The end points of the green column are identified and denoted as vertex cells. To obtain a larger convex hull, the rear corner cells $c^{\mathrm{G}}_{\mathrm{r},1}$ and $c^{\mathrm{G}}_{\mathrm{r},2}$ are moved outside the vehicle bounds if connections are possible to the previously obtained vertex cells without intersecting inadmissible cells.

(c) The light blue area displays the convex hull of admissible cells, the vertex cells are marked with orange cells.

Figure 3.7: Illustration of the steps in Algorithm 1. © 2020 IEEE.

within the prediction horizon N. It is, however, assumed that a convex hull can always be found for $k = 1$. If no admissible convex hull is found at prediction step k, the convex hull of step $k - 1$ is used.

Grid-based Model Predictive Controller

Given the convex hull representing the admissible EV states, we can formulate a tractable SMPC optimal control problem, which is denoted as a standard MPC optimal control problem without prediction uncertainty, as uncertainty is handled within the linear safety constraint. The optimal control problem is given by

$$\begin{aligned}
&\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} \left(\|\Delta \boldsymbol{x}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2 \right) + \|\Delta \boldsymbol{x}_N\|_{\boldsymbol{S}}^2 && \text{(3.41a)}\\
&\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) && \text{(3.41b)}\\
&\qquad \boldsymbol{x}_{k+1}^{\text{TV}} = \boldsymbol{A}\boldsymbol{x}_k^{\text{TV}} + \boldsymbol{B}\boldsymbol{u}_k^{\text{TV}} && \text{(3.41c)}\\
&\qquad \boldsymbol{u}_k^{\text{TV}} = \boldsymbol{K}\left(\boldsymbol{x}_k^{\text{TV}} - \boldsymbol{x}_{\text{ref},k}^{\text{TV}}\right) && \text{(3.41d)}\\
&\qquad \boldsymbol{u}_k \in \mathcal{U}_k, \qquad k \in \mathbb{I}_{0,N-1} && \text{(3.41e)}\\
&\qquad \boldsymbol{x}_k \in \mathcal{X}_k, \qquad k \in \mathbb{I}_{1,N} && \text{(3.41f)}\\
&\qquad \boldsymbol{H}_k^{\text{safe}}\boldsymbol{x}_k \leq \boldsymbol{h}_k^{\text{safe}}, \qquad k \in \mathbb{I}_{1,N} && \text{(3.41g)}
\end{aligned}$$

with the EV input $\boldsymbol{U} = \left(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}\right)^\top$ and $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{\text{ref},k}$ with maneuver-dependent EV reference $\boldsymbol{x}_{\text{ref},k}$, weighting matrices $\boldsymbol{Q}, \boldsymbol{S} \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{R} \in \mathbb{R}^{2\times 2}$, a discrete-time EV prediction model $\boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$, based on (3.31), and the input and state constraints $\mathcal{U}_k$ and $\mathcal{X}_k$. The TV model for the prediction is according to (3.32), (3.33).

In this optimal control problem, the safety constraint (3.39) is formulated as the inequality constraint (3.41g) according to (3.40).

3.3.3 Simulation Study

This SMPC algorithm is now applied to an autonomous driving scenario. In the following, we first provide the general setup that has been used for the simulations. Then, we present the results of a first simulation, a highway scenario where an EV overtakes two TVs, to demonstrate the method. Eventually, we show the results of a second simulation with multiple TVs to analyze the computational cost obtained with the proposed grid-based SMPC method.

Simulation Setup

The presented method has been implemented in MATLAB using the MPC routine developed in [79] as a base implementation. Each vehicle has been modeled as a rectangle with length and width of 6 m and 2 m. Simulations are run with sampling time $\Delta t = 0.2\,\text{s}$. The scenario is a two-lane highway with a lane width $l_{\text{lane}} = 3.5\,\text{m}$. Cell dimensions are $l_x = 0.5\,\text{m}$ and $l_y = 0.25\,\text{m}$.

We consider the continuous-time EV model

$$\dot{x} = v\cos(\psi + \alpha), \tag{3.42a}$$
$$\dot{y} = v\sin(\psi + \alpha), \tag{3.42b}$$
$$\dot{\psi} = \frac{v}{l_{\mathrm{r}}}\sin(\alpha), \tag{3.42c}$$
$$\dot{v} = a, \tag{3.42d}$$
$$\alpha = \arctan\left(\frac{l_{\mathrm{r}}}{l_{\mathrm{r}} + l_{\mathrm{f}}}\tan(\delta_{\mathrm{f}})\right) \tag{3.42e}$$

where x and y represent the longitudinal and lateral position of the vehicle CoG, v and a denote the longitudinal velocity and acceleration, α denotes the body slip angle, δ_{f} is the steering angle of the front wheels, l_{r} and l_{f} are the distances from the vehicle CoG to the rear and front axles, respectively. We denote with $\boldsymbol{x} = (x, y, \psi, v)^\top$ and $\boldsymbol{u} = (\delta_{\mathrm{f}}, a)^\top$ the state and the control input of the EV. Model (3.42) is discretized by the forward-Euler method with sampling time Δt.

The EV employs two simple policies to decide on a reference lane $y_{\mathrm{ref},k}$:

1) if the actual lane is occupied 20 m in front of it, the EV moves to the nearest free lane, otherwise it keeps its actual lane

2) when it passes a TV and the longitudinal distance between their CoGs is larger than 15 m, the EV will overtake the TV by positioning itself in front of the TV.

The EV is subject to the following constraints $\boldsymbol{u} \in \mathcal{U}$,

$$1\,\mathrm{m} \leq y \leq 6\,\mathrm{m}, \tag{3.43a}$$
$$-3\,\mathrm{deg} \leq \delta_{\mathrm{f}} \leq 3\,\mathrm{deg}, \tag{3.43b}$$
$$-5\,\mathrm{m/s^2} \leq a \leq -5\,\mathrm{m/s^2}, \tag{3.43c}$$

in addition to the safety inequality constraints (3.40).

We consider a discrete-time point-mass TV prediction model for (3.32) with

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0.5\,(\Delta t)^2 & 0 \\ \Delta t & 0 \\ 0 & 0.5\,(\Delta t)^2 \\ 0 & \Delta t \end{bmatrix}. \tag{3.44}$$

The selected TV controller matrix values are $(k_{12}, k_{21}, k_{22}) = (-1, -0.8, -2.2)$. We assume Gaussian noise $\boldsymbol{w}_t^{\mathrm{TV}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$ with covariance matrix $\boldsymbol{\Sigma}_w = \mathrm{diag}\,(1,\ 1,\ 1,\ 1)$ and disturbance matrix $\boldsymbol{E} = \mathrm{diag}\,(0.05,\ 0.067,\ 0.013,\ 0.03)$.

Here, only the two most likely TV maneuvers are considered: a lane keeping (LK) and a lane changing (LC) maneuver with constant longitudinal velocity where each of them is weighted with a probability for the respective maneuver being executed. Note that more maneuvers could be considered. Here, we randomly assign a probability in the range of 0.8 to 1 to one of the predicted maneuvers. The second maneuver is given a probability such that the sum equals one.

The SMPC has a prediction horizon $N = 20$, weighting matrices $\boldsymbol{Q} = \mathrm{diag}\,(0,\ 2,\ 0.5,\ 0.1)$ and $\boldsymbol{R} = \mathrm{diag}\,(0.1,\ 1)$, and a probability threshold $p_{\mathrm{th}} = 0.15$. Algorithm 1 is used to find a convex hull at each time step k of the prediction horizon N.

Overtaking Scenario

This scenario consists of a straight two-lane road with two TVs. The center of the right lane is set to 1.75 m, the left lane to 5.25 m. The EV is positioned on the left lane with initial state $\boldsymbol{x} = (10,\ l_{\mathrm{ref}},\ 0,\ 26)$, where $l_{\mathrm{ref}} = 5.25\,\mathrm{m}$. The two TVs start with the initial states $\boldsymbol{x}^{\mathrm{TV}_1} = (40,\ 27,\ 5.25,\ 0)$, and $\boldsymbol{x}^{\mathrm{TV}_2} = (90,\ 27,\ 1.75,\ 0)$. TV_1 is positioned on the left lane, TV_2 on the right one. A probability 0.8 is assigned to the LK maneuver and 0.2 to the LC maneuver for both TVs. In the simulation, the TVs follow the maneuver with the higher probability. Therefore, both TVs will actually proceed along their lane.

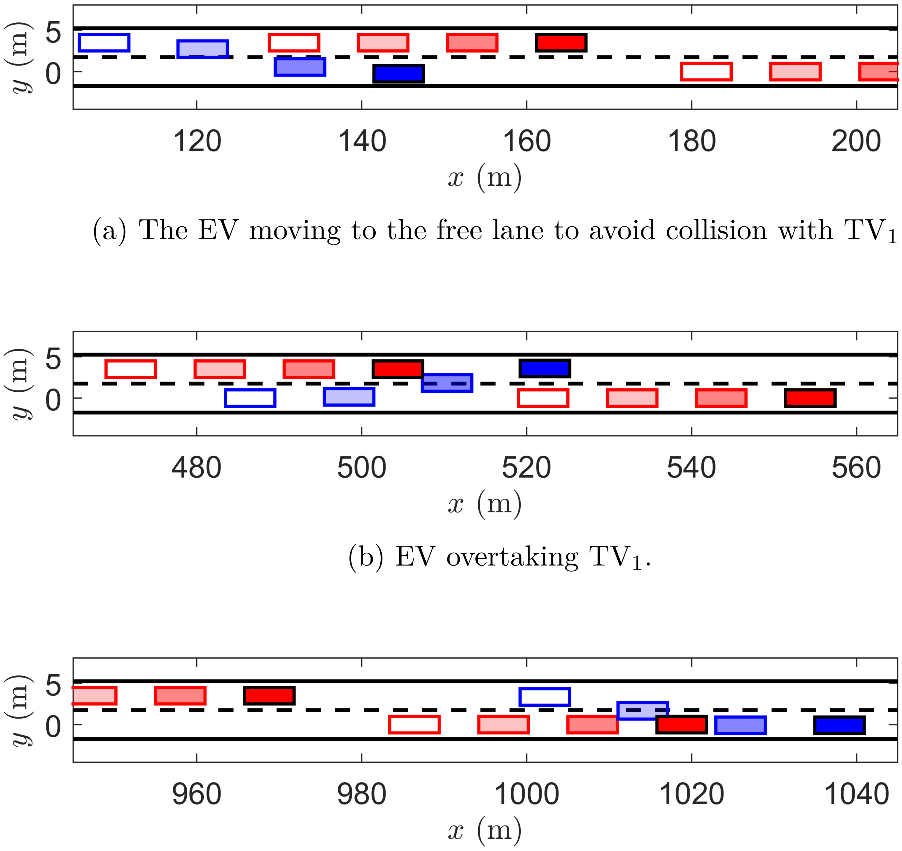
Figures 3.8, 3.9, and 3.10 show the simulation results. Figure 3.8 illustrates the vehicle motion, Figure 3.9 displays the EV velocity and steering angle, and Figure 3.10 shows the distance between the EV and the two TVs. At the beginning, the EV accelerates to reach its reference velocity of 30 m/s, as shown in the first plot of Figure 3.9. When the lane is occupied by TV_1 20 m in front of the EV, the EV starts a LC maneuver, moving to the right lane, as shown in Figure 3.8a. In Figure 3.10a, it can be seen that once the maneuver is completed, the longitudinal distance between EV and TV_1 is approximately 16 m. As soon as it passes TV_1 and the distance between their CoGs is larger than 15 m, the reference lane for the EV changes and the EV starts to move towards the left lane to finish overtaking TV_1. Figure 3.8b shows a sequence of this phase. When the EV has completed the maneuver, i.e., when it lies completely in the left lane, the longitudinal distance with respect to TV_1 is about 17 m. Once it reaches and passes TV_2, the EV performs a new LC maneuver by moving to the right lane again. Once the EV lies on the right lane, the relative distance between the CoGs of the two vehicles is approximately 21 m. This last phase is represented in Figure 3.8c.

For the presented scenario, and with the given policies, we can see that there is no deceleration by the EV while performing LC maneuvers, but only changes in the steering angle δ_{f} and, therefore, in the vehicle heading ψ. Collisions are avoided. As the simulated (not the predicted) TV motion is deterministic in this scenario, one simulation is sufficient for the proposed method. For the presented grid-based SMPC method, additional simulations with identical initialization result in the same behavior. This is in contrast to SMPC methods based on sampling, where the different samples drawn for each simulation change the control behavior of each simulation.

Given this setup and the policy to compute the reference lane for the EV, the MATLAB solver `fmincon` always finds a feasible solution. Therefore, bounds on control signal and space constraints are respected. However, it is important to mention that these results are obtained with the choice $p_{\mathrm{th}} = 0.15$, and no further research has been conducted on different values. The effect of varying risk parameters is studied in other SMPC works, e.g., [30, 50].

Computational Cost Evaluation

To evaluate the computational cost of the algorithm, we use the following more complex setup. The general setup is the one of the overtaking scenario. For each simulation, the EV is randomly positioned on one of the two lanes and a random reference lane is assigned. The same applies to each TV. The first TV is positioned in front of the EV with a longitudinal distance of 40 m. If more TVs are simulated, these are positioned every 50 m. For each TV, one probability is sampled in the range 0.8 to 1 and assigned randomly to one of the two maneuvers, LC or LK, and the second value is assigned to the other maneuver such that the



(a) The EV moving to the free lane to avoid collision with TV_1.

(b) EV overtaking TV_1.

(c) EV overtaking TV_2.

Figure 3.8: Grid-based SMPC vehicle motion during the overtaking maneuvers. Blue boxes represent the EV, red boxes represent TV1 and TV2, respectively. Fading boxes show past states. © 2020 IEEE.

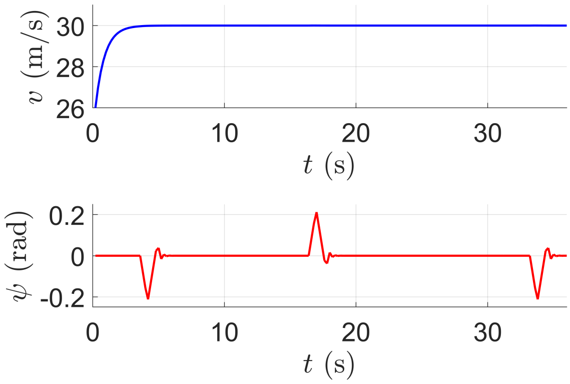
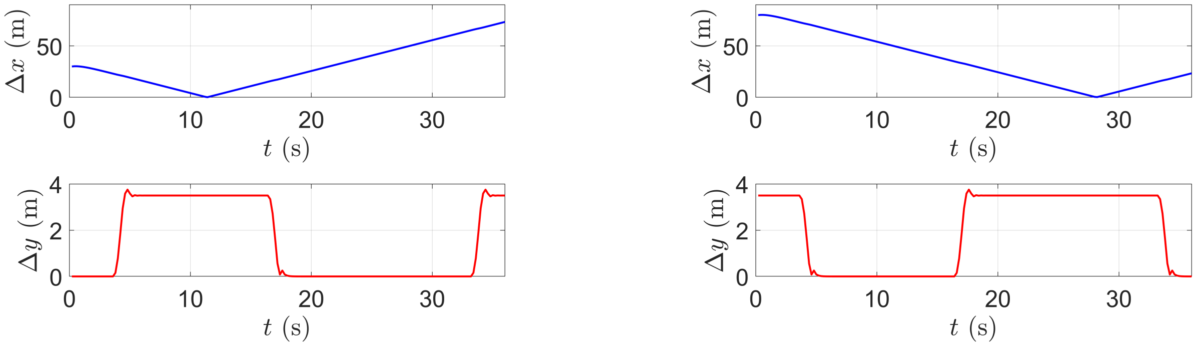


Figure 3.9: Grid-based SMPC EV velocity and steering over time t. © 2020 IEEE.



(a) Distance between EV and TV_1.

(b) Distance between EV and TV_2.

Figure 3.10: Grid-based SMPC longitudinal and lateral distance between EV and TVs in absolute value terms over time t. © 2020 IEEE.

sum of the probabilities equals one. The actual behavior of a TV follows the maneuver with higher probability. We simulated three scenarios with one, two, and three TVs, and each of them has been run 10 times on a standard desktop computer with an Intel i5 processor (3.3 GHz).

Table 3.4 shows the mean μ_{ct} and the standard deviation σ_{ct} of the algorithm computation time per iteration. By increasing the number of TVs in the EV detection range for the scenario, the computation time mean μ_{ct} remains almost constant, while the standard deviation σ_{ct} shows larger variations. The computational cost of the algorithm is mainly due to the complexity of the nonlinear EV model (3.42), and not dependent on the number of TVs in the scenario. The computational effort generating the PG and calculating the convex hull is comparatively small, as this is done prior to solving the optimal control problem. This is a major advantage over for example [30], where the computation time increases significantly with an increasing number of TVs.

# of TVs	μ_{ct} (s)	σ_{ct} (s)
1 TV	0.45	0.48
2 TVs	0.44	0.43
3 TVs	0.42	0.36

Table 3.4: Mean and standard-deviation per algorithm iteration for scenarios with a different number of TVs .

3.3.4 Discussion

In Section 3.3.2, we introduced the probability threshold parameter p_{th}, which allows for transforming the PG into a BG, in order to obtain a deterministic safety constraint expression. This parameter is a trade-off between conservatism and risk. By setting a low value for p_{th}, a high number of cells will be considered occupied. At a certain step, the road can seem fully occupied, resulting in a conservative maneuver for the EV. On the other hand, a high value of p_{th} will reduce the number of occupied cells considered by the algorithm and, at certain step of the prediction horizon, the road can seem free. This can yield a more aggressive controller and potentially a collision between the vehicles.

Therefore, the parameter p_{th} has to be chosen by evaluating a trade-off between conservatism and risk for the planning algorithm, depending on the kind of distribution one decides to adopt. It can also be beneficial to apply a time-varying probability threshold, adapting to different situations. In general, selecting a suitable threshold is challenging, similar to choosing a risk parameter in other SMPC approaches. Note that the risk parameter and the considered probabilities do not perfectly represent true probabilities.

In the simulations, a nonlinear EV model is used to increase the accuracy of the EV predictions, whereas a simpler, linear TV model is used in combination with noise, as the TV behavior is subject to uncertainty. However, applying a linearized EV prediction model allows us to solve a QP problem, given the safety constraint (3.40), a quadratic cost function, as well as linear input and state constraints. This is highly beneficial when a fast algorithm is needed that still considers stochastic behavior of surrounding vehicles.

3.4 Conclusion

The proposed methods in this chapter address application-specific challenges for SMPC, considering different sources and descriptions of uncertainty. First, we propose the S+SC MPC method, which allows us to consider the specific uncertainty structure found in many applications, where both task uncertainty and task execution uncertainty are present. As SCMPC is suitable for non-Gaussian task uncertainty and SMPC copes well with Gaussian execution uncertainty, the combination shows promising results. Whereas in this chapter the S+SC MPC method is applied to a vehicle scenario, the framework is designed in a general way, such that it is applicable also to other applications, e.g., human-robot collaboration. In such robotics settings, a robotic arm may have the option of moving to one of several items, while the exact motion towards the specific item may vary. Without specifically focusing on agents, the proposed framework may also be applicable to process control or finance.

Second, we presented a novel and simple approach to apply SMPC to trajectory planning in uncertain environments, by using a probabilistic grid. This allows for efficient trajectory planning while considering stochastic behavior of surrounding objects. The proposed method scales well with an increasing number of objects considered, here shown for three objects, and can handle arbitrary probability distributions of future object motion. It is still of interest to obtain simulation results for more complex scenarios, as well as combine the proposed approach with occupancy grids. While we applied the proposed method to an automated vehicle example, other applications are possible. It is especially interesting to apply the grid-based SMPC approach to three-dimensional applications, e.g., trajectory planning for robots.

Probabilistic MPC for Extended Prediction Horizons

4

Long prediction horizons and detailed prediction models negatively affect the computational effort of MPC. Here, methods are derived to extend prediction horizons without increasing the computational effort. These methods are based on MPC work using models of different granularity [7] and a non-uniformly spaced optimization horizon [188, 191]. We also consider how system uncertainty may be handled differently for short-term and long-term predictions within the proposed MPC optimal control problems. The content of this chapter was published in [33–35].

4.1 Introduction

When designing an MPC controller, horizon length and model accuracy need to be chosen to fit the control task. Using a precise model over a long prediction horizon generally improves the prediction, but increases the computational effort. Considering uncertainties in the prediction leads to safer control actions, but further increases the computational cost and leads to a more conservative behavior, especially if uncertainty increases over time.

Detailed models provide precise short-term predictions, however, even small model inaccuracies can accumulate over a long prediction horizon, leading to the question how detailed a long-term prediction model needs to be. In certain applications, it is useful to plan precisely for the short-term future while only roughly planning the long-term future. Consider the task of controlling an automated vehicle. Whereas precise planning with a detailed prediction model is fundamental for the immediate future, long-term aims, such as smart lane change decisions, do not require a detailed prediction model. However, accounting for long-term aims is still beneficial, for example switching to the right lane early in dense traffic facilitates a right turn later. This is especially relevant for urban automated driving [12].

In this chapter, we propose two MPC schemes. First, we present an MPC scheme that combines the approaches of [7] and [188, 191] for undisturbed systems. The prediction horizon is divided into two segments: A detailed model with relatively small sampling time is combined with an approximated, coarse model and larger sampling time. This enables using benefits of both individual methods. Computational effort due to model complexity is reduced by using the simplified model for the long-term horizon. Furthermore, the time covered by the prediction horizon is extended by choosing larger sampling times while the amount of decision variables remains constant. Recursive feasibility of the proposed method is guaranteed.

The presented approach is beneficial for tasks requiring precise control for the short-term future, ensured by a small sampling time and a detailed model, where additionally long-term, coarse planning is advantageous. The long-term planning allows us to incorporate long-term goals into short-term planning in such a way that it does not compromise the required short-term precision, i.e., the cost-to-go is improved. We show the effectiveness of the proposed method in a brief collision avoidance simulation.

Second, we focus on systems subject to uncertainty by extending the previously introduced method with robust constraints on the short-term prediction horizon and chance constraints for long-term predictions. For the short term, the overall error of the prediction is reduced by the detailed model, due to a small modeling error and manageable system uncertainty. For long-term predictions, the prediction error increases. This prediction error is increasingly influenced by the propagated system uncertainty, decreasing the benefit of applying a detailed prediction model. Therefore, the less detailed, coarse model is used for long-term predictions to reduce computational complexity. Applying RMPC for the long term would result in a conservative solution of the optimal control problem. Therefore, SMPC with chance constraints is used for the long-term to reduce conservatism, as precise and robust control actions are often not sensible for long-term planning. The proposed method enables robust planning for the immediate future while still considering a longer horizon without overly restrictive solutions due to increased system uncertainty. This approach can be beneficial in safety-critical applications such as autonomous driving, where collision avoidance must be ensured in the immediate future, while considering a longer horizon allows for efficient planning.

The SMPC approach in the long-term horizon is implemented similarly to [116]. This allows us to consider an arbitrarily distributed and bounded disturbance in the coarse model, following from a suitable projection of the disturbance in the detailed model. Furthermore, the SMPC computations caused by the chance constraints are evaluated offline using sampling-based methods or deterministic approaches. The method is then evaluated on a mobile robot control scenario, showing the overall efficient performance. As part of the simulation analysis, the reduction of conservatism compared to RMPC is discussed by examining the tightening of the constraints in detail.

Summarizing, the contributions of this chapter are the following.

- Combination of MPC with models of different granularity and MPC with a non-uniformly spaced horizon for extended MPC prediction horizons.
- Proof of recursive feasibility for an undisturbed system.
- Specific consideration of uncertainty in the short-term and long-term predictions, focusing on safety and efficient planning, respectively.

Here, safety refers to robustly satisfying constraints within the MPC optimal control problem for the short-term horizon.

4.1.1 Related Work

Various approaches have been suggested to tackle the issue of long prediction horizons and model accuracy in MPC. Hierarchical MPC methods [169] use multiple MPC levels with varying complexity. However, the optimal control problems are solved individually, e.g., a high level regulator with slow time scale on a reduced order model and a low level regulator with fast time scale in [69]. Hierarchical MPC schemes are especially popular for chemical applications where different time scales are present.

MPC with move blocking [39, 40, 75, 180] provides an approach to reduce the number of decision variables within the optimal control problem. Regarding input move blocking, certain inputs along the prediction horizon are set equal to previous input values. However, shifting the blocked inputs when solving the optimal control problem is an issue. A flexible move blocking strategy was proposed in [177], adapting the blocking when relevant.

In [7] an MPC scheme is proposed, which uses two different models over the prediction horizon. A detailed model for the short-term horizon is combined with an approximated, coarse prediction model for the long-term horizon. A robust MPC approach [133] is chosen for the long-term horizon to account for model mismatch. While recursive feasibility is guaranteed, stability is not shown. In [206] a real-time iteration scheme for nonlinear MPC is presented, where constraints in the later part of the prediction horizon are replaced by logarithmic barriers.

A different approach to reduce computational effort is presented in [188, 191]. Only a single prediction model is employed over the prediction horizon; however, the sampling time is varied, resulting in an MPC scheme with a non-uniformly spaced horizon (NUSH). The sampling time increases along the prediction horizon, making it possible to extend the time covered by the horizon while keeping the amount of decision variables constant. While stability, based on dissipativity, is shown, recursive feasibility is not addressed. Both MPC with models of different granularity and MPC with a non-uniformly spaced optimization horizon exploit less detailed planning for the long-term future in order to reduce computational complexity. However, both methods only focus on one specific aspect of reducing the computational complexity and do not fully consider system uncertainty.

Figure 4.1 illustrates the different MPC algorithms relevant for this chapter and our contributions. In [34], we adapted the MPC method with models of different granularity of [7] such that RMPC is used for the short-term horizon and SMPC is employed for long-term predictions. In [33], we consider the undisturbed systems and combine the methods of [7] and [188, 191]. Then, in [35] our previous contributions are combined for a modular MPC framework to employ extended prediction horizons.

4.1.2 Chapter Overview

This chapter is structured as follows. Before presenting the contributions, we introduce preliminaries in Section 4.2, addressing MPC with models of different granularity and MPC with a non-uniformly spaced optimization horizon. We present two methods in this chapter. Section 4.3 proposes an MPC method based on models of different granularity and a non-uniformly spaced optimization horizon. After stating the problem formulation in Section 4.3.1, the optimal control problem of this method is shown in Section 4.3.2 including a proof of recursive feasibility. A simulation study and a subsequent discussion are given in Section 4.3.3 and Section 4.3.4, respectively. The first proposed method is then extended to systems subject to uncertainty in Section 4.4. We first formulate the updated problem in Section 4.4.1, then develop the method in Section 4.4.2. Subsequently, we show simulation results in Section 4.4.3 and discuss the findings in Section 4.4.4. We draw conclusive remarks in Section 4.5.

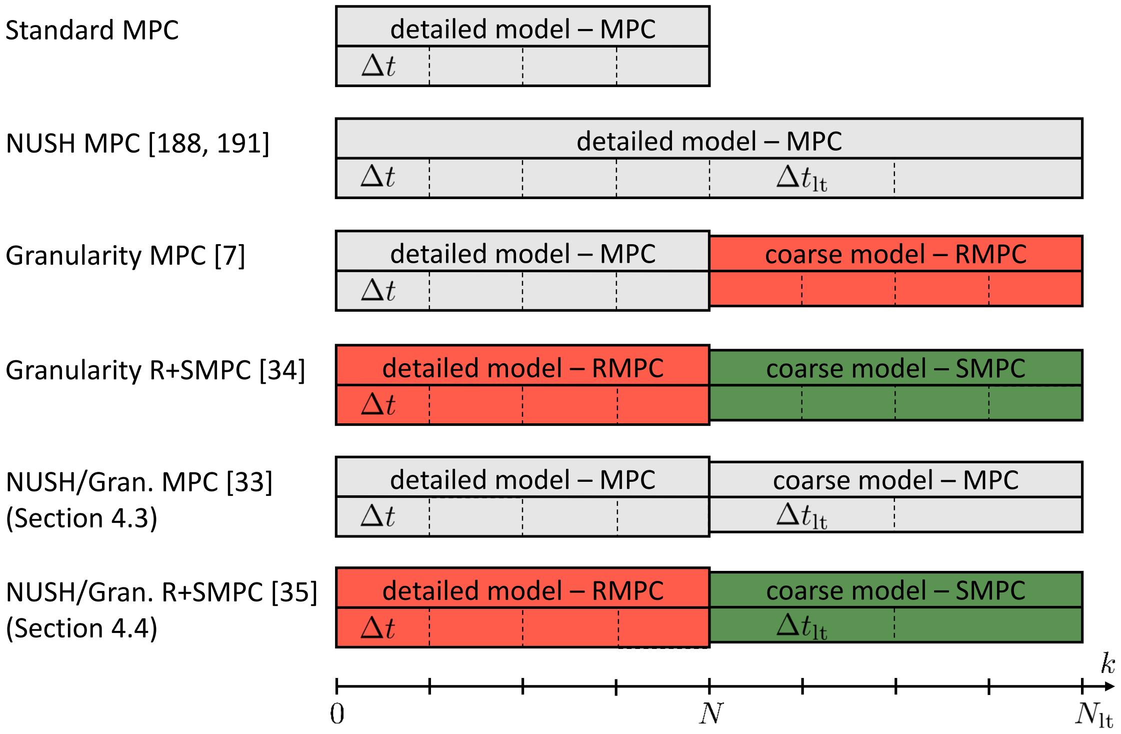


Figure 4.1: Overview of methods for extended MPC horizons with long-term horizon N_{lt} and long-term time step Δt_{lt}.

4.2 Preliminaries

Before presenting the proposed methods, we briefly review MPC with models of different granularity and MPC with a non-uniformly spaced optimization horizon. These preliminaries are simplified versions of [7] and [188].

4.2.1 MPC with Models of Different Granularity

The main idea of MPC with models of different granularity becomes clear when investigating the structure of the MPC optimal control problem

$$\min_{\tilde{\boldsymbol{U}}} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right) + \sum_{k=N}^{N_{\text{lt}}-1} \left(l_{\text{lt}}\left(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k\right)\right) + V_{\text{lt}}\left(\boldsymbol{\xi}_{N_{\text{lt}}}\right) \tag{4.1a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{4.1b}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{4.1c}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N} \tag{4.1d}$$

$$\left(\boldsymbol{\xi}_N, \boldsymbol{\nu}_N\right) = \operatorname{Proj}\left(\boldsymbol{x}_N, \boldsymbol{u}_N\right) \tag{4.1e}$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}_{\text{lt}}\boldsymbol{\xi}_k + \boldsymbol{B}_{\text{lt}}\boldsymbol{\nu}_k \tag{4.1f}$$

$$\boldsymbol{\nu}_k \in \mathcal{U}_{\boldsymbol{\nu}}, \qquad k \in \mathbb{I}_{N,N_{\text{lt}}} \tag{4.1g}$$

$$\boldsymbol{\xi}_k \in \Xi, \qquad k \in \mathbb{I}_{N,N_{\text{lt}}}. \tag{4.1h}$$

In summary, the optimization horizon is split and individual costs, prediction models, and constraints are defined for both the short-term and long-term horizon, i.e., (4.1b) - (4.1d) and (4.1f) - (4.1h), respectively.

The short-term part (4.1b) - (4.1d) is similar to (2.3) and we consider the long-term horizon N_{lt}, the long-term stage cost l_{lt} and terminal cost V_{lt}, the long-term prediction model (4.1f) with input $\boldsymbol{\nu}_k$, state $\boldsymbol{\xi}_k$, system matrix $\boldsymbol{A}_{\mathrm{lt}}$, input matrix $\boldsymbol{B}_{\mathrm{lt}}$, as well as the input constraint set $\mathcal{U}_{\boldsymbol{\nu}}$ and state constraint set Ξ. We denote the overall input sequence as

$$\tilde{\boldsymbol{U}} = (\boldsymbol{u}_0, \dots, \boldsymbol{u}_N, \boldsymbol{\nu}_N, \dots, \boldsymbol{\nu}_{N_{\mathrm{lt}}-1})^\top \tag{4.2}$$

and the state sequence as

$$\tilde{\boldsymbol{X}} = (\boldsymbol{x}_0, ..., \boldsymbol{x}_N, \boldsymbol{\xi}_N, ..., \boldsymbol{\xi}_{N_{\mathrm{lt}}})^\top. \tag{4.3}$$

In general, the coarse long-term prediction model is a simplification of the detailed prediction model. A projection function (4.1e) is used in order to connect these models within the prediction [7].

Assumption 4.1. *(a) There exists a surjective projection function* Proj : $\mathbb{R}^{n_{\boldsymbol{x}}} \times \mathbb{R}^{n_{\boldsymbol{u}}} \rightarrow \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\nu}$*, which maps the detailed model states* $\boldsymbol{x}_k$ *and inputs* $\boldsymbol{u}_k$ *to the coarse model states* $\boldsymbol{\xi}_k$ *and inputs* $\boldsymbol{\nu}_k$*, i.e.,* $(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k) = \mathrm{Proj}\,(\boldsymbol{x}_k, \boldsymbol{u}_k)$.
(b) The projection $(\Xi, \mathcal{U}_{\boldsymbol{\nu}}) = \mathrm{Proj}\,(\mathcal{X}, \mathcal{U})$ *defines the admissible constraint sets.*

This setup allows for the use of a coarse prediction model for long-term predictions, which reduces computational effort. However, a detailed model is still used for short-term predictions. Therefore, accurate predictions are possible, which immediately affect the next applied MPC input, whereas the coarse long-term prediction model allows us to loosely plan the long-term future.

4.2.2 MPC with a Non-uniformly Spaced Prediction Horizon

In [188,191], an approach is presented that uses varying sampling times within the prediction horizon of the MPC scheme. This allows us to extend the prediction horizon time without increasing the number of decision variables or to consider the same horizon time with a reduced number of decision variables. Within an optimal control problem, we achieve the latter by extending the long-term prediction by selecting a different sampling time Δt_{lt} for the long-term prediction horizon, yielding

$$\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t_i\right)\right) + V_{\mathrm{f}}\left(\boldsymbol{x}_N, \Delta t_i\right) \tag{4.4a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{A}(\Delta t_i)\boldsymbol{x}_k + \boldsymbol{B}(\Delta t_i)\boldsymbol{u}_k \tag{4.4b}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{4.4c}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N} \tag{4.4d}$$

with i different sampling time steps Δt_i. Note that the same sampling time step Δt_i can be used for multiple prediction steps. It is often sufficient to only choose two different time steps: Δt for short-term predictions and Δt_{lt} for long-term predictions.

We consider a quadratic cost function

$$l(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t_i) = \boldsymbol{x}_k^\top \boldsymbol{Q}_{\Delta t_i} \boldsymbol{x}_k + \boldsymbol{u}_k^\top \boldsymbol{R}_{\Delta t_i} \boldsymbol{u}_k. \tag{4.5}$$

We need to take into account the different sampling times in the weighting matrices $\boldsymbol{Q}_{\Delta t_i}$ and $\boldsymbol{R}_{\Delta t_i}$ of the quadratic cost functions (4.5). This is achieved by adapting the original weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ for the states and inputs for the long-term prediction to

$$\boldsymbol{Q}_{\Delta t_i} = \frac{\Delta t_i}{\Delta t_0}\boldsymbol{Q}, \quad \boldsymbol{R}_{\Delta t_i} = \frac{\Delta t_i}{\Delta t_0}\boldsymbol{R}. \tag{4.6}$$

This increase of the long-term prediction weighting matrices is necessary to maintain balance in the cost function, as increasing the sampling time decreases the number of steps within a given prediction period.

4.3 MPC with Models of Different Granularity and a Non-uniformly Spaced Prediction Horizon

In the following, we propose an MPC method where the optimization horizon is split into a short-term horizon and long-term horizon. The short-term horizon uses a detailed prediction model and regular sampling time steps. The long-term horizon employs a coarse prediction model and larger sampling time steps. A control invariant set at the last short-term horizon step makes it possible to ensure recursive feasibility of the MPC optimal control problem. This section is based on the work published in [33].

4.3.1 Problem Formulation

Similar to [7], we consider two nonlinear, discrete-time prediction models and constraints

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) && \text{(4.7a)} \\ \text{s.t.} \quad \boldsymbol{x}_k &\in \mathcal{X} && \text{(4.7b)} \\ \boldsymbol{u}_k &\in \mathcal{U} && \text{(4.7c)} \end{aligned} \qquad \begin{aligned} \boldsymbol{\xi}_{k+1} &= \boldsymbol{f}_{\text{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k) && \text{(4.8a)} \\ \text{s.t.} \quad \boldsymbol{\xi}_k &\in \Xi && \text{(4.8b)} \\ \boldsymbol{\nu}_k &\in \mathcal{U}_{\boldsymbol{\nu}} && \text{(4.8c)} \end{aligned}$$

with inputs $\boldsymbol{u}_k \in \mathbb{R}^{n_{\boldsymbol{u}}}$ and $\boldsymbol{\nu}_k \in \mathbb{R}^{n_{\boldsymbol{\nu}}}$, and states $\boldsymbol{x}_k \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and $\boldsymbol{\xi}_k \in \mathbb{R}^{n_{\boldsymbol{\xi}}}$. Here, model (4.8) is considered to be an approximation of model (4.7). The state and input constraints are given by the state and input constraint sets $\mathcal{X}$, $\mathcal{U}$, and Ξ, $\mathcal{U}_{\boldsymbol{\nu}}$, respectively.

We now extend the models (4.7) and (4.8) including different sampling times

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t) && \text{(4.9a)} \\ \text{s.t.} \quad \boldsymbol{x}_k &\in \mathcal{X} && \text{(4.9b)} \\ \boldsymbol{u}_k &\in \mathcal{U} && \text{(4.9c)} \end{aligned} \qquad \begin{aligned} \boldsymbol{\xi}_{k+1} &= \boldsymbol{f}_{\text{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\text{lt}}) && \text{(4.10a)} \\ \text{s.t.} \quad \boldsymbol{\xi}_k &\in \Xi && \text{(4.10b)} \\ \boldsymbol{\nu}_k &\in \mathcal{U}_{\boldsymbol{\nu}} && \text{(4.10c)} \end{aligned}$$

where Δt and Δt_{lt} are the sampling times for the respective prediction models. The two models are linked given a projection function as defined in Assumption 4.1 [7].

Objective 4.1. *The goal is to develop a recursively feasible MPC method that allows us to employ an extended prediction horizon while keeping the computational complexity manageable.*

In order to achieve Objective 4.1, the previously presented approaches, MPC with models of different granularity and MPC with a non-uniformly spaced horizon, are now combined.

4.3.2 Method

In this section, the MPC optimal control problem will be presented, which includes models of different granularity and a non-uniformly spaced prediction horizon. Recursive feasibility of the approach is shown, followed by a discussion.

Given the methods presented in Section 4.2, we combine a detailed model and small sampling time for the short-term horizon with a coarse model and larger sampling time for the long-term horizon. The idea is displayed in Figure 4.1. The MPC optimal control problem is given by

$$\min_{\substack{\{\boldsymbol{u}_0,\ldots,\boldsymbol{u}_N\},\\ \{\boldsymbol{\nu}_N,\ldots,\boldsymbol{\nu}_{N_{\mathrm{lt}}-1}\}}} \sum_{k=0}^{N-1} \left(l(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t)\right) + V_{\mathrm{f}}(\boldsymbol{x}_N) + \sum_{k=N}^{N_{\mathrm{lt}}-1} \left(l_{\mathrm{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\mathrm{lt}})\right) + V_{\mathrm{f,lt}}(\boldsymbol{\xi}_{N_{\mathrm{lt}}}) \tag{4.11a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t) \tag{4.11b}$$

$$\boldsymbol{u}_k \in \mathcal{U}, k \in \mathbb{I}_{0,N} \tag{4.11c}$$

$$\boldsymbol{x}_k \in \mathcal{X}, k \in \mathbb{I}_{1,N-1} \tag{4.11d}$$

$$\boldsymbol{x}_N \in \mathcal{X}_{\mathrm{f}} \subseteq \mathcal{X} \tag{4.11e}$$

$$(\boldsymbol{\xi}_N, \boldsymbol{\nu}_N) = \mathrm{Proj}\,(\boldsymbol{x}_N, \boldsymbol{u}_N) \subseteq \Xi \times \mathcal{U}_{\boldsymbol{\nu}} \tag{4.11f}$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{f}_{\mathrm{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\mathrm{lt}}) \tag{4.11g}$$

$$\boldsymbol{\nu}_k \in \mathcal{U}_{\boldsymbol{\nu}}, k \in \mathbb{I}_{N,N_{\mathrm{lt}}-1} \tag{4.11h}$$

$$\boldsymbol{\xi}_k \in \Xi,\ k \in \mathbb{I}_{N+1,N_{\mathrm{lt}}} \tag{4.11i}$$

with the standard sampling time Δt and a larger sampling time $\Delta t_{\mathrm{lt}} > \Delta t$, as well as a control invariant set $\mathcal{X}_{\mathrm{f}}$. The input $\boldsymbol{u}_N$ is necessary to evaluate (4.11f).

The stage costs $l(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t)$ and $l_{\mathrm{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\mathrm{lt}})$ depend on the respective sampling time. For a quadratic cost function, a similar approach to [188], presented in Section 4.2, can be applied to adapt the weighting matrices, based on Δt and Δt_{lt}. The terminal cost functions are given by V_{f} and $V_{\mathrm{f,lt}}$.

Remark 4.1. *Ideally, the constraint sets Ξ and $\mathcal{U}_{\boldsymbol{\nu}}$ of the coarse model are computed using the projection function $(\Xi, \mathcal{U}_{\boldsymbol{\nu}}) = \mathrm{Proj}(\mathcal{X}, \mathcal{U})$. If the main focus is to improve the cost-to-go and feasibility issues occur, the constraint sets Ξ and $\mathcal{U}_{\boldsymbol{\nu}}$ may be chosen to depend more loosely on $\mathcal{X}$ and $\mathcal{U}$.*

The proposed MPC scheme allows us to apply an accurate prediction model with small sampling time for precise short-term predictions, while still considering long-term aims, with a less accurate long-term prediction. In the following, recursive feasibility of the MPC scheme is shown.

Recursive feasibility. Here, it is not possible to apply standard MPC theory to prove recursive feasibility, e.g., shifting the previous input sequence and a control invariant terminal constraint, as the sampling time changes for the long-term horizon, i.e., $\Delta t < \Delta t_{\mathrm{lt}}$. In the following, we specifically denote the time step t.

We use Definition 2.5 with the input sequence

$$\tilde{\boldsymbol{U}}_t = (\boldsymbol{u}_{t|t}, , ..., \boldsymbol{u}_{t+N|t}, \boldsymbol{\nu}_{t+N|t}, ..., \boldsymbol{\nu}_{t+(N_{\mathrm{lt}}-1)|t}) \tag{4.12}$$

for the following proof of recursive feasibility. To prove recursive feasibility, the initial optimal control problem must be feasible and $\mathcal{X}_{\mathrm{f}}$ must be control invariant.

Assumption 4.2. *The optimal control problem* (4.11) *is initially feasible, i.e., a solution* $\tilde{\boldsymbol{U}}_t$ *to* (4.11)*, according to* (4.12)*, exists for* $t = 0$.

Assumption 4.3. *The terminal set* $\mathcal{X}_\mathrm{f}$ *is control invariant according to Definition 2.4.*

Additionally, the difference in sampling time must be considered.

Assumption 4.4. *A control invariant set* Ξ_cis *can be obtained, such that for all* $\boldsymbol{x}_k \in \mathcal{X}_\mathrm{f}$*, it follows that* $\boldsymbol{\xi}_k \in \Xi_\mathrm{cis}$.

This assumption implicates the following. If a control invariant set for a state $\boldsymbol{x}_k$ with model (4.9) exists, a control invariant set also exists for the corresponding state $\boldsymbol{\xi}_k$ with model (4.10), given the different sampling time.

Theorem 4.1. *Let Assumptions 4.1, 4.2, 4.3, and 4.4 hold. Then, the MPC optimal control problem* (4.11) *is recursively feasible.*

The theorem is proved by showing that a feasible $\tilde{\boldsymbol{U}}_{t+1}$ exists, given a feasible $\tilde{\boldsymbol{U}}_t$.

Proof. Due to Assumption 4.2, an input

$$\tilde{\boldsymbol{U}}_t = \left(\boldsymbol{u}_{t|t}, ..., \boldsymbol{u}_{t+N|t}, \boldsymbol{\nu}_{t+N|t}, ..., \boldsymbol{\nu}_{t+(N_\mathrm{lt}-1)|t}\right) \tag{4.13}$$

exists. First, the focus is on the short-term horizon, using $\boldsymbol{U}_t$. Shifting the initial inputs $(\boldsymbol{u}_{t|t}, ..., \boldsymbol{u}_{t+(N-1)|t})$ by one step yields the input sequence $(\boldsymbol{u}_{t+1|(t+1)}, ..., \boldsymbol{u}_{(t+1)+(N-2)|(t+1)})$, as the previous input sequence remains feasible for step $t+1$. According to (4.11e), $\boldsymbol{x}_{t+N|t}$ lies in the control invariant set $\mathcal{X}_\mathrm{f}$, therefore, $\boldsymbol{x}_{(t+1)+(N-1)|(t+1)} \in \mathcal{X}_\mathrm{f}$ and an input $\boldsymbol{u}_{(t+1)+(N-1)|(t+1)}$ exists such that $\boldsymbol{x}_{(t+1)+N|(t+1)} \in \mathcal{X}_\mathrm{f}$. This implies an input $\boldsymbol{u}_{(t+1)+N|(t+1)}$ exists, yielding the input sequence

$$\boldsymbol{U}_{t+1} = \left(\boldsymbol{u}_{t+1|(t+1)}, ..., \boldsymbol{u}_{(t+1)+(N-1)|(t+1)}, \boldsymbol{u}_{(t+1)+N|(t+1)}\right). \tag{4.14}$$

Next, the long-term horizon is considered. Given Assumption 4.4, $\boldsymbol{\xi}_N \in \Xi_\mathrm{cis}$ and $\boldsymbol{\nu}_N$ exists. As Ξ_cis is a control invariant set, an input sequence

$$\boldsymbol{U}_{\boldsymbol{\nu},t+1} = \left(\boldsymbol{\nu}_{(t+1)+N|(t+1)}, ..., \boldsymbol{\nu}_{(t+1)+(N_\mathrm{lt}-1)|(t+1)}\right) \tag{4.15}$$

exists, yielding

$$\tilde{\boldsymbol{U}}_{t+1} = \left(\boldsymbol{u}_{t+1|(t+1)}, ..., \boldsymbol{u}_{(t+1)+N|(t+1)}, \boldsymbol{\nu}_{(t+1)+N|(t+1)}, ..., v_{(t+1)+(N_\mathrm{lt}-1)|(t+1)}\right). \tag{4.16}$$

Therefore, the MPC optimal control problem (4.11) is recursively feasible. □

Note that $\boldsymbol{u}_{t+N|t}$ is not part of the cost function (4.11a) and does not guarantee $\boldsymbol{x}_{t+(N+1)|t} \in \mathcal{X}_\mathrm{f}$, but $\boldsymbol{u}_{t+N|t}$ is necessary to evaluate (4.11f).

4.3.3 Simulation Study

We evaluate the proposed MPC method in a setting similar to the one described in [7]. A mobile robot is steered along a path with obstacles, as illustrated in Figure 4.2. The aim is to reach the target point while avoiding obstacles. All quantities are given in SI units. The simulations were carried out in MATLAB with the `fmincon` solver on a standard desktop computer.

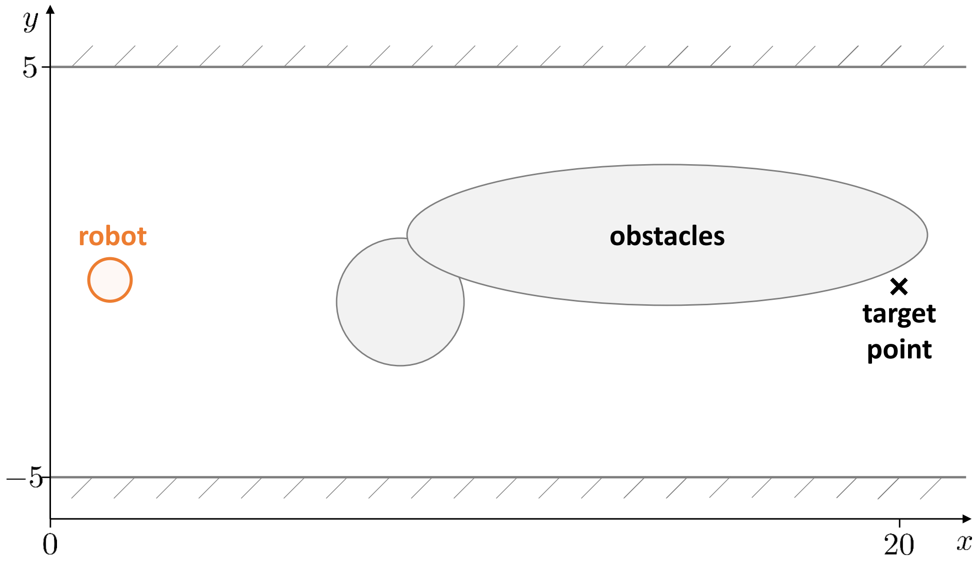


Figure 4.2: NUSH+Granularity MPC simulation scenario. © 2021 IEEE.

Simulation Setup

Two system models are considered, where the nonlinearity is found in the constraints. The detailed prediction model is given by

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 0 & 0 \\ \Delta t \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \Delta t \frac{1}{m} \end{bmatrix} \boldsymbol{u}_k \tag{4.17}$$

with state vector $\boldsymbol{x}_k = (x_k, v_{x,k}, y_k, v_{y,k})^\top$, input $\boldsymbol{u}_k = (F_{x,k}, F_{y,k})^\top$, sampling time Δt, and mass $m = 0.5$. The state vector consists of $x-$ and $y-$positions, as well as $x-$ and $y-$velocity v_x and v_y, the inputs are forces in $x-$ and $y-$direction, F_x and F_y. The following constraints, $\mathcal{X}$ and $\mathcal{U}$, are employed for states and inputs

$$-3 \leq v_{x,k} \leq 3 \tag{4.18a}$$
$$-5 \leq \ y_k \ \leq 5 \tag{4.18b}$$
$$-3 \leq v_{y,k} \leq 3 \tag{4.18c}$$
$$-3 \leq F_{x,k} \leq 3 \tag{4.18d}$$
$$-0.5 \leq F_{x,k} \leq 0.5. \tag{4.18e}$$

The control invariant set $\mathcal{X}_\mathrm{f}$ is given by

$$v_{x,N} = 0, \quad v_{y,N} = 0, \quad -5 \leq y_N \leq 5. \tag{4.19}$$

This ensures that at the end of the first horizon segment, the robot can come to a standstill by applying negative acceleration and a zero lateral acceleration, which avoids any constraint violations.

The approximated, coarse model, based on (4.17), is given by

$$\boldsymbol{\xi}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{\xi}_k + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \boldsymbol{\nu}_k \tag{4.20}$$

with state $\boldsymbol{\xi}_k = (x_k, y_k)^\top$ and input $\boldsymbol{\nu}_k = (v_{x,k}, v_{y,k})^\top$. The models (4.17) and (4.20) are linked by the projection matrix

$$\begin{pmatrix} \boldsymbol{\xi}_N \\ \boldsymbol{\nu}_N \end{pmatrix} = \operatorname{Proj}\left(\begin{pmatrix} \boldsymbol{x}_N \\ \boldsymbol{u}_N \end{pmatrix} \right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_N \\ \boldsymbol{u}_N \end{pmatrix}. \tag{4.21}$$

The coarse model is subject to constraints Ξ and $\mathcal{U}_{\boldsymbol{\nu}}$ similar to (4.18), i.e.,

$$-5 \leq \; y_k \; \leq 5 \tag{4.22a}$$

$$-3 \leq v_{x,k} \leq 3 \tag{4.22b}$$

$$-3 \leq v_{y,k} \leq 3, \tag{4.22c}$$

where the control invariant set Ξ_{cis} is defined as in (4.19).

Additionally, obstacles according to Figure 4.2 are considered as ellipsoidal constraints in both the detailed and the coarse model. Given the ellipse equation

$$\left(\frac{x_k - x^{\text{OB}}}{a_{\text{r}}} \right)^2 + \left(\frac{y_k - y^{\text{OB}}}{b_{\text{r}}} \right)^2 \leq 1 \tag{4.23}$$

with ellipse parameters a_{r} and b_{r} and origin offset $(x^{\text{OB}}, y^{\text{OB}})$, we consider the two overlapping obstacles with parameters $(a_{\text{r},1}, b_{\text{r},1}, x_1^{\text{OB}}, y_1^{\text{OB}}) = (1.5,\ 1.5,\ 10,\ -0.1)$ and $(a_{\text{r},2}, b_{\text{r},2}, x_2^{\text{OB}}, y_2^{\text{OB}}) = (5,\ 1.4,\ 15.2,\ 1.3)$.

The position of the two obstacles allows us to analyze the benefit of a longer prediction horizon. Passing the obstacles above results in a longer path. However, the circular obstacle is positioned in such a way ($y^{\text{OB}} < 0$) that it is more rewarding to pass the obstacle above. A longer prediction horizon now makes it possible to choose the path with higher short-term cost, as it has lower cost in the long-term.

We compare three MPC setups to evaluate the proposed method: standard MPC, MPC with models of different granularity, and the proposed approach. The standard MPC has a shorter horizon to then show the advantage of using a longer prediction horizon. The aim is to reach the reference point $(x_{\text{ref}}, y_{\text{ref}}) = (20,\ 0)$, resulting in the reference states $\boldsymbol{x}_{\text{ref}} = (20,\ 0,\ 0,\ 0)^\top$ and $\boldsymbol{\xi}_{\text{ref}} = (20,\ 0)^\top$. The initial state is $\boldsymbol{x}_0 = (0,\ 0,\ 0,\ 0)$. All stage costs have the quadratic form

$$l(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t) = (\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref}})^\top \boldsymbol{Q} (\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref}}) + \boldsymbol{u}_k^\top \boldsymbol{R} \boldsymbol{u}_k \tag{4.24a}$$

$$l_{\text{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\text{lt}}) = (\boldsymbol{\xi}_k - \boldsymbol{\xi}_{\text{ref}})^\top \boldsymbol{Q}_{\Delta t_{\text{lt}}} (\boldsymbol{\xi}_k - \boldsymbol{\xi}_{\text{ref}}) + \boldsymbol{\nu}_k^\top \boldsymbol{R}_{\Delta t_{\text{lt}}} \boldsymbol{\nu}_k \tag{4.24b}$$

with $\boldsymbol{Q} = \operatorname{diag}(1,\ 0,\ 5,\ 0)$, $\boldsymbol{R} = \operatorname{diag}(0.1,\ 0.1)$ and $\boldsymbol{Q}_{\Delta t_{\text{lt}}} = \operatorname{diag}(1,\ 5)$, $\boldsymbol{R}_{\Delta t_{\text{lt}}} = \operatorname{diag}(0.01,\ 0.01)$. While velocities are not penalized in $l(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t)$, they are penalized slightly in $l_{\text{lt}}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k)$ in order to have a non-zero matrix $\boldsymbol{R}_{\text{lt}}$. Terminal cost functions are chosen as $V_{\text{f}}(\boldsymbol{x}_k) = (\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref}})^\top \boldsymbol{Q} (\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref}})$ and $V_{\text{f,lt}}(\boldsymbol{\xi}_k) = (\boldsymbol{\xi}_k - \boldsymbol{\xi}_{\text{ref}})^\top \boldsymbol{Q}_{\Delta t_{\text{lt}}} (\boldsymbol{\xi}_k - \boldsymbol{\xi}_{\text{ref}})$.

The three controllers have the following characteristics.

A) Standard MPC: uses a prediction horizon $N = 10$ with sampling time $\Delta t = 0.2$ for model (4.17), constraints (4.18), and terminal constraints (4.19), as well as stage cost $l(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and terminal cost $V_\mathrm{f}(\boldsymbol{x}_N)$.

B) MPC with models of different granularity: uses the horizons $N = 10$ and $N_\mathrm{lt} = 16$ with sampling time $\Delta t = 0.2$ with model (4.17) and constraints (4.18) for the short-term horizon $N = 10$, and model (4.20), constraints (4.22), and terminal constraints (4.19), between $N = 10$ and the long-term horizon $N_\mathrm{lt} = 16$. The stage costs are $l(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t)$ and $l_\mathrm{lt}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_\mathrm{lt})$ with terminal costs $V_\mathrm{f}(\boldsymbol{x}_N)$ and $V_\mathrm{f,lt}(\boldsymbol{\xi}_{N_\mathrm{lt}})$.

C) Proposed NUSH+Granularity MPC scheme: uses two horizons. For the short-term horizon $N = 10$ with sampling time $\Delta t = 0.2$, model (4.17) and constraints (4.18) are employed, as well as (4.19) for the control invariant set $\mathcal{X}_\mathrm{f}$. Between $N = 10$ and the long-term horizon $N_\mathrm{lt} = 16$ the increased sampling time $\Delta t_\mathrm{lt} = 0.4$ is chosen with the model (4.20), constraints (4.22), and control invariant set Ξ_cis according to (4.19). Terminal costs $V_\mathrm{f}(\boldsymbol{x}_N)$ and $V_\mathrm{f,lt}(\boldsymbol{\xi}_{N_\mathrm{lt}})$ are used with the stage costs $l(\boldsymbol{x}_k, \boldsymbol{u_k}, \Delta t)$ and

$$l_\mathrm{lt}(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_\mathrm{lt}) = (\boldsymbol{\xi}_k - \boldsymbol{\xi}_\mathrm{ref})^\top \tilde{\boldsymbol{Q}}_{\Delta t_\mathrm{lt}} (\boldsymbol{\xi}_k - \boldsymbol{\xi}_\mathrm{ref}) + \boldsymbol{\nu}_k^\top \tilde{\boldsymbol{R}}_{\Delta t_\mathrm{lt}} \boldsymbol{\nu}_k \tag{4.25}$$

with $\tilde{\boldsymbol{Q}}_{\Delta t_\mathrm{lt}} = \mathrm{diag}(2,\ 10)$, $\tilde{\boldsymbol{R}}_{\Delta t_\mathrm{lt}} = \mathrm{diag}(0.02,\ 0.02)$ according to (4.6). The weights are increased, as the sampling time is larger compared to the short-term horizon, resulting in less states and inputs considered in the cost function.

The main properties of the analyzed MPC schemes (bold font) used for the simulation are summarized in Table 4.1. Properties and results are also provided for further MPC schemes that are not discussed in detail.

method	N	Δt	$N_\mathrm{lt} - N$	Δt_lt	cost J
standard MPC	10	0.2			$5.9 \cdot 10^3$
standard MPC	13	0.2			$5.9 \cdot 10^3$
standard MPC	16	0.2			$5.9 \cdot 10^3$
standard MPC	8	0.4			$6.0 \cdot 10^3$
NUSH MPC [188]	10	0.2	3 (detailed)	0.4	$5.9 \cdot 10^3$
gran. MPC [7]	10	0.2	6 (coarse)	0.2	$5.6 \cdot 10^3$
proposed MPC	10	0.2	3 (coarse)	0.4	$5.6 \cdot 10^3$

Table 4.1: Comparison of MPC setups: short-term horizon (detailed prediction model), long-term horizon (detailed or coarse prediction model), sampling time, and cost.

As shown, the decision variables vary between the three methods. While the horizon of the proposed method and MPC with models of different granularity covers the same horizon, less decision variables are necessary for the proposed approach.

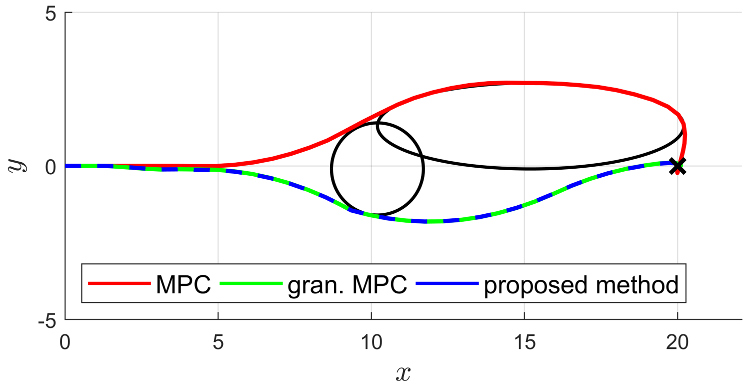


Figure 4.3: Simulation results: the standard MPC controller (red) chooses a longer path due to the shorter horizon, while both other approaches find the shorter path. © 2021 IEEE.

Simulation Results

In this section, we will compare the simulation results of the three methods. Each simulation was run for 50 iterations. We will first focus on the individual simulations and then investigate the overall result.

The simulation results of the individual controllers are illustrated in Figure 4.3. As the center of the circular obstacle is set slightly below $y = 0$, the standard MPC controller moves the robot towards the top. If only the circular obstacle were present, this would be the behavior with the lowest cost. However, due to the short horizon the ellipsoidal obstacle is only detected later. As the cost would be larger to change the path, the robot continues the longer path. Both the MPC with models of different granularity and the proposed MPC scheme detect the ellipsoidal obstacle before deciding on a path. Therefore, both methods select the shorter path below the circular obstacle, resulting in lower overall costs.

The overall cost J_{sim} for each simulation run is analyzed by comparing the real cost which occurred for each step, i.e.,

$$J_{\text{sim}} = \sum_{t=0}^{N_{\text{sim}}-1} l(\boldsymbol{x}_{t+1}, \boldsymbol{u}_t, \Delta t) \tag{4.26}$$

according to (4.24a) with $N_{\text{sim}} = 50$ simulation steps. The overall results are shown in Table 4.1. The standard MPC with $N = 10$ has the lowest computational effort, the average computation time per iteration is 0.27 s (100 %). However, as described before, the shorter horizon results in higher costs, as the longer path is chosen, illustrated by the increased cost compared to the other two methods. In this example, the computational effort of the proposed method is 0.41 s (151 %). Eventually, we compare the proposed method with MPC with models of different granularity. While the costs are equal, the proposed method reduces the computational effort by 33 % compared to MPC with models of different granularity (226 %).

All three controllers reach the target state eventually; however, cost and computational effort vary. Whereas the proposed method proved to be beneficial here, this is highly scenario dependent. It will be of interest to apply the proposed MPC scheme to more challenging automated vehicle scenarios, considering dynamic obstacles with uncertain behavior [30,53].

4.3.4 Discussion

The presented method divides the prediction horizon into two segments. Multiple segments extending the original horizon with different simpler models and larger sampling times are also possible. However, the effort of designing and setting up multiple segments could be higher than the resulting benefit.

The proposed approach can be interpreted and applied in two ways with respect to standard MPC: extending or splitting the horizon. In a first interpretation, the second horizon segment is regarded as an extended horizon compared to the standard MPC horizon. This allows for longer predictions, while the computational effort is only slightly increased due to a simplified model and larger sampling times. A second interpretation is as follows. The time span covered by the prediction horizon is equal for standard MPC and the proposed method. But computational complexity is reduced as the detailed model is only employed for the short-term prediction and less decision variables are used, given the non-uniformly spaced horizon.

In contrast to [7] and other literature, in the optimal control problem (4.11) the control invariant set $\mathcal{X}_\mathrm{f}$ is at the end of the first horizon segment ($\boldsymbol{x}_N$). This is necessary to guarantee recursive feasibility. If the control invariant set were at the end of the overall horizon, recursive feasibility could not be guaranteed, as the different sampling times in the short- and long-term horizon do not allow standard MPC theory to guarantee recursive feasibility, i.e., reusing the shifted horizon for the next time step is not possible. This is similar to guaranteeing stability in MPC with a non-uniformly spaced horizon [188]. However, if we interpret the proposed method as an approach that extends the standard horizon with a long-term horizon to improve the prediction at only slightly increased computational effort, it is suitable to place a control invariant set at the end of the short-term horizon.

The predicted states in the long-term horizon do not affect recursive feasibility, as $\boldsymbol{x}_N \in \mathcal{X}_\mathrm{f}$ ensures that the optimal control problem remains recursively feasible. The long-term horizon is considered as an improvement for the cost-to-go. Therefore, the constraints for the long-term horizon do not necessarily have to exactly match the constraints of the short-term horizon. A set Ξ_cis must still be provided, however, to ensure that the proof of Theorem 4.1 remains valid. Nevertheless, there is a certain degree of freedom to select Ξ_cis.

As stability was not yet shown for MPC with models of different granularity, the focus of this section was to first guarantee recursive feasibility, which is guaranteed for MPC with models of different granularity but not for MPC with a non-uniformly spaced horizon. Dissipativity theory could be of interest, similar to the stability guarantee in [188], when investigating stability for the proposed method.

In [7] a robust MPC scheme was employed for the long-term horizon to address consistency of the models. While this was omitted here to focus on the combination of different models and varying sampling time, a robust MPC scheme could be applied for the long-term prediction together with additive noise to the coarse system model (4.10).

It is also important to note that not any simplified model is suitable to be combined with a detailed model. It must be possible to find a projection function, which is more likely if the coarse model is a reduced model of the detailed model. Finding a reduced model for a detailed nonlinear model is challenging. However, for detailed linear models, it is often straightforward to obtain a reduced model that ensures that Assumption 4.1 is fulfilled. An example was addressed in the simulation study.

4.4 Extending the MPC Prediction Horizon for Systems with Uncertainty

After having considered undisturbed systems, we now present a method for systems subject to uncertainty. Again, the horizon is split into a short-term and a long-term horizon, where the latter employs a coarse prediction model and increasing sampling time steps. In addition, the short-term prediction considers robust constraints and the long-term prediction uses chance constraints. This setup enables safe planning for the immediate future together with less conservative long-term planning. This section is based on the work published in [34,35].

4.4.1 Problem Formulation

In general, a single prediction model and a constant sampling time Δt is employed within MPC. Here, we consider two prediction models of different granularity, i.e., different level of detail, and different sampling time for a linear, discrete-time system with additive disturbance, i.e.,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}(\Delta t)\boldsymbol{x}_k + \boldsymbol{B}(\Delta t)\boldsymbol{u}_k + \boldsymbol{w}_k, \tag{4.27a}$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}_{\mathrm{lt}}(\Delta t_{\mathrm{lt}})\boldsymbol{\xi}_k + \boldsymbol{B}_{\mathrm{lt}}(\Delta t_{\mathrm{lt}})\boldsymbol{\nu}_k + \boldsymbol{\delta}_k, \tag{4.27b}$$

where $\Delta t < \Delta t_{\mathrm{lt}}$ are the sampling times for the respective prediction horizons, $\boldsymbol{x}_k \in \mathbb{R}^{n_{\boldsymbol{x}}}$, $\boldsymbol{\xi}_k \in \mathbb{R}^{n_{\boldsymbol{\xi}}}$ denote the states, and $\boldsymbol{u}_k \in \mathbb{R}^{n_{\boldsymbol{u}}}$, $\boldsymbol{\nu}_k \in \mathbb{R}^{n_{\boldsymbol{\nu}}}$ the inputs at the prediction step k, with $\boldsymbol{A} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$, $\boldsymbol{B} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{u}}}$, $\boldsymbol{A}_{\mathrm{lt}} \in \mathbb{R}^{n_{\boldsymbol{\xi}} \times n_{\boldsymbol{\xi}}}$, $\boldsymbol{B}_{\mathrm{lt}} \in \mathbb{R}^{n_{\boldsymbol{\xi}} \times n_{\boldsymbol{\nu}}}$. The bounded disturbance $\boldsymbol{w}_k \in \mathcal{W} \subset \mathbb{R}^{n_{\boldsymbol{x}}}$ denotes additive uncertainty within the system model, where $\mathcal{W}$ is a compact, convex set and includes the origin. While (4.27a) is a more detailed model, (4.27b) is a coarse representation of system (4.27a) with an additive bounded disturbance $\boldsymbol{\delta}_k$, which acts as a projection, or an over-approximation, of the bounded disturbance $\boldsymbol{w}_k$ from the detailed model. Based on the detailed model, the coarse representation may be obtained by only considering a subset of states of the detailed model [7] or by model reduction techniques [80]. Furthermore, it is assumed that the PDF $f_{\boldsymbol{\delta}}(\boldsymbol{\delta})$ of $\boldsymbol{\delta}_k$ is known, and that the realizations of $\boldsymbol{\delta}_k$ for $k \in \mathbb{N}$ are i.i.d. with zero-mean.

The states and inputs of both models are linearly constrained for all $k \in \mathbb{N}$ by

$$\boldsymbol{x}_k \in \mathcal{X} = \left\{\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \boldsymbol{H}\boldsymbol{x} \leq \boldsymbol{h}\right\}, \tag{4.28a}$$

$$\boldsymbol{u}_k \in \mathcal{U} = \left\{\boldsymbol{u} \in \mathbb{R}^{n_u} \mid \boldsymbol{G}\boldsymbol{u} \leq \boldsymbol{g}\right\}, \tag{4.28b}$$

$$\boldsymbol{\xi}_k \in \boldsymbol{\Xi} = \left\{\boldsymbol{\xi} \in \mathbb{R}^{n_\xi} \mid \boldsymbol{H}_{\mathrm{lt}}\boldsymbol{\xi} \leq \boldsymbol{h}_{\mathrm{lt}}\right\}, \tag{4.28c}$$

$$\boldsymbol{\nu}_k \in \mathcal{U}_{\boldsymbol{\nu}} = \left\{\boldsymbol{\nu} \in \mathbb{R}^{n_v} \mid \boldsymbol{G}_{\mathrm{lt}}\boldsymbol{\nu} \leq \boldsymbol{g}_{\mathrm{lt}}\right\}, \tag{4.28d}$$

where the constraints for the coarse model follow from the constraints of the detailed model. These two models provide the basis of the proposed MPC method of this section.

Objective 4.2. *The aim is to extend the MPC method, developed in the previous Section 4.3, to systems subject to uncertainty.*

Achieving Objective 4.2 requires treating constraints subject to uncertainty differently in short-term and long-term predictions. The focus on short-term predictions is robust constraint satisfaction for safety reasons, whereas long-term predictions must ensure that considering uncertainty does not lead to overly conservative control behavior.

4.4.2 Method

In the following, an MPC optimal control problem is derived with robust constraints and a detailed model for short-term predictions, as well as chance constraints and a coarse model for long-term predictions, where the horizon for the short-term and long-term prediction is spaced differently. The proposed combined robust and stochastic MPC method, using NUSH and models of different granularity, is referred to as NG R+SMPC. We will first present the general structure of the proposed NG R+SMPC optimal control problem, where we briefly introduce the ideas of MPC with models of different granularity and MPC with a non-uniformly spaced optimization horizon. Then, details are provided on the robust constraints and chance constraints. Eventually, the resulting overall NG R+SMPC optimal control problem is shown. The proposed method is illustrated in Figure 4.1.

MPC Optimal Control Problem

We first formulate an MPC optimal control problem with two prediction stages and a total of N_{lt} prediction steps. The detailed model (4.27a) is used for the short-term prediction, with N prediction steps and sampling time Δt, to guarantee robust constraint satisfaction in the immediate future. Additionally, the coarse model (4.27b) is used for the long-term prediction, with $N_{\mathrm{lt}} - N$ prediction steps and sampling time Δt_{lt}, to reduce the computational cost while considering the uncertainties in a probabilistic manner by turning the state constraints into chance constraints. The overall prediction time is $N\Delta t + (N_{\mathrm{lt}} - N)\Delta t_{\mathrm{lt}}$.

The general optimal control problem is given by

$$\min_{\tilde{\boldsymbol{U}}} \mathrm{E}\left(\sum_{k=0}^{N-1}\left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t\right)\right) + \sum_{k=N}^{N_{\mathrm{lt}}-1}\left(l_{\mathrm{lt}}\left(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\mathrm{lt}}\right)\right) + V_{\mathrm{f,lt}}\left(\boldsymbol{\xi}_{N_{\mathrm{lt}}}, \Delta t_{\mathrm{lt}}\right)\right) \tag{4.29a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{A}(\Delta t)\boldsymbol{x}_k + \boldsymbol{B}(\Delta t)\boldsymbol{u}_k + \boldsymbol{w}_k \tag{4.29b}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{4.29c}$$

$$\boldsymbol{x}_k \in \mathcal{X} \quad \forall \boldsymbol{w}_k \in \mathcal{W}, \qquad k \in \mathbb{I}_{1,N} \tag{4.29d}$$

$$\left(\boldsymbol{\xi}_N, \boldsymbol{\nu}_N\right) = \mathrm{Proj}\left(\boldsymbol{x}_N, \boldsymbol{u}_N\right) \tag{4.29e}$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}_{\mathrm{lt}}(\Delta t_{\mathrm{lt}})\boldsymbol{\xi}_k + \boldsymbol{B}_{\mathrm{lt}}(\Delta t_{\mathrm{lt}})\boldsymbol{\nu}_k + \boldsymbol{\delta}_k \tag{4.29f}$$

$$\boldsymbol{\nu}_k \in \mathcal{U}_{\boldsymbol{\nu}}, \qquad k \in \mathbb{I}_{N,N_{\mathrm{lt}}-1} \tag{4.29g}$$

$$\Pr\left(\boldsymbol{\xi}_k \in \Xi\right) \geq \beta, \qquad k \in \mathbb{I}_{N,N_{\mathrm{lt}}} \tag{4.29h}$$

with the input sequence

$$\tilde{\boldsymbol{U}} = \left(\boldsymbol{u}_0, \dots, \boldsymbol{u}_N, \boldsymbol{\nu}_N, \dots, \boldsymbol{\nu}_{N_{\mathrm{lt}}-1}\right)^{\top} \tag{4.30}$$

and state sequences

$$\tilde{\boldsymbol{X}} = \left(\boldsymbol{x}_0, ..., \boldsymbol{x}_N, \boldsymbol{\xi}_N, ..., \boldsymbol{\xi}_{N_{\mathrm{lt}}}\right)^{\top}. \tag{4.31}$$

The cost functions are defined as

$$l\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t\right) = \|\boldsymbol{x}_k\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}_k\|^2_{\boldsymbol{R}}\,, \tag{4.32a}$$

$$l_{\mathrm{lt}}\left(\boldsymbol{\xi}_k, \boldsymbol{\nu}_k, \Delta t_{\mathrm{lt}}\right) = \|\boldsymbol{\xi}_k\|^2_{\boldsymbol{Q}_{\Delta t_{\mathrm{lt}}}} + \|\boldsymbol{\nu}_k\|^2_{\boldsymbol{R}_{\Delta t_{\mathrm{lt}}}}\,, \tag{4.32b}$$

$$V_{\mathrm{f,lt}}\left(\boldsymbol{\xi}_k, \Delta t_{\mathrm{lt}}\right) = \|\boldsymbol{\xi}_k\|^2_{\boldsymbol{P}_{\Delta t_{\mathrm{lt}}}} \tag{4.32c}$$

with the positive definite weighting matrices $\boldsymbol{Q} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$, $\boldsymbol{R} \in \mathbb{R}^{n_{\boldsymbol{u}} \times n_{\boldsymbol{u}}}$, $\boldsymbol{Q}_{\Delta t_{\mathrm{lt}}} \in \mathbb{R}^{n_{\boldsymbol{\xi}} \times n_{\boldsymbol{\xi}}}$, $\boldsymbol{R}_{\Delta t_{\mathrm{lt}}} \in \mathbb{R}^{n_{\boldsymbol{v}} \times n_{\boldsymbol{v}}}$, and $\boldsymbol{P}_{\Delta t_{\mathrm{lt}}} \in \mathbb{R}^{n_{\boldsymbol{\xi}} \times n_{\boldsymbol{\xi}}}$. The risk parameter $\beta \in [0, 1)$ in (4.29h) specifies the desired probability of state constraint satisfaction in the long-term prediction. Note that $\boldsymbol{u}_N$ is needed to compute (4.29e).

In the following, we modify (4.29) to obtain a tractable optimal control problem.

Robust Constraints with a Detailed Model

For the short-term prediction with robust constraints, we define a disturbance-free reference system with tighter constraints [160]. Therefore, the input is decomposed into a stabilizing state feedback and a new decision variable $\boldsymbol{c}_k$ for the controller, resulting in

$$\boldsymbol{u}_k = \boldsymbol{K}\boldsymbol{x}_k + \boldsymbol{c}_k. \tag{4.33}$$

with feedback gain $\boldsymbol{K}$. The actual system model and the nominal system model are given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{c}_k + \boldsymbol{w}_k, \tag{4.34a}$$

$$\overline{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_{\boldsymbol{K}}\overline{\boldsymbol{x}}_k + \boldsymbol{B}\boldsymbol{c}_k, \tag{4.34b}$$

with the nominal states $\overline{\boldsymbol{x}}_k$ and the stabilized system matrix $\boldsymbol{A}_{\boldsymbol{K}} = \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$. The deviation $\boldsymbol{e}_k := \boldsymbol{x}_k - \overline{\boldsymbol{x}}_k$ of the actual system yields the error update

$$\boldsymbol{e}_{k+1} = \boldsymbol{A}_{\boldsymbol{K}}\boldsymbol{e}_k + \boldsymbol{w}_k \tag{4.35}$$

with zero initial deviation, i.e.,

$$\boldsymbol{e}_0 = 0 \;\Leftrightarrow\; \boldsymbol{x}_0 = \overline{\boldsymbol{x}}_0. \tag{4.36}$$

Based on the Minkowski set addition [160], the set containing $\boldsymbol{e}_k$ is given by

$$\mathcal{S}_k := \bigoplus_{i=0}^{k-1} \boldsymbol{A}_{\boldsymbol{K}}^i \mathcal{W} = \mathcal{W} \oplus \boldsymbol{A}_{\boldsymbol{K}}\mathcal{W} \oplus \cdots \oplus \boldsymbol{A}_{\boldsymbol{K}}^{k-1}\mathcal{W}. \tag{4.37}$$

Now, (4.37) allows us to compute the minimal disturbance invariant set $\mathcal{S}_\infty$. This invariant set helps to define an outer-bounding tube around the states of the nominal system $\overline{\boldsymbol{x}}_k$, which includes the states of the actual system $\boldsymbol{x}_k$ for any possible disturbance sequence, i.e.,

$$\boldsymbol{x}_k \in \{\overline{\boldsymbol{x}}_k\} \oplus \mathcal{S}_\infty. \tag{4.38}$$

With (4.38), tighter constraint sets for states and inputs of the nominal system are computed, yielding

$$\overline{\mathcal{X}} = \mathcal{X} \ominus \mathcal{S}_\infty, \tag{4.39a}$$

$$\overline{\mathcal{U}} = \mathcal{U} \ominus \boldsymbol{K}\mathcal{S}_\infty, \tag{4.39b}$$

given that the set $\mathcal{W}$ is small enough such that $\mathcal{S}_\infty \subset \operatorname{int}(\mathcal{X})$ and $\boldsymbol{K}\mathcal{S}_\infty \subset \operatorname{int}(\mathcal{U})$ hold [133].

Although using the nominal system (4.34b) and the tightened constraint sets (4.39) for the short-term prediction would lead to robust constraint satisfaction, it is possible to improve this method by making use of an additional degree of freedom to the controller, namely the initial state of the nominal system $\overline{\boldsymbol{x}}_0$ [133]. There is no guarantee that setting the initial state of the nominal system $\overline{\boldsymbol{x}}_0$ equal to the actual initial state $\boldsymbol{x}_0$ enhances convergence to the reference of the nominal state trajectory. In order to determine an improved center of the tube, the controller considers the initial state of the nominal system $\overline{\boldsymbol{x}}_0$ as an additional decision variable to the inputs $\boldsymbol{c}_k$. It is necessary that the actual initial (current) state $\boldsymbol{x}_0$ remains in the tube with the initial nominal state $\overline{\boldsymbol{x}}_0$ as its center, i.e.,

$$\boldsymbol{x}_0 \in \{\overline{\boldsymbol{x}}_0\} \oplus \mathcal{S}_\infty \;\Leftrightarrow\; \boldsymbol{x}_0 - \overline{\boldsymbol{x}}_0 \in \mathcal{S}_\infty, \tag{4.40}$$

which is treated as a constraint for this decision variable. This method yields faster convergence and additionally has pleasing theoretical properties considering stability [133, 160].

Chance Constraints with a Coarse Model

It is necessary to reformulate the probabilistic constraint (4.29h) into a deterministic statement in order to implement the SMPC scheme. Similar to RMPC, the goal is to find tightened constraints for the (deterministic) nominal states for each prediction step, considering the evolution of the disturbance in the actual system.

We first decompose the states $\boldsymbol{\xi}_k$ into a deterministic and a probabilistic part and the inputs $\boldsymbol{\nu}_k$ into a stabilizing state feedback and a new decision variable $\boldsymbol{\eta}_k$, i.e.,

$$\boldsymbol{\xi}_k = \boldsymbol{\zeta}_k + \boldsymbol{\epsilon}_k, \tag{4.41a}$$

$$\boldsymbol{\nu}_k = \boldsymbol{K}_{\text{lt}}\boldsymbol{\xi}_k + \boldsymbol{\eta}_k, \tag{4.41b}$$

which yields

$$\boldsymbol{\zeta}_{k+1} = \boldsymbol{A}_{\boldsymbol{K},\text{lt}}\boldsymbol{\zeta}_k + \boldsymbol{B}_{\text{lt}}\boldsymbol{\eta}_k, \tag{4.42a}$$

$$\boldsymbol{\epsilon}_{k+1} = \boldsymbol{A}_{\boldsymbol{K},\text{lt}}\boldsymbol{\epsilon}_k + \boldsymbol{\delta}_k, \tag{4.42b}$$

where $\boldsymbol{A}_{\boldsymbol{K},\text{lt}} = \boldsymbol{A}_{\text{lt}} + \boldsymbol{B}_{\text{lt}}\boldsymbol{K}_{\text{lt}}$ is the stabilized system matrix.

Due to $\boldsymbol{\delta}_k$ being a projection, or an over-approximation, of the disturbance in the detailed model, it follows from (4.36) that $\boldsymbol{\epsilon}_0 = \boldsymbol{0}$. Thus, with (4.42b), the probabilistic error $\boldsymbol{\epsilon}_k$ is expressed via the disturbances $\boldsymbol{\delta}_0$ to $\boldsymbol{\delta}_{k-1}$ as

$$\boldsymbol{\epsilon}_k = \sum_{i=0}^{k-1} \boldsymbol{A}_{\boldsymbol{K},\text{lt}}^i \boldsymbol{\delta}_{k-1-i}, \quad k > 0 \tag{4.43}$$

and its probability distribution may be computed if the distribution of all $\boldsymbol{\delta}_i$ is known.

State constraint tightening. The constraint (4.29h) may be rewritten in terms of the decomposed state (4.41a) as

$$\Pr\left([\boldsymbol{H}_{\text{lt}}]_j \,\boldsymbol{\zeta}_k \leq [\boldsymbol{h}_{\text{lt}}]_j - [\boldsymbol{H}_{\text{lt}}]_j \,\boldsymbol{\epsilon}_k\right) \geq \beta, \quad j \in \mathbb{I}_{1,n_{\boldsymbol{h}}}, \tag{4.44}$$

where $[\boldsymbol{H}_{\mathrm{lt}}]_j$ and $[\boldsymbol{h}_{\mathrm{lt}}]_j$ denote the j-th row and entry of a matrix $\boldsymbol{H}_{\mathrm{lt}}$ and vector $\boldsymbol{h}_{\mathrm{lt}}$, and $n_{\boldsymbol{h}}$ is the number of rows and entries of $\boldsymbol{H}_{\mathrm{lt}}$ and $\boldsymbol{h}_{\mathrm{lt}}$. Based on [116], (4.44) may be formulated as

$$\begin{aligned}&\forall j \in \mathbb{I}_{1,n_{\boldsymbol{h}}}, \ \ \exists \tilde{\tau} \in \mathbb{R} \text{ s.t.}\\ &[\boldsymbol{H}_{\mathrm{lt}}]_j \, \boldsymbol{\zeta}_k \leq \tilde{\tau}, \ \Pr\left(\tilde{\tau} \leq [\boldsymbol{h}_{\mathrm{lt}}]_j - [\boldsymbol{H}_{\mathrm{lt}}]_j \, \boldsymbol{\epsilon}_k\right) \geq \beta.\end{aligned} \tag{4.45}$$

Resulting from this expression, suitable tightened state constraints for the nominal state $\boldsymbol{\zeta}_k \in \overline{\Xi}_k$, which guarantee the satisfaction of constraint (4.29h), are defined using

$$\overline{\Xi}_k = \{\boldsymbol{\zeta} \in \mathbb{R}^{n_\xi} | \boldsymbol{H}_{\mathrm{lt}}\boldsymbol{\zeta} \leq \boldsymbol{\tau}_k\}, \quad k \in \mathbb{I}_{N,N_{\mathrm{lt}}}, \tag{4.46}$$

where $\boldsymbol{\tau}_k = \left([\boldsymbol{\tau}_k]_1, \dots, [\boldsymbol{\tau}_k]_{n_\xi}\right)^\top$ is found by solving

$$\begin{aligned}\forall j \in \mathbb{I}_{1,n_{\boldsymbol{h}},}, \ k \in \mathbb{I}_{N,N_{\mathrm{lt}}} : \ [\boldsymbol{\tau}_k]_j &= \max_{\tau} \tau\\ \text{s.t. } \Pr\left(\tau \leq [\boldsymbol{h}_{\mathrm{lt}}]_j - [\boldsymbol{H}_{\mathrm{lt}}]_j \, \boldsymbol{\epsilon}_k\right) &\geq \beta.\end{aligned} \tag{4.47}$$

Input constraint tightening. In addition, tightened constraints for the input $\boldsymbol{\nu}_k$ have to be found, due to the state-feedback of the disturbed state $\boldsymbol{\xi}$ in (4.41b). Conservatism is decreased by applying a stochastic constraint tightening in the predictions, as described in [116].

Similar to (4.46) and (4.47), the tightened input constraint set is defined by

$$\overline{\mathcal{U}}_{\boldsymbol{\nu},k} = \{\boldsymbol{\nu} \in \mathbb{R}^{n_v} | \boldsymbol{G}_{\mathrm{lt}}\boldsymbol{\nu} \leq \boldsymbol{\vartheta}_k\}, \quad k \in \mathbb{I}_{N,N_{\mathrm{lt}}-1}, \tag{4.48}$$

where $\boldsymbol{\vartheta}_k = ([\boldsymbol{\vartheta}_k]_1, \dots, [\boldsymbol{\vartheta}_k]_{n_v})^\top$ is found by solving

$$\begin{aligned}\forall j \in \mathbb{I}_{1,n_{\boldsymbol{g}}}, \ k \in \mathbb{I}_{N,N_{\mathrm{lt}}} : \ [\boldsymbol{\vartheta}_k]_j &= \max_{\vartheta} \vartheta\\ \text{s.t. } \Pr\left(\vartheta \leq [\boldsymbol{g}_{\mathrm{lt}}]_j - [\boldsymbol{G}_{\mathrm{lt}}]_j \, \boldsymbol{K}_{\mathrm{lt}}\boldsymbol{\epsilon}_k\right) &\geq \beta_u,\end{aligned} \tag{4.49}$$

where $n_{\boldsymbol{g}}$ is the number of rows and entries of $\boldsymbol{G}_{\mathrm{lt}}$ and $\boldsymbol{g}_{\mathrm{lt}}$. The probabilistic level β_u may be different from β. If these computations are solved, a deterministic expression of the probabilistic chance constraint is provided.

In comparison to our earlier work [34] where the disturbance was assumed to be normally distributed, here, the proposed general reformulation of the chance constraint is valid for arbitrary probability distributions of the disturbance $\boldsymbol{\delta}_k$. Assuming a normally distributed disturbance simplifies the problems (4.47) and (4.49), due to the fact that the error $\boldsymbol{\epsilon}_k$ is also normally distributed at all times k, and only its covariance matrix has to be computed to retrieve the PDF.

Computational Considerations

Using RMPC or SMPC requires additional computational cost to compute the tightened constraints. For RMPC, it is possible to compute the tightened constraint sets (4.39) offline by first finding a disturbance invariant outer approximation $\mathcal{S}_{\infty,\mathrm{approx}}$ of the minimal disturbance invariant set $\mathcal{S}_\infty$. A method to find such an approximation is described in [158]. The approximated set is chosen as

$$\mathcal{S}_{\infty,\mathrm{approx}} = (1 - \alpha_s)\mathcal{S}_s, \tag{4.50}$$

with $\alpha_s \in [0,1)$, $s \in \mathbb{N}$, and $\mathcal{S}_s$ according to (4.37). The approximation (4.50) is valid, i.e., $\mathcal{S}_{\infty,\text{approx}}$ is a disturbance invariant set and $\mathcal{S}_\infty \subseteq \mathcal{S}_{\infty,\text{approx}}$, if the parameters α_s and s fulfill the condition $(\boldsymbol{A}_{\boldsymbol{K}})^s\,\mathcal{W} \subseteq \alpha_s \mathcal{W}$.

For constraint tightening in the SMPC part, we need to solve the chance-constrained problems (4.47) and (4.49). In order to overcome the difficulty of finding a deterministic solution, sampling techniques to solve chance-constrained problems can be applied offline. They are independent of the underlying distribution, easy to implement, and it is possible to give specific guarantees about their solution [116,117]. In particular, they enable directly using complicated simulations or measurements of the error, instead of determining a PDF. In the following, we elaborate on the offline sampling approach presented in [116].

The idea is to draw a sufficiently large number n_ω of i.i.d. samples of the disturbance $\boldsymbol{\delta}$ and require the constraint to hold for all, but a fixed number ω of samples. Under the condition $\tilde{\beta}_\text{u} n_\omega > k$ with $\tilde{\beta} = 1 - \beta$, the explicit conditions

$$\tilde{\beta}_\text{lb} n_\omega - 1 + \sqrt{3\tilde{\beta}_\text{lb} n_\omega \ln \frac{2}{\beta_\text{conf}}} \leq \omega \leq \tilde{\beta}_\text{ub} n_\omega - \sqrt{2\tilde{\beta}_\text{ub} n_\omega \ln \frac{1}{\beta_\text{conf}}} \tag{4.51}$$

must hold such that the solution to the sampled program is equal to the chance-constrained programs (4.47) and (4.49) for some $\tilde{\beta} \in [1-\tilde{\beta}_\text{ub}, 1-\tilde{\beta}_\text{lb}]$ with confidence $1-\beta_\text{conf}$ [43,116]. It is now possible to compute a set of error evolutions $\mathcal{E}_{j,k} = \left\{ [\boldsymbol{H}_\text{lt}]_j\, \boldsymbol{\epsilon}_k^{(i)} \right\}_{i=1,\dots,n_\omega}$ for each state constraint (i.e., row of $\boldsymbol{H}_\text{lt}$) and prediction step k with $\boldsymbol{\epsilon}_k$ computed according to (4.43) using the drawn samples of the disturbance realizations. The tightening of each constraint at each predicted time step k is then retrieved by

$$[\boldsymbol{\tau}_k]_j = [\boldsymbol{h}_\text{lt}]_j - q_{1-\omega/n_\omega}, \tag{4.52}$$

with $q_{1-\omega/n_\omega}$ being the $(1-\frac{\omega}{n_\omega})$-quantile of the set $\mathcal{E}_{j,k}$. The parameter $q_{1-\omega/n_\omega}$ can be seen as a tightening parameter, which tightens the constraints for the nominal system, such that the chance constraints for the actual disturbed system are fulfilled with the given probability level and confidence. The same procedure is applied to compute the input constraint tightening $\boldsymbol{\vartheta}_k$ by replacing $\boldsymbol{H}_\text{lt}$ and $\boldsymbol{h}_\text{lt}$ with $\boldsymbol{G}_\text{lt}$ and $\boldsymbol{g}_\text{lt}$. However, as shown in [116], if the tightened constraints are derived via a sampling approach, the results on chance constraint satisfaction do not hold with certainty but only with confidence $(1-\beta_\text{conf})^\beta$.

Optimal Control Problem

Using the results from this section for the optimal control problem (4.29), the overall NG R+SMPC optimal control problem is given by

$$\begin{aligned}
\min_{\tilde{\boldsymbol{C}}} \; & \mathrm{E}\left(\sum_{k=0}^{N-1}\left(l\left(\overline{\boldsymbol{x}}_k, \boldsymbol{K}\overline{\boldsymbol{x}}_k+\boldsymbol{c}_k, \Delta t\right)\right)+\sum_{k=N}^{N_{\mathrm{lt}}-1}\left(l_{\mathrm{lt}}\left(\boldsymbol{\zeta}_k, \boldsymbol{K}_{\mathrm{lt}}\boldsymbol{\zeta}_k+\boldsymbol{\eta}_k, \Delta t_{\mathrm{lt}}\right)\right)+V_{\mathrm{f,lt}}\left(\boldsymbol{\zeta}_{N_{\mathrm{lt}}}\right)\right) && \text{(4.53a)}\\
\text{s.t. } & \boldsymbol{x}_0-\overline{\boldsymbol{x}}_0 \in \mathcal{S}_\infty && \text{(4.53b)}\\
& \overline{\boldsymbol{x}}_{k+1}=\boldsymbol{A}_{\boldsymbol{K}}\overline{\boldsymbol{x}}_k+\boldsymbol{B}\boldsymbol{c}_k && \text{(4.53c)}\\
& \boldsymbol{K}\overline{\boldsymbol{x}}_k+\boldsymbol{c}_k \in \overline{\mathcal{U}}, \qquad k \in \mathbb{I}_{0,N-1} && \text{(4.53d)}\\
& \overline{\boldsymbol{x}}_k \in \overline{\mathcal{X}}, \qquad k \in \mathbb{I}_{1,N} && \text{(4.53e)}\\
& \left(\boldsymbol{\zeta}_N, \boldsymbol{\nu}_N\right)=\operatorname{Proj}\left(\overline{\boldsymbol{x}}_N, \boldsymbol{K}\overline{\boldsymbol{x}}_N+\boldsymbol{c}_N\right) && \text{(4.53f)}\\
& \boldsymbol{\eta}_N=\boldsymbol{\nu}_N-\boldsymbol{K}_{\mathrm{lt}}\boldsymbol{\zeta}_N && \text{(4.53g)}\\
& \boldsymbol{\zeta}_{k+1}=\boldsymbol{A}_{\boldsymbol{K},\mathrm{lt}}\boldsymbol{\zeta}_k+\boldsymbol{B}_{\mathrm{lt}}\boldsymbol{\eta}_k && \text{(4.53h)}\\
& \boldsymbol{K}_{\mathrm{lt}}\boldsymbol{\zeta}_k+\boldsymbol{\eta}_k \in \overline{\mathcal{U}}_{\boldsymbol{\nu},k}, \qquad k \in \mathbb{I}_{N,N_{\mathrm{lt}}-1} && \text{(4.53i)}\\
& \boldsymbol{\zeta}_k \in \overline{\Xi}_k, \qquad k \in \mathbb{I}_{N,N_{\mathrm{lt}}} && \text{(4.53j)}
\end{aligned}$$

with the input sequence

$$\tilde{\boldsymbol{C}}=\left(\boldsymbol{c}_0,\ldots,\boldsymbol{c}_N,\boldsymbol{\eta}_N,\ldots,\boldsymbol{\eta}_{N_{\mathrm{lt}}}\right)^\top \tag{4.54}$$

and state sequences

$$\tilde{\overline{\boldsymbol{X}}}=\left(\overline{\boldsymbol{x}}_0,...,\overline{\boldsymbol{x}}_N,\boldsymbol{\zeta}_N,...,\boldsymbol{\zeta}_{N_{\mathrm{lt}}}\right)^\top. \tag{4.55}$$

The stage cost functions and terminal cost function depend on different weighting matrices, according to (4.32). Similar to [133], the feedback control law $\boldsymbol{\pi}^*(\cdot)$, which results from the solution of the above stated optimal control problem (4.53), follows

$$\boldsymbol{\pi}^*(\boldsymbol{x}_0):=\overline{\boldsymbol{u}}_0^*+\boldsymbol{K}\left(\boldsymbol{x}_0-\overline{\boldsymbol{x}}_0^*\right). \tag{4.56}$$

This is simplified by

$$\boldsymbol{\pi}^*(\boldsymbol{x}_0)=\boldsymbol{K}\overline{\boldsymbol{x}}_0^*+\boldsymbol{c}_0^*+\boldsymbol{K}\left(\boldsymbol{x}_0-\overline{\boldsymbol{x}}_0^*\right)=\boldsymbol{K}\boldsymbol{x}_0+\boldsymbol{c}_0^*, \tag{4.57}$$

where $\boldsymbol{c}_0^*$ is the first element of the optimal control sequence $\tilde{\boldsymbol{C}}^*$ and $\boldsymbol{x}_0$ is the actually sampled system state. Except considering (4.53b) within the optimal control problem, constraint tightening is performed offline, as mentioned in Section 4.4.2.

4.4.3 Simulation Study

In this section, we analyze the previously introduced method in a scenario based on [7], i.e., motion control of a mobile robot in a known environment consisting of boundaries, a dynamic obstacle, and a static obstacle. The simulation is based on a kinematic robot model. All stated values and axis types are given in SI units. The robot and the dynamic obstacle both have a radius of 0.5. The objective for the controlled robot is to get from the starting point (0, 0) to the target point (19, 0) without colliding with any obstacles. The simulation scenario is shown in Figure 4.4.

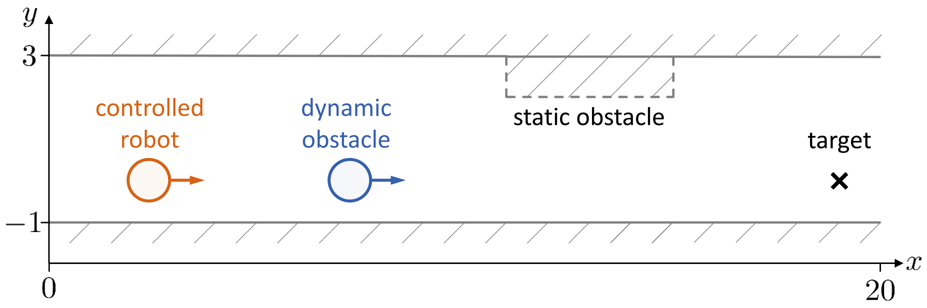


Figure 4.4: NG R+SMPC simulation scenario.

Simulation Setup

The controlled robot knows the velocity and starting position of the dynamic obstacle, and the additional uncertainty for velocities in x- and y-direction, w_{v_x} and w_{v_y}, where $|w_{v_x}| \leq 0.1$ and $\left|w_{v_y}\right| \leq 0.1$.

Detailed model and constraints. The detailed model for the short-term prediction is given by

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ 0 & \Delta t \end{bmatrix} \boldsymbol{u}_k + \boldsymbol{w}_k, \tag{4.58}$$

with sampling time $\Delta t = 0.2$ and states $\boldsymbol{x}_k = (x_k, v_{x,k}, y_k, v_{y,k})^\top$ consisting of the position (x_k, y_k) in the x-y-plane and the velocities $v_{x,k}$, $v_{y,k}$ in both directions, as well as the inputs $\boldsymbol{u}_k = (a_{x,k}, a_{y,k})^\top$ representing the acceleration in both directions. The disturbance $\boldsymbol{w}_k$ has zero mean and is bounded by the set $\mathcal{W} = \{\boldsymbol{w}_k \mid |\boldsymbol{w}_k|_\infty \leq 0.1\}$, accounting for w_{v_x} and w_{v_y}. Actuator limitations are taken into account by box-constraints, bounding the inputs $\boldsymbol{u}_k$ by $|a_{x,k}| \leq 3$ and $|a_{y,k}| \leq 3$. The lateral position is constrained by $-0.5 \leq y_k \leq 2.5$ to ensure that the robot with radius 0.5 does not leave the environment boundaries. The velocities are constrained by $|v_{x,k}| \leq 3$ and $|v_{y,k}| \leq 3$.

Similar to other works, e.g., [23, 89, 172], the safety constraint for collision avoidance with the dynamic obstacle is defined by a nonlinear safety ellipse

$$d_k = \frac{\left(x_k - x_k^{\mathrm{DO}}\right)^2}{a_{\mathrm{r}}^2} + \frac{\left(y_k - y_k^{\mathrm{DO}}\right)^2}{b_{\mathrm{r}}^2} - 1 \geq 0, \tag{4.59}$$

with the ellipse parameters $a_{\mathrm{r}} = b_{\mathrm{r}} = 1$ describing a circle and the obstacle position $\left(x_k^{\mathrm{DO}}, y_k^{\mathrm{DO}}\right)$. If $d_k \geq 0$ holds, the center of the robot is outside of the ellipse and the constraint is satisfied. The corners of the static box-obstacle are at the points $(11, 3)$, $(11, 2)$, $(15, 2)$, and $(15, 3)$, resulting in linear inequality constraints.

In order to determine the tightened constraints for the RMPC approach in the short-term prediction, we compute a disturbance invariant outer approximation of the minimal disturbance invariant set $\mathcal{S}_\infty$ via (4.50) with $\alpha_s = 0.1457$ and $s = 11$ using the Multi-Parametric Toolbox 3 [83] in MATLAB, by applying the input decomposition (4.33) with the feedback gain

$$\boldsymbol{K} = -\begin{bmatrix} 3.77 & 4.67 & 0 & 0 \\ 0 & 0 & 3.77 & 4.67 \end{bmatrix}. \tag{4.60}$$

This yields the slightly adapted constraints for the nominal inputs $\overline{\boldsymbol{u}}_k$, i.e., $|\overline{a}_{x,k}| = |\overline{a}_{y,k}| \leq 1.60$, while the constraints for the state $\overline{\boldsymbol{x}}_k$ of the nominal system are given by $-0.29 \leq \overline{y}_k \leq 2.29$, $|\overline{v}_{x,k}| \leq 2.32$, and $|\overline{v}_{y,k}| \leq 2.32$. The tighter safety constraint is challenging to compute, due to the nonlinearity of (4.59). For simplicity, we over-approximate this constraint by an ellipse similar to (4.59), enlarged by the maximal distance between the boundary of the tube and its center, based on the procedure of Section 4.4.2. Considering the position dimensions of the tube, the reduced tube results in a square with width $2 \cdot 0.71$. The maximal distance between center and boundary results in $\sqrt{2} \cdot 0.71$, which yields the tightened ellipse parameters $\overline{a}_\mathrm{r} = \overline{b}_\mathrm{r} = 2.0$.

Coarse model and constraints. The coarse model of the robot depends on Δt_lt and is given by

$$\boldsymbol{\xi}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{\xi}_k + \begin{bmatrix} \Delta t_\mathrm{lt} & 0 \\ 0 & \Delta t_\mathrm{lt} \end{bmatrix} \boldsymbol{\nu}_k + \boldsymbol{\delta}_k, \tag{4.61}$$

where $\Delta t_\mathrm{lt} = 2\Delta t = 0.4$, the states $\boldsymbol{\xi}_k = (x_k, y_k)^\top$ consider only the position in the x-y-plane, and the velocities in both directions are treated as the inputs, i.e., $\boldsymbol{\nu}_k = (v_{x,k}, v_{y,k})^\top$. The random disturbance $\boldsymbol{\delta}_k$ is assumed to be distributed according to a truncated normal distribution with zero mean and the covariance matrix $\boldsymbol{\Sigma}_\delta = \mathrm{diag}(0.1, 0.1)$, truncated at the bounds of the interval $[-0.1, 0.1] \times [-0.1, 0.1]$.

The projection function, which maps the states and inputs of the detailed model to the states and inputs of the coarse model, is given by

$$\begin{pmatrix} \boldsymbol{\xi}_N \\ \boldsymbol{\nu}_N \end{pmatrix} = \mathrm{Proj}\left(\begin{pmatrix} \boldsymbol{x}_N \\ \boldsymbol{u}_N \end{pmatrix} \right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_N \\ \boldsymbol{u}_N \end{pmatrix}. \tag{4.62}$$

The tightened constraints for the coarse model are computed as described in Section 4.4.2, i.e., solving (4.47) and (4.49), using the proposed sampling approach of Section 4.4.2. In this example, for $\beta = 0.99$, $\beta_\mathrm{conf} = 10^{-4}$, $\tilde{\beta}_\mathrm{l} = 0.95(1-\beta)$, and $\tilde{\beta}_\mathrm{u} = 1.05(1-\beta)$, the conditions are fulfilled for $n_\omega = 941012$ and $\omega = 9454$, and the tightened constraints are computed according to (4.52). For the computation of $\boldsymbol{\epsilon}_k$, we apply the input decomposition (4.42) with the state-feedback gain

$$\boldsymbol{K}_\mathrm{lt} = -\begin{bmatrix} 0.83 & 0 \\ 0 & 0.83 \end{bmatrix}. \tag{4.63}$$

It is not possible to compute the tighter safety constraint using the proposed sampling method, due to the nonlinearity of (4.59). Similar to the robust prediction stage, we over-approximate this constraint by an enlarged ellipse, considering the worst-case constraint tightening of each prediction step k for constraints on the robot position.

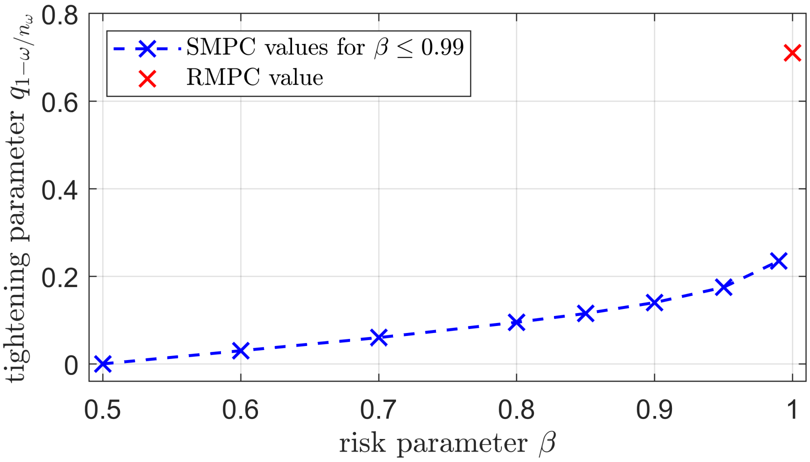


Figure 4.5: Dependency of the constraint tightening parameter on the risk parameter.

Objective function. For the optimal control problem, we use the quadratic cost functions

$$l\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \Delta t\right) = \left\|\boldsymbol{x}_k - (19,\ 0,\ 0,\ 0)^\top\right\|_{\boldsymbol{Q}}^2 + \left\|\boldsymbol{u}_k\right\|_{\boldsymbol{R}}^2, \tag{4.64a}$$

$$l_{\mathrm{lt}}\left(\boldsymbol{\zeta}_k, \boldsymbol{\nu}_k, \Delta t_{\mathrm{lt}}\right) = \left\|\boldsymbol{\zeta}_k - (19,\ 0)^\top\right\|_{\boldsymbol{Q}_{\Delta t_{\mathrm{lt}}}}^2 + \left\|\boldsymbol{\nu}_k\right\|_{\boldsymbol{R}_{\Delta t_{\mathrm{lt}}}}^2 \tag{4.64b}$$

with the terminal cost function $V_{\mathrm{f,lt}}\left(\boldsymbol{\zeta}_k\right) = \left\|\boldsymbol{\zeta}_k - (19,\ 0)^\top\right\|_{\boldsymbol{P}_{\Delta t_{\mathrm{lt}}}}^2$. The weighting matrices are defined as $\boldsymbol{Q} = \operatorname{diag}(1,\ 0.1,\ 1,\ 0.1)$ and $\boldsymbol{R} = \operatorname{diag}(0.1,\ 0.1)$, as well as $\boldsymbol{Q}_{\Delta t_{\mathrm{lt}}} = 2\operatorname{diag}(1,\ 1)$ and $\boldsymbol{R}_{\Delta t_{\mathrm{lt}}} = 2\operatorname{diag}(0.1,\ 0.1)$ according to (4.6), and $\boldsymbol{P}_{\Delta t_{\mathrm{lt}}} = \operatorname{diag}(2.36,\ 2.36)$. The expectation values follow directly from the zero-mean uncertainties. We choose the short-term horizon $N = 8$ and the long-term horizon $N_{\mathrm{lt}} - N = 6$, resulting in $N_{\mathrm{lt}} = 14$ and a horizon time window of $N\Delta t + (N_{\mathrm{lt}} - N)\Delta t_{\mathrm{lt}} = 4$, which is larger than the constant sampling time horizon $(N_{\mathrm{lt}} - N)\Delta t = 2.8$.

Discussion of constraint tightening. According to (4.52), the tighter constraints are computed by subtracting $q_{1-\omega/n_\omega}$ from the constraint bound $\boldsymbol{h}_{\mathrm{lt}}$. For our example and chosen probability distribution of the disturbance $\boldsymbol{\delta}_k$, the maximum value of the quantile, i.e., the constraint tightening parameter, does not exceed the value 0.27 for any k of the prediction, even for a relatively large risk parameter $\beta = 0.99$. For comparison, the equivalent constraint tightening value in the robust case (i.e, $\beta = 1$) equals 0.71. Figure 4.5 illustrates the dependence of the constraint tightening parameter $q_{1-\omega/n_\omega}$ on the risk parameter β for $k = 20$ and $\beta \leq 0.99$. The parameters have been computed for $\beta \leq 0.99$ with confidence $1 - \beta_{\mathrm{conf}} = 0.999$. For higher probabilistic levels, the number of required samples increases drastically, which causes computational difficulties. For this reason, larger values of β are omitted. The robust tightening parameter, i.e., setting $\beta = 1$, was determined using the Multi-Parametric Toolbox 3 [83].

The reason for this significant difference between the constraint tightening of SMPC and RMPC is that the worst-case error evolution scenario caused by the disturbance, and scenarios close to the worst-case, are very unlikely. Using an SMPC scheme for the long-term prediction drastically reduces the conservatism of the controller, compared to using RMPC over the full horizon.

Simulation Results

We will first analyze the behavior of the robot and then evaluate the performance, i.e., cost and computational effort of the proposed method. For an appropriate comparison, we additionally implement the three different methods from the simulations of [34] to evaluate and compare our proposed control scheme, resulting in the following four MPC schemes. Note that these MPC schemes are different than the ones presented in previous simulation in Section 4.3.3.

A) Single model RMPC: RMPC with the detailed model and the sampling time $\Delta t = 0.2$ over the full horizon.

B) Single model R+SMPC: RMPC with the detailed model for short-term predictions and SMPC with the detailed model for long-term predictions, with the sampling time $\Delta t = 0.2$ over the full horizon.

C) Granularity R+SMPC [34]: RMPC with the detailed model for short-term predictions and SMPC with the coarse model for long-term predictions, with the sampling time $\Delta t = 0.2$ over the full horizon.

D) NG R+SMPC (proposed method): RMPC with the detailed model and a sampling time of $\Delta t = 0.2$ for short-term predictions and SMPC with the coarse model and a sampling time of $\Delta t_{\mathrm{lt}} = 0.4$ for long-term predictions.

The simulations were carried out in MATLAB using `FORCES Pro` [62, 206].

Controlled robot behavior. The resulting trajectory for one example simulation of the proposed NG R+SMPC method is displayed in Figure 4.6.

We now compare the behavior of the proposed method to the three other approaches. Both single model R+SMPC (B) and granularity R+SMPC (C) yield a similar trajectory to the NG R+SMPC method (D) simulation. Therefore, a display of the results is omitted.

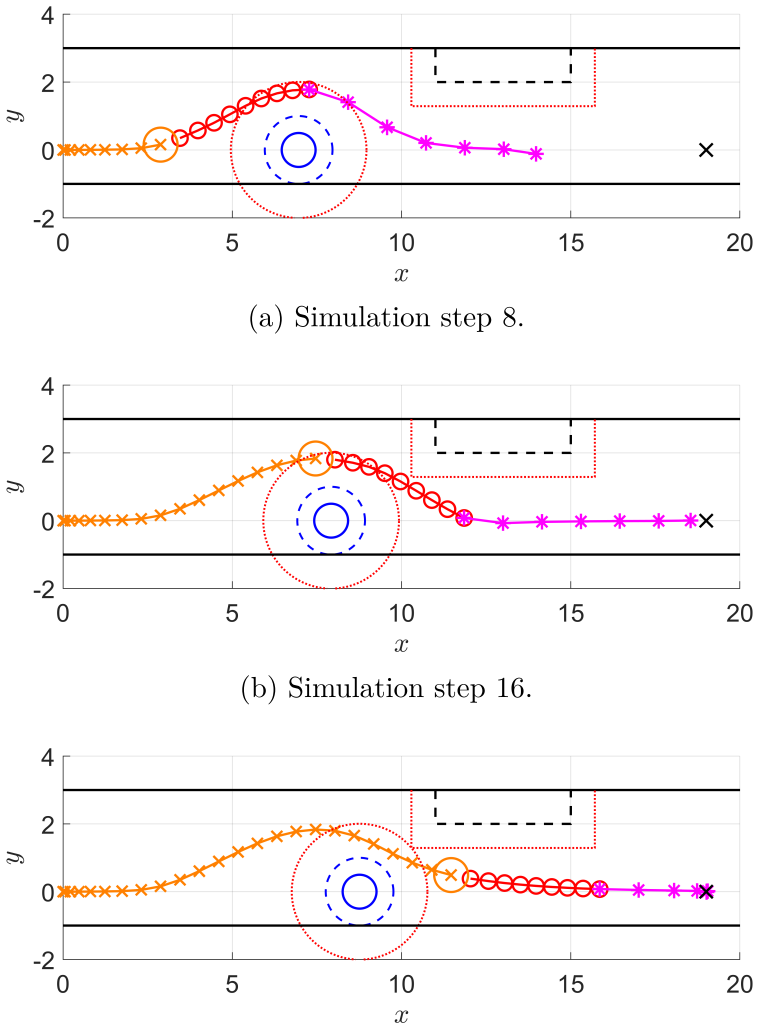
The result of applying the single model RMPC (A) approach is shown in Figure 4.7. Unlike the first simulation with the proposed method, the robust approach is more conservative and fails to pass the dynamic obstacle. Whereas only the constraints for the next predicted step are displayed in Figure 4.7, the robust constraints are enforced on the entire horizon, forcing the robot to stop in front of the static obstacle in order to avoid potential constraint violations.

Cost and computational effort. After having analyzed the behavior of the proposed method, the cost and computational effort is now evaluated for 100 simulations with random initial dynamic obstacle position and velocity. For each time step, a mean value over all simulations is calculated for cost and computational effort.

We compare the cost of the four approaches by evaluating the cost

$$J_{\mathrm{sim}} = \sum_{t=0}^{N_{\mathrm{sim}}-1} l(\boldsymbol{x}_{t+1}, \boldsymbol{u}_t, \Delta t) \tag{4.65}$$

with $l\left(\boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \Delta t\right) = ||\boldsymbol{x}_{k+1} - (19,\, 0,\, 0,\, 0)^\top||^2_{\boldsymbol{Q}_{\Delta t_{\mathrm{lt}}}} + ||\boldsymbol{u}_k||^2_{\boldsymbol{R}_{\Delta t_{\mathrm{lt}}}}$ and N_{sim} simulation steps. Figure 4.8 shows that the costs for single model R+SMPC (B) and granularity R+SMPC (C)



(a) Simulation step 8.

(b) Simulation step 16.

(c) Simulation step 23.

Figure 4.6: Simulation results of the proposed NG R+SMPC method (D). Previous robot steps are shown in orange. The planned RMPC and SMPC trajectory parts are shown in red circles and pink asterisks, respectively. Dotted red lines display the constraints for the (first step) nominal states of the RMPC.

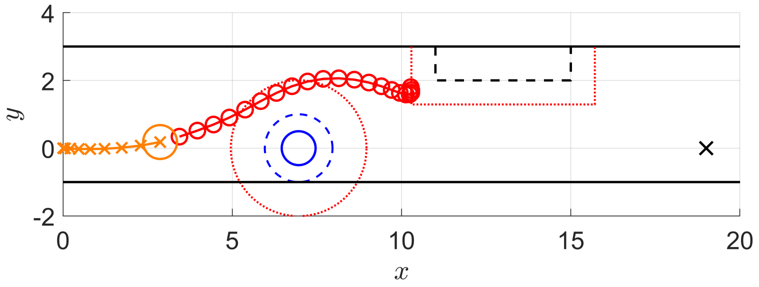


Figure 4.7: Simulation results of the RMPC method (A). Previous robot steps are shown in orange. The planned RMPC trajectory parts are shown in red circles. Dotted red lines display the constraints for the (first step) nominal states of the RMPC.

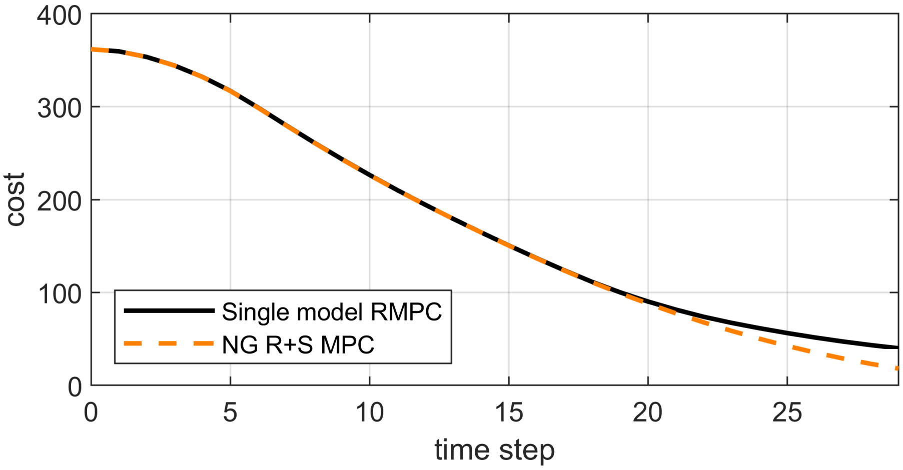


Figure 4.8: Cost per step for the four approaches, averaged over 100 simulation runs. NG R+SMPC, Granularity R+SMPC, and single model R+SMPC yield similar cost; hence, only NG R+SMPC is displayed.

are equal to the cost of the proposed NG R+SMPC method (D). Applying the single model RMPC method (A) results in a more conservative robot behavior, i.e., the robot does not pass the dynamic obstacle in front of the static box-obstacle in 35 of 100 simulation runs. This can also be seen in the cost, which is similar to the other methods at first, but then remains at a higher level as the robot is unable to move closer to the target point in 35 simulation runs. All other methods are able to overtake the dynamical obstacle in all 100 simulations.

We now evaluate the computation time to solve the optimization problems at each time step. The results, based on the 100 simulation runs, are displayed in Figure 4.9. For this simulation setup, the computational effort of the proposed NG R+SMPC method (D) is 32 % lower than single model RMPC (A), 43 % lower than single model R+SMPC (B), and 38 % lower than the Granularity R+SMPC approach (C).

In summary, for this simulation the proposed NG R+SMPC method results in less computational effort compared to a single model R+SMPC or a Granularity R+SMPC approach, while the performance remains similar. The proposed approach is less conservative compared to a single model RMPC approach, in addition to having less computational effort. If recursive feasibility is required, the proposed approach could be incorporated into the failsafe SMPC framework presented in Chapter 5.

4.4.4 Discussion

The proposed NG R+SMPC method enables robust planning for a short-term horizon and consider long-term targets, but with a reduced number of decision variables. Applying chance constraints instead of robust constraints for the long-term prediction reduces conservatism, depending on the underlying probability distribution of the disturbance. Robustly accounting for uncertainties over a long horizon is often highly restrictive and the worst-case scenarios are generally unlikely. Additionally, using the coarse model with a larger sampling time potentially decreases the computational effort. Note that it is possible to combine more than two models of different granularity and to add more segments with differently spaced

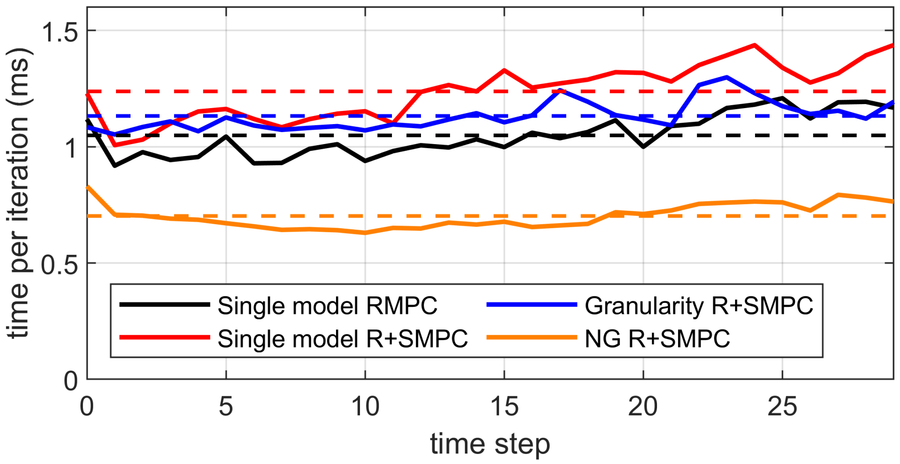


Figure 4.9: Computational effort per step for the four approaches, averaged over 100 simulation runs. Dashed lines represent mean values over all steps.

sampling horizons.

Obtaining the projection mentioned in Assumption 4.1 and guaranteeing consistency between the models via a robust approach as proposed in [7] or by finding additional constraints for the coarse model can be challenging, especially for complex systems. Determining suitable coarse models and the corresponding uncertainties remains a topic for further study.

The bounded disturbance in the coarse model of the proposed method allows us to investigate a proof of recursive feasibility in the future, which may be accomplished in probability with even tighter constraints for the SMPC scheme, similar to [116].

Besides the fact that stability for MPC with models of different granularity was not yet shown, the standard approach to show stability of an MPC scheme, e.g., [160, Chapter 2.4], cannot be applied to methods with non-uniformly spaced prediction horizons. However, dissipativity theory could be of interest for investigating stability of the proposed method, similar to the stability guarantee in [188].

This chapter was mostly based on [35]. Compared to our earlier work [34], there are three main improvements. The computation time is decreased by including the NUSH scheme, the SMPC constraint tightening is now done offline, and the proposed method is able to handle bounded uncertainties with arbitrary distributions instead of only normally distributed uncertainties.

4.5 Conclusion

In this chapter, we proposed two strategies to increase the MPC prediction horizon length while keeping computational complexity manageable. First, the undisturbed case is considered. We proposed an MPC scheme that combines a detailed model with smaller sampling time and an approximated, coarse model with larger sampling time. The presented method allows for precise planning on a short-term horizon while still considering long-term goals by improving the cost-to-go. The coarse model combined with increased sampling time allows for reduced computational effort. While recursive feasibility is guaranteed, stability is still an open challenge.

Second, we extended the previously proposed method by considering uncertainty within the MPC optimal control problem. We derived a mixed RMPC and SMPC method, which uses two models for the prediction horizon with different sampling times. RMPC is used with a detailed model for short-term predictions, while chance-constrained SMPC is combined with a coarse model for long-term predictions. The more general formulation of the disturbance in the SMPC horizon facilitates a more detailed analysis of a suitable projection of the disturbance in the short-term horizon and opens the way for an analysis of recursive feasibility and stability, which is a topic for further research. The proposed MPC method provides benefits for applications where robust constraint satisfaction is required, while considering long-term targets with less restrictive chance constraints using the coarse model and a larger sampling time.

5 Safety for Stochastic MPC

This chapter focuses on guaranteeing safety for SMPC while maintaining efficiency advantages of SMPC. We first design a safe SMPC algorithm that is designed for use in automated driving and then present a general safe SMPC approach. Part of the content of this chapter was published in [31, 32], particularly Section 5.2.

5.1 Introduction

Designing controllers for safety-critical systems requires considering two major challenges. Safety must be ensured for a system subject to uncertainty, and the controller should reduce conservatism to enable efficient system behavior, i.e., maximizing desired objectives. As it is possible to define safety via input and state constraints, MPC is a suitable method to control safety-critical systems subject to uncertainty.

A prominent example of control in safety-critical systems is automated driving. After pioneering work [24, 58, 59] in the 1980s and 1990s, research in automated driving has seen significant progress over the past decade. Improvements in computer vision-based sensing and the use of these sensor data in control algorithms enable automated vehicles to detect and react to hazards in dynamic traffic and a constantly changing environment. A majority of road accidents are still caused by human errors; therefore, increasing the level of vehicle autonomy has great potential to reduce the overall number of accidents. Automated vehicles are especially relevant in critical situations where a significant number of human drivers is incapable of performing necessary maneuvers in time [97]. While there are various control methods for vehicle trajectory planning, MPC has proved to be a suitable approach for automated vehicle trajectory planning [94, 112], which is in large parts due to the ability of MPC to consider uncertainty of other traffic participants and the environment.

In MPC, system uncertainty is addressed by RMPC [129] or SMPC [68, 134]. The conservatism resulting from robustly handling uncertainty in MPC is reduced by SMPC, where robust constraints are reformulated into probabilistic constraints. This probabilistic reformulation increases efficiency, but it also allows for a small probability of constraint violation, i.e, a probability of collision for vehicles. However, guaranteeing safety within SMPC has hardly been investigated.

In comparison to SMPC, trajectory planning based on reachability analysis provides formal safety guarantees [123, 182]. This approach is especially applicable to automated vehicles. Here, worst-case predictions are obtained for other surrounding vehicles in order to plan failsafe vehicle trajectories, referred to as failsafe trajectory planning (FTP), which is closely related to RMPC.

In this chapter, we tackle the challenge of safe and efficient control. We first present a novel MPC trajectory planner that combines the advantages of SMPC and failsafe trajectory planning for automated vehicles in environments with uncertainty. A trajectory is planned with SMPC, providing optimistic and efficient planning. In a regular setting, the first optimized SMPC input is then applied to the vehicle. In addition to SMPC, for every time step a failsafe trajectory is planned, given the first optimized SMPC input. The optimistic SMPC input is only applied to the vehicle if it is still possible to find a failsafe backup trajectory after having applied the first SMPC input. This ensures that the efficient SMPC trajectory is executed as long as a backup exists, therefore guaranteeing safety. The proposed method is referred to as stochastic Model Predictive Control + failsafe trajectory planning (SMPC+FTP). For SMPC+FTP, we present a detailed case differentiation to generate safety constraints with respect to other surrounding vehicles, both for the SMPC and the FTP optimal control problem.

We then generalize the SMPC+FTP approach such that it is applicable to general safety-critical application. In the general safe SMPC algorithm, the SMPC input is only applied if the resulting state remains in a robust positively invariant set, which ensures that a safe trajectory may be obtained in the next step. Whereas the design of both the SMPC+FTP method and the safe SMPC algorithm guarantees recursive feasibility, input-to-state stability is only guaranteed for the safe SMPC algorithm. With this proposed safety algorithm, for a given safety-critical application and based on desired control objectives, the most suitable SMPC approach can be chosen. This choice may be made independently of required properties, which are later ensured by the proposed safety algorithm.

In summary, the contributions of this chapter are as follows:

- A novel SMPC+FTP method providing efficient and safe trajectory planning for automated highway driving, including lane change decisions.
- An elaborate case differentiation for highway safety constraints.
- A proof of recursive feasibility of the proposed SMPC+FTP method.
- A general safe SMPC algorithm, compatible with any SMPC for linear systems with additive uncertainty and polytopic constraints.
- Guarantees for recursive feasibility and input-to-state stability for the safe SMPC algorithm.
- No influence of the SMPC risk parameter on safety for both proposed methods, simplifying risk parameter tuning.

The proposed methods combine advantages of stochastic and robust predictive control. For SMPC+FTP a simulation study of two complex highway scenarios demonstrates the benefits of optimistic trajectory planning in a regular scenario, while the ability of SMPC+FTP to guarantee safety is shown in an emergency scenario. The advantages of the proposed safe SMPC algorithm are demonstrated in a simple simulation example.

5.1.1 Related Work

Whereas general literature on SMPC has been covered in detail in Section 2.1, here, we briefly focus on safety in MPC, especially RMPC. We then present a detailed overview of trajectory planning for automated vehicles.

Safety in MPC

RMPC handles system uncertainty in a robust, but conservative way [9, 132], where tube-based MPC is the most common approach [102,108,130]. Stability and recursive feasibility guarantees are possible if the uncertainty bound is known initially. RMPC has successfully been applied to safety-critical applications such as automated driving [61, 71, 181], autonomous racing [198], and robotic manipulation [144].

Further approaches have recently been proposed to address safety in probability within MPC. A combination of MPC and control barrier functions consider safety similarly to how Lyapunov functions are used for stability [77, 208]. However, guaranteeing recursive feasibility in the presence of uncertainty remains a challenge. In [194, 195] a predictive safety filter is proposed to guarantee safety in probability for reinforcement learning. This is achieved by enforcing that only those reinforcement learning-based inputs may be applied, which allow for satisfaction of a soft-constrained optimal control problem.

Trajectory Planning for Automated Vehicles

Trajectory planning for automated vehicles is a widely studied area of research. There are various methods in non-MPC related fields, such as using partially observable Markov decision processes (POMDP) [87] or reinforcement learning [138]. Learning-based methods are also popular for autonomous racing [92,167,184,197]. When considering automated road vehicles, planning trajectories with MPC has the advantage of iteratively replanning the vehicle trajectory with constraints, accounting for a changing environment.

Standard MPC has been developed for cooperative adaptive cruise control, focusing on cooperative driving [95, 127] and vehicle platooning [52]. MPC is also designed specifically to plan trajectories for a single autonomous vehicle [81] or for combined maneuver and trajectory planning [201]. MPC was also combined with potential-field methods in order to avoid static and dynamic obstacles [159]. Extensive surveys cover MPC for autonomous ground vehicles [203] and MPC for advanced driver-assistance systems [140].

The main focus of this chapter is trajectory planning with robust or stochastic MPC, as well as failsafe trajectory planning (FTP). Failsafe trajectory planning is defined as planning collision-free vehicle trajectories, accounting for any legal future motion of surrounding vehicles [152]. For bounded uncertainties in real-world applications, FTP is applied based on finding worst-case sets. Combined with reachability analysis, formal safety guarantees are given [182]. The computation of these reachable sets is connected to control invariant sets in RMPC as stated in [74]. An approach to include reachability analysis into MPC is given in [175].

In [4] a method is proposed to compute the set of all future locations possibly occupied by traffic participants. The remaining safe space is admissible to plan emergency trajectories. This FTP is presented in [123]. First, given the most likely motion of surrounding vehicles, an optimal trajectory is determined. Then, an emergency trajectory is connected to the last point of the optimal trajectory for which collision avoidance is still guaranteed. The safe space is determined by an over-approximated set of any possible future vehicle motion. The failsafe trajectory is generated in such a way that the controlled vehicle comes to a standstill. In [152] an FTP method is introduced which generates failsafe trajectories in real-time. The method is tested in various simulations based on the CommonRoad benchmark framework [5]. A motion planning framework is introduced in [126] which combines reachability analysis with optimization-based trajectory planning. In [61] an RMPC method is suggested, which uses a combination of a potential field like function and reachability sets to obtain safe zones on the road. A further RMPC method for collision avoidance with moving obstacles is presented in [181].

SMPC has been intensively studied in the context of automated vehicles. These works focus on the trade-off between risk and conservatism, defined by probabilistic constraints, i.e., chance constraints [176]. On the one hand, taking into account unlikely uncertainty realizations drastically reduces efficiency, on the other hand, planning too optimistically increases risk. A major challenge in SMPC is reformulating the probabilistic chance constraint into a tractable constraint, which can be handled by a solver.

An SMPC particle approach is shown in [15] with a simple vehicle braking scenario, where particles approximate the uncertainty. An SMPC trajectory planner for automated vehicles in the presence of fixed obstacles is presented in [111]. In [50] the environment is modeled by an Interacting Multiple Model Kalman filter. Given the most likely prediction for surrounding vehicles, a vehicle trajectory is then planned with SMPC assuming Gaussian uncertainty. Varying risk parameters, denoting the level of accepted risk, are studied, illustrating the trade-off between risk and conservatism. In [187] an SMPC lane change controller is presented, where the lane change risk is considered using predicted time-to-collision.

A different SMPC approach is utilized in [53,170], focusing on SCMPC based on [43,171]. In SCMPC samples of the uncertainty are drawn, which must then satisfy the constraints to find a tractable chance constraint expression. While [170] focuses on simple lane change scenarios, the work is extended in [53] and experimental results are presented.

In summary, SMPC approaches provide efficient vehicle trajectories for the majority of uncertainty realizations in regular situations. However, for unlikely uncertainty realizations, safety issues occur.

5.1.2 Chapter Overview

The chapter is structured as follows. Section 5.2 introduces the SMPC+FTP approach for automated vehicles and Section 5.3 proposes the general safe SMPC algorithm. The basics of SMPC and FTP are introduced in Section 5.2.1, whereas the relevant vehicle models and the problem statement are presented in Section 5.2.2. The proposed SMPC+FTP approach is derived in Section 5.2.3, and details for the respective SMPC and FTP optimal control problems are provided in Section 5.2.4 and Section 5.2.5. The simulation results are presented in Section 5.2.6. A discussion is given in Section 5.2.7. The problem formulation for the safe SMPC algorithm, the derivation of the proposed method, and its properties are presented in Sections 5.3.1, 5.3.2, and 5.3.3, respectively. A simulation example is shown in Section 5.3.4. Conclusive remarks, addressing both methods, are given in Section 5.4.

5.2 Stochastic MPC with a Safety Guarantee for Automated Driving

In the following, we propose an SMPC algorithm with a safety guarantee for automated driving. Safety is guaranteed with a failsafe backup planning algorithm, which overwrites the SMPC input if necessary, based on the first predicted SMPC input.

5.2.1 Preliminaries

In the following, we briefly introduce the general MPC optimal control problems for SMPC and FTP in order to highlight the difference between these two planning approaches.

SMPC with Chance Constraints

Whereas standard MPC considers hard constraints, this is problematic if uncertainties are present. Hard constraints subject to uncertainty can be considered by chance constraints. This yields the SMPC optimal control problem

$$\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} \left(l(\boldsymbol{x}_k, \boldsymbol{u}_k) \right) + V_\mathrm{f}\left(\boldsymbol{x}_N\right) \tag{5.1a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{5.1b}$$

$$\boldsymbol{u}_k \in \mathcal{U}_k, \qquad k \in \mathbb{I}_{0,N-1} \tag{5.1c}$$

$$\boldsymbol{x}_k \in \mathcal{X}_k, \qquad k \in \mathbb{I}_{1,N} \tag{5.1d}$$

$$\Pr\Big(\boldsymbol{x}_k \in \mathcal{X}'_{k,\mathrm{safe}}(\boldsymbol{w})\Big) \geq \beta, \qquad k \in \mathbb{I}_{1,N} \tag{5.1e}$$

with the optimized input sequence $\boldsymbol{U} = \left(\boldsymbol{u}_0, \dots, \boldsymbol{u}_{N-1}\right)^\top$, states $\boldsymbol{x}_k$, prediction model $\boldsymbol{f}$, and the normally distributed, zero mean uncertainty $\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}^w\right)$ with covariance matrix $\boldsymbol{\Sigma}^w$. The cost function consists of the stage cost $l(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and the terminal cost $V_\mathrm{f}\left(\boldsymbol{x}_N\right)$. States and inputs are bounded by the state and input constraint sets $\mathcal{X}_k$ and $\mathcal{U}_k$, respectively, and the safety constraint $\mathcal{X}'_{k,\mathrm{safe}}(\boldsymbol{w})$ depends on the uncertainty $\boldsymbol{w}$. The probabilistic chance constraint is given by (5.1e). The safety constraint $\boldsymbol{x}_k \in \mathcal{X}'_{k,\mathrm{safe}}(\boldsymbol{w})$ is required to hold according to the risk parameter β. For $\beta < 1$ a non-zero constraint violation probability is therefore allowed.

The chance constraint (5.1e) cannot be handled by a solver directly, but is required to be reformulated into a deterministic expression. Details on the reformulation are given in Section 5.2.4.

In this section, uncertainty is considered regarding surrounding vehicles. The safe set $\Xi'_\mathrm{safe}(\boldsymbol{w})$ therefore depends on how the uncertainty $\boldsymbol{w}$ affects the surrounding vehicles. This is described in Section 5.2.2. Note that SMPC optimal control problems can consider an expectation value in the cost function; however, this is omitted here as no expectation value will be necessary for the automated driving optimal control problem, i.e., the prediction model (5.1b) does not include uncertainty. In the simulation study in Section 5.2.6, the uncertainty in the safe set $\Xi'_{k,\mathrm{safe}}(\boldsymbol{w})$ is governed by uncertainty in the behavior of surrounding vehicles.

Failsafe Trajectory Planning

We also consider an MPC optimal control problem for FTP, i.e., a failsafe MPC optimal control problem. In contrast to SMPC, FTP considers the worst-case realizations of the uncertainty, resulting in safe, yet conservative optimized inputs. The FTP optimal control problem is given by

$$
\begin{aligned}
&\min_{\boldsymbol{U}} \sum_{k=0}^{N-1} \left(l(\boldsymbol{\xi}_k, \boldsymbol{u}_k)\right) + V_{\mathrm{f}}\left(\boldsymbol{\xi}_N\right) && (5.2a)\\
&\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}\left(\boldsymbol{\xi}_k, \boldsymbol{u}_k\right) && (5.2b)\\
&\quad \boldsymbol{u}_k \in \mathcal{U}_k, \qquad k \in \mathbb{I}_{0,N-1} && (5.2c)\\
&\quad \boldsymbol{\xi}_k \in \Xi_k, \qquad k \in \mathbb{I}_{1,N} && (5.2d)\\
&\quad \boldsymbol{\xi}_k \in \Xi_{k,\mathrm{safe}}(\boldsymbol{w}), \qquad k \in \mathbb{I}_{1,N-1} && (5.2e)\\
&\quad \boldsymbol{\xi}_N \in \Xi_{N,\mathrm{safe}}(\boldsymbol{w}), && (5.2f)
\end{aligned}
$$

which is similar to the SMPC problem (5.1). However, the safety constraint (5.2e) is not a chance constraint, as in (5.1e), but a hard constraint. In contrast to SMPC, for FTP the safe set $\Xi_{k,\mathrm{safe}}(\boldsymbol{w})$ is constructed based on reachability analysis to ensure formal safety guarantees. This is strongly connected to the computation of invariant sets in RMPC. In addition to constraint (5.2e), a terminal constraint (5.2f) is required, which ensures that the terminal prediction state $\boldsymbol{\xi}_N$ may remain in a safe state beyond the prediction horizon. Based on this safe terminal set $\Xi_{N,\mathrm{safe}}(\boldsymbol{w})$ it is guaranteed that there exist system inputs $\boldsymbol{u}_{k^+}$ with $k^+ > N$ that result in safe states $\boldsymbol{\xi}_{k^+}$. Note that the safe set $\Xi'_{k,\mathrm{safe}}(\boldsymbol{w})$ in (5.1e) is not necessarily computed in the same way as the safe set $\Xi_{k,\mathrm{safe}}(\boldsymbol{w})$ in (5.2e),(5.2f). Details on the FTP optimal control problem are provided in Section 5.2.5.

5.2.2 Problem Formulation

MPC requires a system model for the controlled vehicle, known as the EV, and surrounding vehicles, referred to as TVs, in order to predict future states within the optimal control problem.

Ego Vehicle Model

We use a road-aligned, kinematic bicycle model to predict the EV states on a finite horizon, as suggested in [103]. The continuous-time system, similar to (3.42), is given by

$$
\begin{aligned}
\dot{s} &= v\cos(\psi^{\mathrm{ra}} + \alpha), && (5.3a)\\
\dot{d} &= v\sin(\psi^{\mathrm{ra}} + \alpha), && (5.3b)\\
\dot{\psi}^{\mathrm{ra}} &= \frac{v}{l_{\mathrm{r}}}\sin\alpha, && (5.3c)\\
\dot{v} &= a, && (5.3d)\\
\alpha &= \arctan\left(\frac{l_{\mathrm{r}}}{l_{\mathrm{r}} + l_{\mathrm{f}}}\tan\delta_{\mathrm{f}}\right), && (5.3e)
\end{aligned}
$$

where l_{r} and l_{f} are the distances from the vehicle center of gravity to the rear and front axles, respectively. The state vector is $\boldsymbol{x} = (s, d, \psi^{\mathrm{ra}}, v)^\top$ and the input vector is $\boldsymbol{u} = (a, \delta_{\mathrm{f}})^\top$. The vehicle velocity is given by v, acceleration and steering angle are denoted by a and δ_{f}, respectively. We consider the longitudinal position s of the vehicle along the road, the lateral vehicle deviation d from the centerline of the right lane, and the orientation ψ^{ra} of the vehicle with respect to the road. We refer to [157] and [49, Chapter 3] for further details on vehicle models. The nonlinear vehicle model (5.3) is summarized as $\dot{\boldsymbol{x}} = \boldsymbol{f}^{\mathrm{c}}\left(\boldsymbol{x}, \boldsymbol{u}\right)$.

Each MPC optimal control problem is initialized with a linearization of the nonlinear prediction model (5.3) around the current vehicle state $\boldsymbol{x}^* = \boldsymbol{x}_0$ and the input $\boldsymbol{u}^* = (0,\, 0)^\top$. Selecting a non-zero reference input $\boldsymbol{u}^*$ often results in large differences $\Delta\boldsymbol{u} = \boldsymbol{u}_k - \boldsymbol{u}^*$ for prediction steps far ahead, increasing the inaccuracy of the linearization. The linearized continuous-time vehicle model is then given by

$$\dot{\boldsymbol{x}}^* + \Delta\dot{\boldsymbol{x}} = \boldsymbol{f}^{\mathrm{c}}\left(\boldsymbol{x}^*, \boldsymbol{0}\right) + \boldsymbol{A}_{\mathrm{l}}\left(\boldsymbol{x} - \boldsymbol{x}^*\right) + \boldsymbol{B}_{\mathrm{l}}\boldsymbol{u} \tag{5.4}$$

with the Jacobian matrices

$$\boldsymbol{A}_{\mathrm{l}} = \left.\left[\frac{\partial \boldsymbol{f}^{\mathrm{c}}}{\partial \boldsymbol{x}}\right]\right|_{(\boldsymbol{x}^*,\boldsymbol{u}^*)}, \qquad \boldsymbol{B}_{\mathrm{l}} = \left.\left[\frac{\partial \boldsymbol{f}^{\mathrm{c}}}{\partial \boldsymbol{u}}\right]\right|_{(\boldsymbol{x}^*,\boldsymbol{u}^*)}. \tag{5.5}$$

A discrete-time model is required for MPC, therefore the linearized prediction model (5.4) is discretized with sampling time Δt. This yields the discrete states $\boldsymbol{x}_k = (s_k, d_k, \psi_k^{\mathrm{ra}}, v_k)^\top$ and inputs $\boldsymbol{u}_k = (a_k, \delta_{\mathrm{f},k})^\top$ for prediction step k, as well as the linearized, discretized system

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_0 + \Delta t \boldsymbol{f}^{\mathrm{c}}\left(\boldsymbol{x}_0, \boldsymbol{0}\right) + \boldsymbol{A}_{\mathrm{d}}\left(\boldsymbol{x}_k - \boldsymbol{x}_0\right) + \boldsymbol{B}_{\mathrm{d}}\boldsymbol{u}_k && \text{(5.6a)} \\ &= \boldsymbol{f}^{\mathrm{d}}\left(\boldsymbol{x}_0, \boldsymbol{x}_k, \boldsymbol{u}_k\right) && \text{(5.6b)} \end{aligned}$$

where $\boldsymbol{A}_{\mathrm{d}}$ and $\boldsymbol{B}_{\mathrm{d}}$ are matrices of the linearized system obtained from $\boldsymbol{A}_{\mathrm{l}}$, $\boldsymbol{B}_{\mathrm{l}}$ with zero-order hold. The nonlinear term $\boldsymbol{f}^{\mathrm{c}}\left(\boldsymbol{x}^*, \boldsymbol{u}^*\right)$ in (5.4) is approximated by a forward Euler method since $\boldsymbol{x}_0$ is known. The linearized, discretized matrices $\boldsymbol{A}_{\mathrm{d}}$ and $\boldsymbol{B}_{\mathrm{d}}$ are given in Appendix C.1.

The following sections derive an SMPC method and constraints to avoid collisions with surrounding vehicles. However, even if no other vehicles are present, certain constraints are required. Acceleration and steering angle are bounded by

$$\begin{aligned} \boldsymbol{u}_{\min} \leq{}& \boldsymbol{u}_k \leq \boldsymbol{u}_{\max} && \text{(5.7a)} \\ \Delta\boldsymbol{u}_{\min} \leq{}& \Delta\boldsymbol{u}_k \leq \Delta\boldsymbol{u}_{\max} && \text{(5.7b)} \end{aligned}$$

with $\Delta\boldsymbol{u}_{k+1} = \boldsymbol{u}_{k+1} - \boldsymbol{u}_k$ and $\boldsymbol{u}_{\max} = (a_{\max}, \delta_{\mathrm{f,max}})^\top$, $\boldsymbol{u}_{\min} = (a_{\min}, \delta_{\mathrm{f,min}})^\top$. Further, road and velocity constraints are considered, resulting in

$$\begin{aligned} & d_k \in \mathcal{X}^{\mathrm{lane}} && \text{(5.8a)} \\ & 0 \leq v_k \leq v_{\max} && \text{(5.8b)} \end{aligned}$$

where $\mathcal{X}^{\mathrm{lane}}$ represents road boundaries and $v_{\max}$ is the maximal velocity. Negative velocities are not allowed, i.e., $v_k \geq 0$.

In the following, we refer to input constraints by the set of admissible inputs $\mathcal{U}$ and state constraints are denoted by the set of admissible states $\mathcal{X}$.

Target Vehicle Model

In order to avoid collisions, the EV is also required to predict the future states of surrounding TVs. The prediction model for the TVs used by the EV is a linear, discrete-time point-mass model given by

$$\begin{aligned} \boldsymbol{x}_{k+1}^{\mathrm{TV}} &= \boldsymbol{A}\boldsymbol{x}_k^{\mathrm{TV}} + \boldsymbol{B}\boldsymbol{u}_k^{\mathrm{TV}} && \text{(5.9a)} \\ \boldsymbol{u}_k^{\mathrm{TV}} &= \tilde{\boldsymbol{u}}_k^{\mathrm{TV}} + \boldsymbol{w}_k^{\mathrm{TV}} && \text{(5.9b)} \end{aligned}$$

where $\boldsymbol{x}_k^{\mathrm{TV}} = (x_k^{\mathrm{TV}}, v_{x,k}^{\mathrm{TV}}, y_k^{\mathrm{TV}}, v_{y,k}^{\mathrm{TV}})^\top$ is the TV state with longitudinal position and velocity x_k^{TV}, $v_{x,k}^{\mathrm{TV}}$ and lateral position and velocity y_k^{TV}, $v_{y,k}^{\mathrm{TV}}$. The linear TV model allows us to propagate the uncertainty, which is necessary for the MPC approach in the following sections. The TV model used in this chapter is only one possible option. Other linear TV prediction models can be utilized.

The system and input matrices are

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 0.5\Delta t^2 & 0 \\ \Delta t & 0 \\ 0 & 0.5\Delta t^2 \\ 0 & \Delta t \end{bmatrix} \tag{5.10}$$

with sampling time Δt. The TV input consists of a feedback controller $\tilde{\boldsymbol{u}}_k^{\mathrm{TV}}$ and a perturbation on the input, which is assumed to be an independent, identically distributed disturbance vector $\boldsymbol{w}_k^{\mathrm{TV}}$. This setup assumes that the TV is following a given reference while deviations are allowed. The TV feedback controller is given by

$$\tilde{\boldsymbol{u}}_k^{\mathrm{TV}} = \boldsymbol{K}\left(\boldsymbol{x}_k^{\mathrm{TV}} - \boldsymbol{x}_{\mathrm{ref},k}^{\mathrm{TV}}\right) \tag{5.11}$$

with the TV reference $\boldsymbol{x}_{\mathrm{ref},k}^{\mathrm{TV}}$. The feedback matrix $\boldsymbol{K}$ is obtained by a linear-quadratic regulator strategy. If the TV input computed by (5.11) exceeds the limits $\boldsymbol{u}_{\max}^{\mathrm{TV}} = (a_{\max}, a_{y,\max})^\top$ and $\boldsymbol{u}_{\min}^{\mathrm{TV}} = (a_{\min}, a_{y,\min})^\top$, summarized as $\mathcal{U}^{\mathrm{TV}}$, the TV inputs are bounded to satisfy $\mathcal{U}^{\mathrm{TV}}$.

We assume that $\boldsymbol{w}_k^{\mathrm{TV}}$ is subject to a Gaussian distribution with zero mean and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}^{\mathrm{TV}}$, which is denoted by $\boldsymbol{w}_k^{\mathrm{TV}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\boldsymbol{w}}^{\mathrm{TV}}\right)$. We also consider sensor noise in the measurement of the TV state, i.e.,

$$\hat{\boldsymbol{x}}_0^{\mathrm{TV}} = \boldsymbol{x}_0^{\mathrm{TV}} + \boldsymbol{w}_0^{\mathrm{sens}} \tag{5.12}$$

where $\hat{\boldsymbol{x}}_0^{\mathrm{TV}}$ is the measured initial state of the TV by the EV. The sensor noise $\boldsymbol{w}_0^{\mathrm{sens}} = (w_{0,x}^{\mathrm{sens}}, w_{0,v_x}^{\mathrm{sens}}, w_{0,y}^{\mathrm{sens}}, w_{0,v_y}^{\mathrm{sens}})^\top$ is assumed to be a truncated Gaussian noise with $\boldsymbol{w}_0^{\mathrm{sens}} \in \mathcal{W}^{\mathrm{sens}}$ and $\boldsymbol{w}_0^{\mathrm{sens}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}^{\mathrm{sens}}\right)$, where $\mathcal{W}^{\mathrm{sens}}$ is a compact and convex set

Problem Statement

We now formulate the objective addressed in this section.

Objective 5.1. *The aim is to develop a trajectory planner for automated highway driving that exploits the advantages of SMPC while guaranteeing that no self-inflicted collisions occur.*

We approach Objective 5.1 by developing a safety strategy for SMPC that is applicable to automated vehicles, which is shown in the following.

5.2.3 Method

SMPC and failsafe trajectory planning both have their individual advantages, i.e., efficient trajectories in an uncertain environment and guaranteed safe motion planning, respectively. In the following, we present a combined SMPC and FTP framework, SMPC+FTP, which exploits advantages of both methods to plan efficient and safe trajectories for autonomous vehicles. This section introduces the setup of the SMPC+FTP framework and gives a proof for recursive feasibility.

Before presenting the SMPC+FTP method, we need to define requirements for a safe ego vehicle state as well as a safe input sequence

$$\boldsymbol{U}_{\text{safe}} = (\boldsymbol{u}_{\text{safe},0}, \boldsymbol{u}_{\text{safe},1}, ..., \boldsymbol{u}_{\text{safe},n_{\text{s}}})^\top \tag{5.13}$$

with $n_{\text{s}} + 1$ individual inputs. Note that n_{s} is not directly related to the MPC prediction horizon.

Definition 5.1 (Safe State). *The state of an ego vehicle, fully located in one lane, is considered to be safe if there is no lateral vehicle motion, i.e., $\psi^{\text{ra}} = 0$, and if the ego vehicle velocity is lower than the velocity of the target vehicle in front on the same lane (or if the ego vehicle velocity is zero). The set of safe states is indicated by $\mathcal{X}_{\text{safe}}$.*

Definition 5.2 (Safe Input Sequence). *An input sequence $\boldsymbol{U}_{\text{safe}}$* (5.13) *is considered safe if consecutively applying all elements of $\boldsymbol{U}_{\text{safe}}$ results in a state trajectory that avoids collisions, i.e., $\boldsymbol{x}_t \in \mathcal{X}_{\text{safe}},\ \forall t \in \mathbb{I}_{1,n_{\text{s}}+1}$, and eventually leads to zero velocity, i.e., $v_{n_{\text{s}}+1} = 0$.*

The definition of safe states and safe input sequences requires assumptions for TVs.

Assumption 5.1. *Target vehicles adhere to the traffic rules.*

Assumption 5.2. *The upper bound of the ego vehicle deceleration is at least as large as the upper bound of the target vehicle deceleration.*

Based on Assumptions 5.1 and 5.2 and given a safe initial EV state, there exists a safe input sequence $\boldsymbol{U}_{\text{safe}}$, consisting of deceleration and zero steering, which results in an EV zero velocity state in the current EV lane, i.e., zero velocity in x-direction and y-direction. TVs behaving against traffic rules cannot be reliably accounted for by any prediction and the deceleration assumption is necessary to avoid colliding with a braking TV in front.

At the initialization of each optimal control problem, the current EV state $\boldsymbol{x}_0$ and the current TV state $\boldsymbol{x}_0^{\text{TV}}$ are known to the EV. Additionally, a safe input sequence $\boldsymbol{U}_{\text{safe}}$ is available from the SMPC+FTP problem solved at the previous time step. Later, we will focus on obtaining a safe input sequence for the SMPC+FTP iteration at the next time step, given the safe input sequence of the current time step.

The SMPC+FTP method uses both SMPC and FTP, i.e., at every time step an SMPC optimal control problem and an FTP optimal control problem are solved. The general idea is that the first input $\boldsymbol{u}_{\text{SMPC},0}$ of the SMPC input sequence $\boldsymbol{U}_{\text{SMPC}} = (\boldsymbol{u}_{\text{SMPC},0}, ..., \boldsymbol{u}_{\text{SMPC},N-1})^\top$ must only be applied if, based on the first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$, a failsafe trajectory can be found. Compared to regular SMPC methods, this approach guarantees that applying the optimistic SMPC input $\boldsymbol{u}_{\text{SMPC},0}$ does not lead to unsafe behavior. The algorithm outline is shown in Figure 5.1.

SMPC

In the first phase of SMPC+FTP, an SMPC problem is solved on a finite horizon N_{SMPC}, yielding the input sequence $\boldsymbol{U}_{\text{SMPC}} = (\boldsymbol{u}_{\text{SMPC},0}, ..., \boldsymbol{u}_{\text{SMPC},N_{\text{SMPC}}-1})^\top$. This SMPC optimization takes into account the uncertain environment and constraints due to other traffic participants, i.e., target vehicles. Collision constraints are formulated as chance-constraints,

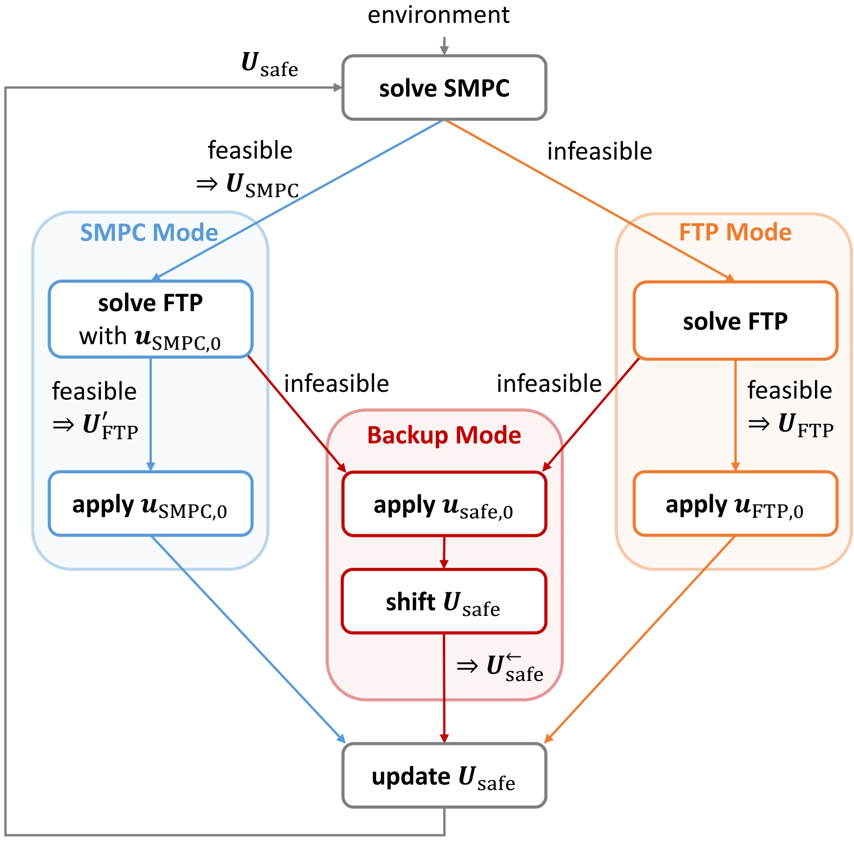

environment
$\boldsymbol{U}_{\text{safe}}$
solve SMPC
feasible
$\Rightarrow \boldsymbol{U}_{\text{SMPC}}$
infeasible
SMPC Mode
solve FTP
with $\boldsymbol{u}_{\text{SMPC},0}$
feasible
$\Rightarrow \boldsymbol{U}'_{\text{FTP}}$
apply $\boldsymbol{u}_{\text{SMPC},0}$
infeasible
infeasible
Backup Mode
apply $\boldsymbol{u}_{\text{safe},0}$
shift $\boldsymbol{U}_{\text{safe}}$
$\Rightarrow \overleftarrow{\boldsymbol{U}}_{\text{safe}}$
FTP Mode
solve FTP
feasible
$\Rightarrow \boldsymbol{U}_{\text{FTP}}$
apply $\boldsymbol{u}_{\text{FTP},0}$
update $\boldsymbol{U}_{\text{safe}}$

Figure 5.1: SMPC+FTP procedure for each time step. Blue shows the ideal mode with an applied SMPC input, orange represents the safe alternative mode with an applied FTP input, and red indicates an infeasible FTP problem, which requires applying a safe backup input. © 2021 IEEE.

based on a probabilistic TV prediction. Therefore, the planned SMPC trajectory provides an efficient and optimistic future trajectory for the EV, as it is not required to avoid collision with TVs for worst-case scenarios.

FTP

The second phase of SMPC+FTP is based on FTP to ensure that the planned EV trajectory remains safe. A failsafe MPC optimal control problem with the horizon N_{FTP} is solved, using a worst-case TV prediction, resulting in an input sequence $\boldsymbol{U}_{\text{FTP}} = (\boldsymbol{u}_{\text{FTP},0}, ..., \boldsymbol{u}_{\text{FTP},N_{\text{FTP}}-1})^\top$. The failsafe trajectory is required to avoid collision with the worst-case TV prediction and after applying the full failsafe input sequence $\boldsymbol{U}_{\text{FTP}}$, the terminal state $\boldsymbol{x}_{N_{\text{FTP}}}$ must be a safe state according to Definition 5.1. The exact FTP formulation depends on the feasibility of the SMPC optimal control problem.

Feasible SMPC (SMPC mode). If the SMPC optimal control problem yields a solution, FTP is used to decide whether applying the first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$ is safe. Therefore, an FTP optimal control problem is formulated starting with the EV state obtained by applying the first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$, i.e., the initial FTP optimal control problem state is

$$\boldsymbol{x}_0' = \boldsymbol{f}\left(\boldsymbol{x}_0, \boldsymbol{x}_0, \boldsymbol{u}_{\text{SMPC},0}\right) \tag{5.14}$$

with $\boldsymbol{f}\left(\boldsymbol{x}_0, \boldsymbol{x}_0, \boldsymbol{u}_{\text{SMPC},0}\right)$ according to (5.6).

If feasible, the FTP optimal control problem yields a failsafe input sequence $\boldsymbol{U}_{\text{FTP}}'$, based on $\boldsymbol{x}_0'$. Therefore, the first element $\boldsymbol{u}_{\text{SMPC},0}$ of the SMPC input sequence is applied safely, as shown by the blue path in Figure 5.1. The resulting new safe input sequence is given by

$$\boldsymbol{U}_{\text{safe}} = \left(\boldsymbol{U}_{\text{FTP}}', \boldsymbol{U}_{\text{brake}}\right) \tag{5.15a}$$

$$\boldsymbol{U}_{\text{brake}} = \left(\left(a_{\min},\, 0\right)^\top, \left(a_{\min},\, 0\right)\top, \dots\right) \tag{5.15b}$$

where $a_{\min}$ is the maximal deceleration and $\boldsymbol{U}_{\text{brake}}$ is a braking sequence to bring the EV to a standstill. The safe input sequence $\boldsymbol{U}_{\text{safe}}$ ensures a safe state after the full failsafe input sequence $\boldsymbol{U}_{\text{FTP}}'$ was applied and then initiates braking to reach zero velocity. Note that $a_{\min}$ is only applied in $\boldsymbol{U}_{\text{brake}}$ until a standstill is reached; subsequently no deceleration is applied.

Infeasible SMPC (FTP mode). If the SMPC optimal control problem is infeasible, the FTP optimal control problem is solved with initial state $\boldsymbol{x}_0$ for the FTP optimal control problem. If an FTP solution $\boldsymbol{U}_{\text{FTP}}$ is found, the first element of $\boldsymbol{U}_{\text{FTP}}$, i.e., $\boldsymbol{u}_{\text{FTP},0}$, is applied, as indicated by the orange path in Figure 5.1. The updated safe input sequence follows from

$$\boldsymbol{U}_{\text{safe}} = \left(\boldsymbol{U}_{\text{FTP},1:N_{\text{FTP}}-1}, \boldsymbol{U}_{\text{brake}}\right) \tag{5.16}$$

with $\boldsymbol{U}_{\text{brake}}$ according to (5.15b) where

$$\boldsymbol{U}_{\text{FTP},1:N_{\text{FTP}}-1} = \left(\boldsymbol{u}_{\text{FTP},1}, ..., \boldsymbol{u}_{\text{FTP},N_{\text{FTP}}-1}\right) \tag{5.17}$$

consists of all input elements of $\boldsymbol{U}_{\text{FTP}}$ except the first input $\boldsymbol{u}_{\text{FTP},0}$.

Infeasible FTP (Backup mode). In case of an infeasible FTP optimal control problem, no new input is generated at the current time step t. However, by definition the safe input sequence obtained at the previous time step $t-1$ remains safe for the current time step t. Therefore, in case that no solution exists to the FTP optimal control problem, the first element of the still valid, safe input sequence $\boldsymbol{U}_{\text{safe}}$ is applied, which is denoted by $\boldsymbol{u}_{\text{safe},0}$. This procedure is highlighted in red in Figure 5.1.

Continuously applying the elements of $\boldsymbol{U}_{\text{safe}}$ results in a safe trajectory according to Definition 5.2. If the FTP optimal control problem remains infeasible for consecutive time steps, multiple subsequent input elements of a single safe input sequence are potentially applied until the FTP optimal control problem becomes feasible again.

This procedure requires shifting $\boldsymbol{U}_{\text{safe}}$ after each SMPC+FTP iteration where the FTP optimal control problem was infeasible, i.e., if the first input element $\boldsymbol{u}_{\text{safe},0}$ of $\boldsymbol{U}_{\text{safe}}$ was applied. The shifted updated input sequence is obtained by

$$\boldsymbol{U}_{\text{safe}}^{\leftarrow} = \boldsymbol{U}_{\text{safe}} \begin{bmatrix} \boldsymbol{0}_{n_\text{s}} \\ \boldsymbol{I}_{n_\text{s}} \end{bmatrix} = \left(\boldsymbol{u}_{\text{safe},1}, \boldsymbol{u}_{\text{safe},2}, ..., \boldsymbol{u}_{\text{safe},n_\text{s}}\right) \tag{5.18}$$

with $\boldsymbol{U}_{\text{safe}} \in \mathbb{R}^{2\times(n_\text{s}+1)}$, identity matrix $\boldsymbol{I}_{n_\text{s}} \in \mathbb{R}^{n_\text{s}\times n_\text{s}}$, and $\boldsymbol{0}_{n_\text{s}} \in \mathbb{R}^{1\times n_\text{s}}$. The shifted safe input sequence $\boldsymbol{U}_{\text{safe}}^{\leftarrow}$ consists of all elements of $\boldsymbol{U}_{\text{safe}}$ except the already applied input $\boldsymbol{u}_{\text{safe},0}$.

Then, the safe input sequence is updated at the end of the SMPC+FTP iteration by selecting

$$\boldsymbol{U}_{\text{safe}} := \boldsymbol{U}_{\text{safe}}^{\leftarrow}, \tag{5.19}$$

which initializes the safe input sequence for the next SMPC+FTP iteration.

Summary of SMPC+FTP

Within the SMPC+FTP method, four cases are considered. These cases are summarized in the following.

SMPC and FTP feasible (SMPC mode). At time step t, the first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$, obtained at time step t, is applied and a new safe input sequence $\boldsymbol{U}_{\text{safe}}$ is obtained according to (5.15).

SMPC infeasible and FTP feasible (FTP mode). At time step t, the first FTP input $\boldsymbol{u}_{\text{FTP},0}$, obtained at time step t, is applied and a new safe input sequence $\boldsymbol{U}_{\text{safe}}$ is obtained according to (5.16).

SMPC feasible and FTP infeasible (Backup mode). At time step t, no new input sequence is obtained. The first input element of the safe input sequence $\boldsymbol{u}_{\text{safe},0}$, obtained at time step $t-1$, is applied. The safe input sequence $\boldsymbol{U}_{\text{safe}}$ remains valid for the next time step $t+1$ and is updated according to (5.19).

SMPC infeasible and FTP infeasible (Backup mode). As in the previous case, at time step t, no new input sequence is obtained. The input $\boldsymbol{u}_{\text{safe},0}$, obtained at time step $t-1$, is applied and $\boldsymbol{U}_{\text{safe}}$ is generated based on (5.19) for the next time step $t+1$.

Following this procedure, in regular cases the SMPC inputs are applied, resulting in efficient performance, whereas FTP guarantees safety for all possible cases, including rare events.

h

Recursive Feasibility

A disadvantage of various SMPC algorithms is that recursive feasibility of the optimal control problem cannot be guaranteed. In this section, recursive feasibility of the SMPC+FTP method is proved, which follows directly from the design of the proposed approach. We use Definition 2.5, i.e., if the optimization problem can be solved at step t, it can also be solved at step $t+1$ for all $t \in \mathbb{N}$. In this section, it is necessary to denote the time step t. The safe input sequence updated at time step t is denoted by $\boldsymbol{U}_{\text{safe},t}$.

Definition 5.3 (Safe Feasible Trajectory)**.** *Let there exist a safe set $\mathcal{X}_{\text{safe}}$ and let $\mathcal{X}_{\text{f}}$ be a control invariant set. Let $\boldsymbol{\chi}_{\boldsymbol{x}_t}^{\boldsymbol{U}_t} = (\boldsymbol{x}_t, ..., \boldsymbol{x}_{t+N})$ denote a trajectory starting at initial state $\boldsymbol{x}_t$ at time step t with N trajectory steps obtained by applying the input sequence $\boldsymbol{U}_t = (\boldsymbol{u}_t, ..., \boldsymbol{u}_{t+N-1})$ with $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t)$. Then, the set $\boldsymbol{\Gamma}_t$ of safe feasible trajectories, eventually converging into the set $\mathcal{X}_{\text{f}}$, is defined as*

$$\boldsymbol{\Gamma}_t = \left\{ \boldsymbol{\chi}_{\boldsymbol{x}_t}^{\boldsymbol{U}_t} \,\middle|\, \boldsymbol{x}_{t+i} \in \mathcal{X}_{\text{safe}},\ i \in \mathbb{I}_{0,N},\ \boldsymbol{x}_{t+N} \in \mathcal{X}_{\text{f}} \right\}. \tag{5.20}$$

A safe feasible trajectory satisfies all constraints given by $\mathcal{X}_{\text{safe}}$ and ends in the control invariant set $\mathcal{X}_{\text{f}}$.

Assumption 5.3. *The ego vehicle prediction models* (5.3) *and* (5.6) *correspond to the dynamics of the real system. The target vehicle model* (5.9) *represents an over-approximation of the real target vehicle dynamics.*

Here, over-approximation means that the possible states reachable with the TV model include all possible states obtained with the real TV dynamics.

Assumption 5.4. *At the initial time step $t = 0$, the initial ego vehicle state is safe and there exists a known initial safe input sequence $\boldsymbol{U}_{\text{safe,init}}$, such that $\boldsymbol{\chi}_{\boldsymbol{x}_0}^{\boldsymbol{U}_{\text{safe,init}}}$ is a safe feasible trajectory, i.e., $\boldsymbol{\chi}_{\boldsymbol{x}_0}^{\boldsymbol{U}_{\text{safe,init}}} \in \boldsymbol{\Gamma}_0$.*

We now show recursive feasibility of the proposed method.

Theorem 5.1. *Let Assumptions 5.3 and 5.4 hold. Then, for the SMPC+FTP approach there exists a feasible trajectory $\boldsymbol{\chi}_{\boldsymbol{x}_t}^{\boldsymbol{U}_t} \in \boldsymbol{\Gamma}_t$ that is guaranteed to be safe at all time steps $t \in \mathbb{N}$.*

Proof. Recursive feasibility is proved by induction by showing that $\boldsymbol{\Gamma}_t \neq \emptyset \Rightarrow \boldsymbol{\Gamma}_{t+1} \neq \emptyset$ for all $t \in \mathbb{N}$.

At time step $t = 0$, it holds that $\boldsymbol{\chi}_{\boldsymbol{x}_0}^{\boldsymbol{U}_{\text{safe,init}}} \in \boldsymbol{\Gamma}_0$, i.e., an initially safe trajectory exists according to Assumption 5.4. If the FTP optimal control problem can be solved at step $t = 0$, a new safe input set $\boldsymbol{U}_{\text{safe},0}$ is obtained according to (5.15) or (5.16). This new safe input set $\boldsymbol{U}_{\text{safe},0}$ remains valid at step $t = 1$ and ensures that a safe trajectory exists, i.e., $\boldsymbol{\chi}_{\boldsymbol{x}_1}^{\boldsymbol{U}_{\text{safe},0}} \in \boldsymbol{\Gamma}_1$. If the FTP optimal control problem is infeasible at step $t = 0$, the shifted previous safe input set remains valid, i.e., $\boldsymbol{U}_{\text{safe},0} = \boldsymbol{U}^{\leftarrow}_{\text{safe,init}}$. In this case, the shifted safe input set $\boldsymbol{U}_{\text{safe},0} = \boldsymbol{U}^{\leftarrow}_{\text{safe,init}}$ guarantees that $\boldsymbol{\chi}_{\boldsymbol{x}_1}^{\boldsymbol{U}_{\text{safe},0}} \in \boldsymbol{\Gamma}_1$. Therefore, $\boldsymbol{\Gamma}_0 \neq \emptyset \Rightarrow \boldsymbol{\Gamma}_1 \neq \emptyset$.

For $t = 1$ it holds that $\boldsymbol{\chi}_{\boldsymbol{x}_1}^{\boldsymbol{U}_{\text{safe},0}} \in \boldsymbol{\Gamma}_1$. A feasible FTP optimal control problem yields the new safe input sequences $\boldsymbol{U}_{\text{safe},1}$, such that there exists a safe trajectory $\boldsymbol{\chi}_{\boldsymbol{x}_2}^{\boldsymbol{U}_{\text{safe},1}} \in \boldsymbol{\Gamma}_2$. If the FTP optimal control problem is infeasible, reusing the still valid previous safe input set $\boldsymbol{U}_{\text{safe},0}$, i.e., setting $\boldsymbol{U}_{\text{safe},1} = \boldsymbol{U}^{\leftarrow}_{\text{safe},0}$, ensures that $\boldsymbol{\chi}_{\boldsymbol{x}_2}^{\boldsymbol{U}_{\text{safe},1}} \in \boldsymbol{\Gamma}_2$.

For time step $t \geq 2$, it holds that $\boldsymbol{\chi}_{t}^{\boldsymbol{U}_{\text{safe},t-1}} \in \boldsymbol{\Gamma}_t$. If the FTP optimal control problem is feasible, this yields the new safe input sequences $\boldsymbol{U}_{\text{safe},t}$, such that there exists a safe trajectory $\boldsymbol{\chi}_{\boldsymbol{x}_{t+1}}^{\boldsymbol{U}_{\text{safe},t}} \in \boldsymbol{\Gamma}_{t+1}$. If the FTP optimal control problem is infeasible, the previous safe input set $\boldsymbol{U}_{\text{safe},t-1}$ is still valid and choosing $\boldsymbol{U}_{\text{safe},t} = \boldsymbol{U}^{\leftarrow}_{\text{safe},t-1}$ ensures that $\boldsymbol{\chi}_{\boldsymbol{x}_{t+1}}^{\boldsymbol{U}_{\text{safe},t}} \in \boldsymbol{\Gamma}_{t+1}$.

Therefore, $\boldsymbol{\chi}_{\boldsymbol{x}_{t+1}}^{\boldsymbol{U}_{\text{save},t}} \in \boldsymbol{\Gamma}_{t+1}$ holds for all $t \in \mathbb{N}$, i.e., the proposed method is safe and recursively feasible. □

Note that the worst-case behavior of the TVs depends on the traffic rules. Therefore, safety and recursive feasibility of the SMPC+FTP method can only be guaranteed if surrounding TVs adhere to the underlying traffic rules, as stated in Assumption 5.1. However, no specific traffic rules are required to prove Theorem 5.1.

The two MPC optimal control problems, SMPC and FTP, are solved successively. In the following, the respective optimal control problems are derived.

5.2.4 Method Details - Stochastic MPC

SMPC solves an optimal control problem with chance constraints, accounting for TV uncertainty, depending on a risk factor β. First, a safety area is defined around each predicted TV state, which accounts for the EV and TV shape. Then, this safety area is increased to account for TV uncertainty, given a predefined risk parameter. Eventually, a linear constraint is generated for each TV, depending on the positioning of the EV and the TV.

Deterministic Target Vehicle Prediction

For SMPC a simple TV prediction is applied, representing the most likely TV behavior with $\boldsymbol{w}_k^{\text{TV}} = \boldsymbol{0}$, i.e., $\boldsymbol{u}_k^{\text{TV}} = \tilde{\boldsymbol{u}}_k^{\text{TV}}$. It is assumed that the current TV maneuver continues for the prediction horizon N_{SMPC}. Therefore, TV model (5.9) is applied where the TV reference $\boldsymbol{x}_{\text{ref},k}^{\text{TV}}$ depends on the current TV maneuver. The reference velocity $v_{x,\text{ref},k}^{\text{TV}}$ is set to the current TV velocity $v_{x,0}^{\text{TV}}$. The TV reference lateral velocity is chosen to be $v_{y,\text{ref},k}^{\text{TV}} = 0$. The reference lateral position $y_{\text{ref},k}^{\text{TV}}$ is the current TV lane center. A new reference lane is selected if part of the TV shape lies in this adjacent lane and the lateral velocity moves the TV towards this adjacent lane.

Target Vehicle Safety Area

Collisions with TVs are avoided by ensuring the necessary distance between the EV and TV. Here, a safety rectangle around the TV is defined. While it is possible to choose other shapes, rectangles makes it possible to easily generate linear constraints, as described later in this section.

The safety rectangle with length a_{r} and width b_{r} is illustrated in Figure 5.2. In order to

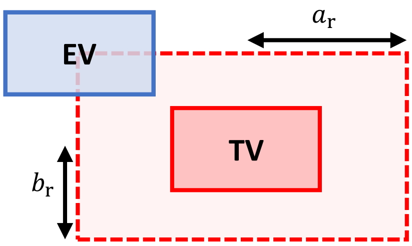


Figure 5.2: Target vehicle safety rectangle. © 2021 IEEE.

ensure that the vehicle shapes do not intersect, the vehicle centers need to be distanced at least by the vehicle length l_veh and width w_veh. For the safety rectangle width this yields

$$b_\mathrm{r} = w_\mathrm{veh} + \iota_\mathrm{safe} \tag{5.21}$$

where ι_safe is a possible additional safety margin.

Calculating the safety rectangle length a_r requires a velocity dependent part $\tilde{a}_\mathrm{r}\left(\boldsymbol{x}, \boldsymbol{x}^\mathrm{TV}\right)$, compensating for a potential velocity difference between the EV and the TV, resulting in

$$a_\mathrm{r} = l_\mathrm{veh} + \iota_\mathrm{safe} + \tilde{a}_\mathrm{r}\left(\boldsymbol{x}, \boldsymbol{x}^\mathrm{TV}\right). \tag{5.22}$$

The velocity dependent part $\tilde{a}_\mathrm{r}$ needs to account for the difference in traveled distance between the EV and TV if both vehicles initiate maximal braking, e.g., in an emergency braking scenario. Here, in addition to Assumption 5.2, zero reaction time is assumed. The traveled distance Δx of a vehicle until standstill is described by

$$\begin{aligned} \Delta x\left(t_\mathrm{stop}\right) &= v_x t_\mathrm{stop} + 0.5 a_\mathrm{min}\left(t_\mathrm{stop}\right)^2 \\ &= -\frac{1}{a_\mathrm{min}}\left(v_x\right)^2 && \text{(5.23a)} \\ t_\mathrm{stop} &= -\frac{v_x}{a_\mathrm{min}} && \text{(5.23b)} \end{aligned}$$

with maximal longitudinal deceleration a_min, time to standstill t_stop, and initial EV and TV velocity v and v_x^TV, respectively. Based on the difference in traveled distance

$$\Delta x^\mathrm{EV}\left(t_\mathrm{stop}^\mathrm{EV}\right) - \Delta x^\mathrm{TV}\left(t_\mathrm{stop}^\mathrm{TV}\right) = -\frac{1}{a_\mathrm{min}}\left(v^2 - \left(v_x^\mathrm{TV}\right)^2\right), \tag{5.24}$$

assuming similar maximal deceleration for the EV and TV, the velocity dependent safety distance is obtained by

$$\tilde{a}_\mathrm{r}\left(\boldsymbol{x}, \boldsymbol{x}^\mathrm{TV}\right) = -\frac{1}{2a_\mathrm{min}} \max\left\{0, \left(v^2 - \left(v_x^\mathrm{TV}\right)^2\right)\right\} \tag{5.25}$$

where the max-operator ensures that the safety rectangle length does not decrease for $v_x^\mathrm{TV} > v$.

For the SMPC optimal control problem, the safety rectangle is calculated for prediction time step k, based on the TV prediction $\boldsymbol{x}_k^\mathrm{TV}$. However, only the initial EV state $\boldsymbol{x}_0$ is considered in the velocity depended part $\tilde{a}_\mathrm{r}$. This is necessary in order to generate linear safety constraints. The resulting safety rectangle parameters are

$$\begin{aligned} b_{\mathrm{r},k} &= w_\mathrm{veh} + \iota_\mathrm{safe} && \text{(5.26a)} \\ a_{\mathrm{r},k} &= l_\mathrm{veh} + \iota_\mathrm{safe} + \tilde{a}_\mathrm{r}\left(\boldsymbol{x}_0, \boldsymbol{x}_k^\mathrm{TV}\right). && \text{(5.26b)} \end{aligned}$$

Chance Constraint Reformulation

The TV safety rectangle given by (5.26) does not account for TV uncertainty. In the following, the safety rectangle is enlarged, depending on the TV uncertainty and a risk parameter β. The chance constraint, similar to (5.1e), is given by

$$\Pr\left(\boldsymbol{x}_k \in \mathcal{X}'_{k,\text{safe}}\left(\boldsymbol{w}_k^{\text{TV}}\right)\right) \geq \beta \tag{5.27}$$

where the safe set $\mathcal{X}'_{k,\text{safe}}\left(\boldsymbol{w}_k^{\text{TV}}\right)$ for the EV state depends on the previously defined safety rectangle parameters of (5.26) and the TV uncertainty $\boldsymbol{w}_k^{\text{TV}}$.

The chance constraint (5.27) cannot be handled by a solver directly. We derive a deterministic approximation for this probabilistic expression, which is inspired by the SMPC approach [50].

According to (5.9) the TV state follows

$$\boldsymbol{x}_{k+1}^{\text{TV}} = \boldsymbol{A}\boldsymbol{x}_k^{\text{TV}} + \boldsymbol{B}\boldsymbol{K}\left(\boldsymbol{x}_k^{\text{TV}} - \boldsymbol{x}_{\text{ref},k}^{\text{TV}}\right) + \boldsymbol{B}\boldsymbol{w}_k^{\text{TV}}, \tag{5.28}$$

while the predicted TV state is given by

$$\hat{\boldsymbol{x}}_{k+1}^{\text{TV}} = \boldsymbol{A}\hat{\boldsymbol{x}}_k^{\text{TV}} + \boldsymbol{B}\boldsymbol{K}\left(\hat{\boldsymbol{x}}_k^{\text{TV}} - \boldsymbol{x}_{\text{ref},k}^{\text{TV}}\right), \tag{5.29}$$

yielding the prediction error

$$\boldsymbol{e}_k = \hat{\boldsymbol{x}}_k^{\text{TV}} - \boldsymbol{x}_k^{\text{TV}}. \tag{5.30}$$

Similar as described in Section 2.2.4, the TV prediction (5.29) is now split into a deterministic and a stochastic part

$$\hat{\boldsymbol{x}}_{k+1}^{\text{TV}} = \boldsymbol{x}_{k+1}^{\text{TV}} + \left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)\boldsymbol{e}_k - \boldsymbol{B}\boldsymbol{w}_k^{\text{TV}} = \boldsymbol{x}_{k+1}^{\text{TV}} + \boldsymbol{e}_{k+1} \tag{5.31}$$

which results in the prediction error update

$$\boldsymbol{e}_{k+1} = \left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)\boldsymbol{e}_k - \boldsymbol{B}\boldsymbol{w}_k^{\text{TV}}. \tag{5.32}$$

Given the sensor noise $\boldsymbol{w}_0^{\text{sens}}$ according to (5.12), the initial error follows $\boldsymbol{e}_0 \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_0^{\boldsymbol{e}}\right)$ with $\boldsymbol{\Sigma}_0^{\boldsymbol{e}} = \boldsymbol{\Sigma}^{\text{sens}}$. As we consider Gaussian distributions, a recursive computation of the prediction error covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$ is possible, yielding

$$\boldsymbol{\Sigma}_{k+1}^{\boldsymbol{e}} = \boldsymbol{B}\boldsymbol{\Sigma}_{\boldsymbol{w}}^{\text{TV}}\boldsymbol{B}^\top + \left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)\boldsymbol{\Sigma}_k^{\boldsymbol{e}}\left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right)^\top. \tag{5.33}$$

Based on the prediction error covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$, the TV safety rectangle is increased. Given a predefined SMPC risk parameter β, the aim is to find a region around the predicted TV state that contains the true TV state with probability β. As the TV safety rectangle only considers positions, we define the reduced error $\tilde{\boldsymbol{e}}_k = \left(e_{x,k}, e_{y,k}\right)^\top$ with the reduced covariance matrix

$$\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}} = \text{diag}(\sigma_{x,k}^2, \sigma_{y,k}^2) \tag{5.34}$$

with variances $\sigma_{x,k}^2$ and $\sigma_{y,k}^2$ for the longitudinal and lateral TV position, corresponding to the first and third diagonal element of $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$.

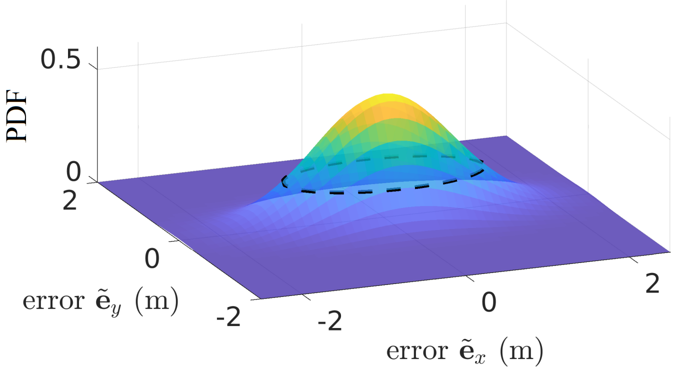


Figure 5.3: Exemplary bivariate Gaussian probability distribution function of the prediction error $\tilde{\boldsymbol{e}}$, including an isoline (dotted black line). © 2021 IEEE.

Lemma 5.1. *The reduced error covariance matrix $\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}$ corresponding to the position coordinates is obtained from $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$ by omitting the correlation to the respective velocities.*

Proof. The proof is given in Appendix C.2. □

The reduced error covariance matrix $\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}$ is now used to enlarge the safety rectangle to account for uncertainty. Note that the error covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$ is still required to compute (5.33).

The bivariate Gaussian distribution described by $\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}$ with mean $\boldsymbol{\mu} = (\mu_x, \mu_y)^\top = \mathbf{0}$ consists of independent random variables for longitudinal and lateral position. This allows us to find a confidence region around the predicted TV state mean, bounded by an ellipsoidal isoline enclosing the highest density region, as illustrated in Figure 5.3. The aim is to find an isoline that contains the prediction error with a probability according to risk parameter β. The isoline ellipse equation is denoted by

$$(\tilde{\boldsymbol{e}}_k - \boldsymbol{\mu})^\top \left(\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}\right)^{-1} (\tilde{\boldsymbol{e}}_k - \boldsymbol{\mu}) = \varkappa \tag{5.35a}$$

$$\frac{(e_{x,k} - \mu_x)^2}{\sigma_{x,k}^2} + \frac{(e_{y,k} - \mu_y)^2}{\sigma_{y,k}^2} = \varkappa \tag{5.35b}$$

with tolerance level $\varkappa$. The tolerance level $\varkappa$ depends on the risk parameter β and indicates the necessary constraint tightening in order to ensure that the prediction error remains below a probability β. The tolerance level $\varkappa$ is determined based on the cumulative distribution function $F(\varkappa, n_\varkappa)$ of the *chi-square distribution* $\chi^2_{n_\varkappa}$ with $n_\varkappa$ degrees of freedom. In this case, $n_\varkappa = 2$ as the reduced error $\tilde{\boldsymbol{e}}_k$ consists of two elements. Given the risk parameter β and the quantile function F^{-1} of the *chi-square distribution* χ^2_2, it follows that

$$\varkappa = F^{-1}(\beta, 2), \tag{5.36}$$

which ensures that the probability of the true TV state lying within the isoline is $\beta \cdot 100\,\%$. The ellipse semi-major and semi-minor axes are then given by

$$e_{\varkappa,x,k} = \sigma_{x,k}\sqrt{\varkappa} \tag{5.37a}$$

$$e_{\varkappa,y,k} = \sigma_{y,k}\sqrt{\varkappa}. \tag{5.37b}$$

While an ellipse, according to (5.35), describes the desired confidence region, the constraint generation method used in this chapter requires a rectangular TV safety area. We therefore over-approximate the ellipse by a rectangle. In order to include this uncertainty consideration in the rectangle parameters $a_{\mathrm{r},k}$ and $b_{\mathrm{r},k}$ of (5.26), the rectangle parameters are increased based on the ellipse semi-major axis $e_{\varkappa,x,k}$ and semi-minor axis $e_{\varkappa,x,k}$, resulting in

$$\begin{aligned} b_{\mathrm{r},k} &= w_{\mathrm{veh}} + \iota_{\mathrm{safe}} + e_{\varkappa,x,k} && (5.38\mathrm{a}) \\ a_{\mathrm{r},k} &= l_{\mathrm{veh}} + \iota_{\mathrm{safe}} + \tilde{a}_{\mathrm{r}}\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\mathrm{TV}}\right) + e_{\varkappa,y,k}. && (5.38\mathrm{b}) \end{aligned}$$

Note that this chance constraint reformulation is not typical for SMPC, but was specifically designed for this application task. The updated safety rectangle parameters are now used to generate the safety constraints for the SMPC optimal control problem.

SMPC Constraint Generation

Given the safety rectangles for each TV, linear constraints to avoid collisions can be defined for each prediction step and for each TV. Each linear constraint has the form

$$0 \geq q_y\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\mathrm{TV}}\right) y_k + q_x\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\mathrm{TV}}\right) x_k + q_t\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\mathrm{TV}}\right) \tag{5.39}$$

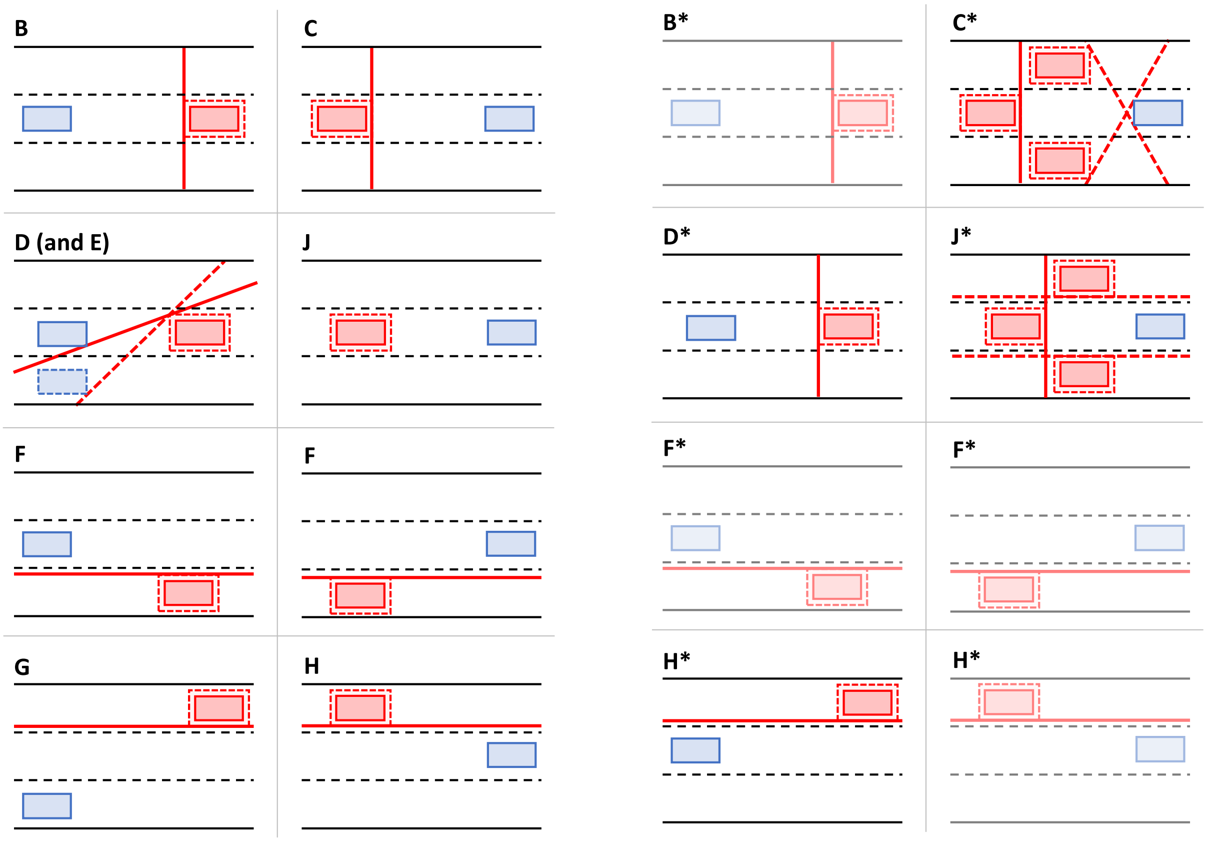
where q_y and q_x are the coefficients for the EV states y_k and x_k, and q_t is the intercept. The coefficients q_y, q_x, and q_t of the linear constraint depend on the current EV state $\boldsymbol{x}_0$ and the predicted mean TV states $\boldsymbol{x}_k^{\mathrm{TV}}$. This results in multiple constraint generation cases, extending the case differentiation in [53].

The cases are distinguished based on the initial vehicle configuration at the beginning of the optimal control problem, i.e., $k = 0$. While the predicted TV state $\boldsymbol{x}_k^{\mathrm{TV}}$ is considered to build the constraint (5.39) at prediction step k for a specific case, only the initial EV state $\boldsymbol{x}_0$ is considered in order to generate linear constraints.

We briefly discuss a shortened overview of constraint cases that are considered, summarized in Table 5.1. Example cases are illustrated in Figure 5.4a. A complete overview of cases, requirements, and constraint parameters q_x, q_y, q_t from (5.39) is found in Appendix C.3.

In summary, no constraints are generated if the longitudinal distance between the EV and TV is larger than ρ_{lar} (case A). If the EV is close enough to the TV (longitudinal distance smaller than ρ_{close}), overtaking is possible by employing an inclined constraint (cases D and E). If the TV is located behind the EV, no constraints are necessary as it is the responsibility of the TV to avoid a collision (case J). For all other cases, horizontal and vertical constraints are employed.

These constraint generation cases are now used to formulate safety constraints in the SMPC optimal control problem.



(a) Constraint generation cases for SMPC.

(b) Constraint generation cases for FTP.

Figure 5.4: Selected constraint generation cases. Driving direction is from left to right. The EV and TV are shown in blue and red, respectively. The dashed red line represents the safety area around the TV. © 2021 IEEE.

case	EV setting (w.r.t. TV)	SMPC	FTP
A, A*	large dist. ($> \rho_{\text{lar}}$)	no constraint	no constraint
B, B*	behind TV ($> \rho_{\text{close}}$)	vertical constraint	vertical constraint
C, C*	ahead of TV ($> \rho_{\text{close}}$)	vertical constraint	virtual TVs mixed constraints
D, D*	same lane as TV behind TV ($\leq \rho_{\text{close}}$)	inclined constraint	vertical constraint
E, E*	right lane next to TV behind TV ($\leq \rho_{\text{close}}$)	inclined constraint	vertical constraint
F, F*	left of TV close to TV ($\leq \rho_{\text{close}}$)	horizontal constraint	horizontal constraint
G, G*	2 lanes right of TV behind TV ($\leq \rho_{\text{close}}$)	horizontal constraint	horizontal constraint
H, H*	right of TV ahead of TV ($\leq \rho_{\text{close}}$)	horizontal constraint	horizontal constraint
J, J*	same lane as TV ahead of TV ($\leq \rho_{\text{close}}$)	no constraint	virtual TVs mixed constraints

Table 5.1: Constraint generation cases.

SMPC Optimal Control Problem

With the definition of the safety constraints, the deterministic optimal control problem replacing the SMPC problem is given by

$$\min_{\boldsymbol{U}} \sum_{k=1}^{N_{\text{SMPC}}} \left(\|\Delta \boldsymbol{x}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_{k-1}\|_{\boldsymbol{R}}^2 + \|\Delta \boldsymbol{u}_{k-1}\|_{\boldsymbol{S}}^2 \right) \tag{5.40a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}^{\text{d}}\left(\boldsymbol{x}_0, \boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{5.40b}$$

$$\boldsymbol{x}_{k+1}^{\text{TV}} = \boldsymbol{A}\boldsymbol{x}_k^{\text{TV}} + \boldsymbol{B}\tilde{\boldsymbol{u}}_k^{\text{TV}} \tag{5.40c}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N_{\text{SMPC}}-1} \tag{5.40d}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{0,N_{\text{SMPC}}} \tag{5.40e}$$

$$0 \geq q_y\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\text{TV}}\right) y_k + q_x\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\text{TV}}\right) x_k + q_t\left(\boldsymbol{x}_0, \boldsymbol{x}_k^{\text{TV}}\right), \qquad k \in \mathbb{I}_{0,N_{\text{SMPC}}} \tag{5.40f}$$

with $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k,\text{ref}}$, EV reference state $\boldsymbol{x}_{k,\text{ref}}$, and the linear function $\boldsymbol{f}^{\text{d}}$ according to (5.6). For the input difference $\Delta \boldsymbol{u}$, we set $\boldsymbol{u}_{-1}$ to the applied input of the previous time step. The cost function sum limits are shifted to include a terminal cost for $\boldsymbol{x}_N$. The weighting matrices are given by $\boldsymbol{Q}$, $\boldsymbol{S}$, and $\boldsymbol{R}$. We consider constant input constraints $\mathcal{U}$ according to (5.7) and state constraints $\mathcal{X}$ according to (5.8).

Remark 5.1. *For clarity, safety constraint* (5.40f) *only considers one TV. If multiple TVs need to be considered, an individual constraint* (5.40f) *must be generated for each TV.*

The resulting SMPC optimal control problem (5.40) is a quadratic program with linear constraints, accounting for uncertainty with the chance constraint reformulation described previously. This optimal control problem can be solved efficiently, where the major calculation steps to obtain the linear constraints (5.40f) are performed offline.

5.2.5 Method Details - Failsafe MPC

While the SMPC algorithm only accounts for all possible TV uncertainty in order to plan an optimistic trajectory, the backup FTP algorithm needs to consider worst-case uncertainty realizations. This is achieved based on reachability analysis. First, the worst-case TV occupancy prediction is determined. Then, linear safety constraints are generated. Eventually, given a safe invariant terminal set, the FTP optimal control problem is solved.

Target Vehicle Occupancy Prediction

Similar to the SMPC algorithm, a rectangular safety area surrounding each TV is defined. However, for the FTP the maximal reachable area needs to be determined. First, it is necessary to define certain traffic rules to which the TV adheres, according to Assumption 5.1:

- Road boundaries apply.
- Negative velocities are forbidden.
- Collisions with vehicles directly in front of the TV (in the same lane) must be avoided.
- Only a single lane change is allowed (within the prediction horizon).
- No lane change is allowed if the TV velocity is below a predefined minimal lane change velocity $v_{\text{LC,min}}$.
- No lane change is allowed if the distance to a vehicle on the new lane becomes too small.

As linear dynamics are assumed for the TV motion, the minimal and maximal possible TV inputs are used to determine the maximal reachable set, inspired by [4, 123, 152].

The set of all possible locations reachable for a TV at prediction step k is denoted by the reachable set $\mathcal{R}_k^{\text{TV}}$, including the TV and shape. While referring to $\mathcal{R}_k^{\text{TV}}$ as the reachable set of the TV, we additionally enlarge this set accounting for the EV shape. This is necessary as the set $\mathcal{R}_k^{\text{TV}}$ is later used to avoid collisions by keeping the EV center outside of $\mathcal{R}_k^{\text{TV}}$. Given the state sequence $\boldsymbol{\chi}_{\hat{\boldsymbol{x}}_0^{\text{TV}}}^{\boldsymbol{U}^{\text{TV}}}$ to the TV dynamics (5.9) starting at the initial state $\hat{\boldsymbol{x}}_0^{\text{TV}}$ applying an input sequence $\boldsymbol{U}_{0:k-1}^{\text{TV}} = \left(\boldsymbol{u}_0^{\text{TV}}, ..., \boldsymbol{u}_{k-1}^{\text{TV}}\right)$, we define the reachable set

$$\mathcal{R}_k^{\text{TV}} = \left\{ \boldsymbol{\chi}_{\hat{\boldsymbol{x}}_0^{\text{TV}}}^{\boldsymbol{U}^{\text{TV}}} \;\middle|\; \boldsymbol{u}_i^{\text{TV}} \in \mathcal{U}^{\text{TV}} \;\forall i \in \mathbb{I}_{0,k-1}, \;\; \hat{\boldsymbol{x}}_0^{\text{TV}} \in \mathcal{X}_0^{\text{TV}} \right\}. \tag{5.41}$$

The initial state for the reachable set $\mathcal{R}_k^{\text{TV}}$ is not the TV state $\boldsymbol{x}_0^{\text{TV}}$, but depends on the sensor uncertainty as well as the TV and EV shape. This initial set is given by

$$\begin{aligned} \mathcal{X}_0^{\text{TV}} = \Big\{ \hat{\boldsymbol{x}}_0^{\text{TV}} \;\Big|\; & \boldsymbol{x}_0^{\text{TV}} + \min_{\boldsymbol{w}_0^{\text{sens}} \in \mathcal{W}^{\text{sens}}} \left(\boldsymbol{w}_0^{\text{sens}}\right) - \left(l_{\text{veh}},\, 0,\, w_{\text{veh}},\, 0\right)^\top \leq \hat{\boldsymbol{x}}_0^{\text{TV}}, \\ & \hat{\boldsymbol{x}}_0^{\text{TV}} \leq \boldsymbol{x}_0^{\text{TV}} + \max_{\boldsymbol{w}_0^{\text{sens}} \in \mathcal{W}^{\text{sens}}} \left(\boldsymbol{w}_0^{\text{sens}}\right) + \left(l_{\text{veh}},\, 0,\, w_{\text{veh}},\, 0\right)^\top \Big\}. \end{aligned} \tag{5.42}$$

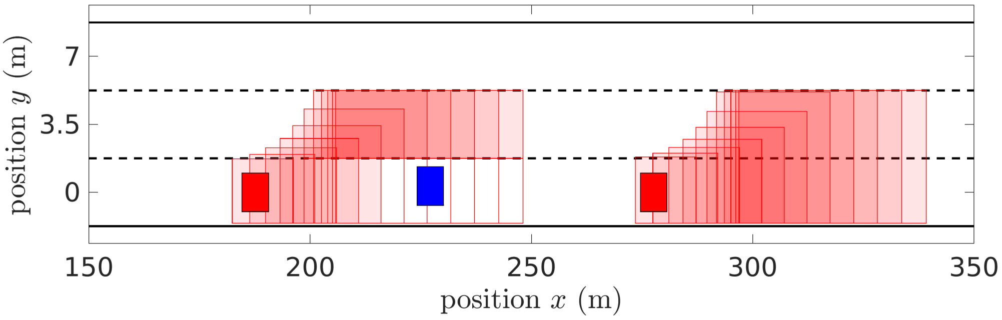


Figure 5.5: Target vehicle occupancy sets for multiple prediction steps with the EV in blue and TVs in red. Areas shaded in red depict areas possibly occupied by a TV. As the TVs must avoid collision with vehicles in front, the left TV cannot occupy the area in the EV lane close to the EV. A TV double lane change is not considered. © 2021 IEEE.

As we assume a linear TV prediction model, the reachable set $\mathcal{R}_k^{\mathrm{TV}}$ is calculated for prediction steps $k > 0$ by applying the maximal and minimal inputs $\boldsymbol{u}^{\mathrm{TV}} \in \mathcal{U}^{\mathrm{TV}}$ while adhering to traffic rules.

The reachable set is only calculated at discrete time steps. In order to account for a continuous system, the final reachable set $\overline{\mathcal{R}}_k^{\mathrm{TV}}$ is obtained by building a rectangular convex hull, covering two consecutive prediction steps, i.e.,

$$\overline{\mathcal{R}}_k^{\mathrm{TV}} = \operatorname{conv}\left(\mathcal{R}_{k-1}^{\mathrm{TV}}, \mathcal{R}_k^{\mathrm{TV}}\right) \tag{5.43}$$

where conv$(\cdot)$ denotes the convex hull operation.

A special case is considered if the TV is located behind the EV in the same lane. The TV must not collide with the EV in the same lane, however, the TV is allowed to switch lanes in order to pass the EV. Here, this is accounted for by treating this special case in the following way. Three placeholder TV reachable sets describe the possible TV behavior. The first placeholder TV reachable set is based in the EV lane such that collisions with the EV are avoided. The other two placeholder TV reachable sets cover the admissible adjacent lanes left and right of the EV, representing the reachable sets for a potential TV lane change. Figure 5.5 shows an example of areas possibly occupied by TVs for multiple prediction steps.

FTP Constraint Generation

Once the reachable sets $\overline{\mathcal{R}}_k^{\mathrm{TV}}$ for each TV are determined, linear constraints are generated. We again consider different cases regarding varying EV and TV positions. The cases are similar to those of SMPC with a few variations as stated in Table 5.1. FTP cases are denoted with an asterisk. Note that, in contrast to other parts of this thesis, the asterisk does not denote an optimal solution here. Exemplary FTP cases are illustrated in Figure 5.4b. Again, a complete overview of the FTP cases is found in Appendix C.3.

We briefly discuss the major differences to the SMPC constraint generation cases. Overtaking is not initiated given the vertical constraints in cases D* and E*. If the TV is located behind the EV, we consider possible TV lane changes by introducing placeholder TVs (cases C* and J*). Here, $\rho_{\mathrm{close}}^{\mathrm{FTP}}$ is used instead of ρ_{close}.

Overall, the constraints generated for FTP are more conservative than for SMPC. This is due to the FTP aim of finding a trajectory that ends in a safe state. This would be complicated by incentivizing FTP to plan overtaking maneuvers. Details on finding a safe terminal state for the FTP optimal control problem are given in the following.

Safe Invariant Terminal Set

In addition to the regular safety constraints, a safe invariant terminal set is required to ensure safe EV inputs after the finite MPC prediction horizon. The FTP inputs are designed in such a way that they remain safe over the prediction horizon. However, after N_{FTP} inputs are applied and no new FTP solution is obtained, an emergency strategy has to be applied to come to a standstill. This is achieved by braking, while maintaining a constant steering angle $\delta_{\text{f}} = 0$, according to (5.15) and (5.16). Therefore, the terminal state of the FTP optimal control problem needs to fulfill certain requirements. First, the vehicle orientation must be aligned with the road, i.e., $\psi^{\text{ra}} = 0$. This guarantees that braking and a constant steering angle $\delta_{\text{f}} = 0$ keep the EV within its current lane. Second, the distance to a TV in front of the EV must be large enough that no collision occurs if both vehicles initiate maximal deceleration. This is accounted for by

$$x_N \leq x_N^{\text{TV}} - \Delta s^{\text{ra}}_{N_{\text{FTP}},\text{min}} \tag{5.44a}$$

$$v_N \leq v_{N_{\text{FTP}},\text{max}} \tag{5.44b}$$

with the minimal terminal safety distance $\Delta s^{\text{ra}}_{N_{\text{FTP}},\text{min}}$ and the maximal terminal safety velocity

$$v_{N_{\text{FTP}},\text{max}} = v^{\text{TV}}_{N_{\text{FTP}},\text{min}} - \sqrt{2\Delta s^{\text{ra}}_{N_{\text{FTP}},\text{min}} a_{x,\text{min}}} \tag{5.45}$$

where $v^{\text{TV}}_{N_{\text{FTP}},\text{min}}$ is the lowest predicted longitudinal TV velocity. Both (5.44) and (5.45) combined ensure that the minimal terminal safety distance $\Delta s^{\text{ra}}_{N_{\text{FTP}},\text{min}}$ is large enough such that, given a maximal EV velocity $v_{N_{\text{FTP}},\text{max}}$, maximal deceleration of the EV guarantees collision avoidance for $k > N_{\text{FTP}}$. This less intuitive terminal constraint again has the advantage of yielding linear constraints.

FTP Optimal Control Problem

An optimal control problem with a similar structure compared to (5.40) is applied for the FTP, yielding

$$\min_{\boldsymbol{U}} \sum_{k=1}^{N_{\text{FTP}}} \left(\|\Delta \boldsymbol{x}_k\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}_{k-1}\|^2_{\boldsymbol{R}} + \|\Delta \boldsymbol{u}_{k-1}\|^2_{\boldsymbol{S}} \right) \tag{5.46a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}^{\text{d}}\left(\boldsymbol{x}_0, \boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{5.46b}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N_{\text{FTP}}-1} \tag{5.46c}$$

$$\boldsymbol{x}_k \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N_{\text{FTP}}} \tag{5.46d}$$

$$0 \geq q_y\left(\boldsymbol{x}_0, \overline{\mathcal{R}}^{\text{TV}}_k\right) y_k + q_x\left(\boldsymbol{x}_0, \overline{\mathcal{R}}^{\text{TV}}_k\right) x_k + q_t\left(\boldsymbol{x}_0, \overline{\mathcal{R}}^{\text{TV}}_k\right), \quad k \in \mathbb{I}_{0,N_{\text{FTP}}} \tag{5.46e}$$

$$x_N \leq x_N^{\text{TV}} - \Delta s^{\text{ra}}_{N_{\text{FTP}},\text{min}}, \quad v_N \leq v_{N_{\text{FTP}},\text{max}} \tag{5.46f}$$

with the linear function $\boldsymbol{f}^{\text{d}}$ according to (5.6). The safety constraint (5.40f) is now changed to constraint (5.46e), accounting for the worst-case TV uncertainty realizations. Similar to

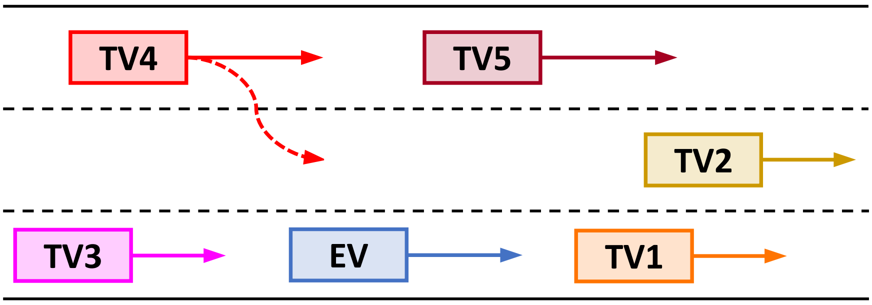


Figure 5.6: Setup for both investigated scenarios (regular and emergency scenario). © 2021 IEEE.

the SMPC optimal control problem, (5.46) is a quadratic program with linear constraints, which can be solved efficiently.

Remark 5.2. *Similar to Remark 5.1,* (5.46) *only considers one TV. If multiple TVs need to be considered, an individual constraints must be generated for each TV.*

5.2.6 Simulation Study

We evaluate the proposed SMPC+FTP algorithm in different settings. In the following, the simulation setup is introduced first. Then, SMPC+FTP is analyzed and compared to a standard SMPC approach and an FTP approach in two scenarios.

Simulation Setup

In this simulation section, we analyze the scenario illustrated in Figure 5.6. The EV is located on the right lane on a three-lane highway. We consider five TVs surrounding the EV on the highway. The goal for the EV is to maneuver safely and efficiently through traffic. The specific aims are to avoid collisions while maintaining a velocity close to a chosen reference velocity.

We consider two different scenarios:

1) Regular scenario: All TVs keep their initial velocities and lanes.

2) Emergency scenario: One of the TVs (TV5) performs an emergency braking maneuver. This causes TV4 to avoid TV5 by moving to the center lane. This is followed by a soft braking maneuver of TV1 to account for possible hazards. Eventually, TV4 moves to the left lane again to pass TV2.

The first scenario represents a regular scenario with no unexpected TV behavior. The second scenario covers a rare case, where a series of unexpected TV actions results in a challenging situation for the autonomous EV.

The simulations are carried out in Matlab using the `fmincon` solver on a computer with an AMD Ryzen 7 1700X processor. The algorithms are based on the NMPC toolbox [79]. In the following, setup parameters are introduced that remain constant throughout the different simulations. All quantities are given in SI units. Units are omitted if clear by context.

scalars	vectors	matrices
$w_{\text{lane}} = 3.5$	$\boldsymbol{u}_{\text{max}} = (5,\ 0.2)^\top$	$\boldsymbol{K} = \begin{bmatrix} 0 & -0.55 & 0 & 0 \\ 0 & 0 & -0.63 & -1.15 \end{bmatrix}$
$l_{\text{veh}} = 5$	$\boldsymbol{u}_{\text{min}} = (-9,\ -0.2)^\top$	$\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{w}}^{\text{TV}} = \text{diag}(0.44,\ 0.09)$
$w_{\text{veh}} = 2$	$\boldsymbol{u}_{\text{max}}^{\text{TV}} = (5,\ 0.4)^\top$	$\boldsymbol{Q} = \text{diag}(0,\ 0.2,\ 10,\ 0.25)$
$l_{\text{f}} = l_{\text{r}} = 2$	$\boldsymbol{u}_{\text{min}}^{\text{TV}} = (-9,\ -0.4)^\top$	$\boldsymbol{R} = \text{diag}(0.33,\ 5)$
$v_{\text{max}} = 35$	$\boldsymbol{w}_0^{\text{sens}\top} = (0.25,\ 0.03,\ 0.25,\ 0.03)^\top$	$\boldsymbol{S} = \text{diag}(0.33,\ 15)$

Table 5.2: General simulation parameters.

vehicle	initial state	vehicle	initial state
EV	$(0,\ 0,\ 0,\ 27)^\top$	TV3	$(-245,\ 20,\ 0,\ 0)^\top$
TV1	$(70,\ 20,\ 0,\ 0)^\top$	TV4	$(-35,\ 32,\ 7,\ 0)^\top$
TV2	$(125,\ 20,\ 3.5,\ 0)^\top$	TV5	$(40,\ 32,\ 7,\ 0)^\top$

Table 5.3: Initial vehicle states.

All MPC algorithms use a sampling time $\Delta t = 0.2\,\text{s}$ with SMPC horizon $N_{\text{SMPC}} = 10$ and the FTP horizon $N_{\text{FTP}} = 10$. The linearized, discrete-time EV prediction model and constraints follow (5.6)-(5.8), whereas the TV prediction model is given by (5.9)-(5.11). Table 5.2 shows the other main simulation parameters. The lane boundaries follow from the lane and vehicle width. Additionally, the safety parameters are $\iota_{\text{safe}} = 0.01$, $\rho_{\text{lar}} = 200$, $\rho_{\text{close}} = 90$, $\rho_{\text{close}}^{\text{FTP}} = \max\{10, |v_0 N_{\text{FTP}} \Delta t|\}$, $v_{\text{LC, min}} = 10$, and $\Delta s_{N,\text{min}}^{\text{ra}} = 22.5$.

In all scenarios, the initial EV reference is set to $(d_{\text{ref}}^{\text{ra}}, \psi^{\text{ra}}{}_{\text{ref}}, v_{\text{ref}}) = (0,\ 0,\ 27)$. While the reference orientation and velocity remain constant throughout the simulation, the EV reference for the lateral position is always set to the current EV lane center.

Whereas the MPC optimal control problems use the linearized, discrete-time prediction model (5.6), the inputs are applied to a simulation using the continuous-time system (5.3).

Given this simulation setup, we now investigate the individual scenarios and analyze the proposed SMPC+FTP method.

Regular Highway Scenario

We first analyze a regular highway scenario. The initial states of the vehicles are given in Table 5.3. The five TVs shown in Figure 5.6 all maintain their initial velocities and lanes, therefore, $\boldsymbol{x}_{\text{ref},k}^{\text{TV}} = \boldsymbol{x}_0^{\text{TV}}$.

In the following, the SMPC+FTP solution is shown in detail and comparisons are made to an SMPC and an FTP method.

SMPC+FTP. Applying the proposed SMPC+FTP approach to the regular highway scenario yields efficient EV behavior in traffic. The SMPC risk parameter is chosen to be

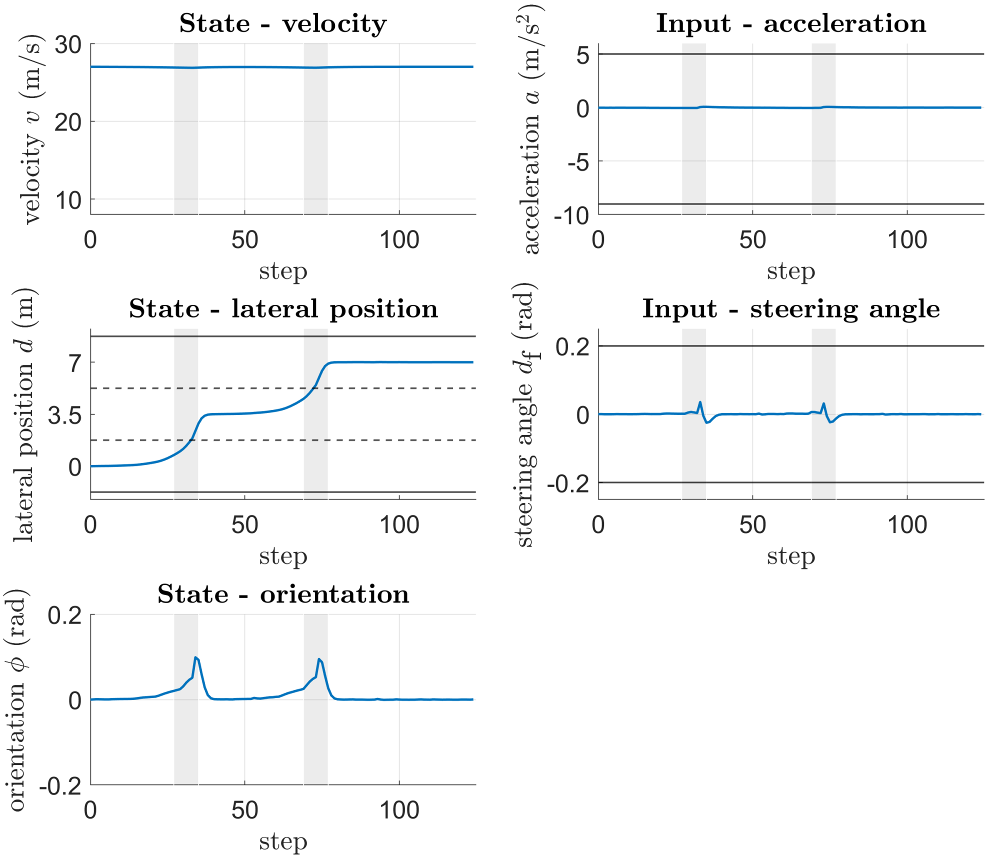


Figure 5.7: SMPC+FTP states and inputs for the regular scenario. Vehicle motion in the gray areas is illustrated in Figure 5.8. © 2021 IEEE.

$\beta = 0.8$. The inputs and important states are shown in Figure 5.7, vehicle motion is displayed in Figure 5.8.

The EV approaches TV1 due to the velocity difference. The EV then changes lanes to the center lane with a moderate steering angle of $\delta_\mathrm{f} < 0.04$. Once TV2 is reached, the EV again changes lanes and eventually passes TV2. The vehicle orientation remains at a limited level, i.e., $\psi^\mathrm{ra} < 0.11$. Throughout the scenario, the EV maintains the reference velocity, and acceleration inputs are small.

The average computation time to solve the SMPC and FTP optimal control problems are 0.11 s and 0.15 s, respectively. Lower computational effort is possible with other solvers. If applied in a setting that requires online computation, optimal control problems with computation times exceeding the requirements are considered as infeasible. In this case, the previously calculated, still valid safe input sequence would be used.

We will now take a closer look at the constraints for SMPC and FTP. SMPC constraints for time step $t = 22$ are illustrated in Figure 5.9. For TV1 in the same lane as the EV, an inclined constraint is generated (case D). At each prediction step, the constraint connects the initial EV shape with the TV1 safety rectangle at the predicted position. The predicted SMPC trajectory for the EV stays above the constraint line. It is to note that only the respective predicted state must satisfy the illustrated constraint. Predicted states farther in the future satisfy respective constraints depending on a TV safety rectangle for a predicted TV position farther ahead. For TV2 case E is active, also resulting in an inclined constraint.

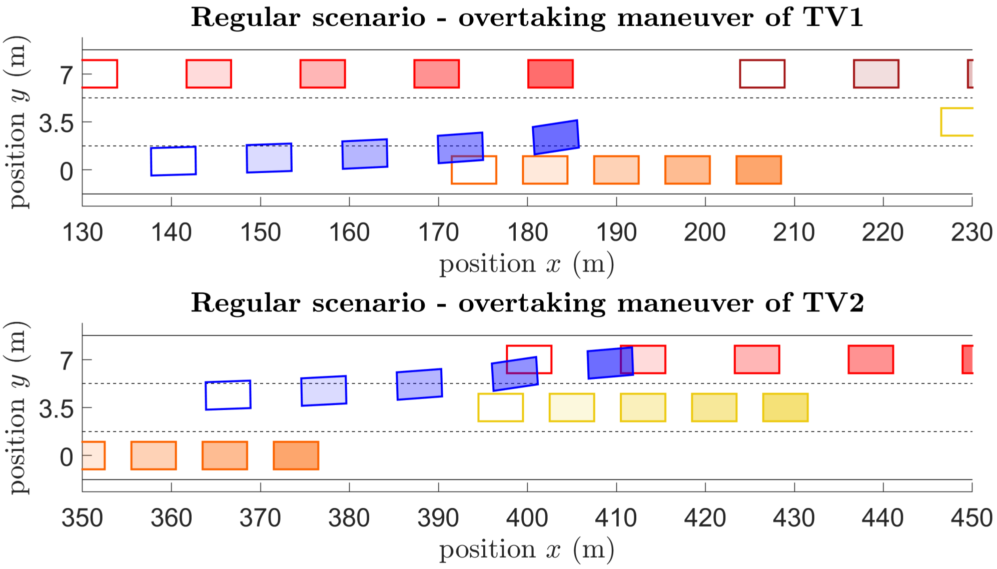


Figure 5.8: Sequences of the regular scenario with SMPC+FTP. Fading boxes show past states. The EV is shown in blue. © 2021 IEEE.

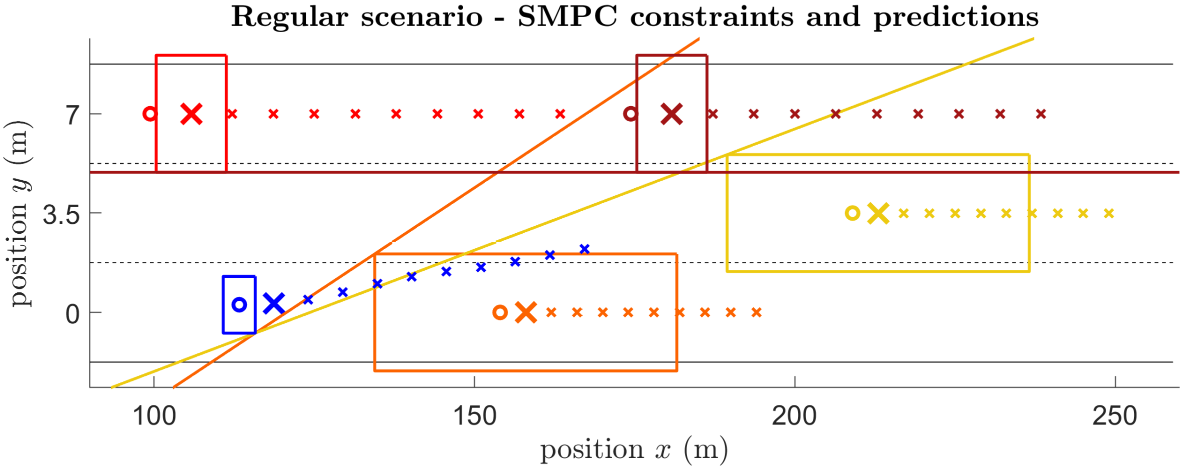


Figure 5.9: SMPC constraints for the regular scenario at time step $t = 22$ and prediction step $k = 1$. The EV shape and planned trajectory are shown in blue. TVs as well as respective safety rectangles and constraints have the same color. Initial states are marked by a circle, prediction states are represented by crosses with a bold cross indicating the displayed prediction step. © 2021 IEEE.

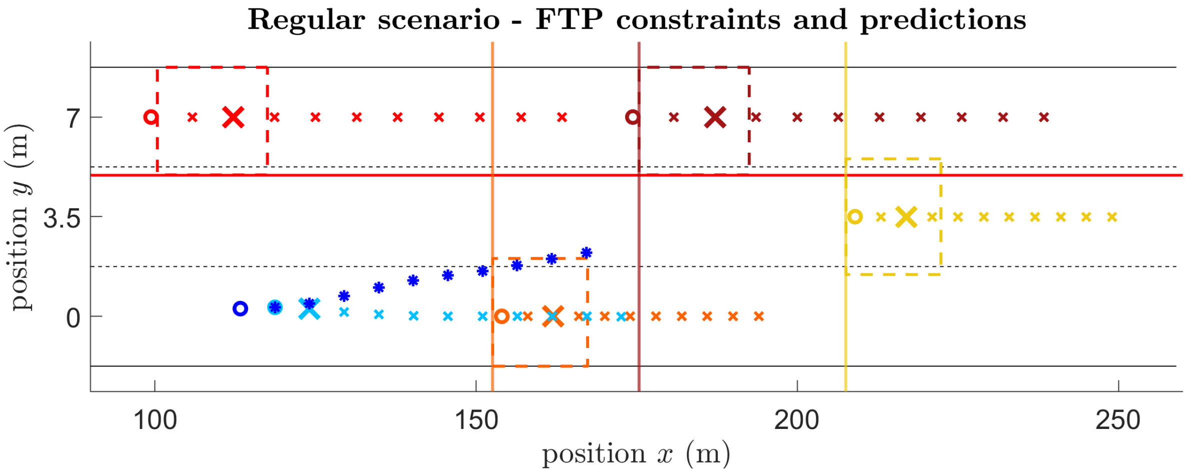


Figure 5.10: FTP constraints for the regular scenario at time step $t = 22$ and prediction step $k = 1$. The EV is shown in blue. TVs as well as respective reachable sets and constraints have the same color. Initial states are marked by a circle. The initial FTP state starts after the first SMPC input is applied. Prediction states are represented by crosses with a bold cross indicating the displayed prediction step. For reference, the planned SMPC trajectory is given by dark blue asterisks with a dark blue circle indicating the initial EV state. © 2021 IEEE.

Both TV4 and TV5 are two lanes left of the EV, yielding cases G and H, resulting in horizontal constraints to the right side of the TVs. TV3 is not shown in Figure 5.9 due to clarity.

The FTP constraints at step $t = 22$ are shown in Figure 5.10. The constraints are more conservative compared to the SMPC constraints. The reachable TV sets extend further to the back than the front, as maximal deceleration is larger than maximal acceleration. Additionally, the convex hull of reachable sets over two consecutive steps is considered. Constraints for TV1 and TV2 are built according to cases D* and B*, respectively. Both constraints for TV4 and TV5 are generated given cases H* and G*. While the SMPC trajectory moves towards the center lane to overtake TV1, the FTP trajectory finds a vehicle motion that, for the final prediction step, remains in the current lane with $\psi^{\text{ra}} = 0$ and enough distance to TV1, i.e., a safe terminal state. As the FTP optimal control problem yields a solution, the first input $\boldsymbol{u}_{\text{SMPC},0}$ of the planned SMPC trajectory is then applied.

Comparison to SMPC and FTP. Throughout the entire simulation, both the SMPC and FTP optimal control problems remain feasible. Therefore, only SMPC inputs are applied. Only using an SMPC algorithm without FTP would therefore yield the same result for this regular scenario.

Unlike SMPC, applying only FTP results in a different solution. As the constraints are more conservative compared to SMPC, the EV never changes lanes to overtake. As indicated by the FTP prediction in Figure 5.10, the FTP constraints keep the EV in its current lane.

We will use the following metric to compare the performance of SMPC+FTP and FTP. Based on the cost function of the optimal control problem, the applied inputs and resulting

risk parameter β	0.8	0.9	0.95	0.99	0.999
cost J_{sim}	11.21	11.35	11.58	11.34	11.31

Table 5.4: Risk parameter analysis.

states for the entire simulation are analyzed according to

$$J_{\text{sim}} = \sum_{t=0}^{N_{\text{sim}}-1} \left(\|\Delta \boldsymbol{x}_{t+1}\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_t\|_{\boldsymbol{R}}^2 + \|\Delta \boldsymbol{u}_t\|_{\boldsymbol{S}}^2 \right) \tag{5.47}$$

with the simulation steps N_{sim}.

The overall cost for SMPC+FTP is $J_{\text{sim}} = 11.32$, while the overall FTP cost is $J_{\text{sim}} = 4.03 \cdot 10^4$. The cost comparison shows that the SMPC+FTP approach yields a more efficient behavior than an FTP approach. In this case, increased efficiency results from keeping the velocity close to the reference velocity.

Risk parameter variation. In the previously discussed simulation, the risk parameter was chosen to be $\beta = 0.8$. Here, we briefly analyze the effect of varying risk parameters on the EV performance. The analyzed risk parameters range from $\beta = 0.8$ to $\beta = 0.999$. The overall simulation cost, according to (5.47), for each risk parameter is given in Table 5.4. The overall costs of the simulation results show that the SMPC behavior and costs for this regular scenario are very similar. However, it can be beneficial regarding the cost to choose a larger risk parameter, as inputs are changed more smoothly. In all five examples the EV behavior is almost similar.

Varying simulation settings. In the previous analysis, only one vehicle configuration is considered. In order to show that the SMPC+FTP method is suitable for various scenarios, we ran 1000 simulations, each consisting of 125 simulation steps, with randomly selected initial vehicle positions and velocities for each simulation run. The EV is located on one of the three lanes, i.e., $d_0 \in \{0, 3.5, 7\}$, with initial longitudinal position $s_0 = 0$ and velocity $v_0 = 27$. The five TVs are randomly placed on one of the three lanes with an initial longitudinal position $x_0^{\text{TV}} \in [-100, 200]$, constant velocity $v_x^{\text{TV}} \in [20, 32]$, and constant $v_y^{\text{TV}} = 0$. It is ensured that all vehicles positioned on the same lane have an initial longitudinal distance $\Delta x \geq 50$ and that TV velocities are chosen such that TVs do not collide with each other.

The SMPC+FTP method successfully handled all 1000 simulation runs and no collisions occurred.

Emergency Highway Scenario

After having shown the efficient SMPC+FTP planning for a regular highway scenario, we now illustrate the safety property of the proposed algorithm in an emergency scenario. The initial vehicle states are the same as in the regular scenario. However, in this emergency scenario the TVs change their velocities and lateral positions. Starting at time step $t = 20$, TV5 initiates an emergency braking maneuver with maximal deceleration until reaching a

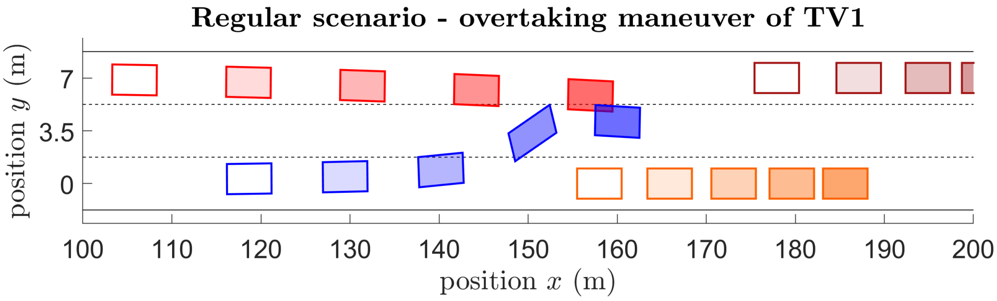


Figure 5.11: Sequences of the emergency scenario collision applying only SMPC. Fading boxes show past states. The EV is shown in blue. © 2021 IEEE.

complete halt. This causes TV4 to change lanes to the center lane in order to avoid TV5. TV1 reduces its velocity to $v_x^{\mathrm{TV1}} = 10\,\mathrm{m\,s^{-1}}$. After having passed TV5, TV4 moves to the left lane to then pass the slower TV2. TV1 also increases its velocity to $v_x^{\mathrm{TV1}} = 20\,\mathrm{m\,s^{-1}}$.

In the following, SMPC without FTP is analyzed first. Then, we present the solution of the SMPC+FTP algorithm.

SMPC. Applying only SMPC results in optimistic EV trajectory planning, while not considering highly unlikely events. Even though TV4 is slowly moving to the center lane, the EV still moves to the center lane to overtake TV1, as a TV4 lane change is still unlikely. However, at step $t = 25$, TV4 continues to increase its lateral velocity towards the center lane. At this point, there exists no feasible SMPC solution anymore that satisfies the chance constraint. This causes the EV to collide with TV4. The collision sequence is illustrated in Figure 5.11. While SMPC performs well in regular scenarios without unlikely uncertainty realizations, these rare situations cause major safety issues.

SMPC+FTP. We now show how the proposed SMPC+FTP method handles the emergency scenario. The EV states and inputs are given in Figure 5.12.

Initially, the EV attempts to switch lanes and overtake TV1. However, at step $t = 27$, the SMPC is unable to find a solution. The FTP problem is still solved successfully and the first planned FTP input is applied. For the next four steps, the SMPC optimal control problem remains infeasible, indicated by the pink lines in Figure 5.12, and the FTP inputs are applied, which are obtained by successfully solving the FTP optimal control problems (FTP Mode). The EV slows down and returns to the right lane, as illustrated in the first sequence of Figure 5.13. At step $t = 37$, the SMPC problem is feasible and the EV plans to overtake TV1 again. However, as TV4 is still too close, the FTP is unable to find a new safe backup trajectory if the next planned SMPC input were applied, i.e., the FTP optimal control problem becomes infeasible. The safe input sequence obtained at the previous time step $t = 36$ is applied to the EV (Backup Mode), as indicated by the red line in Figure 5.12. The EV remains in the right lane until TV4 is far enough away to safely change to the center lane, as shown in the second sequence of Figure 5.13. Eventually, the EV passes TV2 by smoothly switching to the left lane with a small steering angle change. The average computation time for solving the SMPC and FTP optimal control problems are 0.15 s and 0.22 s, respectively. The values are higher compared to the regular scenario, as the computation time for infeasible optimal control problems is significantly larger.

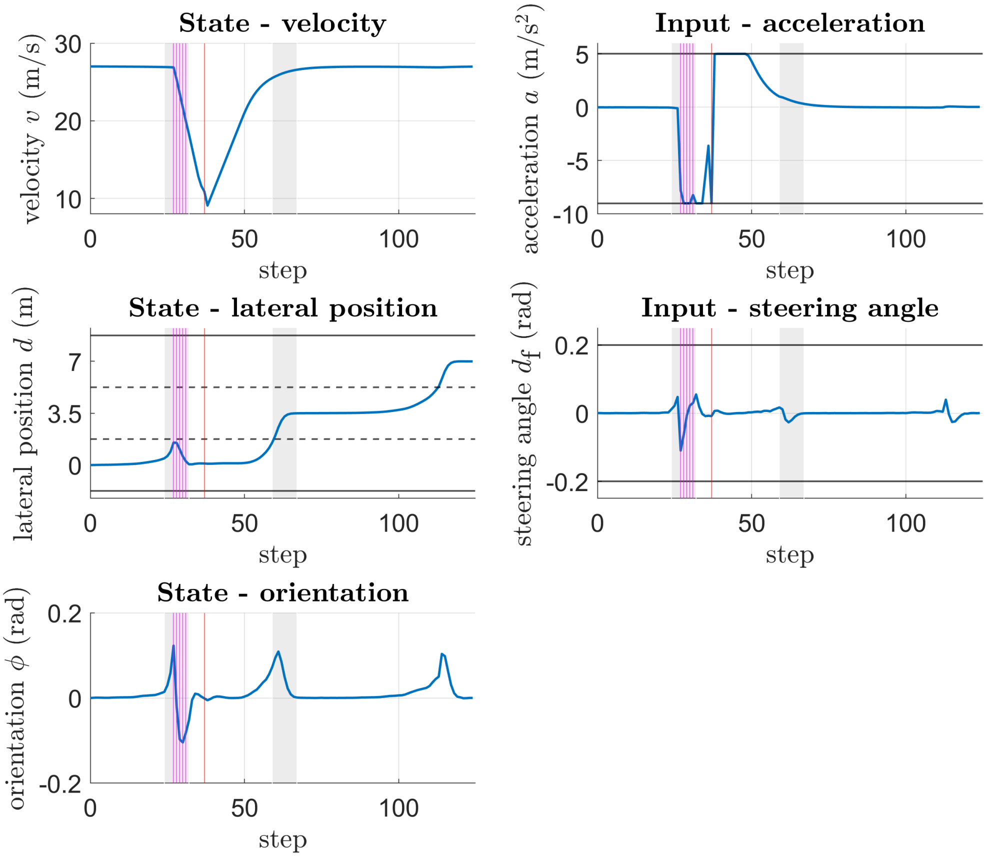


Figure 5.12: SMPC+FTP states and inputs for the emergency scenario. Pink vertical lines represent infeasible SMPC and feasible FTP solutions (FTP Mode), red vertical lines show infeasible FTP solutions (Backup Mode). Vehicle motion in the gray areas is illustrated in Figure 5.13. © 2021 IEEE.

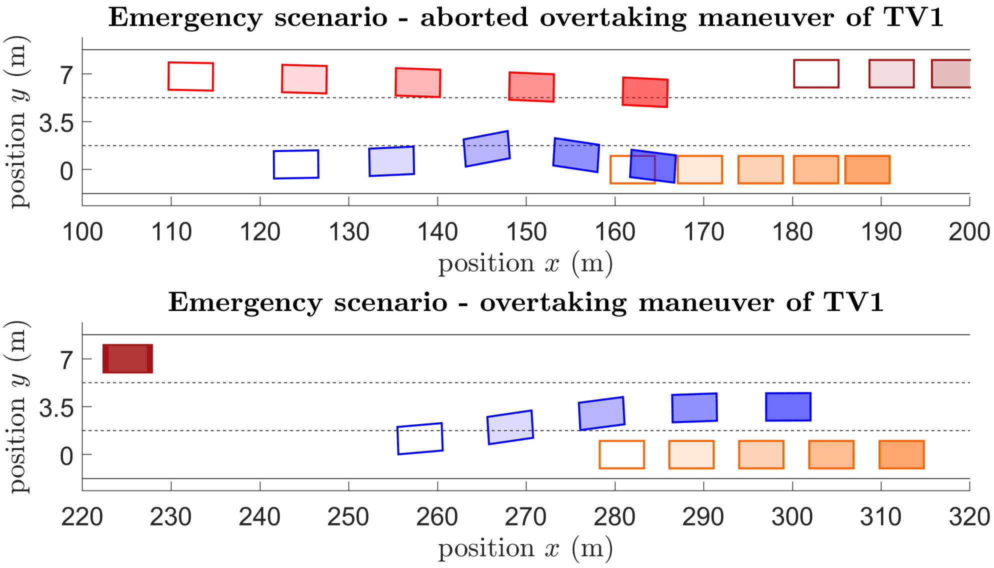


Figure 5.13: Sequences of the emergency scenario with SMPC+FTP. Fading boxes show past states. The EV is shown in blue. © 2021 IEEE.

It is also possible to only apply FTP in this emergency scenario. While this leads to safe vehicle behavior throughout the simulation, the EV does not overtake TV1 and TV2. Comparing the cost yields the following result. Applying FTP to the emergency scenario yields a cost of $J_{\text{sim}} = 4.28 \cdot 10^4$, while the SMPC+FTP cost is $J_{\text{sim}} = 3.34 \cdot 10^4$.

In summary, the simulation scenarios in this section have shown the benefits of the proposed SMPC+FTP method. SMPC optimistically plans trajectories, which are executed as long as there always exists a safe backup trajectory, computed by FTP. In regular scenarios, SMPC+FTP provides benefits known from SMPC. In emergency scenarios, the safety guarantee of FTP holds while the EV is still more efficient compared to applying pure FTP.

5.2.7 Discussion

In some FTP approaches it is required that the vehicle comes to a standstill at the end of the failsafe trajectory. Here, we only require a certain distance to vehicles ahead and zero orientation offset with respect to the road for the terminal state. This enables the use of a relatively short FTP horizon.

It is possible to get oscillating behavior between applied SMPC inputs and the activation of FTP. In other words, in step one the SMPC input is applied, which potentially causes the FTP to intervene in the next step. Then, a safe state is again achieved, leading to another, potentially over-aggressive SMPC input, again requiring FTP in the subsequent step. This can be avoided by designing the SMPC controller and its constraints less aggressively, as done in the simulation study.

Regarding the simulation, simulating each scenario once is adequate. While the TV is assumed to behave probabilistically by the EV, the actual TV behavior here is deterministic. And whereas SCMPC depends on drawn samples, which vary between simulations, the applied SMPC approach uses a chance constraint reformulation that always yields the same constraint, given the same uncertainty distribution.

In the emergency scenario, multiple TVs change velocities or lanes. This scenario was chosen such that the SMPC method causes a collision, which usually does not happen even for highly unlikely TV trajectories. The chance constraint within SMPC does allow for a small probability of constraint violation, however, in most cases the iterative structure of MPC handles potential future constraint violations. Furthermore, constraint violations do not necessarily cause collisions, as the safety area around a TV is larger than the actual TV shape.

Comparing the planned SMPC trajectories for the EV at two consecutive time steps without any major environment changes, one would assume that the planned trajectory remains similar. However, this is not the case. The constraints with respect to other TVs are generated based on the EV state at the beginning of the optimal control problem in order to formulate linear constraints. Therefore, in the next step, the constraint generation is based on an updated initial EV state, resulting in a slightly different planned SMPC trajectory compared to the previously planned trajectory. This could be addressed by using EV predictions for the constraint generation, however, this would require nonlinear constraints.

The applied vehicle inputs in the emergency scenario lead to relatively high steering angles. This is not ideal for a smooth vehicle motion. Even though this behavior is acceptable in rare cases, the motion could be optimized by defining more cases for the constraint generation.

The individual SMPC and FTP algorithms in this chapter are possible controller realizations, specifically designed for highway scenarios with multiple TVs. The properties of the combined SMPC+FTP method are not restricted to the suggested SMPC and FTP trajectory planners described in Section 5.2.4 and Section 5.2.5, respectively. Other SMPC or FTP approaches can be applied.

In dense traffic or unclear traffic situations, humans often do not wait until the desired maneuver is entirely realizable. Instead, humans often slowly initiate maneuvers, causing other vehicles to react. For example, cutting into a lane is often preceded by slight motion towards the other lane so that other vehicles leave extra space. Therefore, it is possible to execute the lane change maneuver successfully, even though it was not possible to safely plan the entire lane change maneuver initially. The SMPC+FTP framework enables automated vehicle motion that comes close to this efficient human behavior.

5.3 General Safe Stochastic MPC

In the following, we extend the previously presented SMPC+FTP approach to a general safe SMPC alorithm for linear systems subject to additive uncertainty.

5.3.1 Problem Formulation

We consider a linear, discrete-time system

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{E}\boldsymbol{w}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) \tag{5.48}$$

with states $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$, inputs $\boldsymbol{u}_t \in \mathbb{R}^{n_{\boldsymbol{u}}}$, and uncertainties $\boldsymbol{w}_t \in \mathbb{R}^{n_{\boldsymbol{w}}}$ at time step t, as well as the known system matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{E}$ with appropriate dimensions. System (5.48) is subject to input constraints $\boldsymbol{u}_t \in \mathcal{U}$ and state constraints $\boldsymbol{x}_t \in \mathcal{X}$.

Assumption 5.5. *The uncertainty $\boldsymbol{w}_t$ is i.i.d. and bounded by $\boldsymbol{w}_t \in \mathcal{W}$.*

The general task is to drive the state of system (5.48) to the origin while keeping inputs low. We achieve this by repeatedly solving an optimal control problem

$$\min_{\boldsymbol{U}_t} \; J(\boldsymbol{x}_t, \boldsymbol{U}_t) \tag{5.49a}$$

$$\text{s.t. } \boldsymbol{x}_{t+k+1|t} = \boldsymbol{f}(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}, \boldsymbol{w}_{t+k|t}) \tag{5.49b}$$

$$\boldsymbol{u}_{t+k|t} \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{5.49c}$$

$$\boldsymbol{x}_{t+k|t} \in \mathcal{X}, \qquad k \in \mathbb{I}_{1,N} \tag{5.49d}$$

with the finite input sequence $\boldsymbol{U}_t = (\boldsymbol{u}_{t|t}, ..., \boldsymbol{u}_{t+N-1|t})$ and the objective function

$$J(\boldsymbol{x}_t, \boldsymbol{U}_t) = \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}\right)\right) + V_{\mathrm{f}}(\boldsymbol{x}_{t+N|t}) \tag{5.50}$$

with prediction horizon N, stage cost l, and the terminal cost function V_{f}. At each time step, the first element $\boldsymbol{u}_t = \boldsymbol{u}_{t|t}$ is applied to the system. This may be expressed as a control law $\boldsymbol{u}_t = \boldsymbol{u}(\boldsymbol{x}_t)$.

We consider uncertainty in the state constraint (5.49d) with the chance constraint

$$\Pr\left(\boldsymbol{x}_{t+k|t} \in \mathcal{X}\right) \geq \beta, \tag{5.51}$$

where a lower risk parameter β, yields a higher probability of constraint violations.

Property Definitions

If MPC is employed in safety-critical applications, three properties are required. First, safety must be ensured. Second, if the MPC optimal control problem is feasible at a time step, a solution must also exist at the next time step, known as recursive feasibility of the optimal control problem. Third, the closed-loop system behavior must be stable. We refer to Section 2.3 for definitions of a robustly positively invariant set $\mathcal{X}_{\text{ris}}$ (Definition 2.2), recursive feasibility (Definition 2.5), and input-to-state stability (Definition 2.6). Additionally, we use the following definitions.

Definition 5.4 (Safety). *The state $\boldsymbol{x}_{t_0}$ of system* (5.48) *is safe at time step t_0 if it is guaranteed that there exist inputs $\boldsymbol{u}_t$, $t \geq t_0$ such that the constraints $\boldsymbol{u}_t \in \mathcal{U}$ and $\boldsymbol{x}_t \in \mathcal{X}$ are satisfied for all $t \geq t_0$.*

Definition 5.5 (Safe Input Sequence). *An input sequence $\boldsymbol{U}_t = (\boldsymbol{u}_t, ..., \boldsymbol{u}_{t+N-1})$ with $\boldsymbol{u}_{t+i} \in \mathcal{U}$ $\forall i = \mathbb{I}_{0,N-1}$ is safe for system* (5.48) *if consecutively applying the individual input elements yields the safe state sequence $(\boldsymbol{x}_{t+1}, ..., \boldsymbol{x}_{t+N})$ with individual safe states $\boldsymbol{x}_{t+i}$ $\forall i = \mathbb{I}_{1,N}$ according to Definition 5.4.*

Problem Statement

The aim of this section is to design a general SMPC algorithm that exploits the advantage of reduced conservatism in SMPC while ensuring the previously mentioned properties.

Objective 5.2. *The SMPC algorithm* (5.49) *for system* (5.48)*, where the chance constraint* (5.51) *replaces the hard constraint* (5.49d)*, must maximize the control objective* (5.50) *while guaranteeing safety (Definition 5.4), recursive feasibility (Definition 2.5), and stability (Definition 2.6).*

In the following, we propose an SMPC algorithm including a safe backup (predictive) controller that ensures satisfaction of all requirements listed in Objective 5.2.

5.3.2 Method

We first describe the general safe SMPC algorithm and then briefly address possible suitable MPC variants.

General Safe SMPC Algorithm

SMPC allows for a certain probability of constraint violation. Therefore, in order to use SMPC in a safe way, it needs to be ensured that applying an SMPC input is safe.

We propose an MPC algorithm that consists of an SMPC part and a backup predictive controller. The general safe SMPC algorithm, illustrated in Figure 5.14, yields an input $\boldsymbol{u}_t$ at each time step t, which is determined based on the following two modes:

- Stochastic mode (with optimal control problem $\mathbb{P}^{\text{s}}(\boldsymbol{x}_t)$)
- Backup mode (with optimal control problem $\mathbb{P}^{\text{b}}(\boldsymbol{x}_t)$)

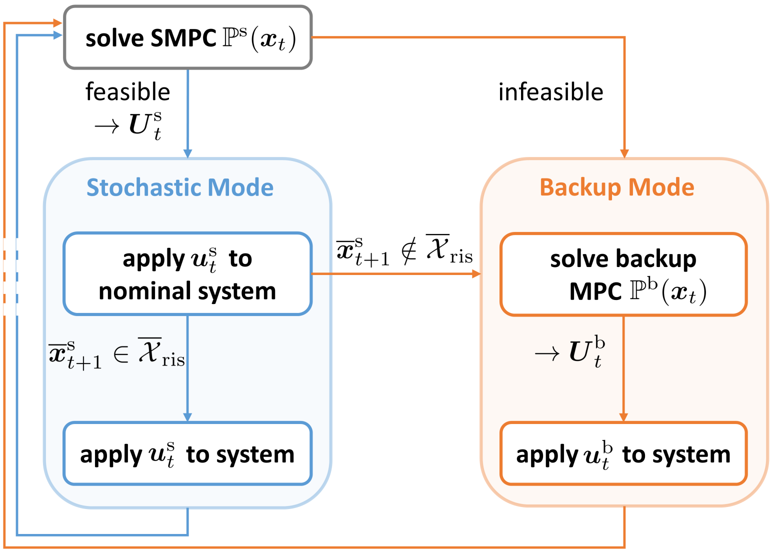

solve SMPC $\mathbb{P}^{\mathrm{s}}(\boldsymbol{x}_t)$
feasible
$\rightarrow \boldsymbol{U}_t^{\mathrm{s}}$
infeasible
Stochastic Mode
apply $\boldsymbol{u}_t^{\mathrm{s}}$ to nominal system
$\overline{\boldsymbol{x}}_{t+1}^{\mathrm{s}} \notin \overline{\mathcal{X}}_{\mathrm{ris}}$
$\overline{\boldsymbol{x}}_{t+1}^{\mathrm{s}} \in \overline{\mathcal{X}}_{\mathrm{ris}}$
apply $\boldsymbol{u}_t^{\mathrm{s}}$ to system
Backup Mode
solve backup MPC $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_t)$
$\rightarrow \boldsymbol{U}_t^{\mathrm{b}}$
apply $\boldsymbol{u}_t^{\mathrm{b}}$ to system

Figure 5.14: Safe SMPC algorithm.

We consider the general SMPC optimal control problem $\mathbb{P}^{\mathrm{s}}(\boldsymbol{x}_t)$ with horizon N given by

$$\min_{\boldsymbol{U}_t^{\mathrm{s}}} \; J(\boldsymbol{x}_t, \boldsymbol{U}_t^{\mathrm{s}}) \tag{5.52a}$$

$$\text{s.t. } \boldsymbol{x}_{t+k+1} = \boldsymbol{f}(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}^{\mathrm{s}}, \boldsymbol{w}_{t+k|t}) \tag{5.52b}$$

$$\boldsymbol{u}_{t+k|t}^{\mathrm{s}} \in \mathcal{U}, \qquad k \in \mathbb{I}_{0,N-1} \tag{5.52c}$$

$$\Pr\left(\boldsymbol{x}_{t+k|t} \in \mathcal{X}\right) \geq \beta, \qquad k \in \mathbb{I}_{1,N} \tag{5.52d}$$

yielding the optimal input sequence $\boldsymbol{U}_t^{\mathrm{s}*} = \left(\boldsymbol{u}_{t|t}^{\mathrm{s}*}, ..., \boldsymbol{u}_{t+N-1|t}^{\mathrm{s}*}\right)$ with the SMPC control law $\boldsymbol{u}^{\mathrm{s}}(\boldsymbol{x}_t) = \boldsymbol{u}_{t|t}^{\mathrm{s}*}$. Any SMPC method may be used to reformulate the chance constraint (5.52d) into a tractable formulation, depending on the uncertainty distribution.

We consider a backup MPC controller with horizon N^{b} and optimal control problem $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_t)$ with cost function $J^{\mathrm{b}}(\boldsymbol{x}_t, \boldsymbol{U}_t^{\mathrm{b}})$, yielding the optimal cost $J^{\mathrm{b}*} = J^{\mathrm{b}}(\boldsymbol{x}_t, \boldsymbol{U}_t^{\mathrm{b}*})$ with input sequence $\boldsymbol{U}_t^{\mathrm{b}*} = \left(\boldsymbol{u}_{t|t}^{\mathrm{b}*}, ..., \boldsymbol{u}_{t+N^{\mathrm{b}}-1|t}^{\mathrm{b}*}\right)$, and control law $\boldsymbol{u}^{\mathrm{b}}(\boldsymbol{x}_t) = \boldsymbol{u}_{t|t}^{\mathrm{b}*}$, resulting in the closed-loop system

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}^{\mathrm{b}}(\boldsymbol{x}_t), \boldsymbol{w}_t\right) \tag{5.53}$$

for system (5.48). Multiple backup controllers are possible in this algorithm, as long as they fulfill the following assumption.

Assumption 5.6. *The backup MPC optimal control problem $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_t)$, with value function $J^{\mathrm{b}*}$ and control law $\boldsymbol{u}^{\mathrm{b}}(\boldsymbol{x}_t)$, is chosen such that $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_t)$ is recursively feasible, $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{u}_t \in \mathcal{U}$ for all t, and such that the origin of the closed-loop system* (5.53) *is ISS with region of attraction $\mathcal{X}_{\mathrm{ris}}$, where $\mathcal{X}_{\mathrm{ris}}$ is robustly positively invariant for all $\boldsymbol{w}_t \in \mathcal{W}$.*

Various MPC schemes exist that fulfill Assumption 5.6, as discussed in Section 5.3.2. The safe SMPC algorithm only applies SMPC inputs if it is guaranteed that the backup optimal control problem $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_t)$ may still be solved at the next time step. Applying the first SMPC input $\boldsymbol{u}^{\mathrm{s}}(\boldsymbol{x}_t)$ to the nominal system yields the next nominal state

$$\overline{\boldsymbol{x}}^{\mathrm{s}}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}^{\mathrm{s}}(\boldsymbol{x}_t). \tag{5.54}$$

It is guaranteed that the backup optimal control problem is feasible for any first step uncertainty $\boldsymbol{w}_t \in \mathcal{W}$ if

$$\overline{\boldsymbol{x}}^{\mathrm{s}}_{t+1} \in \overline{\mathcal{X}}_{\mathrm{ris}} \tag{5.55}$$

where $\overline{\mathcal{X}}_{\mathrm{ris}} = \mathcal{X}_{\mathrm{ris}} \ominus \mathcal{W}$, which ensures that $\boldsymbol{x}^{\mathrm{s}}_{t+1} \in \mathcal{X}_{\mathrm{ris}}$.

We are now able to propose two different modes within the safe SMPC algorithm, both evaluated at time step t.

Stochastic mode. The control law $\boldsymbol{u}_t = \boldsymbol{u}^{\mathrm{s}}(\boldsymbol{x}_t)$ is applied if the SMPC optimal control problem is feasible and if (5.55) is fulfilled, i.e., $\boldsymbol{U}^{\mathrm{s}*}_t \neq \emptyset$ and $\overline{\boldsymbol{x}}^{\mathrm{s}}_{t+1} \in \overline{\mathcal{X}}_{\mathrm{ris}}$.

Backup mode. If the SMPC optimal control problem is infeasible or if (5.55) is not satisfied, i.e., $\boldsymbol{U}^{\mathrm{s}*}_t = \emptyset$ or $\overline{\boldsymbol{x}}^{\mathrm{s}}_{t+1} \notin \overline{\mathcal{X}}_{\mathrm{ris}}$, the backup MPC optimal control problem is solved and the control law $\boldsymbol{u}_t = \boldsymbol{u}^{\mathrm{b}}(\boldsymbol{x}_t)$ is applied.

Note that in the stochastic mode, only one optimal control problem is solved, while the backup mode requires solving two optimal control problems.

MPC Details

The proposed algorithm allows us to consider any SMPC approach to solve (5.52), e.g., SMPC with exact chance constraint reformulations based on normal distributions, affine disturbance feedback SMPC, or sampling-based SMPC. A suitable SMPC method may be chosen depending on the type of uncertainty and the type of system.

Assumption 5.6 allows us to employ various MPC schemes for the backup controller, which enables application of a wider class of backup controllers compared to [32]. The most intuitive choice are RMPC approaches that guarantee recursive feasibility and stability for a bounded uncertainty. The backup MPC can also be based on other approaches, such as MPC based on reachability analysis [175] or the failsafe MPC idea described in Section 5.2.1. It is even possible to consider recursively feasible SMPC approaches as backup controllers, e.g., [116], if Assumption 5.6 may be satisfied.

5.3.3 Properties

In the following, we show that the proposed SMPC algorithm is recursively feasible, safe, and ISS.

Recursive Feasibility

Based on Definition 2.5, we first prove recursive feasibility of the optimal control problem of the safe SMPC algorithm described in Section 5.3.2.

Theorem 5.2. *Let Assumptions 5.5 and 5.6 hold and let the system input $\boldsymbol{u}_t$ be determined based on the proposed safe SMPC algorithm in Section 5.3.2. Then, for an admissible $\boldsymbol{u}_0$, obtaining a solution $\boldsymbol{u}_t$ is feasible for all $t > 0$.*

Proof. The proof is based on showing that at any time step t it is guaranteed that an admissible input $\boldsymbol{u}_t$ is applied at t and an admissible input $\boldsymbol{u}_{t^+}$ may be obtained at the next time step t^+.

If $\boldsymbol{U}_t^{\mathrm{s}*} \neq \emptyset$ and $\overline{\boldsymbol{x}}_{t+1}^{\mathrm{s}} \in \overline{\mathcal{X}}_{\mathrm{ris}}$, $\boldsymbol{u}_t = \boldsymbol{u}^{\mathrm{s}}(\boldsymbol{x}_t)$ exists and $\boldsymbol{x}_{t+1}^{\mathrm{s}} \in \mathcal{X}_{\mathrm{ris}}$, which guarantees that a solution $\boldsymbol{U}_{t+1}^{\mathrm{b}*}$ exists for $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_{t+1})$.

In the backup mode, $\boldsymbol{U}_t^{\mathrm{b}*}$ exists for $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_t)$ as $\boldsymbol{x}_t \in \mathcal{X}_{\mathrm{ris}}$ and Assumption 5.6 ensures that $\mathbb{P}^{\mathrm{b}}(\boldsymbol{x}_{t+1})$ remains feasible.

Hence, all possibilities are covered. This holds for all $t > 0$, i.e., admissible inputs are guaranteed at subsequent time steps, which concludes the proof. □

Note that no terminal constraint is necessary for the SMPC optimal control problem to ensure recursive feasibility of the overall algorithm. Based on guaranteed recursive feasibility of the safe SMPC algorithm, safety and stability are now discussed.

Safety

We require that the SMPC algorithm described in Section 5.3.2 is safe. Based on Definition 5.4, this requirement demands that all constraints are met at all time steps, which we show in the following.

Theorem 5.3. *Let Assumptions 5.5 and 5.6 hold and let the system input $\boldsymbol{u}_t$ be determined based on the proposed safe SMPC algorithm in Section 5.3.2. Then, for a safe initial state $\boldsymbol{x}_0$, safety according to Definition 5.4 is guaranteed for $t > 0$.*

Proof. The proof is based on Theorem 5.2; hence, it is guaranteed that one of the two modes is applicable at each time step t. In the stochastic mode, $\boldsymbol{u}_t \in \mathcal{U}$ and (5.55) ensures that $\boldsymbol{x}_t \in \mathcal{X}$ for all $\boldsymbol{w}_t \in \mathcal{W}$. The backup mode guarantees, by design, that $\boldsymbol{u}_t \in \mathcal{U}$ and $\boldsymbol{x}_t \in \mathcal{X}$ for all $\boldsymbol{w}_t \in \mathcal{W}$. Hence, in both modes it is guaranteed that $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{u}_t \in \mathcal{U}$, which holds for all $t > 0$ as the proposed SMPC algorithm is recursively feasible. □

As shown, safety is ensured by the backup predictive controller and (5.55). This enables the use of chance constraints within the SMPC optimal control problem $\mathbb{P}^{\mathrm{s}}(\boldsymbol{x}_t)$, which allow for constraint violations in the open-loop prediction.

Stability

In MPC, stability is often proved by showing that the value function is decreasing for subsequent time steps, also known as the descent property. These proofs are based on the MPC idea of a moving horizon, where the previously planned input sequence remains valid and only one additional input element is added to the input sequence for the next time step. For the proposed safe SMPC algorithm, however, this assumption does not hold. Since switching between different modes is possible, the predicted input and state trajectories may vary at each time step. We tackle this challenge by using Lemma 2.1 based on [76]. This lemma ensures that the origin of a system subject to uncertainty is ISS if the undisturbed system is asymptotically stable and the system is Lipschitz continuous with respect to state $\boldsymbol{x}$ and uncertainty $\boldsymbol{w}$.

Assumption 5.7. *The cost function J^{b} is selected according to* (5.50). *The stage cost is chosen as $l\left(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}\right) = \overline{\boldsymbol{x}}_{t+k|t}^\top \boldsymbol{Q} \overline{\boldsymbol{x}}_{t+k|t} + {\boldsymbol{u}_{t+k|t}}^\top \boldsymbol{R} \boldsymbol{u}_{t+k|t}$ with $\boldsymbol{Q} = \boldsymbol{Q}^\top \succ 0$ and $\boldsymbol{R} =$*

$\boldsymbol{R}^\top \succ 0$ where $\overline{\boldsymbol{x}}_{t+k|t}$ denotes the nominal states. The terminal cost $V_\mathrm{f}(\boldsymbol{x}_{t+N|t})$ is chosen as a Lyapunov function in a terminal set $\mathcal{X}_\mathrm{f}$ for the undisturbed closed-loop system $\boldsymbol{x}_{t+1} = (\boldsymbol{A}+\boldsymbol{BK})\boldsymbol{x}_t$ such that for all $\boldsymbol{x}_t \in \mathcal{X}_\mathrm{f}$

$$V_\mathrm{f}((\boldsymbol{A}+\boldsymbol{BK})\boldsymbol{x}_t) - V_\mathrm{f}(\boldsymbol{x}_t) \leq -\boldsymbol{x}_t{}^\top(\boldsymbol{Q}+\boldsymbol{K}^\top\boldsymbol{RK})\boldsymbol{x}_t \tag{5.56}$$

where $\boldsymbol{K}$ is a stabilizing feedback matrix.

We can now formulate the ISS property of a system controlled with the proposed algorithm.

Theorem 5.4. *Let Assumptions 5.5-5.7 hold and let the system input $\boldsymbol{u}_t$ be determined based on the proposed safe SMPC algorithm in Section 5.3.2. Then, for $\boldsymbol{x}_0 \in \mathcal{X}_\mathrm{ris}$, the origin is ISS for system* (5.48) *and $\boldsymbol{x}_t \in \mathcal{X}_\mathrm{ris}, \quad t > 0$.*

Proof. We prove ISS by showing that $V(\boldsymbol{x}_t) = J^{\mathrm{b}*}$ is an ISS Lyapunov function for (5.48) where $V(\cdot)$ satisfies (2.21a) and (2.21b). Any input prediction in either of the two modes can be described by $\boldsymbol{u}_{t+k|t}{}' = \boldsymbol{u}^\mathrm{b}_{t+k|t} - \tilde{\boldsymbol{u}}_{t+k|t},\ k \in \mathbb{I}_{0,N-1}$ where $\tilde{\boldsymbol{u}}_{t+k|t}$ represents the offset between the backup MPC input element $\boldsymbol{u}^\mathrm{b}_{t+k|t}$ and the input element obtained in the safe SMPC algorithm $\boldsymbol{u}_{t+k|t}{}'$. As $\boldsymbol{u}_{t+k|t}{}'$ and $\boldsymbol{u}^\mathrm{b}_{t+k|t}$ are bounded, $\tilde{\boldsymbol{u}}_{t+k|t}$ is bounded, allowing us to define the new bounded uncertainty $\tilde{\boldsymbol{w}}_{t+k|t} = \left(\tilde{\boldsymbol{u}}^\top_{t+k|t}, \boldsymbol{w}_{t+k|t}{}^\top\right)^\top$. This yields the closed loop system

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) &= \boldsymbol{Ax}_t + \boldsymbol{Bu}_t + \boldsymbol{Ew}_t &\text{(5.57a)}\\ &= \boldsymbol{Ax}_t + \boldsymbol{Bu}^\mathrm{b}(\boldsymbol{x}_t) - \boldsymbol{B}\tilde{\boldsymbol{u}}_t + \boldsymbol{Ew}_t &\text{(5.57b)}\\ &= \boldsymbol{Ax}_t + \boldsymbol{Bu}^\mathrm{b}(\boldsymbol{x}_t) + [-\boldsymbol{B}, \boldsymbol{E}]\tilde{\boldsymbol{w}}_t &\text{(5.57c)}\end{aligned}$$

which can be abbreviated by $\boldsymbol{f}'(\boldsymbol{x}_t, \tilde{\boldsymbol{w}}_t)$.

Function $\boldsymbol{f}'$ is continuous and $\boldsymbol{f}'(\boldsymbol{0},\boldsymbol{0}) = 0$. With Assumption 5.7, it holds that $V(\cdot)$ is positive definite and continuous on $\mathcal{X}_\mathrm{ris}$. Hence, based on [99, Lemma 4.3], functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ exist such that $\alpha_1(||\boldsymbol{x}_t||) \leq V(\boldsymbol{x}_t) \leq \alpha_2(||\boldsymbol{x}_t||)$, i.e., (2.21a) is fulfilled.

Due to Assumption 5.6, $V(\boldsymbol{x}_t) = J^{\mathrm{b}*}$ is an ISS Lyapunov function for the undisturbed system with $\tilde{\boldsymbol{w}}_t = \boldsymbol{0}$, i.e., $V(\boldsymbol{f}'(\boldsymbol{x}_t, \boldsymbol{0})) - V(\boldsymbol{x}_t) \leq -\alpha_3(||\boldsymbol{x}_t||)$. With J^b designed according to Assumption 5.7 and a bounded $\mathcal{X}_\mathrm{ris}$, $J^{\mathrm{b}*}$ is Lipschitz continuous. Hence, Lemma 2.1 is fulfilled and $V(\cdot)$ is an ISS Lyapunov function for $\boldsymbol{f}'(\boldsymbol{x}_t, \tilde{\boldsymbol{w}}_t)$ with $\boldsymbol{x}_t \in \mathcal{X}_\mathrm{ris}$, i.e., it holds that $V(\boldsymbol{f}'(\boldsymbol{x}_t, \tilde{\boldsymbol{w}}_t)) - V(\boldsymbol{x}_t) \leq -\alpha_3(||\boldsymbol{x}_t||) + \gamma(||\tilde{\boldsymbol{w}}_t||)$. It follows that the origin is ISS with the safe SMPC algorithm. □

Note that tuning the risk parameter in the SMPC optimal control problem does not affect recursive feasibility, safety, or stability. This allows us to choose a risk parameter that yields the most efficient behavior.

5.3.4 Simulation Study

We analyze the proposed algorithm in a numerical example, similar to the example in Section 2.4, and specifically elaborate on the advantages over SMPC and RMPC. Simulations are carried out in Matlab where the set calculations are done with the Mutli-Parametric Toolbox 3 and the MPC routine is based on [79].

Simulation Setup

We consider the discrete-time system

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1 & 0.0075 \\ -0.143 & 0.996 \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} 4.798 \\ 0.115 \end{bmatrix} u_t + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{w}_t \tag{5.58}$$

with $\boldsymbol{x}_t = (x_{1,t},\, x_{2,t})^\top$ and the normally distributed uncertainty $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$, $\boldsymbol{w}_t \in \mathcal{W}$ where $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \operatorname{diag}(0.06,\, 0.06)$ and $\mathcal{W} = \{\boldsymbol{w}_t \mid ||\boldsymbol{w}_t||_\infty \leq 0.07\}$. The input is bounded by $|u_t| \leq 0.2$ and we employ the state constraint $x_{1,t} \leq 2.8$. Additionally, we define $|x_{1,t}| \leq 10$ and $|x_{2,t}| \leq 10$ to obtain a bounded set $\mathcal{X}$, however, in the following simulation only $x_{1,t} \leq 2.8$ is relevant. The initial state is $\boldsymbol{x}_0 = (-1.3,\, 3.5)^\top$

For the SMPC optimal control problem, we approximate the uncertainty with the non-truncated normal distribution $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ and use an analytic chance constraint reformulation approach as described in Section 2.2.4. We split the state into a deterministic and a probabilistic part $\boldsymbol{x}_t = \boldsymbol{z}_t + \boldsymbol{e}_t$, yielding an adapted input $u_t = \boldsymbol{K}\boldsymbol{x}_t + c_t$ with a stabilizing feedback matrix $\boldsymbol{K}$ and the new input decision variable c_t. The state constraint is considered as the chance constraint $\Pr(x_1 \in 2.8) \geq \beta$ with $\beta = 0.8$. The normal distribution $\boldsymbol{w}$ yields the constraint tightening

$$x_{1,k} \leq 2.8 - \gamma_k \tag{5.59a}$$

$$\gamma_k = \sqrt{2(1,\, 0)^\top \boldsymbol{\Sigma}^e_k (1,\, 0)}\, \operatorname{erf}^{-1}(2\beta - 1) \tag{5.59b}$$

with the error covariance matrix

$$\boldsymbol{\Sigma}^{\boldsymbol{e}}_{k+1} = \boldsymbol{A}_{\boldsymbol{K}} \boldsymbol{\Sigma}^{\boldsymbol{e}}_k \boldsymbol{A}_{\boldsymbol{K}}^\top + \boldsymbol{\Sigma}_{\boldsymbol{w}} \tag{5.60}$$

where $\boldsymbol{\Sigma}^e_0 = \operatorname{diag}(0,\, 0)$ and $\boldsymbol{A}_{\boldsymbol{K}} = \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$.

For the backup MPC, we use an RMPC approach according to [128], satisfying Assumption 5.6. This approach yields the tightened state constraint $\overline{x}_1 \leq 1.72$ and tightened input constraint $-0.018 \leq \overline{u}_t \leq 0.025$. The terminal constraint $\mathcal{X}_\mathrm{f}$ is chosen to be a maximal robust control invariant set.

For SMPC and RMPC we employ a sampling time $\Delta t = 0.1$, a horizon $N = N^\mathrm{b} = 11$, and we use the stabilizing feedback gain $\boldsymbol{K} = [-0.29,\, 0.49]$.

For both the SMPC and RMPC optimal control problem, we use the cost according to Assumption 5.7 with $\boldsymbol{Q} = \operatorname{diag}(1,\, 10)$ and $\boldsymbol{R} = 1$. We choose $V_\mathrm{f}(\boldsymbol{x}) = ||\boldsymbol{x}||^2_{\boldsymbol{Q}_\mathrm{f}}$ with $\boldsymbol{Q}_\mathrm{f} = \begin{bmatrix} 1.91 & -5.06 \\ -5.06 & 39.54 \end{bmatrix}$, which satisfies the discrete-time algebraic Riccati equation.

Simulation Results

We first analyze the resulting trajectories of a simulation with zero uncertainty using the proposed safe SMPC algorithm as well as pure RMPC and SMPC, where $\beta = 0.8$. The pure RMPC is based on the backup RMPC controller in Section 5.3.4 and the pure SMPC uses the SMPC part described in Section 5.3.4. The results are shown in Figure 5.15.

Whereas the autonomous system with $u_t = 0$ violates the state constraint $x_1 \leq 2.8$, the pure SMPC trajectory moves as close towards the state constraint as the constraint tightening γ_k, according to (5.59), allows. State and input constraints are tightened more conservatively

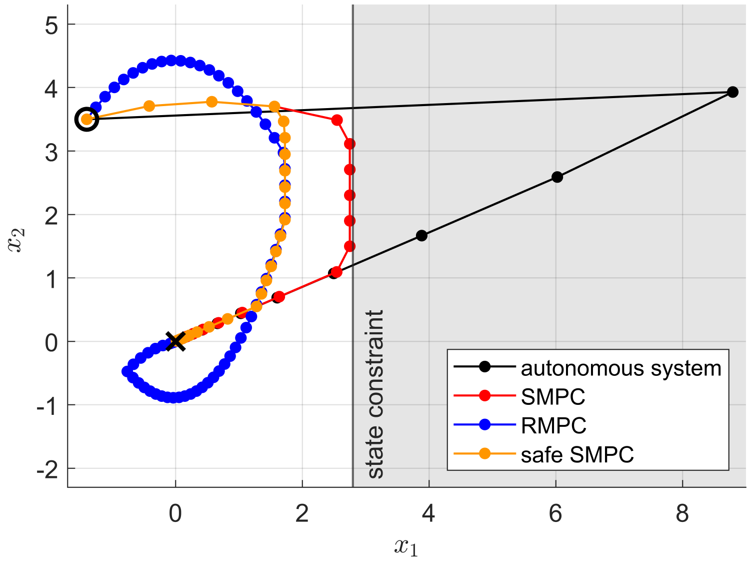


Figure 5.15: Simulation results for the autonomous system, RMPC, SMPC, and safe SMPC, all without applied system uncertainty.

method	avg. cost	avg. violations per run
RMPC	3.56e3	0
SMPC	0.88e3	0.89
safe SMPC	1.13e3	0

Table 5.5: Safe SMPC comparison.

for pure RMPC, requiring a large number of steps to eventually reach the origin. The safe SMPC approach is initially similar to pure SMPC, as x_1 is at a large enough distance to the state constraint. Once (5.55) is not satisfied anymore, the safety mechanism of the proposed algorithm is triggered and the backup mode becomes active. For the following steps, the input of the safe backup RMPC is applied. SMPC inputs are applied as soon as it is possible to satisfy (5.55) again.

The procedure of the safe SMPC approach is illustrated in Figure 5.16, showing the resulting trajectories of 10 simulation runs subject to uncertainty. Switching to the backup RMPC inputs ensures that the state constraint is never violated.

Analyzing 100 simulation runs of each safe SMPC, pure RMPC, and pure SMPC with $N_{\text{sim}} = 80$ simulation steps highlights the advantages of the proposed method. The results are given in Table 5.5. The cost is determined by

$$J_{\text{sim}} = \sum_{k=1}^{N_{\text{sim}}} \left(\|\boldsymbol{x}_k\|_{\boldsymbol{Q}} + \|\boldsymbol{u}_{k-1}\|_{\boldsymbol{R}} \right). \tag{5.61}$$

Whereas SMPC yields the lowest cost, constraint violations occur regularly. RMPC avoids constraint violations but the cost is majorly increased compared to SMPC. The proposed safe SMPC guarantees constraint satisfaction with only slightly higher cost compared to SMPC, combining the advantages of SMPC and RMPC.

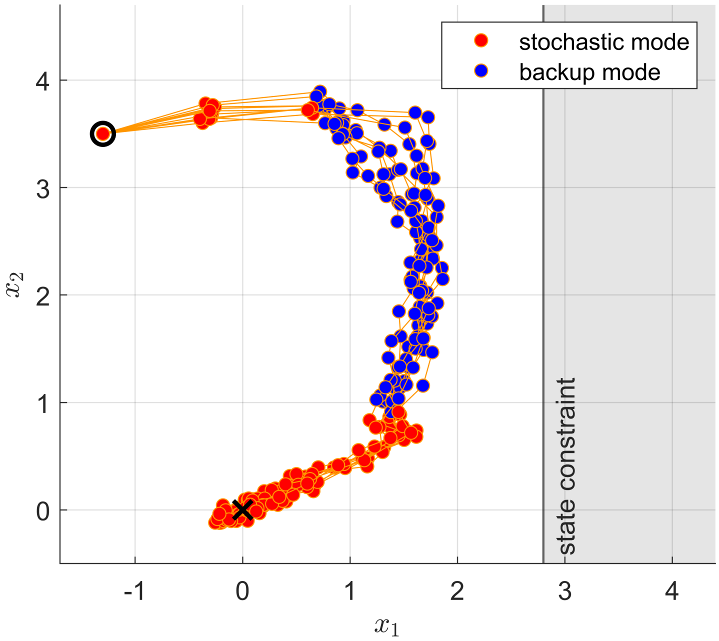


Figure 5.16: Resulting safe SMPC trajectories for 10 simulation runs with uncertainty; SMPC inputs in red, safe backup inputs in blue.

5.4 Conclusion

In this chapter, we presented safety strategies for SMPC. We first showed a safe and efficient SMPC+FTP method for self-driving vehicles. While SMPC is used to plan optimistic and efficient vehicle trajectories, a failsafe trajectory planning MPC problem ensures that only those SMPC inputs are applied, which keep the vehicle in a safe state. We then generalized the SMPC+FTP method to linear systems subject to additive uncertainty. This safe SMPC algorithm offers the possibility to provide a safety guarantee for general SMPC approaches. In addition, recursive feasibility and stability is guaranteed, without the need of a terminal constraint for SMPC.

The efficiency of the SMPC+FTP method depends on the proposed constraint generation. Extending and refining the case differentiation will have a positive effect on efficiency. Considering urban automated driving, the SMPC+FTP approach remains valid, however, the case differentiation must be adapted to fit the urban traffic environment.

The safe SMPC algorithm is not limited to SMPC. Instead of using SMPC, other controllers, e.g., learning-based methods, can be used. This would allow us to ensure safety and stability for learning-based controllers.

The presented safety methods for SMPC are suitable to be applied to further safety-critical transportation applications, such as currently developed air taxis. However, it is also possible to extend the application area to non-transportation applications, such as human-robot collaboration, where uncertainty is always present while safety must still be guaranteed.

6 Minimizing Constraint Violation Probability in MPC

SMPC ensures that constraints are only violated up to a level allowed by the risk parameter. This chapter focuses on minimizing constraint violation probability in MPC, which is especially relevant for safety-critical systems. Part of the content of this chapter was published in [29], particularly Section 5.2.

6.1 Introduction

Autonomous systems in safety-critical applications, such as autonomous driving or human-robot interaction, depend on controllers that are able to safely and efficiently cope with uncertainties. In these applications, autonomous vehicles and robots must avoid collisions to ensure safety while also optimizing other objectives, e.g., energy consumption. For this problem setup, MPC is a promising method due to its ability to cope with hard constraints while optimizing an objective function.

When uncertainty is present, constraints are handled in a robust way by RMPC [9]. Initially known bounds on the uncertainty enable a stability guarantee and recursive feasibility of the RMPC optimal control problem. Nevertheless, robustly accounting for uncertainty comes with issues. On the one hand, if the uncertainty bound was initially not estimated or chosen large enough, all guarantees are lost. On the other hand, if uncertainty bounds are chosen too large, potentially to account for rare worst-case events, RMPC becomes highly conservative.

The issue of conservatism in RMPC is addressed by SMPC [68,134]. In SMPC, constraints subject to uncertainty are handled as chance constraints. Again, multiple issues arise. Similar to RMPC, wrong initial assumptions for the uncertainty cause feasibility issues of the SMPC optimal control problem. A further drawback of chance constraints in SMPC appears if the optimal solution is ‘on the chance constraint’ even though other solutions are possible with no or only minimal effect on the cost function. In other words, a solution to the SMPC optimal control problem minimizes the cost function and satisfies the required probability for the chance constraint. There might be other solutions with low cost that have a chance constraint violation probability less than required by the risk parameter or even zero. However, the SMPC optimal control problem is considered to be solved once a solution is found with minimal cost, which satisfies the chance constraint. This means that the solution with a lower constraint violation probability is not found. Furthermore, choosing a suitable risk parameter is challenging, as high values increase risk while low values reduce efficiency.

These issues are especially relevant in safety-critical systems. One example is an autonomous vehicle that plans to avoid collisions in an uncertain environment, e.g., a car avoiding a bicycle with uncertain behavior. If the support of the uncertainty is not known a priori, RMPC algorithms are either not applicable or require that the vehicle does not move until all surrounding vehicles are distanced enough. This, however, is not practical. Therefore, the collision constraint, for example realized with a norm constraint, could be transformed into a chance constraint in an SMPC approach, allowing for a small collision probability. While this yields a more efficient solution than RMPC, a collision might result. Further, if the chance constraint in SMPC cannot be satisfied anymore because an unlikely scenario occurred or the uncertainty support changed, the optimal control problem becomes infeasible. Alternative control laws and recovery strategies can then be used to regain a feasible controller, or trajectories can be calculated that yield minimal intrusion [211]. However, in such scenarios where the chance constraint cannot be satisfied, the controller ideally yields the safest solution possible, which is not guaranteed with standard recovery strategies. In the example of the autonomous vehicle, the safest solution is the one with the lowest collision probability.

Therefore, when designing MPC for safety-critical systems, the following requirements must be addressed:

- Minimal constraint violation probability.
- Ability to cope with time-variant uncertainty bounds and constraints.
- Recursive feasibility of the MPC optimal control problem.
- Stability.

Whereas RMPC and SMPC partially consider these requirements, both methods are impractical for safety-critical systems due to the above-mentioned issues.

In this chapter, we propose a novel MPC strategy that minimizes the probability that constraint are violated. In the following, we will refer to the proposed method as CVPM-MPC, i.e., MPC with constraint violation probability minimization (CVPM).

We first present an approach specifically suited for collision avoidance, which satisfies general hard constraints over the entire prediction horizon, but additionally minimizes the probability of violating a norm constraint in the next predicted step while also optimizing for other control objectives. This is achieved by first calculating a set that constrains the system inputs such that only those inputs are allowed that minimize the constraint violation probability. This is then followed by an optimal control problem that optimizes further required objectives such as fast reference tracking or energy consumption. In this subsequent optimal control problem, only those inputs are admissible that minimize the norm constraint violation probability. It can be difficult to determine a tightened set of admissible inputs that guarantee minimal constraint violation probability. Therefore, we suggest an approach that allows for the computation of this tightened input set, given uncertainties with symmetric, unimodal probability density function, i.e., the relative likelihood of uncertainty realizations decreases with increased distance to the mean. This tractable approach yields a convex set of inputs that minimize the constraint violation probability. Guarantees are provided for recursive feasibility and convergence of the proposed MPC algorithm. A simulation for a vehicle collision avoidance scenario is shown to display the effectiveness of the proposed method and highlight its advantages compared to SMPC and RMPC.

Second, we generalize the norm-based CVPM-MPC approach for linear systems with polytopic constraints. Additionally, the general CVPM-MPC method makes it possible to minimize constraint violation probability over multiple prediction steps. We guarantee recursive feasibility of the MPC algorithm for time-variant uncertainty bounds and time-variant constraints. Furthermore, we prove input-to-state stability of the system origin, assuming a constant uncertainty bound. The benefits of the proposed approach is demonstrated in a numerical simulation.

In summary, the contributions are as follows.

- A novel CVPM-MPC method for norm constraints or linear constraints, which minimizes the constraint violation probability while optimizing further objectives.
- Control behavior with safer solutions than SMPC and less conservatism than RMPC.
- Ability to handle time-varying uncertainty bounds and constraints.
- Guarantee of recursive feasibility when using CVPM-MPC with norm constraints or linear constraints, and input-to-state stability for CVPM-MPC with linear state constraints.

The proposed CVPM-MPC method can be beneficial in multiple applications, especially in safety-critical applications such as autonomous driving or human-robot interaction where the risk measure regarding collision is norm-based [50, 173, 205]. In these safety-critical applications, there is a clear priority on maximizing the probability of safety, i.e., the constraint violation probability of safety constraints needs to be minimal, before optimizing other objectives, e.g., energy consumption.

6.1.1 Related Work

Classic MPC methods deal well with deterministic systems and provide guarantees for stability as well as recursive feasibility [78,132,155], where recursive feasibility ensures that the MPC optimal control problem remains feasible at future time steps if it is initially feasible.

More advanced MPC algorithms are necessary in the presence of uncertainty in the system. RMPC methods [129] provide control laws that satisfy the control objectives and constraints by accounting for the worst-case realization of the uncertainty, assuming that the bound, i.e., the support, of the probability distribution for the uncertainty is known a priori [124, 133]. The most prominent RMPC approaches are min-max MPC [156], considering maximal uncertainties, and tube-based MPC [102,108,130], which defines a tube around the nominal state trajectory in order to account for uncertainty. A different approach is proposed in [175] where MPC is combined with reachability analysis, designed for safety-critical systems and specifically considering computation time. In contrast to other methods, no Lyapunov function is required, which might be hard to obtain. Recursive feasibility is ensured based on the solution of the previous time step, whereas convergence is shown based on dual-mode MPC [137]. A major drawback of RMPC is its conservative control law due to accounting for the worst-case uncertainty realization. This can be problematic in applications with high levels of uncertainty, e.g., autonomous driving in dense traffic.

This issue is addressed by SMPC methods, which exploit knowledge of the uncertainty by introducing chance constraints and potentially applying an expectation value based objective function. In many applications, it is acceptable to allow for a small probability of constraint violation. This results in a positive effect on performance, as the control law is no longer required to account for unlikely uncertainty realizations.

Recursive feasibility of SMPC for bounded disturbances is addressed in [104]. A further approach addressing recursive feasibility in SMPC are stochastic tube methods [44,105] using constraint tightening. In [116] an approach is suggested that combines the works of [104] and [105] where a tuning parameter is introduced that allows for shifting priority between performance and an increased feasible region. Covariance steering-based SMPC [145, 146] is a further SMPC approach, which ensures recursive feasibility for linear systems with unbounded noise. Recursive feasibility in SMPC for probabilistically constrained Markovian jump linear systems is addressed in [119]

Due to its ability to efficiently cope with environments subject to uncertainty, SMPC has become increasingly popular in applications. In most SMPC applications, constraint violations are not critical, e.g., energy control in buildings [121, 148] or hybrid electrical vehicles [57,209]. However, the possible constraint violation and the resulting infeasibility of the optimal control problem are limiting factors when designing an efficient SMPC algorithm in practice, especially in safety-critical applications.

However, safety within SMPC is only specifically addressed in few works. In [211] an approach is presented, where a least-intrusive trajectory is found if a collision is inevitable. Recently, robust and stochastic MPC have also been used to provide safety for reinforcement learning [194,195,207]. A different, yet promising way to include safety into MPC are the recently proposed methods combining control barrier functions with MPC [77,165,208]. However, major challenges remain, e.g., considering uncertainty and the guarantee of recursive feasibility.

All the previously described SMPC approaches fail, however, once unexpected changes arise, potentially due to changing uncertainties or changing constraints. While slack variables can be introduced or alternative problems can be solved [45,46] to regain feasibility, there is no guarantee that the obtained solution is optimal from a safety point of view.

In summary, previous MPC approaches only cover parts of the requirements for safety-critical systems. The major challenge, reasonably minimizing constraint violation probability, is still an open problem and, therefore, addressed in the following.

6.1.2 Chapter Overview

The remainder of the chapter is structured as follows. Section 6.2 introduces CVPM-MPC for norm constraints and in Section 6.3 the CVPM-MPC method for linear constraints is proposed. A problem formulation is given in Section 6.2.1. The norm-constrained CVPM-MPC method is introduced in Section 6.2.2, first focusing on minimizing the constraint violation probability, then introducing the resulting MPC algorithm. Section 6.2.3 analyzes the properties of the norm-based CVPM-MPC method, guarantees on recursive feasibility and convergence. An example of the applied method is given in Section 6.2.4, simulating a vehicle collision avoidance scenario. The norm-based CVPM-MPC method is discussed in Section 6.2.5. Then, after providing an updated, more general problem formulation in Section 6.3.1, Section 6.3.2 proposes the CVPM-MPC method for linear constraints. This is followed by details on the properties in Section 6.3.3. A numerical example is shown in Section 6.3.4, demonstrating the benefits of applying CVPM-MPC to safety-critical systems. A discussion is given in Section 6.3.5. Section 6.4 provides conclusive remarks.

6.2 CVPM-MPC for Norm Constraints

The safety-critical aspect of safety-critical systems usually addresses collision avoidance, e.g., in robotics or automated driving. Here, we present an MPC method that minimizes the constraint violation probability of a norm-based constraint, which can be used to represent collision avoidance.

6.2.1 Problem Formulation

In this section, we define the system class and the general MPC algorithm. Additionally, a probabilistic norm constraint is introduced. Based on these preliminaries, the problem statement is given subsequently.

System Dynamics and Control Objective

Consider the controlled linear, time-invariant, discrete-time system

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t, \tag{6.1a}$$

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t \tag{6.1b}$$

with time step t, states $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$, control input $\boldsymbol{u}_t \in \mathbb{R}^{n_{\boldsymbol{u}}}$, output $\boldsymbol{y}_t \in \mathbb{R}^{n_{\boldsymbol{y}}}$, and matrices $\boldsymbol{A} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}}$, $\boldsymbol{B} \in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{u}}}$, $\boldsymbol{C} \in \mathbb{R}^{n_{\boldsymbol{y}} \times n_{\boldsymbol{x}}}$.

Furthermore, we consider a dynamic obstacle (DO), which is a system subject to uncertainty. The uncertain system dynamics are given by

$$\boldsymbol{y}_{t+1}^{\mathrm{DO}} = \boldsymbol{y}_t^{\mathrm{DO}} + \boldsymbol{u}_t^{\mathrm{DO}} + \boldsymbol{w}_t \tag{6.2a}$$

$$= \overline{\boldsymbol{y}}_{t+1}^{\mathrm{DO}} + \boldsymbol{w}_t \tag{6.2b}$$

depending on the output $\boldsymbol{y}_t^{\mathrm{DO}} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ at time step t, a deterministic, known input $\boldsymbol{u}_t^{\mathrm{DO}} \in \mathbb{R}^{n_{\boldsymbol{y}}}$, and a stochastic part $\boldsymbol{w}_t \in \mathbb{R}^{n_{\boldsymbol{y}}}$, which is the realization of a random variable $\boldsymbol{W}_t$. The nominal prediction of $\boldsymbol{y}_{t+1}^{\mathrm{DO}}$ is indicated by $\overline{\boldsymbol{y}}_{t+1}^{\mathrm{DO}} = \boldsymbol{y}_t^{\mathrm{DO}} + \boldsymbol{u}_t^{\mathrm{DO}}$, consisting of the previous output $\boldsymbol{y}_t^{\mathrm{DO}}$ and the deterministic, known input $\boldsymbol{u}_t^{\mathrm{DO}}$.

Assumption 6.1. *The random variables $\boldsymbol{W}_t\,(\boldsymbol{w}_t) \sim f_{\boldsymbol{W}_t}$ with the probability distribution $p_{\boldsymbol{W}_t}$ and density function $f_{\boldsymbol{W}_t}$ have zero mean and are truncated with the initially known, convex and bounded support* $\mathrm{supp}\,(f_{\boldsymbol{W}_t})$.

The support of $f_{\boldsymbol{W}_t}$ is given by

$$\mathrm{supp}\,(f_{\boldsymbol{W}_t}) = \{\boldsymbol{w}_t \mid \|\boldsymbol{w}_t\|_2 \leq w_{\max,t}\} \tag{6.3}$$

where $w_{\max,t} \in \mathbb{R}_{\geq 0}$.

The controller for (6.1) is designed to optimize a finite horizon objective function while accounting for input and state constraints.

Model Predictive Control

We consider an MPC algorithm with a finite horizon objective function J, similar to Section 2.2.3. The finite horizon cost $J\left(\boldsymbol{x}_0, \boldsymbol{U}\right)$, with the MPC horizon N, continuous stage cost $l\left(\boldsymbol{x}_j, \boldsymbol{u}_j\right) = \boldsymbol{x}_j^\top \boldsymbol{Q} \boldsymbol{x}_j + \boldsymbol{u}_j^\top \boldsymbol{R} \boldsymbol{u}_j$ with $l\left(\boldsymbol{0}, \boldsymbol{0}\right) = 0$, $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$, $\boldsymbol{R} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{Q} \succeq 0$, $\boldsymbol{R} \succ 0$, and continuous terminal cost V_f with $V_\mathrm{f}\left(\boldsymbol{0}\right) = 0$, is then given by

$$J\left(\boldsymbol{x}_0, \boldsymbol{U}\right) = \sum_{k=0}^{N-1} \left(l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right) + V_\mathrm{f}(\boldsymbol{x}_N) \tag{6.4}$$

with the input sequence $\boldsymbol{U} = \left(\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1}\right)$.

We first formulate the MPC optimal control problem, including a terminal state constraint $\mathcal{X}_\mathrm{f}$, resulting in

$$\begin{aligned} \min_{\boldsymbol{U}} \ & J\left(\boldsymbol{x}_0, \boldsymbol{U}\right) && \text{(6.5a)} \\ \text{s.t. } & \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k && \text{(6.5b)} \\ & \boldsymbol{u}_k \in \mathcal{U}, \quad k \in \mathbb{I}_{0,N-1} && \text{(6.5c)} \\ & \boldsymbol{x}_k \in \mathcal{X}, \quad k \in \mathbb{I}_{1,N} && \text{(6.5d)} \\ & \boldsymbol{x}_N \in \mathcal{X}_\mathrm{f}. && \text{(6.5e)} \end{aligned}$$

The input $\boldsymbol{u}_k$ is bounded by the non-empty input value space $\mathcal{U} \subseteq \mathbb{R}^{n_{\boldsymbol{u}}}$. The convex state constraint is given by $\mathcal{X}$.

Assumption 6.2. *For all $\boldsymbol{x}_k \in \mathcal{X}_\mathrm{f}$, there exists an admissible $\boldsymbol{u}_k$ such that $\boldsymbol{x}_{k+1} \in \mathcal{X}_\mathrm{f}$.*

Assumption 6.3. *The stage cost $l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$ is continuous, positive definite, and $l(\boldsymbol{0}, \boldsymbol{0}) = 0$. The terminal cost V_f is a Lyapunov function in $\mathcal{X}_\mathrm{f}$.*

We denote with $\mathcal{U}_{\boldsymbol{x},k}$ the set of admissible inputs $\boldsymbol{u}_k$ such that all constraints of (6.5) are satisfied for $j \in \mathbb{I}_{k:N-1}$, i.e.,

$$\mathcal{U}_{\boldsymbol{x},k} = \left\{\boldsymbol{u}_k \mid \boldsymbol{u}_j \in \mathcal{U} \ \forall \ j \in \mathbb{I}_{k,N-1}, \quad \boldsymbol{x}_j \in \mathcal{X} \ \forall \ j \in \mathbb{I}_{k+1,N}, \quad \boldsymbol{x}_N \in \mathcal{X}_\mathrm{f}\right\}, \quad k \in \mathbb{I}_{0:N-1} \tag{6.6}$$

where the special case $\mathcal{U}_{\boldsymbol{x},0}$ is given by

$$\mathcal{U}_{\boldsymbol{x},k} = \left\{\boldsymbol{u}_0 \mid \text{(6.5c)}, \text{(6.5d)}, \text{(6.5e)}\right\}. \tag{6.7}$$

Remark 6.1. *Instead of steering $\boldsymbol{x}_t$ to the origin as in* (6.4)*, specific references $\boldsymbol{x}_{\mathrm{ref},t}$ can be tracked, as described in Rawlings et al. [160, Chapter 1.5].*

MPC with a Probabilistic Norm Constraint

In the following, the uncertain system (6.2) is considered and a probabilistic norm constraint is introduced.

Assumption 6.4. *The initial state $\boldsymbol{y}_0^\mathrm{DO}$ and deterministic input $\boldsymbol{u}_0^\mathrm{DO}$ are known at the beginning of each MPC optimal control problem.*

Here, we consider an additional constraint for the MPC problem (6.5), which is the norm constraint

$$\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2 \geq h_t \tag{6.8}$$

representing a constraint on the 2-norm $\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2$, e.g., the distance between two points must not be smaller than a minimal value h_t that can be used for collision avoidance. While (6.8) is a hard constraint, we will first transform (6.8) into a chance constraint and later, in Section 6.2.2, we will minimize the probability that this norm constraint is violated.

Remark 6.2. *It is also possible to consider a p-norm constraint with* $\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_p$ *instead of the* 2*-norm. Similar to the* 2*-norm, all p-norms are convex. Without loss of generality we will consider the* 2*-norm as most applications require a* 2*-norm to represent the Euclidean distance.*

As $\boldsymbol{y}_t^{\mathrm{DO}}$ is subject to uncertainty, the norm constraint (6.8) is difficult, potentially impossible, to fulfill, or it might lead to overly conservative control inputs. The hard norm constraint (6.8) can be relaxed if substituted by the chance constraint

$$\Pr\left(\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2 < h_t\right) \leq \beta_t \tag{6.9}$$

with

$$p_t \coloneqq \Pr\left(\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2 < h_t\right) \tag{6.10}$$

where β_t is a risk parameter and p_t denotes the constraint violation probability for the norm constraint (6.8). We obtain $p_t = 0$ if the maximal uncertainty value $w_{\max,t-1}$ cannot cause $\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2 < h_t$.

The following example will illustrate the idea of the chance constraint. A similar example is analyzed in a simulation in Section 6.2.4.

Example 6.1. *We consider a controlled object with position* $\boldsymbol{y}_t$ *and a dynamic obstacle with position* $\boldsymbol{y}_t^{\mathrm{DO}}$ *where* $\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2$ *is the distance between both objects. The objects collide if* $\left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2 < h_t$*. An interpretation for* (6.9) *is that* p_t *represents the probability of a collision and this constraint violation probability is bounded by a predefined risk parameter* β_t*.*

We now consider the chance constraint (6.9) within the MPC optimal control problem. Whereas it is possible to consider the norm constraint (6.8) over multiple steps, within the MPC optimal control problem, we only consider the norm constraint for the next predicted step $k = 1$ with a horizon $N \geq 1$. Applying (6.8) over the entire horizon N results in a conservative control law similar to RMPC. The one-step chance constraint is given by

$$\Pr\left(\left\|\boldsymbol{y}_1 - \boldsymbol{y}_1^{\mathrm{DO}}\right\|_2 < h_1\right) \leq \beta_1 \tag{6.11}$$

where we define

$$p_1\left(\boldsymbol{u}_0\right) \coloneqq \Pr\left(\left\|\boldsymbol{y}_1 - \boldsymbol{y}_1^{\mathrm{DO}}\right\|_2 < h_1\right). \tag{6.12}$$

Note that the dependence of $p_1\left(\boldsymbol{u}_0\right)$ on $\boldsymbol{y}_0$ and $\boldsymbol{y}_0^{\mathrm{DO}}$ is omitted due to clarity, as these values will not be decision variables in the MPC optimal control problem.

Only the general MPC problem (6.5) is addressed in Assumptions 6.2 and 6.3. The norm constraint (6.8) is not considered in these assumptions, as it is specifically addressed in the method presented in Sections 6.2.2 and 6.2.3.

Remark 6.3. *The norm constraint* (6.8) *is only considered in the first prediction step, i.e., at step $k = 1$, as we later minimize the probability of constraint violation for the first step. However, if this norm constraint is required to be considered at future steps $k \in \mathbb{I}_{2:N}$, this can be achieved by treating* (6.8) *as a chance constraint, similar to* (6.11)*, resulting in*

$$\Pr\left(\left\|\boldsymbol{y}_k - \boldsymbol{y}_k^{\mathrm{DO}}\right\|_2 < h_k\right) \leq \beta_k, \quad k \in \mathbb{I}_{2:N}. \tag{6.13}$$

This chance constraint (6.13) *is then added to* (6.5) *and subsequently needs to be considered in* (6.6)*. Assumptions 6.2 and 6.3 still need to be fulfilled if chance constraints are included for $k \in \mathbb{I}_{2,N}$ in the optimal control problem.*

Problem Statement

Instead of only bounding the chance constraint (6.11) by the risk parameter β_1, we aim at minimizing the constraint violation probability p_1 within the MPC optimal control problem.

Objective 6.1. *The goal is to solve the MPC problem*

$$\begin{aligned} J^* = \min_{\boldsymbol{U}} \; & J\left(\boldsymbol{x}_0, \boldsymbol{U}\right) && (6.14a)\\ \text{s.t. } & \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k && (6.14b)\\ & \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k && (6.14c)\\ & \boldsymbol{u}_k \in \mathcal{U}_{\boldsymbol{x},k}, \quad k \in \mathbb{I}_{0,N-1} && (6.14d) \end{aligned}$$

while it needs to be guaranteed that

$$\boldsymbol{u}_0 = \arg\min_{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}} \Pr\left(\left\|\boldsymbol{y}_1 - \boldsymbol{y}_1^{\mathrm{DO}}\right\|_2 < h_1\right) \tag{6.15}$$

and that the MPC problem remains recursively feasible.

In Objective 6.1, (6.14d) summarizes the constraints of the initial MPC problem (6.5), according to the definition of $\mathcal{U}_{\boldsymbol{x},k}$ in (6.6), and (6.15) ensures that p_1 is minimized.

This problem formulation allows us to find a method that additionally handles a further important challenge in safety-critical applications, which are problematic in SMPC and RMPC. It is fundamental to maintain recursive feasibility of the MPC optimal control problem for an unexpectedly increasing uncertainty support.

Here, we propose a novel MPC approach, CVPM-MPC, that first ensures the minimal constraint violation probability p_1, but then still optimizes the cost function $J(\boldsymbol{x}_0, \boldsymbol{U})$. This approach yields a control input resulting in the lowest possible constraint violation probability, given input and state constraints, while still optimizing further objectives. The CVPM-MPC method guarantees recursive feasibility, also for a changing uncertainty support, and ensures convergence of the MPC algorithm.

6.2.2 Method

In this section, we derive the CVPM-MPC method to minimize the constraint violation probability p_1 for the first predicted step $k = 1$ in an MPC problem. First, a general approach is presented to find a tightened admissible input set that minimizes the first step constraint violation probability. In the following part, it is shown how this approach can

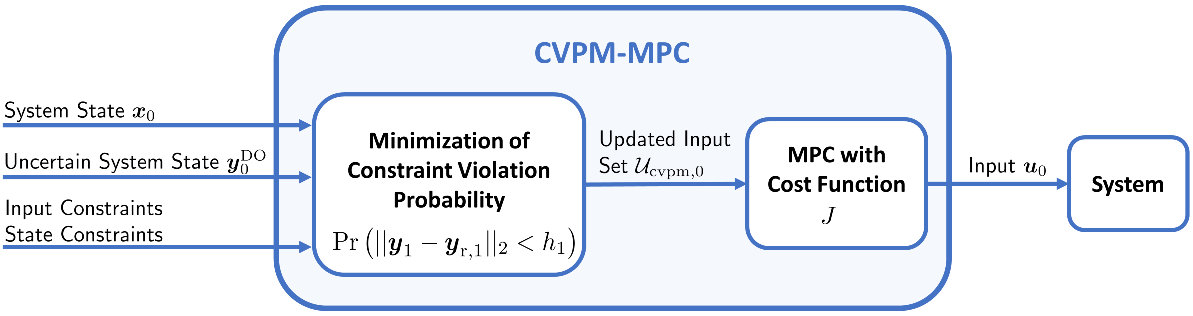

CVPM-MPC
System State $\boldsymbol{x}_0$
Uncertain System State $\boldsymbol{y}_0^{\mathrm{DO}}$
Input Constraints
State Constraints
Minimization of Constraint Violation Probability
$\Pr\left(\|\boldsymbol{y}_1 - \boldsymbol{y}_{\mathrm{r},1}\|_2 < h_1\right)$
Updated Input Set $\mathcal{U}_{\mathrm{cvpm},0}$
MPC with Cost Function
J
Input $\boldsymbol{u}_0$
System

Figure 6.1: Visualization of the CVPM-MPC method: Given an input set and state constraints, as well as the current system state and uncertain system state, an updated input set is determined. This updated input set minimizes the norm constraint violation probability for the next step. Then, an MPC optimal control problem is solved. The updated input set ensures constraint violation probability minimization while optimizing for other objectives.

be incorporated into MPC. A visualization of the method is displayed in Figure 6.1. As determining the tightened input set within the CVPM-MPC method is difficult in general, we then provide an alternative, computable approach, assuming an uncertainty with symmetric, unimodal PDF. A solution approach for a multi-step CVPM-MPC is described in [29].

General Method to Minimize Constraint Violation Probability for One-Step Problem

When minimizing p_1 over $\boldsymbol{u}_0$ within the MPC algorithm, three different cases need to be considered. In each case, a set $\mathcal{U}_{\mathrm{cvpm},0}$ is determined, which consists of inputs $\boldsymbol{u}_0$ that minimize the constraint violation probability. Ideally, even considering the bounded uncertainty, satisfaction of the constraint (6.8) may be guaranteed in the next step, for all choices of $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$, which will be referred to as case 1. However, for stochastic systems we potentially have the situation that case 1 cannot be guaranteed. Here, two cases need to be distinguished. First, given the uncertainty, there is no choice for $\boldsymbol{u}_0$ that guarantees constraint satisfaction (case 2). Second, some choices for $\boldsymbol{u}_0$ guarantee constraint satisfaction, while other choices do not lead to such a guarantee (case 3). Depending on the case, $\mathcal{U}_{\mathrm{cvpm},0}$ is determined differently as described in the following.

Case 1 (guaranteed constraint satisfaction). The probability of violating the norm constraint is zero independent of the choice for $\boldsymbol{u}_0$, i.e.,

$$p_1\left(\boldsymbol{u}_0\right) = 0 \quad \forall \boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}. \tag{6.16}$$

Therefore, every $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ is a valid input, resulting in

$$\mathcal{U}_{\mathrm{cvpm},0} = \mathcal{U}_{\boldsymbol{x},0}. \tag{6.17}$$

Case 2 (impossible constraint satisfaction guarantee). There is no choice for $\boldsymbol{u}_0$ such that constraint satisfaction can be guaranteed in the presence of uncertainty, i.e.,

$$p_1\left(\boldsymbol{u}_0\right) > 0 \quad \forall \boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}. \tag{6.18}$$

As it is impossible to guarantee $p_1 = 0$, the aim is to minimize p_1. Selecting

$$\mathcal{U}_{\mathrm{cvpm},0} = \left\{ \boldsymbol{u}_0 \,\middle|\, \boldsymbol{u}_0 = \underset{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}}{\arg\min}\, p_1\left(\boldsymbol{u}_0\right) \right\} \tag{6.19}$$

yields the set $\mathcal{U}_{\mathrm{cvpm},0}$, which only consists of inputs $\boldsymbol{u}_0$ that minimize p_1.

Case 3 (possible constraint satisfaction guarantee). If only some inputs $\boldsymbol{u}_0$ guarantee satisfaction of the norm constraint (6.8), i.e.,

$$\exists\ \boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0} \ \text{ s.t. } \ p_1\left(\boldsymbol{u}_0\right) = 0, \tag{6.20}$$

then the set

$$\mathcal{U}_{\mathrm{cvpm},0} = \left\{ \boldsymbol{u}_0 \mid \left(p_1\left(\boldsymbol{u}_0\right) = 0\right) \wedge \left(\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}\right) \right\} \tag{6.21}$$

consists of these inputs that yield constraint satisfaction.

In all three cases $\mathcal{U}_{\mathrm{cvpm},0}$ needs to be found, requiring the following strong assumption.

Assumption 6.5. *The set $\mathcal{U}_{\mathrm{cvpm},0}$ can be determined for all cases 1-3.*

While it is possible to approximate $\mathcal{U}_{\mathrm{cvpm},0}$ by sampling, finding an analytic solution for $\mathcal{U}_{\mathrm{cvpm},0}$ highly depends on the probability distribution. However, if $\mathcal{U}_{\mathrm{cvpm},0}$ can be determined, the CVPM-MPC method guarantees minimal constraint violation probability for p_1.

Theorem 6.1. *If Assumption 6.5 holds, minimization of the constraint violation probability of p_1 is guaranteed by selecting $\mathcal{U}_{\mathrm{cvpm},0}$ according to cases 1-3.*

Proof. The proof follows straightforward from the definition of the three cases. All possibilities are covered regarding the guarantee of constraint satisfaction, i.e., guaranteed constraint satisfaction (case 1), impossible constraint satisfaction guarantee (case 2), and the case where constraint satisfaction is only guaranteed for some but not all $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ (case 3). If $p_1 = 0$ is possible, i.e., case 1 or 3, (6.17) and (6.21) guarantee that $\mathcal{U}_{\mathrm{cvpm},0}$ consists only of inputs $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ that yield $p_1 = 0$. If no $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ guarantees $p_1 = 0$, minimal constraint violation is guaranteed by only allowing inputs $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ that minimize p_1 according to (6.19). □

In dynamic environments the worst-case uncertainty $w_{\max,t}$ may change over time, which influences the probability of constraint violations. If the support changes, the CVPM-MPC approach still minimizes this constraint violation probability.

Corollary 6.1. *If the uncertainty support* $\operatorname{supp}\left(f_{\boldsymbol{W}_t}\right)$ *changes from step t to $t+1$, the CVPM-MPC problem solved at step $t+1$ guarantees that the constraint violation probability p_{t+2} is minimized.*

Proof. The proof follows directly from the problem definition. First, the CVPM-MPC approach ensures that the constraint violation probability is minimized for each step, which allows $p_{t+2} > p_{t+1}$ if the uncertainty support increases. Second, minimizing p_{t+2} is independent of minimizing p_{t+1}. □

The MPC problem (6.14) is now adapted given the set $\mathcal{U}_{\mathrm{cvpm},0}$ to guarantee minimal constraint violation probability of the norm constraint while still optimizing for further objectives.

Model Predictive Control with Minimal First Step Constraint Violation Probability

Applying the previously determined $\mathcal{U}_{\text{cvpm},0}$ yields the CVPM-MPC problem

$$\begin{aligned}
\min_{\boldsymbol{U}}\ & J\left(\boldsymbol{x}_0, \boldsymbol{U}\right) && \text{(6.22a)}\\
\text{s.t. } & \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k && \text{(6.22b)}\\
& \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k && \text{(6.22c)}\\
& \boldsymbol{U} \in \mathcal{U}_0^*. && \text{(6.22d)}
\end{aligned}$$

The set $\mathcal{U}_0^*$ defines the admissible inputs that yield minimal constraint violation probability combined with keeping the inputs and states within the input and state constraint sets. The set $\mathcal{U}_0^*$ is given by

$$\mathcal{U}_0^* = \left\{\boldsymbol{U} \mid \left(\boldsymbol{u}_0 \in \mathcal{U}_{\text{cvpm},0}\right) \wedge \left(\boldsymbol{u}_k \in \mathcal{U}_{\boldsymbol{x},k}, k \in \mathbb{I}_{1:N-1}\right)\right\} \tag{6.23}$$

where $\mathcal{U}_{\boldsymbol{x},k}$ is defined in (6.6) and $\mathcal{U}_{\text{cvpm},0}$ is obtained as described previously.

The complete CVPM-MPC problem (6.22) allows us to optimize a cost function and satisfy state and input constraints while minimization of the constraint violation probability p_1 is ensured.

Minimal Constraint Violation Probability for One-Step Problem with Symmetric Unimodal PDF

The previously proposed CVMP-MPC method only guarantees minimal constraint violation probability if Assumption 6.5 is fulfilled. Therefore, it must be possible to always determine $\mathcal{U}_{\text{cvpm},0}$, which is a strong assumption. In the following, we provide an adapted approach of the CVMP-MPC method, which guarantees minimal constraint violation probability if the PDF of the uncertainty is symmetric and unimodal.

We first give a definition of symmetric, unimodal PDFs. Further, we introduce a substitute expression for the constraint violation probability p_t. Then, the three cases are adapted in order to minimize p_1 for the PDF addressed in the following. For each case a convex set of admissible inputs $\mathcal{U}_{\text{cvpm},0}$ is determined.

Symmetric unimodal PDF. We first define the class of symmetric, unimodal probability distributions.

Definition 6.1 (Symmetric Unimodal Distribution)**.** *A probability distribution is symmetric and unimodal if its PDF has a single mode, i.e., a single global maximum, which coincides with its mean $\boldsymbol{\mu}$ and*

$$f\left(\boldsymbol{\mu} + \boldsymbol{\tau}_1\right) = f\left(\boldsymbol{\mu} + \boldsymbol{\tau}_2\right) \ \forall \ \left\|\boldsymbol{\tau}_1\right\|_2 = \left\|\boldsymbol{\tau}_2\right\|_2. \tag{6.24}$$

With Definition 6.1 it is ensured that the PDF has its peak at mean $\boldsymbol{\mu}$ and that the PDF is strictly radially decreasing. As the probability distribution is symmetric, all realizations with similar distance to $\boldsymbol{\mu}$ have the same relative likelihood. Since there is only one global maximum of the PDF at $\boldsymbol{\mu}$, realizations with increasing distance to $\boldsymbol{\mu}$ have a lower relative likelihood.

The constraint violation probability p_t is a probabilistic expression and cannot directly be used in the optimal control problem. The following assumption will allow us to find a deterministic substitute for p_t.

Assumption 6.6. *The PDF $f_{\boldsymbol{W}_t}$ for $\boldsymbol{W}_t$ in* (6.2) *is symmetric and unimodal with mean $\boldsymbol{\mu} = \boldsymbol{0}$.*

Example 6.2. *An example for an admissible probability distribution $p_{\boldsymbol{W}_t}$ with symmetric, unimodal PDF is a truncated isotropic bivariate normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ with covariance matrix*

$$\Sigma = \operatorname{diag}\left(\sigma_1^2, \sigma_2^2\right) = \sigma^2 \boldsymbol{I},\ \ \sigma = \sigma_1 = \sigma_2 \tag{6.25}$$

with variance σ^2 and identity matrix $\boldsymbol{I}$. The support in each direction is required to be equal, which can be achieved by over-approximating. Distributions with $\sigma_1 \neq \sigma_2$ can be over-approximated by choosing

$$\boldsymbol{\Sigma} = \sigma_{\max} \boldsymbol{I},\ \ \sigma_{\max} = \max\left(\sigma_1, \sigma_2\right). \tag{6.26}$$

We now address the relation between p_t and $f_{\boldsymbol{W}_t}$ considering Assumption 6.6. The following lemma shows that the constraint violation probability p_t can be decreased by choosing $\boldsymbol{u}_{t-1}$ such that the distance is increased between the next system output $\boldsymbol{y}_t$ and the next known, nominal random system output $\overline{\boldsymbol{y}}_t^{\mathrm{DO}}$.

Lemma 6.1. *If Assumption 6.6 holds, the probability p_t is decreasing for an increasing norm $\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\mathrm{DO}}\right\|_2$.*

Proof. The proof is derived in Appendix D.1. □

The lemma shows that the probability of violating the norm constraint (6.8) decreases if the difference between $\boldsymbol{y}_t$ and $\overline{\boldsymbol{y}}_t^{\mathrm{DO}}$ increases. Lemma 6.1 now allows us to find a substitute function for p_t.

Substitute probability function. The probability p_k cannot be used directly to obtain the set $\mathcal{U}_{\mathrm{cvpm},0}$. Therefore, a substitution is required for p_k. Based on Lemma 6.1, the probability p_k decreases for an increasing norm $\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2$. This property is used to choose a substitution for the constraint violation probability p_k. Here, the substitute function is selected to be

$$\varrho\left(\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2\right) = \left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2^2. \tag{6.27}$$

While p_k is decreasing with the norm $\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2$, the function $\varrho\left(\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2\right)$ is increasing with $\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2$. Therefore, increasing the value of function $\varrho\left(\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2\right)$ yields a reduced probability p_k, which is exploited in the following to minimize constraint violation probability.

Remark 6.4. *While $\varrho\left(\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2\right) = \left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2^2$ is adequate for most safety-critical applications, other scalar functions $\varrho'\left(\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2\right)$ are possible, as long as they are twice differentiable and strictly monotonically increasing.*

Considering the constraint violation probability for the first predicted step $k = 1$, this probability p_1 is minimized for a maximal $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$. However, since $f_{\boldsymbol{W}_t}$ is truncated and p_t is bounded, there potentially are multiple admissible inputs that result in an equal constraint violation probability. The aim is now to find the convex set $\mathcal{U}_{\mathrm{cvpm},0}$ including all inputs $\boldsymbol{u}_{\mathrm{cvpm},0} \in \mathcal{U}_{\mathrm{cvpm},0}$ that result in a minimal p_1. As $\boldsymbol{u}_0^{\mathrm{DO}}$ is deterministic and known according to Assumption 6.4, $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$ is a deterministic expression that can be evaluated.

The set $\mathcal{U}_{\mathrm{cvpm},0}$ can then be found by comparing the worst-case uncertainty $w_{\max,0}$ with the minimum and maximum possible values of $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$, i.e., $\varrho_{\min,1}$ and $\varrho_{\max,1}$, respectively. The maximal value $\varrho_{\max,1}$ is determined by

$$\varrho_{\max,1} \coloneqq \max_{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}} \varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) = \varrho\left(\max_{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}} \left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)\right) \tag{6.28}$$

corresponding to the largest distance between $\boldsymbol{y}_1$ and $\overline{\boldsymbol{y}}_1^{\mathrm{DO}}$. Analogously $\varrho_{\min,1}$ can be found by

$$\varrho_{\min,1} \coloneqq \min_{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}} \varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) = \varrho\left(\min_{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}} \left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)\right). \tag{6.29}$$

The result for $\varrho_{\min,1}$ can be obtained by determining the minimum value of $\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2$, as the objective function $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$ and $\mathcal{U}_{\boldsymbol{x},0}$ are convex. The following lemma provides a strategy to find $\varrho_{\max,1}$.

Lemma 6.2. *Let the non-empty convex polytope $\mathcal{V} \subset \mathbb{R}^{n_v}$, $n_{\boldsymbol{v}} \in \mathbb{N}$, be bounded by a finite set of hyperplanes, such that $\mathcal{V}$ has a finite number of edge vertices with a convex function $\varphi : \mathcal{V} \to \mathbb{R}$. Then, a global maximum*

$$\varphi_{\max} = \max_{\boldsymbol{v} \in \mathcal{V}} \varphi(\boldsymbol{v}) \tag{6.30}$$

is obtained by searching for the maximum value of φ on the boundary $\partial\mathcal{V}$ of its domain $\mathcal{V}$.

Proof. This proof is based on Bauer's maximum principle [8]. We consider any two points $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \partial\mathcal{V}$ on the boundary of $\mathcal{V}$. Any point on the line between $\boldsymbol{v}_1, \boldsymbol{v}_2$ can be described by $\boldsymbol{b} = \lambda_{\boldsymbol{b}}\boldsymbol{v}_1 + (1 - \lambda_{\boldsymbol{b}})\boldsymbol{v}_2$, using the definition of convexity. Due to the convexity of φ it holds that $\varphi(\boldsymbol{b}) \leq \max\{\varphi(\boldsymbol{v}_1), \varphi(\boldsymbol{v}_2)\}$. Any point on the line between $\boldsymbol{v}_1, \boldsymbol{v}_2$ can be reached by a convex combination. Since $\boldsymbol{v}_1, \boldsymbol{v}_2$ can be chosen arbitrarily, every point $\boldsymbol{b}$ in the interior of $\mathcal{V}$ can be reached. Therefore, a global maximum $\varphi_{\max}$ is found on the boundary $\partial\mathcal{V}$. □

Determination of the updated admissible input set. We again investigate three cases. The resulting set $\mathcal{U}_{\mathrm{cvpm},0}$, depending on the three cases, is then used in the CVPM-MPC problem (6.22) to guarantee minimal constraint violation probability of the norm constraint. In order to distinguish between the cases, we will consider the relation

$$\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2 \geq h_1 + w_{\max,0} \Rightarrow \left\|\boldsymbol{y}_1 - \boldsymbol{y}_1^{\mathrm{DO}}\right\|_2 \geq h_1, \tag{6.31}$$

which follows using a reverse triangle inequality as shown in Appendix D.2, given the dynamics of the uncertain system (6.2). Here, $h_1 + w_{\max,0}$ represents the necessary distance between

$\boldsymbol{y}_1$ and $\overline{\boldsymbol{y}}_1^{\mathrm{DO}}$, consisting of the required minimal distance h_1 at step $k = 1$ and the maximal random system step $w_{\max,0}$ at $k = 0$, such that $\left\|\boldsymbol{y}_1 - \boldsymbol{y}_1^{\mathrm{DO}}\right\|_2 \geq h_1$ for all $\|\boldsymbol{w}_0\|_2 \leq w_{\max,0}$.

Case 1 (guaranteed constraint satisfaction). For any $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ constraint satisfaction is guaranteed, i.e., $p_1 = 0$ for

$$\varrho_{\min,1} \geq \varrho\left(h_1 + w_{\max,0}\right). \tag{6.32}$$

The initial state configuration of the controlled and stochastic system is such that the minimum value possible for $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$, $\varrho_{\min,1}$, still yields a larger value than inserting h_1 combined with the worst-case $w_{\max,0}$ into ϱ, which moves $\boldsymbol{y}_1^{\mathrm{DO}}$ closest to $\boldsymbol{y}_1$. This results in a guaranteed constraint satisfaction $p_1 = 0$. Therefore, every $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ is an admissible input, i.e.,

$$\mathcal{U}_{\mathrm{cvpm},0} = \mathcal{U}_{\boldsymbol{x},0}. \tag{6.33}$$

Case 2 (impossible constraint satisfaction guarantee). There is no input $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ that can guarantee $p_1 = 0$, i.e.,

$$\varrho_{\max,1} < \varrho\left(h_1 + w_{\max,0}\right). \tag{6.34}$$

The largest value for $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$ that can be achieved with $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ is $\varrho_{\max,1}$, corresponding to the lowest possible p_1. However, to guarantee constraint satisfaction of (6.8), $\varrho_{\max,1}$ is required to be larger or at least equal to $\varrho(h_1 + w_{\max,0})$, with the worst-case absolute value $w_{\max,0}$ for the realization of $\boldsymbol{w}_0$. Constraint satisfaction cannot be guaranteed here.

The solution corresponding to $\varrho_{\max,1}$ is denoted by $\boldsymbol{u}_{\mathrm{cvpm},0}$. Minimal p_1 is achieved with

$$\boldsymbol{u}_{\mathrm{cvpm},0} = \underset{\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}}{\arg\max}\ \varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) \tag{6.35}$$

as $\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)$ increases and p_1 decreases with an increasing norm.

Therefore,

$$\mathcal{U}_{\mathrm{cvpm},0} = \left\{\boldsymbol{u}_{\mathrm{cvpm},0}\right\} \tag{6.36}$$

is selected since the input choice $\boldsymbol{u}_{\mathrm{cvpm},0}$ guarantees the lowest constraint violation probability when $p_1 > 0$.

Remark 6.5. *If* (6.35) *yields more than one solution,* $\mathcal{U}_{\mathrm{cvpm},0}$ *in* (6.36) *may also consist of more than one element, i.e., all solutions of* (6.35)*. However, there can be restrictions if convexity of* $\mathcal{U}_{\mathrm{cvpm},0}$ *is required.*

Case 3 (possible constraint satisfaction guarantee). The final case yields $p_1 = 0$ for some $\boldsymbol{u}_0$ and applies if

$$\varrho_{\max,1} \geq \varrho\left(h_1 + w_{\max,0}\right) > \varrho_{\min,1}. \tag{6.37}$$

While some $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ cannot guarantee zero constraint violation probability, it is possible to find $\boldsymbol{u}_0$ such that

$$\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) \geq \varrho\left(h_1 + w_{\max,0}\right). \tag{6.38}$$

Therefore, for some $\boldsymbol{u}_0$ constraint satisfaction can be guaranteed in the presence of uncertainty. Hence, the task is to find a set

$$\mathcal{U}_{\mathrm{cvpm},0} = \left\{\boldsymbol{u}_0 \,\middle|\, \left(\varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) \geq \varrho\left(h_1 + w_{\max,0}\right)\right) \wedge \left(\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}\right)\right\}, \tag{6.39}$$

which consists of all inputs $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$ that yield constraint satisfaction and therefore $p_1 = 0$.

The first part of the set in (6.39),

$$\mathcal{U}_{\mathrm{mode3},0} = \left\{\boldsymbol{u}_0 \,\middle|\, \varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) \geq \varrho\left(h_1 + w_{\max,0}\right)\right\}, \tag{6.40}$$

describes a super-level set, including only inputs $\boldsymbol{u}_0$ that lead to $p_1 = 0$. This super-level set is generally non-convex. In order to receive a convex set $\mathcal{U}_{\mathrm{cvpm},0}$ for the optimal control problem, an approximation is performed, based on the boundary

$$\partial\mathcal{U}_{\mathrm{mode3},0} = \left\{\boldsymbol{u}_0 \,\middle|\, \varrho\left(\left\|\boldsymbol{y}_1 - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) = \varrho\left(h_1 + w_{\max,0}\right)\right\}. \tag{6.41}$$

Proposition 6.1. *An approximated, convex solution of* (6.39) *in case 3 is obtained by*

$$\mathcal{U}_{\mathrm{cvpm},0} = \hat{\mathcal{U}}_{\mathrm{cvpm},0} = \left\{\boldsymbol{u}_0 \,\middle|\, \hat{\mathcal{U}}_0\left(\boldsymbol{u}_0^*\right) \cap \mathcal{U}_{\boldsymbol{x},0}\right\} \tag{6.42}$$

with

$$\hat{\mathcal{U}}_0\left(\boldsymbol{u}_0^*\right) = \left\{\boldsymbol{u}_0 \,\middle|\, \left(\nabla_{\boldsymbol{u}_0^*}\varrho\left(\left\|\boldsymbol{y}_1\left(\boldsymbol{u}_0^*\right) - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)\right)^\top \left(\boldsymbol{u}_0 - \boldsymbol{u}_0^*\right) \geq 0\right\}, \tag{6.43}$$

the gradient operator $\nabla_{\boldsymbol{u}_0^*}$, and a point $\boldsymbol{u}_0^* \in \partial\mathcal{U}_{\mathrm{mode3},0} \cap \mathcal{U}_{\boldsymbol{x},0}$ that is an admissible input.

Remark 6.6. *While it was previously not explicitly stated that $\boldsymbol{y}_1$ depends on $\boldsymbol{u}_0$, in Proposition 6.1 the dependence of $\boldsymbol{y}_1$ on $\boldsymbol{u}_0^*$ is stated for clarity.*

Proof. The set $\mathcal{U}_{\mathrm{mode3},0}$ is non-empty and non-convex with the boundary point $\boldsymbol{u}_0^* \in \partial\mathcal{U}_{\mathrm{mode3},0}$ of $\mathcal{U}_{\mathrm{mode3},0}$. There exists a supporting hyperplane to $\mathcal{U}_{\mathrm{mode3},0}$ at $\boldsymbol{u}_0^*$ [21]. This supporting hyperplane is then used to approximate the non-convex set $\mathcal{U}_{\mathrm{mode3},0}$. The gradient $\nabla_{\boldsymbol{u}_0^*}\left(\varrho\left(\left\|\boldsymbol{y}_1\left(\boldsymbol{u}_0^*\right) - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)\right)$ is a vector that is orthogonal to the hyperplane on the boundary $\partial\mathcal{U}_{\mathrm{mode3},0}$ at $\boldsymbol{u}_0^*$, pointing away from the convex set $\mathcal{U}_{\mathrm{mode3},0}$. The scalar product of $\nabla_{\boldsymbol{u}_0^*}\left(\varrho\left(\left\|\boldsymbol{y}_1\left(\boldsymbol{u}_0^*\right) - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)\right)$ and any point $\boldsymbol{u}_0$ on this hyperplane is zero, whereas the scalar product of $\nabla_{\boldsymbol{u}_0^*}\left(\varrho\left(\left\|\boldsymbol{y}_1\left(\boldsymbol{u}_0^*\right) - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right)\right)$ and any point in the half plane not containing $\mathcal{U}_{\mathrm{mode3},0}$ is positive. Therefore, (6.43) approximates $\mathcal{U}_{\mathrm{mode3},0}$. As the intersection of two convex sets yields a convex set [21], the resulting approximated set $\hat{\mathcal{U}}_{\mathrm{cvpm},0}$ is convex as well. □

An approach to finding $\boldsymbol{u}_0^*$ is solving the system

$$\varrho\left(\left\|\boldsymbol{y}_1\left(\boldsymbol{u}_0^*\right) - \overline{\boldsymbol{y}}_1^{\mathrm{DO}}\right\|_2\right) = \varrho\left(h_1 + w_{\max,0}\right) \tag{6.44}$$

with $\boldsymbol{u}_0^* \in \mathcal{U}_{\boldsymbol{x},0}$. The choice of $\boldsymbol{u}_0^*$ is not unique. It is possible that $\hat{\mathcal{U}}_{\mathrm{cvpm},0}$ is empty due to approximating even though case 3 applies.

Remark 6.7. *If $\hat{\mathcal{U}}_{\mathrm{cvpm},0} = \emptyset$ in case 3, then $\boldsymbol{u}_0$ can be determined by following the procedure of case 2.*

Following the approach in Remark 6.7 still provides a solution that minimizes p_1. However, in case 2 only a single option $\mathcal{U}_{\mathrm{cvpm},0} = \boldsymbol{u}_{\mathrm{cvpm},0}$ is given, whereas case 3 has the advantage of providing a set $\mathcal{U}_{\mathrm{cvpm},0}$ with multiple possible inputs $\boldsymbol{u}_0$. Case 3 therefore offers the possibility to then optimize to account for further objectives, given the set of admissible inputs $\mathcal{U}_{\mathrm{cvpm},0}$.

6.2.3 Properties

In the following, two important properties are analyzed. First, recursive feasibility of the proposed method is shown, which follows directly from the design of the CVPM-MPC method. This is followed by a proof of convergence, requires certain assumptions. The main challenge for these proofs is that the set of admissible inputs for the MPC optimal control problem changes at each time step.

Recursive Feasibility

Recursive feasibility guarantees that if the MPC optimal control problem is solvable at step t, it is also solvable at step $t+1$. This needs to hold as MPC requires the solution of an optimal control problem at every time step. We use Definition 2.5 for recursive feasibility. In the following, recursive feasibility will be established for the proposed method.

Theorem 6.2. *The CVPM-MPC algorithm in* (6.22) *is recursively feasible with the general CVPM approach of Section 6.2.2.*

The proof is divided into two parts. First it is shown that $\mathcal{U}_{\text{cvpm},0} \neq \emptyset$ at any step. Then, recursive feasibility of the optimal control problem (6.22) is shown.

Proof. As shown in the proof of Theorem 6.1, the three cases (6.16), (6.18), and (6.20) cover all possibilities with individual, nonempty sets $\mathcal{U}_{\text{cvpm},0}$. This yields that there always exists a $\boldsymbol{u}_0 \in \mathcal{U}_{\text{cvpm},0}$.

As $\boldsymbol{u}_0 \in \mathcal{U}_{\boldsymbol{x},0}$, it holds that conditions (6.5c), (6.5d), (6.5e) can be fulfilled with $\boldsymbol{u}_k \in \mathcal{U}_{\boldsymbol{x},k}$ for $k \in \mathbb{I}_{1:N-1}$ according to (6.6) and due to Assumption 6.2. No input $\boldsymbol{u}_0$ is possible that would cause $\mathcal{U}_{\boldsymbol{x},k} = \emptyset$ for $k \in \mathbb{I}_{1:N-1}$. Therefore, feasible solutions $\boldsymbol{u}_k$ exist and $\mathcal{U}_{\boldsymbol{x},k}$ is a non-empty set for $k \in \mathbb{I}_{1:N-1}$.

The first condition in (6.23) considers the first input $\boldsymbol{u}_0$, while the second condition covers the following inputs $\boldsymbol{u}_k$ with $k \in \mathbb{I}_{1:N-1}$. Therefore, the two conditions are independent and $\mathcal{U}_0^* \neq \emptyset$ for any MPC optimal control problem. The MPC algorithm (6.22) is guaranteed recursively feasible. □

The proof for the general CVPM-MPC method can be extended for the CVPM-MPC approach for uncertainties with symmetric, unimodal PDFs.

Corollary 6.2. *If Assumption 6.6 holds, the CVPM-MPC algorithm in* (6.22) *is recursively feasible with the CVPM approach for uncertainties with symmetric, unimodal PDFs.*

Proof. The proof follows straightforward from Theorem 6.2, showing that $\mathcal{U}_{\text{cvpm},t} \neq \emptyset$ for all three cases (6.32), (6.34), and (6.37). According to Lemma 6.2, $\varrho_{\text{min},1}$ and $\varrho_{\text{max},1}$ can always be found. Given any value for $\varrho\,(h_1 + w_{\text{max},0})$, exactly one of the three cases is applicable, yielding $\mathcal{U}_{\text{cvpm},0} \neq \emptyset$. For cases 1 and 2 no approximation is necessary. If $\hat{\mathcal{U}}_{\text{cvpm},0} = \emptyset$ for case 3, the approach of case 2 is used according to Remark 6.7, i.e., $\mathcal{U}_{\text{cvpm},0} = \left\{\boldsymbol{u}_{\text{cvpm},0}\right\}$. Therefore, $\mathcal{U}_{\text{cvpm},t} \neq \emptyset$ for all three cases. □

Theorem 6.2 and Corollary 6.2 show that if the MPC problem (6.5) is designed to be recursively feasible, the CVPM-MPC algorithm (6.22), based on (6.5), remains recursively feasible. According to Corollary 6.1, minimizing p_1 is independent of the uncertainty support, therefore, recursive feasibility is guaranteed if the uncertainty support changes.

Convergence

In the following, convergence of the proposed method is shown. Here, the MPC optimal control problem starts at $\boldsymbol{x}_t$. It is possible to track a reference varying from the origin, however, without loss of generality we will only consider the regulation of the origin here.

The uncertain output $\boldsymbol{y}_t^{\mathrm{DO}}$ can potentially lie close to the origin or even directly in the origin. In order to minimize p_t, an area around $\boldsymbol{y}_t^{\mathrm{DO}}$ is then inadmissible for the system output $\boldsymbol{y}_t$. This can lead to the scenario where the origin is inadmissible for the controlled system, i.e., $\mathbf{0} \in \mathcal{X}_{\mathrm{cv},t}$, where

$$\mathcal{X}_{\mathrm{cv},t} = \{\boldsymbol{x}_t \mid p_t(\boldsymbol{u}_{t-1}) > 0,\ \boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{B}\boldsymbol{u}_{t-1}\} \tag{6.45}$$

denotes the bounded and open set of states $\boldsymbol{x}_t$ with $p_t > 0$, i.e., constraint violation is possible for all $\boldsymbol{x}_t \in \mathcal{X}_{\mathrm{cv},t}$. An inadmissible origin is an issue when investigating the stability of the proposed algorithm. However, we will provide a convergence guarantee under the following two Assumptions concerning the stochastic nature of $\boldsymbol{y}_t^{\mathrm{DO}}$.

Assumption 6.7. *(a) There exists a $t_0 < \infty$ such that for all $t \geq t_0$ it holds that*

$$\mathbf{0} \notin \mathcal{X}_{\mathrm{cv},t} \quad \forall\, t \geq t_0. \tag{6.46}$$

(b) There exists a $t_{y0} < \infty$ and a finite sequence of inputs $\boldsymbol{u}_t$ such that $\boldsymbol{y}_t = \mathbf{0}$ for all $t \geq t_{y0} \geq t_0$.
(c) There exists a $t_{\mathrm{case1,3}} < \infty$ and for all $t \geq t_{\mathrm{case1,3}} \geq t_0$

$$\exists\ \boldsymbol{u}_{t-1} \ \text{ s.t. } \ p_t(\boldsymbol{u}_{t-1}) = 0 \tag{6.47}$$

and $\mathcal{U}_{\mathrm{cvpm},t} \neq \emptyset$.

Assumption 6.7 (a) is important such that even if $\boldsymbol{y}_t^{\mathrm{DO}}$ is occupying the space around the origin for some time, eventually $\boldsymbol{y}_t^{\mathrm{DO}}$ has enough distance to the origin and, therefore, the origin becomes admissible for the controlled system, as the boundedness of the stochastic system state yields a closed admissible space for the controlled system. Assumption 6.7 (b) ensures that there is a possibility for the controlled system to reach the origin.

With Assumption 6.7 (c) it is guaranteed that either case 1 or case 3 is applicable if Assumption 6.7 (a) holds. This ensures that $p_t = 0$ at some time after the origin becomes admissible for the controlled system.

Lemma 6.3. *If Assumption 6.7 holds, there exists a closed, control invariant set $\tilde{\mathcal{X}}_t = \mathcal{X} \setminus \mathcal{X}_{\mathrm{cv},t}$ for $t \geq t_{\mathrm{case1,3}}$, which contains the origin.*

Proof. As cases 1 or 3 are applied, the space blocked by $\mathcal{X}_{\mathrm{cv},t}$ around $\boldsymbol{y}_t^{\mathrm{DO}}$ with non-zero constraint violation probability can be regarded as a hard constraint. This yields $\boldsymbol{x}_t \notin \mathcal{X}_{\mathrm{cv},t}$ for all $t \geq t_{\mathrm{case1,3}}$. As $\mathcal{X}$ is closed and $\mathcal{X}_{\mathrm{cv},t}$ is open, the resulting set $\tilde{\mathcal{X}}_t$ is closed. As $\boldsymbol{x}_t \in \tilde{\mathcal{X}}_t \subseteq \mathcal{X}$, there exists a $\boldsymbol{u}_t$ such that $\boldsymbol{x}_{t+1} \in \mathcal{X}$ according to Theorem 6.2. Additionally, Assumption 6.7 (c) ensures that $\mathcal{U}_{\mathrm{cvpm},t}$ is not empty, therefore $\boldsymbol{x}_{t+1} \in \tilde{\mathcal{X}}_t$ and $\tilde{\mathcal{X}}_t$ is control invariant. □

The set $\tilde{\mathcal{X}}$ consists of the states that ensure constraint satisfaction of $\mathcal{X}$ and yield $p_t = 0$ for $t \geq t_{\mathrm{case1,3}}$.

Assumption 6.8. *The terminal constraint set $\mathcal{X}_\mathrm{f}$ is a subset of $\tilde{\mathcal{X}}_t$, i.e., $\mathcal{X}_\mathrm{f} \subset \tilde{\mathcal{X}}_t$.*

In the following, convergence of the proposed method is addressed.

Theorem 6.3. *If Assumptions 6.3 and 6.7 hold, the proposed CVPM-MPC method in Section 6.2.2 satisfies that $\boldsymbol{x}_t$ converges to $\mathbf{0}$ for $t \rightarrow \infty$.*

Proof. First, the MPC algorithm in (6.5) will be considered without the norm constraint (6.8). As $V\left(\boldsymbol{x}_0, \boldsymbol{U}\right)$ is a Lyapunov function in $\mathcal{X}$, given Assumption 6.3, the MPC algorithm of (6.22) without (6.8) is asymptotically stable, following the MPC stability proof of [160, Chapter 2.4].

Now the CVPM-MPC method is considered. According to Theorem 6.2, for all t, $\boldsymbol{x}_t \in \mathcal{X}$ there exists a feasible $\boldsymbol{U}_t$ such that $\boldsymbol{x}_{t+1}$ remains in $\mathcal{X}$. Lemma 6.3 ensures that $\boldsymbol{x}_{t'} \in \tilde{\mathcal{X}}_t$ for $t' \geq t_{\mathrm{case1,3}}$, where $\tilde{\mathcal{X}}_t$ replaces $\mathcal{X}$ to ensure constraint satisfaction of the norm constraint. The set $\tilde{\mathcal{X}}_t$ is closed, control invariant, contains the origin according to Assumption 6.7, and $\mathcal{X}_\mathrm{f} \subseteq \tilde{\mathcal{X}}_t$, given Assumption 6.8. Therefore, the system (6.1), controlled by the CVPM-MPC algorithm in (6.22), is asymptotically stable and converges to $\mathbf{0}$ for $t > t'$ and $t \rightarrow \infty$, similar to the MPC algorithm in (6.5). □

In Theorem 6.3 it is only shown that the system converges to the origin once the random system fulfills Assumption 6.7. However, every time the stochastic output allows the system to reach the origin, the system will move towards the origin. Once the origin is reached, the system state $\boldsymbol{x}_t$ remains at $\mathbf{0}$ until $\boldsymbol{y}_t^{\mathrm{DO}}$ changes in such a way that the origin has non-zero constraint violation probability. As the main goal is to ensure minimum constraint violation probability of (6.9), $\boldsymbol{y}_t$ will move away from the origin to minimize p_t if $\boldsymbol{y}_t^{\mathrm{DO}}$ behaves in such a way that it causes $p_t > 0$ in the origin.

Corollary 6.3. *If Assumptions 6.7 holds, the proposed CVPM-MPC method for uncertainties with symmetric, unimodal PDFs satisfies that $\boldsymbol{x}_t \in \mathcal{X}$ for all t and that $\boldsymbol{x}_t$ converges to $\mathbf{0}$ for $t \rightarrow \infty$.*

Proof. The proof is similar to the proof of Theorem 6.3. The set $\mathcal{X}_{\mathrm{cv},t}$ in (6.45) can be expressed as

$$\mathcal{X}_{\mathrm{cv},t} = \left\{ \boldsymbol{x}_t \;\middle|\; \varrho\left(\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\mathrm{DO}}\right\|_2\right) < \varrho\left(h_t + w_{\max,t-1}\right), \; \boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t \right\}. \tag{6.48}$$

Equation (6.46) is satisfied by

$$\varrho\left(\left\|\mathbf{0} - \overline{\boldsymbol{y}}_t^{\mathrm{DO}}\right\|\right) \geq \varrho\left(h_t + w_{\max,t-1}\right) \quad \forall\, t \geq t_0 \tag{6.49}$$

while (6.47) transforms into

$$\exists\; \boldsymbol{u}_{t-1} \;\; \text{s.t.} \;\; \varrho\left(\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\mathrm{DO}}\right\|_2\right) \geq \varrho\left(h_t + w_{\max,t-1}\right) \tag{6.50}$$

for the CVPM-MPC method for uncertainties with symmetric, unimodal PDFs.

Similar to Lemma 6.3, given the open and constant set $\mathcal{X}_{\mathrm{cv},t}$, $\tilde{\mathcal{X}}_t$ is closed, constant, control invariant, and contains the origin given Assumption 6.7. With the MPC algorithm (6.5) and $t > t'$, $t \rightarrow \infty$ the system (6.1) is asymptotically stable and therefore converges to $\mathbf{0}$. □

Therefore, if the origin is admissible, the controlled system will converge. However, satisfying the norm constraint has priority over converging to the origin. Compared to standard MPC methods, the origin is not necessarily within the constraint set $\mathcal{X}$. Therefore, the standard MPC stability approach cannot be applied, but convergence under mild assumptions is proved in Theorem 6.3.

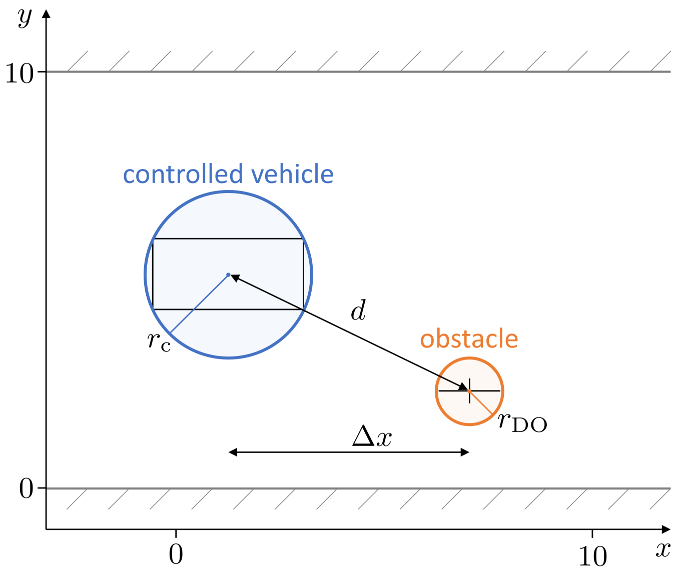


Figure 6.2: Vehicle avoidance scenario. Approximated shapes of the controlled vehicle (car) and obstacle (bicycle) are indicated by black lines within the objects.

6.2.4 Simulation Study

In the following, a simulation is presented and discussed to further explain the general idea and its application. This collision avoidance scenario with two vehicles illustrates an application where the proposed method is beneficial. The simulations were run in MATLAB on a standard desktop computer using MPT3 [83] and YALMIP [115]. Solving a single optimal control problem of the MPC algorithm takes 54 ms on average. All quantities are given in SI units.

Collision Avoidance Simulation

A collision occurs if the distance between two objects becomes too small. This distance can be represented by a norm constraint. The priority is then enforcing the norm constraint, or if not feasible, minimizing the probability of violating the norm constraint.

We consider the example mentioned in Sections 6.1 where a controlled vehicle avoids collision with a bicycle, referred to as obstacle in the following. The controlled vehicle is approximated by the radius $r_\mathrm{c} = 2.0$ and the obstacle is approximated by the radius $r_\mathrm{DO} = 0.8$ and is subject to stochastic motion in a bounded area, e.g., a road. The circles are chosen to fully cover the individual shapes of the controlled vehicle and obstacle. The scenario setup is shown in Figure 6.2. The discrete-time system dynamics of the controlled vehicle in x- and y-direction are given by

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} \exp(\Delta t) - 1 & 0 \\ 0 & \exp(\Delta t) - 1 \end{bmatrix} \boldsymbol{u}_t, \tag{6.51a}$$

$$\boldsymbol{y}_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_t, \tag{6.51b}$$

where $\boldsymbol{x} = (x, y)^\top$ and $(v_x, v_y)^\top$ are the position and velocity in a two-dimensional environment, respectively. The inputs are given by $(u_1, u_2)^\top$. System (6.51) is similar to (6.1). We will consider the input constraints

$$\mathcal{U} = \left\{ \boldsymbol{u} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \middle| \, 1 \leq v_x \leq 9, \ |v_y| \leq 3.5 \right\}. \tag{6.52}$$

In x-direction there exists a minimum velocity $v_{x,\mathrm{min}} = 1$ to ensure that the controlled vehicle is always moving forward, which also limits the potential oscillating behavior due to the CVPM-MPC approach. We also consider the state constraint

$$\mathcal{X} = \left\{ \boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix} \middle| \, y_{\mathrm{lb}} \leq y \leq y_{\mathrm{ub}} \right\} \tag{6.53}$$

where $y_{\mathrm{lb}} = 2.0$ and $y_{\mathrm{ub}} = 8.0$ are the boundaries of the road minus the radius r_{c}.

The assumed behavior of the obstacle with random behavior is given by

$$\boldsymbol{y}_t^{\mathrm{DO}} = \boldsymbol{y}_0^{\mathrm{DO}} + \sum_{i=0}^{t-1} \left(\boldsymbol{u}_i^{\mathrm{DO}} + \boldsymbol{w}_i \right) \tag{6.54}$$

depending on the initial output $\boldsymbol{y}_0^{\mathrm{DO}}$, the input $\boldsymbol{u}_t^{\mathrm{DO}}$, and the realization $\boldsymbol{w}_t$ of the random variable $\boldsymbol{W}_t \sim f_{\boldsymbol{W}_t}$ and $\boldsymbol{y}_{\mathrm{r}} = (x^{\mathrm{DO}}, y^{\mathrm{DO}})^\top$. We assume $f_{\boldsymbol{W}_t}$ to be symmetric, unimodal, and truncated, resulting in the support of $f_{\boldsymbol{W}_t}$

$$\operatorname{supp}\left(f_{\boldsymbol{W}_t}\right) = \left\{ \boldsymbol{w}_t \mid \|\boldsymbol{w}_t\|_2 \leq w_{\mathrm{max},t} \right\} \tag{6.55}$$

where $w_{\mathrm{max},t}$ is the radius of the support boundary of $\boldsymbol{W}_t$. The physical interpretation of $w_{\mathrm{max},t}$ is that it is the maximum uncertain distance the obstacle can move in one step, additionally to the deterministic distance $\boldsymbol{u}_t^{\mathrm{DO}}$. At step t, the controlled vehicle knows the obstacle position $\boldsymbol{y}_t^{\mathrm{DO}}$ and deterministic input $\boldsymbol{u}_t^{\mathrm{DO}}$, but $\boldsymbol{w}_t$ is unknown. The deterministic input represents the forward motion of the dynamic obstacle; the random variable denotes the uncertainty within the forward motion. Without the deterministic input the dynamic obstacle would only move around its initial position $\boldsymbol{y}_0^{\mathrm{DO}}$ due to the zero-mean random variable. The combination of a deterministic input with the random variable with zero mean is similar to a random variable with non-zero mean, where the controlled vehicle knows the mean. In general, the exact PDF is not required to be known. The proposed CVPM-MPC approach is applicable as long as the actual PDF adheres to Assumption 6.6 and the uncertainty bounds are known.

As the main aim of this simulation is to minimize the collision probability, an expression for this probability is necessary in order to analyze the simulation results. The collision probability at step t between the two vehicles will be denoted by $p_{\mathrm{col},t}$ and it has finite support as $f_{\boldsymbol{W}_t}$ is truncated. In this example, a norm constraint is used to avoid a collision, i.e., the norm constraint violation probability is minimized. Therefore, the probability of a collision $p_{\mathrm{col},t}$ is defined analogous to p_t in Section 6.2.1. The derivation and expression for the collision probability $p_{\mathrm{col},t}$ is omitted here due to readability. Details can be found in Appendix D.3.

The collision probability $p_{\mathrm{col},t}$ depends on the Euclidean distance

$$d_t = \left\| \boldsymbol{y}_t - \overline{\boldsymbol{y}}_{\mathrm{r},k} \right\|_2 \tag{6.56}$$

between the controlled vehicle and obstacle. Similar to (6.8) a norm constraint can be formulated where $h_t = d_{\text{safe},t}$ can be interpreted as the minimal distance between the controlled vehicle and the obstacle such that a collision is avoided. The support of $p_{\text{col},t}$ results from adding the radius of the controlled vehicle and the obstacle to $\operatorname{supp}(f_{\boldsymbol{W}_t})$, i.e.,

$$\operatorname{supp}(p_{\text{col},t}) = \{\boldsymbol{y}_t \mid d_t \leq d_{\text{safe},t}\} \tag{6.57}$$

where $d_{\text{safe},t} = w_{\max,t-1} + r_{\text{DO}} + r_{\text{c}}$ is the safety distance required to avoid a collision between the controlled vehicle and the obstacle, taking into account the radius of both vehicles, r_{DO} and r_{c}, and the maximal obstacle step $w_{\max,t-1}$. Similar to Lemma 6.1 for p_t, $p_{\text{col},t}$ is decreasing for increasing d_t.

We choose a quadratic function for $\varrho(\cdot)$, which is strictly monotonically increasing with ξ. This yields

$$\varrho\left(\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\text{DO}}\right\|_2\right) = \left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\text{DO}}\right\|_2^2, \tag{6.58}$$

which can be considered a substitution of the probability function $p_{\text{col},t}$.

The controlled vehicle uses the CVPM-MPC algorithm (6.22) with $N = 10$ and

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}. \tag{6.59}$$

The x-position references for the controlled vehicle are obtained by $x_{\text{ref},t} = x_0 + v_{x,\text{ref}} t \Delta t$, where $v_{x,\text{ref}}$ is the reference velocity in x-direction.

In the following, two scenarios will be analyzed. In the first scenario, the controlled vehicle is located close to its state boundary, i.e., the road boundary, showing that the norm constraint can be minimized in the presence of state constraints. In the second scenario, the obstacle uncertainty support will suddenly increase. The orientation ψ of the controlled vehicle in Figure 6.3 and Figure 6.5 is approximated by

$$\psi = \arctan \frac{u_2}{u_1}. \tag{6.60}$$

Active state constraints. In the first simulation, it is shown that the proposed method is applicable if state constraints are active. The reference velocity and y-position for the controlled vehicle are set to $v_{x,\text{ref}} = 5.0$ and $y_{\text{ref}} = 8.0$, respectively, with initial position $\boldsymbol{y}_0 = (0, 8)^\top$. The obstacle motion consists of a deterministic part $\boldsymbol{u}_t^{\text{DO}} = (0.5, 0)^\top$ combined with random behavior subject to a Gaussian uncertainty with $w_{\max,t} = 0.15$, with mean y-position $y_{\text{r}} = 4.0$ and a mean x-velocity $v_{\text{r},x} = 5.0$. Therefore, the x-position reference of the controlled vehicle is the same as the mean x-position of the obstacle in every step. Here, a sine motion is applied to the y-position of the obstacle with constant x-velocity, which is one possible outcome given the Gaussian uncertainty. The sine motion ensures that the maximal uncertainty values appear in the simulation, while the constant obstacle x-velocity keeps the controlled vehicle and the obstacle close together.

The vehicle configurations at different time steps are shown in Figure 6.3 and the further results of the simulation are displayed in Figure 6.4. Initially, the controlled vehicle and obstacle have the same x-position. Starting at time $t = 3.9\,\text{s}$ the controlled vehicle needs to slow down to maintain a safe distance to the obstacle. As the maximal obstacle uncertainty

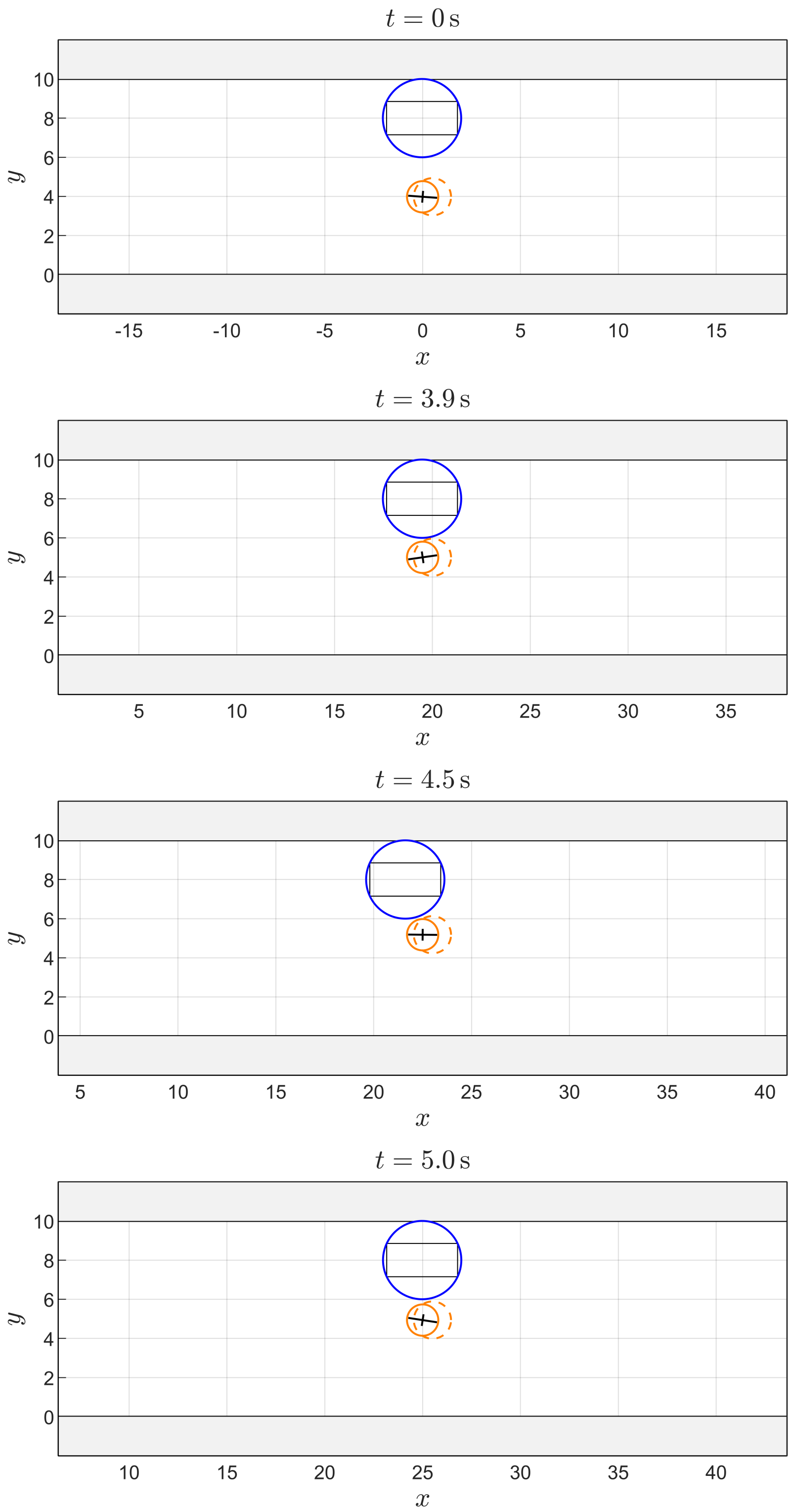


Figure 6.3: Vehicle configurations for the simulation with active state constraints. The controlled vehicle boundary is shown as a solid blue line and the obstacle boundary is a solid orange line. The dashed orange circle represents the possible obstacle location at the next time step.

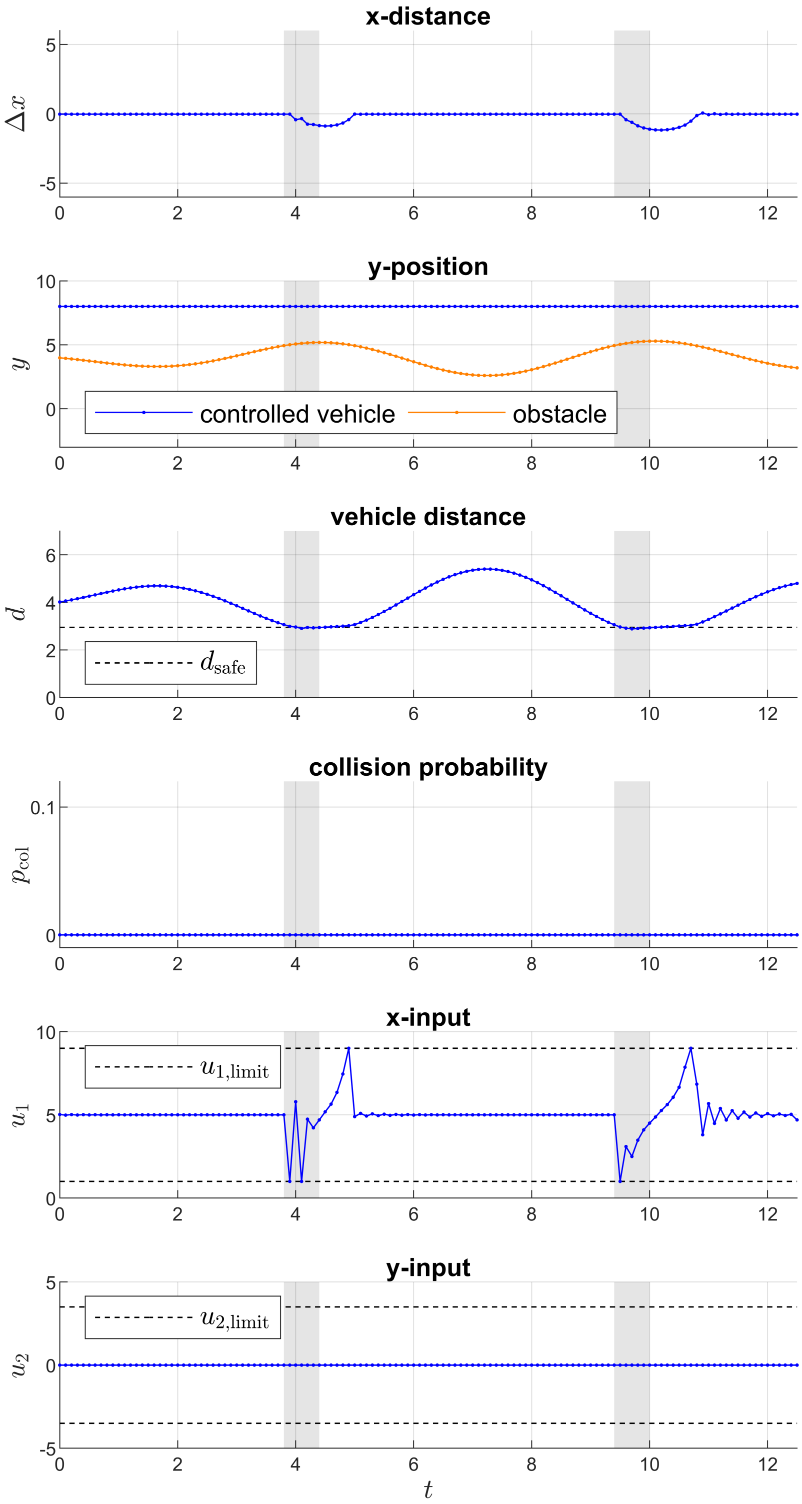


Figure 6.4: Simulation results for the simulation with active state constraint. The controlled vehicle is close to the state constraint. The gray area denotes actions by the controlled vehicle to avoid collision. The collision probability remains 0.

is known by the controlled vehicle, the collision probability is kept at zero. After time $t = 4.5\,\mathrm{s}$, the obstacle moves away from the controlled vehicle, resulting in increased input u_1 in order to get closer to the x-position reference. At time $t = 5.0\,\mathrm{s}$, the controlled vehicle catches up with its x-position reference, which is then followed by constant inputs. Between time $t = 9.0\,\mathrm{s}$ and $t = 11.0\,\mathrm{s}$ similar behavior can be observed. It can be seen that the CVPM-MPC ensures $p_t = 0$ with active state constraints. As mentioned in Section 6.2.3, the motion of the obstacle can result in an inadmissible origin, i.e., Assumption 6.7 (c) is violated and the controlled vehicle cannot keep its reference velocity. However, as shown in Theorem 6.3, once the obstacle moves away the velocity of the controlled vehicle again reaches the reference velocity.

This first simulation scenario is also used for a Monte Carlo simulation with 2000 simulation runs to evaluate the effectiveness of constraint violation probability minimization. Instead of applying the deterministic sine motion to the obstacle y-position and keeping the x-velocity constant, in addition to the deterministic part $\boldsymbol{u}_t^{\mathrm{DO}} = (0.5,\, 0)^\top$ a random step is applied to the obstacle. This random step is based on a truncated bivariate normal distribution with covariance matrix $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma^2, \sigma^2)$, $\sigma = 0.05$, mean $\mathbf{0}$, and $w_{\max,t} = 0.15$. The covariance matrix was chosen in such a way that truncating the normal distribution does not have a large effect, i.e., the non-truncated normal distribution has a probability of less than $1\,\%$ that $\|\boldsymbol{w}_t\|_2 > w_{\max,t}$. The results underline the effectiveness of the CVPM-MPC method. In $98.8\,\%$ of the simulations the constraint violation probability remained at $p_{\mathrm{col},t} = 0$. The maximal constraint violation observed was $p_{\mathrm{col},t} = 0.21\,\%$. No collisions occurred in any of the 2000 simulation runs.

Change of uncertainty property. In the second simulation, we show that the proposed method is capable of dealing with varying uncertainty support of the obstacle. The controlled vehicle aims to obtain the reference velocity $v_{x,\mathrm{ref}} = 4.0$ while maintaining $y_{\mathrm{ref}} = 4.0$ with the initial position $\boldsymbol{y}_0 = (0, 4)^\top$. The obstacle moves with a constant input $\boldsymbol{u}_t^{\mathrm{DO}} = (0.25,\, 0)^\top$ at $y^{\mathrm{DO}} = 4.0$. We consider here that the obstacle uncertainty support suddenly changes, for example due to a changing environment. At first the expected uncertainty support is $w_{\max,t} = 0.15$ and at time $t = 2.0\,\mathrm{s}$ it changes to $w_{\max,t} = 0.9$, while returning to $w_{\max,t} = 0.15$ at time $t = 4.0\,\mathrm{s}$. In the simulation, the obstacle does not move randomly, which helps to better understand the action of the controlled vehicle once the uncertainty support changes. At each time step, the controlled vehicle knows the current uncertainty support of the obstacle.

The vehicle configurations at different time steps are shown in Figure 6.5 and the results of the simulation are displayed in Figure 6.6. As the controlled vehicle has a higher velocity it will eventually pass the obstacle, therefore, the distance $\Delta x = x - x^{\mathrm{DO}}$ turns positive. At time $t = 0.8\,\mathrm{s}$, the controlled vehicle gets close enough to the obstacle that the controlled vehicle moves away from y_{ref} to maintain $v_{x,\mathrm{ref}}$ and ensures that the distance $d_t = \left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2 \geq d_{\mathrm{safe},t}$. As $w_{\max,t}$ increases at time $t = 2.0\,\mathrm{s}$, so does the required distance between the controlled vehicle and obstacle, causing the controlled vehicle to move further away from y_{ref}. Due to input limitations, the controlled vehicle cannot move fast enough. This results in $d_t < d_{\mathrm{safe},t}$, i.e., $p_{\mathrm{col},t} > 0$ at time $t = 2.0\,\mathrm{s}$, i.e., there is a probability of collision for the next time step. However, d_t is increased to a maximal level, given $\boldsymbol{u}_t \in \mathcal{U}_t$, resulting in a minimal constraint violation probability $p_{\mathrm{col},t}$. Once the distance satisfies $d_t \geq d_{\mathrm{safe},t}$ at time $t = 2.2\,\mathrm{s}$, $p_{\mathrm{col},t}$ becomes zero, and the controlled vehicle moves along the obstacle

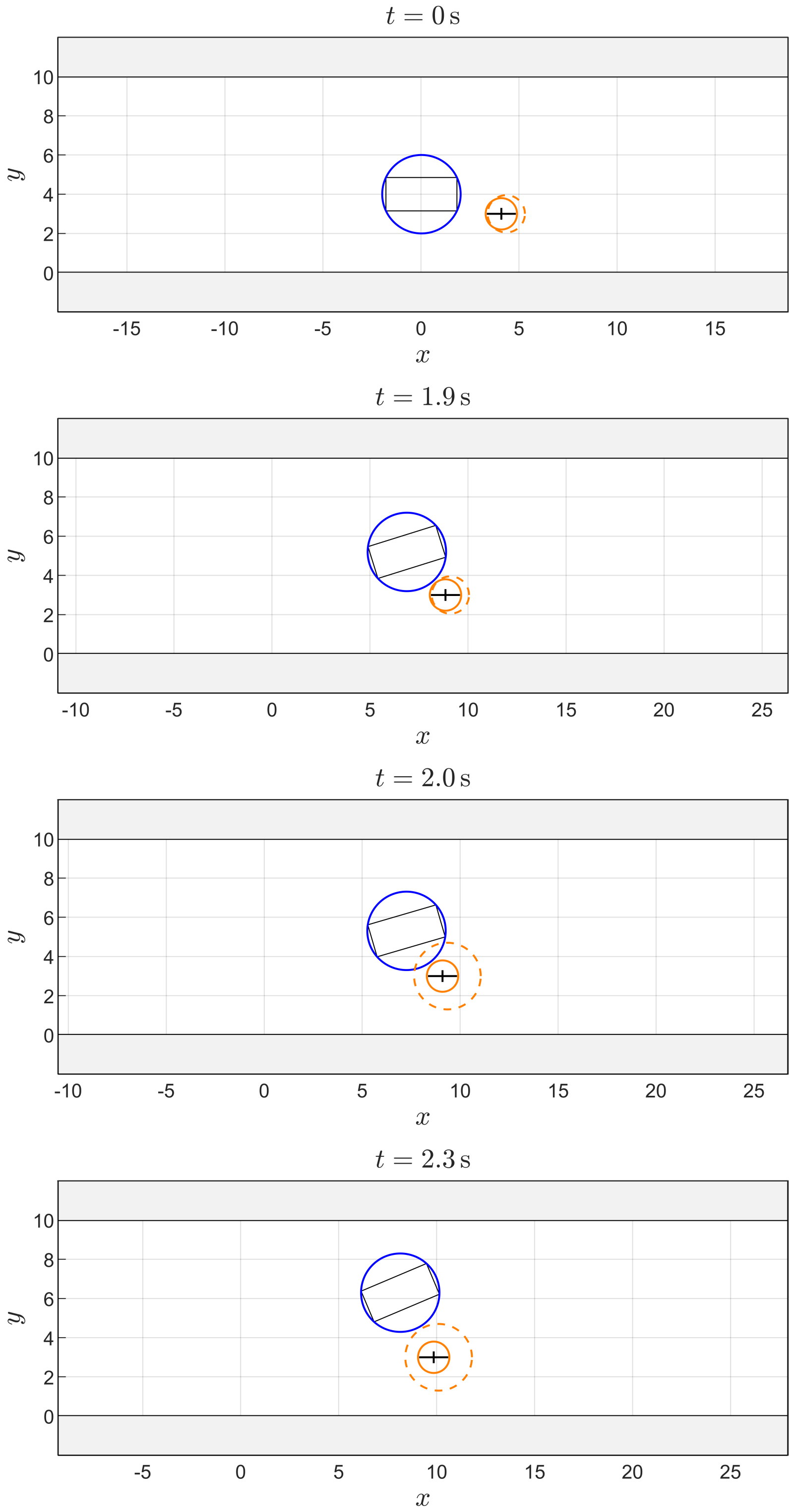


Figure 6.5: Vehicle configurations for the simulation with changing uncertainty support. The controlled vehicle and obstacle boundaries are shown as solid blue and orange lines, respectively. The dashed orange circle represents the possible obstacle location at the next time step.

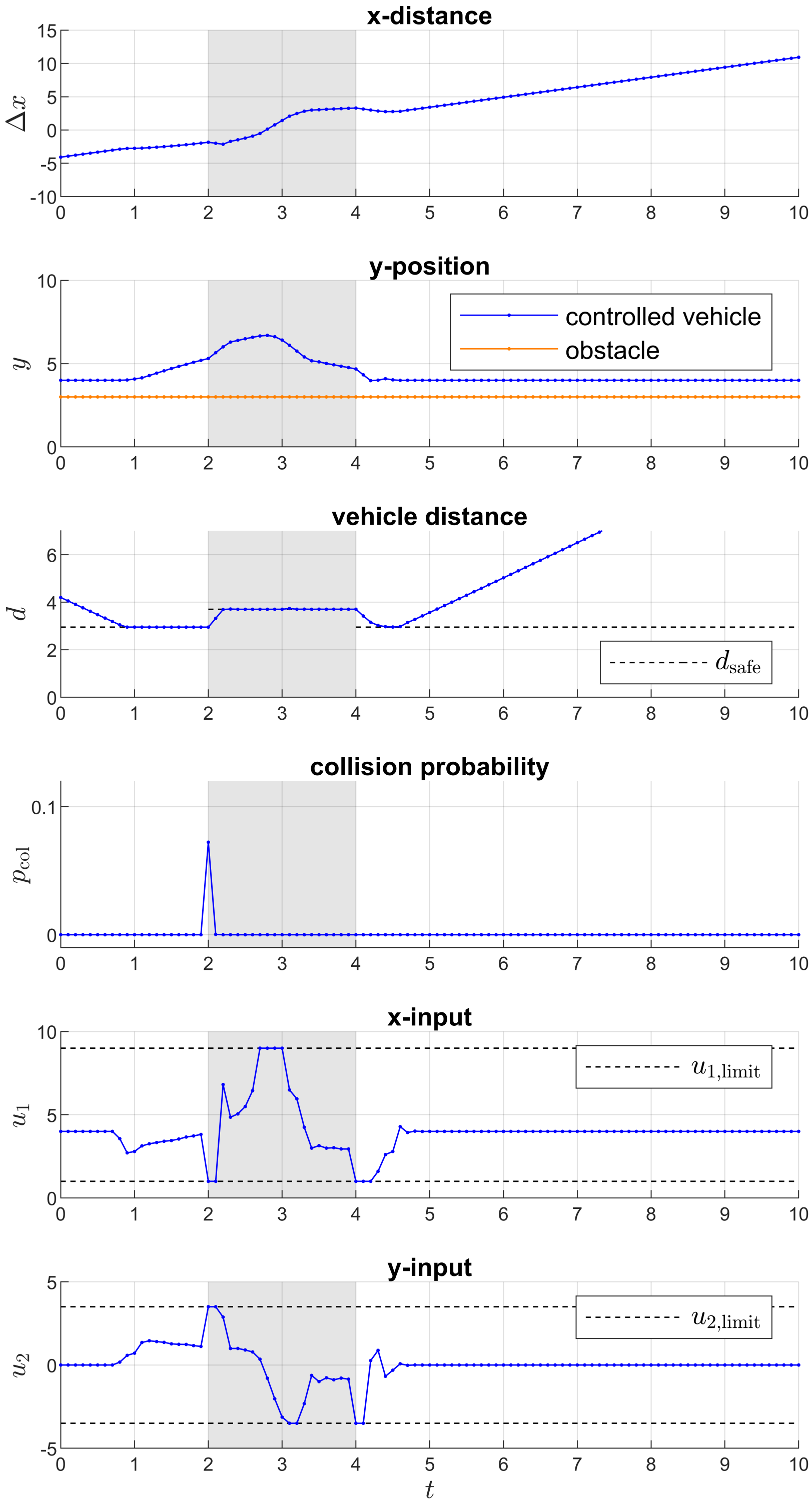


Figure 6.6: Simulation results for the simulation with changing uncertainty support. The gray area represents a higher uncertainty support. Once the uncertainty support changes, the collision probability temporarily increases to the minimal level possible.

boundary for the next step, as seen for time $t = 2.3\,\mathrm{s}$. At time $t = 4.0\,\mathrm{s}$, $w_{\max,t}$ decreases, and the controlled vehicle converges to y_{ref} at time $t = 4.8\,\mathrm{s}$.

In order to validate the probability of constraint violation, the simulation was run 2000 times with an arbitrary random obstacle step at time $t = 2.0\,\mathrm{s}$, which is the first step with the increased uncertainty bound $w_{\max,t} = 0.9$. The vehicles collided in 144 simulations, yielding a collision probability of 0.072 compared to the calculated collision probability 0.0723, as described in Appendix D.3.

Comparison to Robust MPC and Stochastic MPC

If RMPC and SMPC are applied in the previous simulations, certain problems arise, mainly due to infeasibility of the optimal control problem. This could be solved by providing rigorous alternative optimal control problems, predefined alternative inputs, or highly conservative worst-case considerations. However, there is no ideal RMPC or SMPC approach to deal with the scenario in the simulation. In the following, we will compare the simulation results of the proposed method to RMPC and SMPC qualitatively and quantitatively.

We will first consider the behavior with RMPC [160] applied to the controlled vehicle. In the first simulation, RMPC delivers safe results similar to the CVPM-MPC method while remaining behind the obstacle in order to account for the worst-case obstacle behavior. In the second simulation, two cases can be distinguished. If the initially considered uncertainty support is $w_{\max,t} = 0.15$, the behavior is similar to the proposed method until the uncertainty support changes. As it is impossible to find a state with zero collision probability after the uncertainty support is altered, the RMPC optimal control problem becomes infeasible. If the considered uncertainty support is initially chosen such that the larger support after time $t = 2.0\,\mathrm{s}$ is covered, RMPC yields a safe solution, however, it is passing the obstacle at a larger distance than initially required. In many applications it is also difficult to choose the worst-case uncertainty support a priori, as higher supports might occur later, resulting in even more conservative RMPC solutions.

It is now assumed that the controlled vehicle is controlled using SMPC [50] with a chance constraint with risk parameter $\beta_t > 0$ for collision avoidance. In the second simulation, before the uncertainty support changes, the controlled vehicle passes the obstacle a little closer than with the proposed CVPM-MPC method, as the chance constraint allows for small constraint violations. However, whereas the proposed CVPM-MPC method ensures safety while only passing the vehicle with little more distance, the SMPC approach would pass the obstacle ‘on the chance constraint’, i.e., as close as β_t allows, sacrificing guaranteed safety for small cost improvements. In other words, leaving slightly more space between the controlled vehicle and the obstacle would result in $p_t = 0$ with only little higher cost.

When the uncertainty support changes, the SMPC solution is as close to the obstacle as β_t allowed in the previous step. The chance constraint cannot be met anymore because the uncertainty support increased, resulting in a constraint violation probability larger than allowed by β_t. The SMPC optimal control problem then becomes infeasible, requiring an alternative optimal control problem to be defined beforehand. In the first simulation, a similar situation occurs. If the chance constraint allows the controlled vehicle to be in a position that yields $p_{\mathrm{col},t} > \beta_t$ due to the unconsidered worst-case obstacle motion, this leads to infeasibility of the optimal control problem.

We evaluate the performance of CVPM-MPC by comparing the overall cost to implementations of RMPC [160] and SMPC [50] for the simulation scenario with a changing uncertainty

method	SMPC risk parameter β_t	changing support	constant support $w_{\max,t} = 0.15$	constant support $w_{\max,t} = 0.90$
CVPM-MPC	-	$1.43 \cdot 10^2$	-	-
RMPC	-	$5.93 \cdot 10^2$	$1.63 \cdot 10^2$	$1.01 \cdot 10^4$
SMPC	0.8	$1.03 \cdot 10^2$	$0.87 \cdot 10^2$	$1.12 \cdot 10^2$
SMPC	0.9	$1.16 \cdot 10^2$	$0.89 \cdot 10^2$	$1.32 \cdot 10^2$
SMPC	0.95	$1.31 \cdot 10^2$	$0.91 \cdot 10^2$	$1.51 \cdot 10^2$
SMPC	0.99	$1.61 \cdot 10^2$	$0.95 \cdot 10^2$	$1.97 \cdot 10^2$

Table 6.1: Performance Comparison.

bound. Performance is compared based on the cost function of the optimal control problem, where the applied states and resulting states of the entire simulation are evaluated according to

$$J_{\text{sim}} = \sum_{t=0}^{N_{\text{sim}}-1} \left(\boldsymbol{x}_{t+1}^\top \boldsymbol{Q} \boldsymbol{x}_{t+1} + \boldsymbol{u}_t^\top \boldsymbol{R} \boldsymbol{u}_t \right) \tag{6.61}$$

with N_{sim} simulation steps. For SMPC four different risk parameters are analyzed. The results are shown in Table 6.1.

We first consider the changing uncertainty support as in the previously presented simulation. The CVPM-MPC method performs significantly better than RMPC, while SMPC has slightly lower cost, except for a conservative choice $\beta_t = 0.99$, where the SMPC cost is even larger.

However, both the RMPC and SMPC optimal control problems become infeasible for some steps when the uncertainty support changes at time $t = 2.0\,\text{s}$. Therefore, we analyze how RMPC and SMPC perform if only a small, constant support of $w_{\max,t} = 0.15$ or only a large, constant support of $w_{\max,t} = 0.9$ is assumed and applied. For SMPC a non-truncated Gaussian distribution is necessary to compute an analytic solution. The distribution is selected to have zero mean and covariance matrix $\boldsymbol{\Sigma} = \text{diag}\left(\sigma^2, \sigma^2\right)$ with $\sigma_{0.15} = 0.05$ and $\sigma_{0.9} = 0.3$. The covariance values are choses such that $\sigma_{0.15}$ and $\sigma_{0.9}$ approximate a distribution with support $w_{\max,t} = 0.15$ or $w_{\max,t} = 0.90$, respectively. For both support cases the CVPM-MPC performance is always better compared to the RMPC performance, with a significant advantage if $w_{\max,t} = 0.90$ is assumed for RMPC. If a small support is assumed, SMPC is less conservative. If SMPC considers a large support, SMPC is only less conservative for lower risk parameters. Nevertheless, assuming $w_{\max,t} = 0.90$ for RMPC and SMPC is only a partly satisfactory solution. In addition to increased cost, feasibility becomes an issue in the case that the uncertainty support increases again.

The comparison shows that the proposed method offers certain advantages over RMPC and SMPC, especially guaranteeing recursive feasibility of the optimal control problem in the presence of a changing uncertainty support.

6.2.5 Discussion

The proposed algorithm is not a combination of RMPC in the first step and, potentially, SMPC in the following steps. While there are some similarities to this combination, we solve a different problem. The most important difference is that the constraint violation probability is minimized in the first predicted step and the initial uncertainty probability is not required to be zero. RMPC approaches require constraint satisfaction initially and ensure that constraints are satisfied throughout the prediction horizon.

Our proposed CVPM-MPC method is more closely related to SMPC than RMPC, as constraint violations are possible. Nevertheless, the suggested method can be interpreted as lying between SMPC and RMPC. The results are more conservative than SMPC, as a zero percent constraint violation probability is found if possible, i.e., $p_t = 0$ in (6.9), but less conservative than RMPC. An advantage over both, SMPC and RMPC, is the ability to minimize the constraint violation probability and to successfully cope with sudden uncertainty support changes, as recursive feasibility can still be guaranteed. The uncertainty support can change due to unexpected events or modeling inaccuracies.

In SMPC with chance constraints, recursive feasibility is a major issue. For example, an unexpected realization of the uncertainty at step t, where the uncertainty realization likelihood lies below the chance constraint risk parameter at step t, leads to a state at step $t + 1$ with no solution to the optimal control problem if the required risk parameter of the chance constraint cannot be met. An option to regain feasibility is to solve an alternative optimal control problem or apply an input that was previously defined. However, these alternatives do not necessarily lead to a solution that yields the lowest constraint violation probability. Furthermore, it is possible to soften chance constraints by using slack variables in the cost function. However, this approach is not acceptable in applications where the chance constraint represents a safety constraint. If a slack variable is introduced, it competes with other objectives within the cost function and does not ensure constraint satisfaction. The proposed CVPM-MPC method always finds the optimal input that results in the lowest constraint violation probability while remaining recursively feasible.

RMPC guarantees recursive feasibility but at the cost of reduced efficiency, as worst-case scenarios need to be taken into account. Additionally, if the support of the uncertainty can suddenly change over time, e.g., the future motion of an object becomes more uncertain due to a changing environment, RMPC can become too conservative to be applicable. A robust solution can only be obtained by always considering the largest possible uncertainty support. The proposed method deals with this by adjusting to changing uncertainty supports at every step. A suddenly or unexpectedly increasing uncertainty support, e.g., due to an inaccurate prediction model, may lead to increased constraint violation probability for a limited time after the support changes. Before the support changes, the optimized inputs of the proposed algorithm lead to a less conservative result than RMPC while ensuring that the constraint violation probability is kept at a minimal level immediately after the change.

In the proposed method, we only consider minimizing the constraint violation for the first predicted step. It is possible to consider multiple steps by increasing the uncertainty support for each considered step; however, this leads to a more conservative solution. For every additionally predicted step in which the constraint violation probability is minimized, the maximal possible uncertainty value must be considered. This yields a highly restrictive set of admissible inputs that minimize the constraint violation probability over multiple predicted steps. As it is assumed that the support of the uncertainty PDF can change over time, considering multiple steps with the initially known support does not guarantee lower constraint violation probability for multiple steps. If the support increases the previously obtained multi-step CVPM-MPC solution becomes invalid. Therefore, given an updated uncertainty support at each step, it is a reasonable approach to only minimize the constraint violation probability for the first predicted step, resulting in the safest solution at the current step. It is possible to consider the norm constraint for collision avoidance in multiple predicted steps by either formulating a chance constraint, as mentioned in Remark 6.3, or a robust constraint. However, this can result in infeasibility of the optimal control problem, particularly if the uncertainty support varies over time. Despite only considering the norm constraint for the next predicted step, it is still beneficial to use an MPC horizon $N > 1$. Other objectives are optimized over the entire horizon, given that the first input is included in the set $\mathcal{U}_{\text{cvpm},0}$, which potentially consists of multiple admissible inputs that all minimize the constraint violation for the next step.

Applying the CVPM-MPC approach possibly results in oscillating behavior. As long as case 1 is valid, the proposed method does not affect the optimal control problem, as $\mathcal{U}_{\text{cvpm},0} = \mathcal{U}_{\boldsymbol{x},0}$. Once case 2 is active, a solution is found that minimizes the probability of constraint violation, ignoring the reference and potentially moving from the reference, as only one input is admissible. When case 1 is valid again, the reference is tracked again until, possibly, case 2 becomes active again. This can be improved by considering the norm constraint as a chance constraint for multiple predicted steps, however, recursive feasibility is not guaranteed, as mentioned before.

The main focus of the suggested method is to minimize the constraint violation probability. It is clear that stability cannot always be guaranteed, as the origin can be excluded from the admissible state set. We consider a narrow road where a bicycle is between the controlled vehicle and the vehicle reference point. If the road is too narrow for the vehicle to pass, it will remain behind the bicycle and never reach the reference point, i.e., Assumption 6.7 (b) is violated. However, Assumption 6.7 implies that the origin is not inadmissible at all times, and once the origin is admissible, the controlled system converges.

It is also important to note that minimizing the constraint violation probability has priority over other optimization objectives. Especially in safety-critical applications this can be of major interest, e.g., an autonomous car must ensure that the collision probability is always minimal, prior to reducing energy or increasing passenger comfort. If SMPC were to be applied in such scenarios, the question would arise of how to choose the SMPC risk parameter β_t. A large β_t yields efficient behavior but might be unacceptable due to an insufficient safety level. Finding a reduced value for β_t in SMPC is challenging, as even very small risk parameters allow for constraint violations, while $\beta_t = 0$ does not yield a chance constraint and the advantages of SMPC are lost. In the proposed CVPM-MPC, the task of appropriately choosing the risk parameter is not required. However, there is also no bound for the possible constraint violation probability.

Additionally, in safety-critical systems a further aspect reduces the usability of chance constraints in SMPC. A solution is valid as long as the probability of violating the safety constraint satisfies the risk parameter. Assuming there exists a solution with lower, or even zero, constraint violation probability, the optimal control problem solution will still be 'on the chance constraint' if this results in lower objective costs, i.e., allow for constraint violations according to the risk parameter. We consider again the example in the introduction of a car overtaking a bicycle. Using a chance constraint with $\beta_t > 0$, the car will pass the bicycle but will choose a trajectory around the bicycle that allows for a collision with a low probability due to $\beta_t > 0$. Given a finite bicycle uncertainty support, passing the bicycle with slightly more distance yields zero collision probability with only a small increase of cost. However, in practice, this slightly increased cost is acceptable if thereby safety is guaranteed.

For the approach in Section 6.2.2, the PDF $f_{\boldsymbol{W}_t}$ does not need to be known exactly as long as it fulfills Assumption 6.6. If $f_{\boldsymbol{W}_t}$ is symmetric and unimodal, it is ensured that increasing $\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\mathrm{DO}}\right\|_2$ results in a lower constraint violation probability p_t.

The proposed method is especially useful in collision avoidance applications, which are either in two- or three-dimensional space. While applying the proposed method in two-dimensional space is straightforward, three-dimensional applications can be more challenging to solve, especially finding $\boldsymbol{u}_0^*$ in (6.42). For collision avoidance scenarios, possible uncertainty in (6.1) can be considered by increased uncertainty in (6.2).

The structure of CVPM-MPC considers two main aims: minimizing constraint violation probability and optimizing additional objectives, such as energy consumption. Multi-objective MPC [10, 199] is an MPC scheme to trade-off multiple opposing objectives within one optimal control problem, where each individual objective is assigned a weighting factor. In our case, we do not aim at a trade-off, but one of the two objectives is regarded first. Only if optimizing the first objective, i.e., minimizing constraint violation probability, enables multiple possible inputs, further objectives are taken into account. In multi-objective MPC this translates to assigning a weight of zero to all additional objectives. While this could be an approach to minimize constraint violation probability, other objectives would not be optimized. Therefore, compared to multi-objective MPC, the proposed CVPM-MPC method provides a solution to safety-critical problems where constraint violation probability needs to be minimized and further objectives should be optimized if possible.

6.3 CVPM-MPC for Linear Systems with Linear Constraints

In the following, we generalize the CVPM-MPC method for linear systems with additive uncertainty and linear constraints. In contrast to Section 6.2, only two cases are necessary, which simplifies the method.

6.3.1 Problem Formulation

In the following, we first state the system dynamics and uncertainty considered. Then, the MPC setup is introduced, including the constraint for which the violation probability must be minimized.

System Dynamics

We consider the following linear, discrete-time system

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{E}\boldsymbol{w}_t \tag{6.62}$$

with state $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and input $\boldsymbol{u}_t \in \mathbb{R}^{n_{\boldsymbol{u}}}$ at time step t, as well as the bounded uncertainty $\boldsymbol{w}_t \in \mathcal{W}_t \subseteq \mathbb{R}^{n_{\boldsymbol{w}}}$, where $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{E}$ have appropriate dimensions and $\mathcal{W}_t$ may vary over time.

Assumption 6.9. *The uncertainty $\boldsymbol{w}_t$ is a truncated Gaussian uncertainty with $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}}$, bounded by $\mathcal{W}_t$.*

Remark 6.8. *The uncertainty $\boldsymbol{w}_t$ may have any symmetric unimodal distribution. However, here we focus on a Gaussian distribution for clarity.*

In the following, a system prediction is required for MPC. Based on the initial state $\boldsymbol{x}_0$, we denote the augmented system dynamics, delivering all predictions, by

$$\underline{\boldsymbol{x}}_k = \underline{\boldsymbol{A}}_k \boldsymbol{x}_0 + \underline{\boldsymbol{B}}_k \underline{\boldsymbol{u}}_{k-1} + \underline{\boldsymbol{E}}_k \underline{\boldsymbol{w}}_{k-1} \tag{6.63}$$

where

$$\underline{\boldsymbol{x}}_k = (\boldsymbol{x}_1, ..., \boldsymbol{x}_k)^\top, \quad \underline{\boldsymbol{u}}_{k-1} = (\boldsymbol{u}_0, ..., \boldsymbol{u}_{k-1})^\top, \quad \underline{\boldsymbol{w}}_{k-1} = (\boldsymbol{w}_0, ..., \boldsymbol{w}_{k-1})^\top \tag{6.64}$$

and

$$\underline{\boldsymbol{A}}_k = \left[\boldsymbol{A}^\top, \left(\boldsymbol{A}^2\right)^\top, \cdots, \left(\boldsymbol{A}^k\right)^\top\right]^\top, \tag{6.65a}$$

$$\underline{\boldsymbol{B}}_k = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{A}^{k-1}\boldsymbol{B} & \boldsymbol{A}^{k-2}\boldsymbol{B} & \dots & \boldsymbol{B} \end{bmatrix}, \tag{6.65b}$$

$$\underline{\boldsymbol{E}}_k = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{E} & \boldsymbol{E} & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{A}^{k-1}\boldsymbol{E} & \boldsymbol{A}^{k-2}\boldsymbol{E} & \dots & \boldsymbol{E} \end{bmatrix}. \tag{6.65c}$$

The augmented system dynamics (6.63) allows us to express all resulting prediction states in a concise way.

Model Predictive Control

MPC is applied to control system (6.62). The MPC cost function of this section is given by

$$J(\boldsymbol{x}_0, \underline{\boldsymbol{u}}_{N-1}) = \sum_{k=0}^{N-1} \left(\overline{\boldsymbol{x}}_k^\top \boldsymbol{Q} \overline{\boldsymbol{x}}_k + \boldsymbol{u}_k^\top \boldsymbol{R} \boldsymbol{u}_k\right) + \overline{\boldsymbol{x}}_N^\top \boldsymbol{Q}_\mathrm{t} \overline{\boldsymbol{x}}_N \tag{6.66}$$

with weighting matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ and the terminal weighting matrix $\boldsymbol{Q}_\mathrm{t}$, where $\boldsymbol{x}_0$ is known and $\overline{\boldsymbol{x}}_k$ denotes the mean of $\boldsymbol{x}_k$.

Remark 6.9. *In* (6.66) *a simple cost function is chosen based on the predicted mean states; however, other cost functions are possible that specifically consider the uncertainty of the state prediction [68].*

Furthermore, we define the set of admissible inputs

$$\underline{\mathcal{U}} = \left\{\underline{\boldsymbol{u}}_{N-1} \in \mathbb{R}^{n_{\boldsymbol{u}}N} \,\middle|\, \boldsymbol{u}_k \in \mathcal{U}\ \forall k \in \mathbb{I}_{0,N-1}\right\} \tag{6.67}$$

for $\underline{\boldsymbol{u}}_{N-1}$ with the bounded polytopic constraint set $\mathcal{U}$. In addition, we introduce the time-varying constraint

$$\boldsymbol{x}_k \in \mathcal{X}_t. \tag{6.68}$$

Assumption 6.10. *The constraint set $\mathcal{X}_t$ is closed, bounded, and contains the origin.*

The constraint (6.68) may be expressed in augmented form by

$$\underline{\boldsymbol{x}}_{N_\mathrm{c}} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{N_\mathrm{c}})^\top \in \mathcal{X}_t{}^{N_\mathrm{c}} = \mathcal{X}_t \times \mathcal{X}_t \times \cdots \times \mathcal{X}_t \tag{6.69}$$

where $N_\mathrm{c} \leq N$ defines a sub-horizon where the state constraint (6.68) is considered.

Problem Statement

As the predicted state $\boldsymbol{x}_k$ is subject to uncertainty, this uncertainty must be specifically addressed. RMPC is designed in such a way that the constraint is satisfied for any possible worst-case uncertainty realization. In SMPC, constraint (6.68) is treated as a chance constraint, which is only required to hold up to a predefined confidence level.

However, both RMPC and SMPC do not provide adequate solutions if the probability of violating constraint (6.69) must be minimized. This problem may be formulated as

$$\min_{\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}} \Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t{}^{N_\mathrm{c}}\right) \tag{6.70}$$

where the input sequence $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}$ ensures that the probability of constraint violation is minimal. Given this constraint violation probability minimization, we now formulate the problem to be addressed within this section.

Objective 6.2. *The MPC problem to be solved is*

$$\min_{\underline{\boldsymbol{u}}_{N-1}} J(\boldsymbol{x}_0, \underline{\boldsymbol{u}}_{N-1}) \tag{6.71a}$$

$$\text{s.t} \quad \underline{\boldsymbol{u}}_{N-1} \in \underline{\mathcal{U}} \tag{6.71b}$$

where it must hold that

$$\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} = \arg\min_{\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}} \Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t{}^{N_\mathrm{c}}\right). \tag{6.72}$$

Note that $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}$ is part of $\underline{\boldsymbol{u}}_{N-1}$ with

$$\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} = (\boldsymbol{u}_0, ..., \boldsymbol{u}_{N_\mathrm{c}-1})^\top, \tag{6.73a}$$

$$\underline{\boldsymbol{u}}_{N-1} = (\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1})^\top, \tag{6.73b}$$

i.e., for $N_\mathrm{c} \leq N$ we obtain

$$\underline{\boldsymbol{u}}_{N-1} = (\boldsymbol{u}_0, ..., \boldsymbol{u}_{N_\mathrm{c}-1}, ..., \boldsymbol{u}_{N-1})^\top. \tag{6.74}$$

In the following section, an MPC method is derived that provides a strategy to solve Objective 6.2.

6.3.2 Method

We next describe the CVPM-MPC method to solve Objective 6.2. Major properties of the CVPM-MPC method are derived in Section 6.3.3.

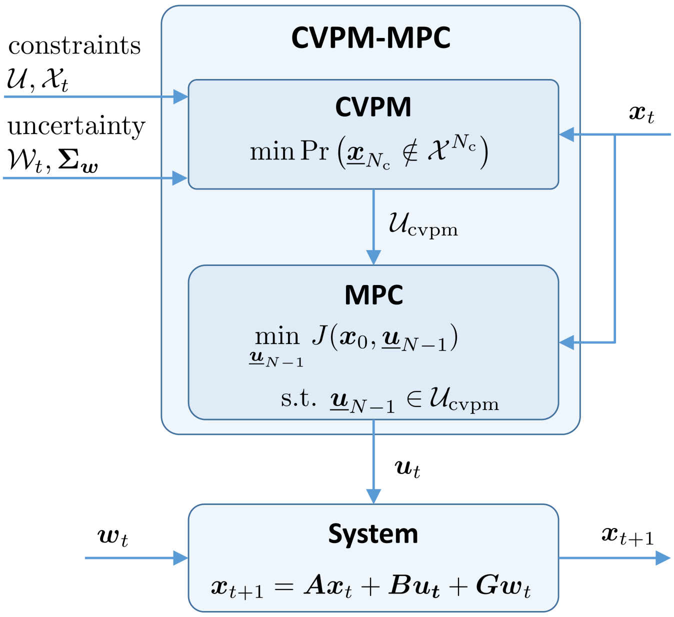

constraints
$\mathcal{U}, \mathcal{X}_t$
uncertainty
$\mathcal{W}_t, \boldsymbol{\Sigma}_{\boldsymbol{w}}$
CVPM-MPC
CVPM
$\min \Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}^{N_\mathrm{c}}\right)$
$\boldsymbol{x}_t$
$\mathcal{U}_{\mathrm{cvpm}}$
MPC
$\min\limits_{\underline{\boldsymbol{u}}_{N-1}} J(\boldsymbol{x}_0, \underline{\boldsymbol{u}}_{N-1})$
s.t. $\underline{\boldsymbol{u}}_{N-1} \in \mathcal{U}_{\mathrm{cvpm}}$
$\boldsymbol{u}_t$
$\boldsymbol{w}_t$
System
$\boldsymbol{x}_{t+1}$
$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{t}} + \boldsymbol{G}\boldsymbol{w}_t$

Figure 6.7: CVPM-MPC method for linear systems with linear constraints.

CVPM-MPC

The general idea of CVPM-MPC is to solve an MPC optimal control problem where only those inputs are allowed that enable minimal constraint violation probability. Therefore, in order to set up the MPC formulation, a set must be found that includes all inputs that minimize constraint violation probability.

Definition 6.2 (Optimal CVPM Input Set)**.** *The optimal CVPM input set* $\mathcal{U}_{\mathrm{cvpm}}$ *consists only of admissible input sequences* $\underline{\boldsymbol{u}}_{N-1}$ *that minimize the constraint violation probability* $\Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t^{N_\mathrm{c}}\right)$. *The optimal CVPM input set containing only the input sequences* $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}$ *is denoted by* $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$.

The MPC problem is then solved with $\mathcal{U}_{\mathrm{cvpm}}$ as the constraint set for control inputs. The structure of the CVPM-MPC method is illustrated in Figure 6.7.

Determining the set $\mathcal{U}_{\mathrm{cvpm}}$ is based on two cases, depending on whether an input sequence $\underline{\boldsymbol{u}}_{N-1}$ exists that guarantees constraint satisfaction.

Definition 6.3 (CVPM Safe Case)**.** *The safe case refers to the case where an admissible input sequence* $\underline{\boldsymbol{u}}_{N-1}$ *exists that guarantees constraint satisfaction, i.e.,*

$$\exists\, \underline{\boldsymbol{u}}_{N-1} \in \underline{\mathcal{U}} \text{ s.t. } \Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t^{N_\mathrm{c}}\right) = 0. \tag{6.75}$$

Definition 6.4 (CVPM Probabilistic Case)**.** *The probabilistic case refers to the case where no input sequence* $\underline{\boldsymbol{u}}_{N-1}$ *exists that guarantees constraint satisfaction, i.e.,*

$$\Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t^{N_\mathrm{c}}\right) > 0 \;\forall\, \underline{\boldsymbol{u}}_{N-1} \in \underline{\mathcal{U}}. \tag{6.76}$$

If the state constraint is not considered for the entire MPC horizon, i.e., $N_\mathrm{c} < N$, it must hold that the first N_c inputs $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \underline{\mathcal{U}}_{N_\mathrm{c}-1}$. Here, $\underline{\mathcal{U}}_{N_\mathrm{c}-1}$ is a projection of the set $\underline{\mathcal{U}}$ spanning over the first N_c inputs, i.e., $\mathrm{proj}_{N_\mathrm{c}-1}(\underline{\mathcal{U}})$.

The applicable case depends on the possibility of satisfying the constraint (6.69), i.e., in each MPC iteration, one of the two cases is applied. We first define a set containing all input sequences that result in constraint satisfaction.

Definition 6.5 (Zero Violation Set)**.** *The zero violation set* $\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}$*, with respect to the state constraint* $\mathcal{X}_t$*, contains those inputs* $\underline{\boldsymbol{u}}_{N-1}$ *where the resulting predicted state trajectory satisfies* (6.69) *and the inputs satisfy* (6.67)*.*

These definitions allow us to determine which of the two cases occurs. We consider the optimal CVPM input set

$$\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} = \underline{\mathcal{U}}_{N_\mathrm{c}-1} \cap \underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}, \tag{6.77}$$

which consists of the intersection of the admissible input set $\underline{\mathcal{U}}_{N_\mathrm{c}-1}$ and the set $\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}$, whose input sequences guarantee constraint satisfaction.

If the optimal input set is not empty, i.e., $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \neq \emptyset$, the safe case is active, as an admissible input sequence exists that allows for constraint satisfaction. However, if $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} = \emptyset$, any admissible input sequence leads to a constraint violation probability larger than zero, which requires the probabilistic case.

For both cases, it is necessary to obtain the set $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$, which is addressed in the following.

Safe case. For the safe case, the set $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ is determined according to (6.77). While the set of admissible inputs $\underline{\mathcal{U}}_{N_\mathrm{c}-1}$ is given according to (6.67), the set zero violation set $\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}$ must be determined.

Lemma 6.4. *For the safe case, the set* $\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}$ *is obtained by*

$$\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1} = \left({\mathcal{X}_t}^{N_\mathrm{c}} \ominus \left(\underline{\boldsymbol{E}}_{N_\mathrm{c}} \circ {\mathcal{W}_t}^{N_\mathrm{c}}\right) \oplus \{-\underline{\boldsymbol{A}}_{N_\mathrm{c}}\boldsymbol{x}_0\}\right) \circ \underline{\boldsymbol{B}}_{N_\mathrm{c}} \tag{6.78}$$

where $\{-\underline{\boldsymbol{A}}_{N_\mathrm{c}}\boldsymbol{x}_0\}$ *is a singleton.*

Proof. The proof is given in Appendix D.4. □

However, it is not guaranteed that the elements of $\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}$ are also part of $\underline{\mathcal{U}}_{N_\mathrm{c}-1}$, requiring the intersection (6.77). If the resulting set $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ is not empty, i.e., $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \neq \emptyset$, an input sequence $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ exists guaranteeing constraint satisfaction of (6.69) for N_c.

Based on the previous results, it is possible to obtain a set of states $\mathcal{X}_\mathrm{caseS}$ (for the current MPC iteration) for which case 1 is applicable. This set is given by

$$\mathcal{X}_\mathrm{caseS} = \left({\mathcal{X}_t}^{N_\mathrm{c}} \ominus \left(\underline{\boldsymbol{E}}_{N_\mathrm{c}} \circ {\mathcal{W}_t}^{N_\mathrm{c}}\right) \oplus \left((-\underline{\boldsymbol{B}}_{N_\mathrm{c}}) \circ \mathcal{U}^{N_\mathrm{c}}\right)\right) \circ \underline{\boldsymbol{A}}_{N_\mathrm{c}}, \tag{6.79}$$

which is obtained similarly to (6.78).

Lemma 6.5. *For time-invariant* $\mathcal{W}_t = \mathcal{W}$ *and* $\mathcal{X}_t = \mathcal{X}$*, the set* $\mathcal{X}_\mathrm{caseS}$ *is robustly positively invariant.*

Proof. If $\boldsymbol{x}_t \in \mathcal{X}_\mathrm{caseS}$, where $\mathcal{X}_\mathrm{caseS}$ is defined in (6.79), it follows that $\boldsymbol{x}_{t+1} \in \mathcal{X} \subseteq \mathcal{X}_\mathrm{caseS}$, i.e., $\mathcal{X}_\mathrm{caseS}$ is robustly positively invariant. □

Probabilistic case. If determining $\underline{\mathcal{U}}_{\mathcal{X},N_\mathrm{c}-1}$ results in an empty set $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ according to (6.77), $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ must be determined differently. In the safe case, $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ consists of those input sequences that guarantee constraint satisfaction. However, if such an input sequence does not exist, minimal constraint violation probability must be ensured. Therefore, in the probabilistic case, $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ comprises the input sequence $\underline{\boldsymbol{u}}^*_{N_\mathrm{c}-1}$ that result in minimal constraint violation probability.

This input sequence $\underline{\boldsymbol{u}}^*_{N_\mathrm{c}-1}$ is obtained by solving the optimal control problem

$$\underline{\boldsymbol{u}}^*_{N_\mathrm{c}-1} = \arg \min_{\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}} \Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin {\mathcal{X}_t}^{N_\mathrm{c}}\right) \tag{6.80a}$$

$$\text{s.t.} \quad \underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \underline{\mathcal{U}}_{N_\mathrm{c}-1}. \tag{6.80b}$$

Later in this section, we elaborate on finding a solution to the optimal control problem (6.80). As previously mentioned, once $\underline{\boldsymbol{u}}^*_{N_\mathrm{c}-1}$ is obtained, it follows that

$$\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} = \{\underline{\boldsymbol{u}}^*_{N_\mathrm{c}-1}\}. \tag{6.81}$$

Now that the set $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ has been determined for both cases, we may formulate the CVPM-MPC optimal control problem.

CVPM-MPC formulation. Given the set $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$, the optimal CVPM-MPC set $\mathcal{U}_{\mathrm{cvpm}}$ is constructed by

$$\mathcal{U}_{\mathrm{cvpm}} = \left\{\underline{\boldsymbol{u}}_{N-1} \mid \underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \ \wedge \ \underline{\boldsymbol{u}}_{N-1} \in \underline{\mathcal{U}}\right\}, \tag{6.82}$$

where the first N_c inputs ensure constraint violation probability minimization of (6.69).

With this updated input constraint set, the CVPM-MPC optimal control problem is given by

$$\underline{\boldsymbol{u}}^*_{N-1} = \arg \min_{\underline{\boldsymbol{u}}_{N-1}} J(\boldsymbol{x}_0, \underline{\boldsymbol{u}}_{N-1}) \tag{6.83a}$$

$$\text{s.t} \quad \underline{\boldsymbol{u}}_{N-1} \in \mathcal{U}_{\mathrm{cvpm}} \tag{6.83b}$$

with $\mathcal{U}_{\mathrm{cvpm}}$ according to (6.82). The CVPM-MPC optimal control problem (6.83) yields satisfaction of the input constraint $\underline{\mathcal{U}}$ as well as minimal constraint violation probability. The closed-loop system to (6.62) is then given by

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}^*_t + \boldsymbol{E}\boldsymbol{w}_t \tag{6.84}$$

where $\boldsymbol{u}^*_t$ is the first element of $\underline{\boldsymbol{u}}^*_{N-1}$ obtained at time step t.

Remark 6.10. *For $N_\mathrm{c} = N$ and the probabilistic case, no optimization is necessary, as $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ only consists of one input sequence, which minimizes constraint violation probability. Even for $N_\mathrm{c} < N$ no optimization is necessary, as only the first input is applied. However, obtaining an optimized input sequence is useful to warm start the optimal control problem at the next time step.*

Probability Optimization in the Probabilistic Case

In the probabilistic case, the input sequence $\underline{\boldsymbol{u}}^{*}_{N_\mathrm{c}-1}$, which leads to minimal constraint violation probability, is defined as the solution to the optimal control problem (6.80). As it is challenging to solve this optimal control problem online, we propose an approximation in the following.

Given the Gaussian uncertainty $\boldsymbol{w}_t \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}}\right)$, the mean $\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}}$ of the state sequence $\underline{\boldsymbol{x}}_{N_\mathrm{c}}$ is

$$\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}} = \underline{\boldsymbol{A}}_{N_\mathrm{c}} \boldsymbol{x}_0 + \underline{\boldsymbol{B}}_{N_\mathrm{c}} \underline{\boldsymbol{u}}_{N_\mathrm{c}-1}. \tag{6.85}$$

The covariance matrix of the state sequence $\underline{\boldsymbol{x}}_{N_\mathrm{c}}$ is determined by

$$\boldsymbol{\Sigma}_{\underline{\boldsymbol{x}}_{N_\mathrm{c}}} = \underline{\boldsymbol{E}}_{N_\mathrm{c}} \boldsymbol{\Sigma}_{\underline{\boldsymbol{w}}_{N_\mathrm{c}}} \underline{\boldsymbol{E}}^{\top}_{N_\mathrm{c}} \tag{6.86}$$

with

$$\boldsymbol{\Sigma}_{\underline{\boldsymbol{w}}_{N_\mathrm{c}}} = \operatorname{diag}\left(\boldsymbol{\Sigma}_{\boldsymbol{w}}, \cdots, \boldsymbol{\Sigma}_{\boldsymbol{w}}\right) \tag{6.87}$$

where $\boldsymbol{\Sigma}_{\underline{\boldsymbol{w}}_{N_\mathrm{c}}}$ refers to the covariance matrix of the disturbance sequence $\underline{\boldsymbol{w}}_{N_\mathrm{c}}$. Given the mean and the covariance matrix for the state sequence $\underline{\boldsymbol{x}}_{N_\mathrm{c}}$, we obtain

$$\underline{\boldsymbol{x}}_{N_\mathrm{c}} \sim \mathcal{N}\left(\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}}, \boldsymbol{\Sigma}_{\underline{\boldsymbol{x}}_{N_\mathrm{c}}}\right), \tag{6.88}$$

showing that the state sequence is subject to a Gaussian distribution.

The constraint violation probability is determined by integrating the PDF values in outside of $\mathcal{X}_t{}^{N_\mathrm{c}}$, i.e.,

$$\begin{aligned} &\Pr\Big(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t{}^{N_\mathrm{c}}\Big) = \\ &= c \int\limits_{\mathbb{R}^{n_{\boldsymbol{x}} N_\mathrm{c}} \setminus \mathcal{X}_t{}^{N_\mathrm{c}}} \exp\left(-\frac{1}{2}\left(\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}} - \underline{\boldsymbol{\xi}}\right)^{\top} \boldsymbol{\Sigma}^{-1}_{\underline{\boldsymbol{x}}_{N_\mathrm{c}}} \left(\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}} - \underline{\boldsymbol{\xi}}\right)\right) \mathrm{d}\underline{\boldsymbol{\xi}}, \qquad (6.89\text{a}) \\ &c = \left((2\pi)^{n_{\boldsymbol{x}} N_\mathrm{c}} \det \boldsymbol{\Sigma}_{\underline{\boldsymbol{x}}_{N_\mathrm{c}}}\right)^{-1}, \qquad (6.89\text{b}) \end{aligned}$$

which must be minimized. However, instead of minimizing $\Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \notin \mathcal{X}_t{}^{N_\mathrm{c}}\right)$, it is equivalent to solve

$$\max_{\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}} \Pr\left(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \in \mathcal{X}_t{}^{N_\mathrm{c}}\right) \tag{6.90}$$

with

$$\begin{aligned} &\Pr\Big(\underline{\boldsymbol{x}}_{N_\mathrm{c}} \in \mathcal{X}_t{}^{N_\mathrm{c}}\Big) = \\ &= c \int\limits_{\mathcal{X}_t{}^{N_\mathrm{c}}} \exp\left(-\frac{1}{2}\left(\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}} - \underline{\boldsymbol{\xi}}\right)^{\top} \boldsymbol{\Sigma}^{-1}_{\underline{\boldsymbol{x}}_{N_\mathrm{c}}} \left(\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}} - \underline{\boldsymbol{\xi}}\right)\right) \mathrm{d}\underline{\boldsymbol{\xi}}. \qquad (6.91) \end{aligned}$$

As we consider the probabilistic case, part of the mean state sequence $\bar{\underline{\boldsymbol{x}}}_{N_\mathrm{c}}$ is not in $\mathcal{X}_t{}^{N_\mathrm{c}}$. As an optimization using (6.91) is non-convex, we approximate (6.91). For this approximation, we first define a set with sufficiently small probability density values. An example of such a low-probability set is given in Figure 6.8.

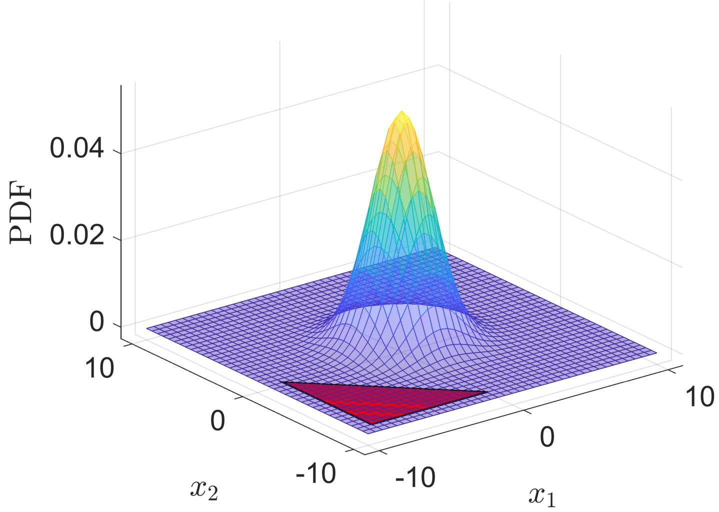


Figure 6.8: Example illustration of PDF values and a low-probability set $\mathcal{X}_{\mathrm{lp}}$ (dark red) given a bivariate normal distribution.

Definition 6.6 (Low-probability Set)**.** *The low-probability set $\mathcal{X}_{\mathrm{lp}} \in \mathbb{R}^{n_{\boldsymbol{x}} N_{\mathrm{c}}}$ is located sufficiently far enough from the mean $\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}}$ such that the PDF values within $\mathcal{X}_{\mathrm{lp}}$ are small compared to the PDF values around $\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}}$.*

We can now approximate (6.91), based on a low-probability set $\mathcal{X}_{\mathrm{lp}}$. We assume that the probabilities within $\mathcal{X}_{\mathrm{lp}}$ are of equal value relative to probabilities around $\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}}$. We then approximate the integral in (6.91) with a multiplication of the PDF value evaluated at any point $\boldsymbol{\xi} \in \mathcal{X}_{\mathrm{lp}}$ within the polytope $\mathcal{P}\left(\mathcal{X}_{\mathrm{lp}}\right)$ of $\mathcal{X}_{\mathrm{lp}}$, yielding

$$\Pr\left(\underline{\boldsymbol{x}}_{N_{\mathrm{c}}} \in \mathcal{X}_t{}^{N_{\mathrm{c}}}\right) \approx c \exp\left(-\frac{1}{2}\left(\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}} - \underline{\boldsymbol{\xi}}\right)^{\top} \boldsymbol{\Sigma}_{\underline{\boldsymbol{x}}_{N_{\mathrm{c}}}}^{-1}\left(\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}} - \underline{\boldsymbol{\xi}}\right)\right)\mathcal{P}\left(\mathcal{X}_{\mathrm{lp}}\right). \tag{6.92}$$

Note that $\underline{\boldsymbol{\xi}}$ is the augmented form of a point $\boldsymbol{\xi}$. In (6.91), $\underline{\boldsymbol{\xi}}$ is an integration variable. In (6.92), $\underline{\boldsymbol{\xi}}$ represents a sequence of elements in $\mathcal{X}_{\mathrm{lp}}$, and is later determined as part of an optimization.

Remark 6.11. *As we consider a bounded uncertainty, and therefore, the PDF is zero within $\mathcal{X}_t$, a straightforward choice for the low-probability set is $\mathcal{X}_{\mathrm{lp}} \subseteq \mathcal{X}_t{}^{N_{\mathrm{c}}}$.*

As we aim at maximizing $\Pr\left(\underline{\boldsymbol{x}}_{N_{\mathrm{c}}} \in \mathcal{X}_t{}^{N_{\mathrm{c}}}\right)$, it is sufficient to solve the quadratic optimal control problem

$$\begin{aligned}
\underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1}^{*} = & \arg\min_{\underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1}} \left(\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}} - \underline{\boldsymbol{\xi}}\right)^{\top} \boldsymbol{\Sigma}_{\underline{\boldsymbol{x}}_{N_{\mathrm{c}}}}^{-1}\left(\bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}} - \underline{\boldsymbol{\xi}}\right) && \text{(6.93a)}\\
\text{s.t.} \quad & \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \in \underline{\mathcal{U}}_{N_{\mathrm{c}}-1}, && \text{(6.93b)}\\
& \underline{\boldsymbol{\xi}} \in \mathcal{X}_{\mathrm{lp}} && \text{(6.93c)}\\
& \bar{\underline{\boldsymbol{x}}}_{N_{\mathrm{c}}} = \underline{\boldsymbol{A}}_{N_{\mathrm{c}}}\boldsymbol{x}_0 + \underline{\boldsymbol{B}}_{N_{\mathrm{c}}}\underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1}. && \text{(6.93d)}
\end{aligned}$$

Assumption 6.11. *The CVPM-MPC method is recursively feasible.*

Recursive feasibility of the proposed MPC method is shown in Section 6.3.3. However, the following Lemma requires recursive feasibility as a prerequisite. Therefore, recursive feasibility is introduced as an assumption here, which is later proved.

Lemma 6.6. *Let Assumptions 6.10 and 6.11 hold. Then, the optimization* (6.93) *always has a solution.*

Proof. The optimal control problem (6.93) is a convex quadratic program because the Hessian of the optimization is the inverse of a covariance matrix and is, therefore, always positive definite. Hence, a single minimum exists if the set ${\mathcal{X}_t}^{N_\mathrm{c}}$ and the admissible input set $\underline{\mathcal{U}}_{N_\mathrm{c}-1}$ are non-empty. The set ${\mathcal{X}_t}^{N_\mathrm{c}}$ is non-empty as it contains at least the origin due to Assumption 6.10 and the admissible input set $\underline{\mathcal{U}}_{N_\mathrm{c}-1}$ is non-empty if Assumption 6.11 holds. □

The presented approach in this section ensures that a solution may always be obtained for the probabilistic case.

6.3.3 Properties

In the following, properties of the CVPM-MPC method are analyzed. First, recursive feasibility is shown. Then, we focus on stability. In order to ensure these two properties, an assumption is required on the base MPC problem (6.71).

Assumption 6.12. *The system matrix $\boldsymbol{A}$ is stable, i.e. the eigenvalues of $\boldsymbol{A}$ are within the unit circle.*

If Assumption 6.12 is not fulfilled, the system may be stabilized with a feedback controller before MPC is applied.

Recursive Feasibility

Recursive feasibility is a fundamental property to be proved for MPC algorithms. We use Definition 2.5 for recursive feasibility. We first focus on showing recursive feasibility in case of regular disturbances $\boldsymbol{w}_t \in \mathcal{W}$, i.e., at this point we first consider a constant uncertainty bound $\mathcal{W}$. The proof of recursive feasibility for CVPM-MPC is based on the recursive feasibility of the base MPC problem (6.71).

Assumption 6.13. *The base MPC problem* (6.71) *is recursively feasible.*

As (6.71) does not consider state constraints, the zero input is feasible at any time step, which ensures recursive feasibility of the base MPC problem. Hence, a feasible initial state $\boldsymbol{x}_0$ leads to a non-empty admissible input set $\underline{\mathcal{U}}$.

Inputs not included in $\underline{\mathcal{U}}$ result in a potential loss of recursive feasibility. Therefore, in order to show recursive feasibility of CVPM-MPC, it must be guaranteed that $\mathcal{U}_\mathrm{cvpm}$ is a subset of $\underline{\mathcal{U}}$.

Lemma 6.7. *Let Assumption 6.13 hold. Then, the CVPM-MPC input set $\mathcal{U}_\mathrm{cvpm}$ is a non-empty subset of $\underline{\mathcal{U}}$.*

Proof. We initially focus on the inputs $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$. In the safe case, $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \subseteq \underline{\mathcal{U}}_{N_\mathrm{c}-1}$ according to (6.77). Therefore, for all input sequences $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1}$ it holds that $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \underline{\mathcal{U}}_{N_\mathrm{c}-1}$. In the probabilistic case, $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \underline{\mathcal{U}}_{N_\mathrm{c}-1}$ according to (6.80b). If $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} = \emptyset$ in the safe case, the probabilistic case ensures that $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \neq \emptyset$ given Lemma 6.6 and $\underline{\mathcal{U}}_{N_\mathrm{c}-1} \neq \emptyset$ according to Assumption 6.13.

For $N_\mathrm{c} = N$, $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} = \underline{\boldsymbol{u}}_{N-1}$ and $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} = \mathcal{U}_{\mathrm{cvpm}} \subseteq \underline{\mathcal{U}}$ in (6.82), ensuring that an input sequence $\underline{\boldsymbol{u}}_{N-1}$ exists at each time step.

For $N_\mathrm{c} < N$, we now focus on the remaining inputs of the input sequence $\underline{\boldsymbol{u}}_{N-1}$, given $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1}$, which consists of the first N_c input elements. As $\underline{\boldsymbol{u}}_{N_\mathrm{c}-1} \in \mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \subseteq \underline{\mathcal{U}}_{N_\mathrm{c}-1}$, it is guaranteed that $\underline{\boldsymbol{u}}_{N-1} \in \underline{\mathcal{U}}$ exists, obtained according to (6.82), given Assumption 6.13. This implies that $\mathcal{U}_{\mathrm{cvpm}} \subseteq \underline{\mathcal{U}}$.

Therefore, it holds that $\mathcal{U}_{\mathrm{cvpm}} \subseteq \underline{\mathcal{U}}$ and $\mathcal{U}_{\mathrm{cvpm}} \neq \emptyset$ for all $t \in \mathbb{N}_{\geq 0}$. □

This preliminary result allows us to conclude recursive feasibility of the CVPM-MPC method.

Theorem 6.4. *Let Assumption 6.13 hold. Then the CVPM-MPC problem* (6.83) *is recursively feasible.*

Proof. If Assumption 6.13 holds, the base MPC method is recursively feasible. According to Lemma 6.7, the CVPM-MPC method always determines an input sequence that is part of the admissible input set $\underline{\mathcal{U}}$ of the base MPC problem. Therefore, recursive feasibility is maintained. □

The CVPM-MPC method is designed such that unmodeled disturbances $\boldsymbol{w}_t \notin \mathcal{W}$ may also be handled, which is similar to considering a time-varying uncertainty bound $\mathcal{W}_t$.

Corollary 6.4. *The proposed CVPM-MPC method remains recursively feasible if an unmodeled disturbance occurs.*

Proof. The base MPC problem (6.71) does not consider $\boldsymbol{w}_t$ and is, therefore, recursively feasible for $\boldsymbol{w}_t \notin \mathcal{W}$. If $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} = \emptyset$ in the safe case after $\boldsymbol{w}_t \notin \mathcal{W}$, Lemma 6.6 ensures that $\mathcal{U}_{\mathrm{cvpm},N_\mathrm{c}-1} \neq \emptyset$ in the probabilistic case, as $\boldsymbol{w}_t \in \mathcal{W}$ is not required for the probabilistic case. Therefore, similar to Theorem 6.4, the CVPM-MPC method remains recursively feasible for $\boldsymbol{w}_t \notin \mathcal{W}$, i.e., the CVPM-MPC method remains recursively feasible for a time-variant uncertainty bound $\mathcal{W}_t$. □

As previously mentioned, by proving that the CVPM-MPC method is recursively feasible, Assumption 6.11 is reasonable.

Stability

Stability of the CVPM-MPC method is guaranteed by two steps. We show that both the safe case and the probabilistic case ensure input-to-state stability, proving that the CVPM-MPC method is ISS. In the following, we use Definition 2.6 for input-to-state stability and Lemma 2.1 to prove that our proposed method is ISS.

In this section, we consider the following notation. The input sequence

$$\underline{\boldsymbol{u}}_{N-1|t} = (\boldsymbol{u}_{t|t}, ..., \boldsymbol{u}_{t+N-1|t}) \tag{6.94}$$

obtained at time step t yields the state sequence $(\boldsymbol{x}_{t+1|t}, ..., \boldsymbol{x}_{t+N|t})$ where $\boldsymbol{x}_{t|t} = \boldsymbol{x}_t$. The following three assumptions are necessary.

Assumption 6.14. *The uncertainty bound and the constraint set are constant, i.e.,* $\mathcal{W}_t = \mathcal{W}$ *and* $\mathcal{X}_t = \mathcal{X}$*, respectively.*

Assumption 6.15. *For the stage cost it holds that* $\boldsymbol{Q} = \boldsymbol{Q}^\top \succ 0$ *and* $\boldsymbol{R} = \boldsymbol{R}^\top \succ 0$.

Assumption 6.16. *The terminal cost weight* $\boldsymbol{Q}_\mathrm{t}$ *is a solution of the Lyapunov equation*

$$\boldsymbol{A}\boldsymbol{Q}_\mathrm{t}\boldsymbol{A}^\top + \boldsymbol{Q} - \boldsymbol{Q}_\mathrm{t} = \boldsymbol{0}. \tag{6.95}$$

Based on these assumptions, we first show that for the safe case the origin of the closed-loop system (6.84) is ISS.

Lemma 6.8. *Let Assumptions 6.14, 6.15, 6.16 hold. Then, for* $\boldsymbol{x}_t \in \mathcal{X}_{\mathrm{caseS}}$*, the origin of the closed-loop system* (6.84) *is ISS.*

Proof. The proof is given in Appendix D.5. □

Consecutively applying the safe case yields similar behavior to robust MPC solutions. If the safe case is not applicable, i.e., $\boldsymbol{x}_t \notin \mathcal{X}_{\mathrm{caseS}}$, it needs to be ensured that the system is still ISS.

Lemma 6.9. *Let Assumptions 6.12, 6.14, 6.15, 6.16 hold. For* $\boldsymbol{x}_t \notin \mathcal{X}_{\mathrm{caseS}}$*, the origin of the closed-loop system* (6.84) *is ISS.*

Proof. The proof is given in Appendix D.6. □

Based on Lemma 6.8 and Lemma 6.9, we can now formulate the stability theorem for CVPM-MPC.

Theorem 6.5. *Let Assumptions 6.12, 6.14, 6.15, 6.16 hold. The origin of the closed-loop system* (6.84)*, controlled by the CVPM-MPC method, is ISS.*

Proof. Based on Lemma 6.8 and Lemma 6.9, the origin of system (6.84) is ISS for all $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$. □

In the following section, the properties are analyzed in a simulation example.

6.3.4 Simulation Study

In the following, we introduce a brief numerical example to show the advantages of CVPM-MPC. We demonstrate the capability of CVPM-MPC handling changing uncertainty bounds and highlight recursive feasibility and stability of the proposed method.

Simulation Setup

We consider a discrete-time linear system. The system matrix, input matrix, and disturbance matrix are given by

$$\boldsymbol{A} = \begin{bmatrix} 0.61 & 0.55 \\ 0.62 & \text{-}0.62 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0.85 & \text{-}0.78 \\ 0.61 & 0.53 \end{bmatrix}, \boldsymbol{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{6.96}$$

respectively. The system matrix $\boldsymbol{A}$ is stable, satisfying Assumption 6.12. The inputs are limited by a box constraint with a maximum value of 3 with the polytopic description of the input constraint

$$\mathcal{U} = \left\{ \boldsymbol{u} \,\middle|\, \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u} \leq \begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix} \right\}. \tag{6.97}$$

In the simulation, two different kinds of disturbances are considered: modeled and unmodeled disturbances. The uncertainty set is given by

$$\mathcal{W}_t = \left\{ \boldsymbol{w} \,\middle|\, \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{w} \leq \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \right\} \tag{6.98}$$

with the covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{w}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{6.99}$$

where a time-variance is considered later.

The MPC employs a horizon of $N_\mathrm{c} = N = 10$ with sampling time $\Delta t = 0.1$ and the state weighting matrix and input weighting matrix are

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{6.100}$$

respectively, according to Assumption 6.15. The terminal cost $\boldsymbol{Q}_\mathrm{t}$ is determined according to Assumption 6.16.

The simulation was run in MATLAB on a standard desktop computer. The computation of the polyhedra is done with the MPT3 toolbox presented in [83]. The average computation time of one step is 60 ms.

We consider the constant state constraint set

$$\mathcal{X}_t = \mathcal{X} = \left\{ \boldsymbol{x} \,\middle|\, \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x} \leq \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \right\}. \tag{6.101}$$

Note that the uncertainty and state constraint satisfy Assumptions 6.9 and 6.10, respectively.

Simulation Results

The following simulation shows convergence to the origin from an initial state with non-zero constraint violation probability. At time step $t = 8$, an unmodeled disturbance affects the system for one time step, which is handled by the CVPM-MPC method. An unmodeled disturbance may be interpreted as an increase of $\mathcal{W}_t$ for one time step or as $\boldsymbol{w}_t \notin \mathcal{W}_t$.

Figure 6.9 illustrates the simulation results at four different time steps. The state con-

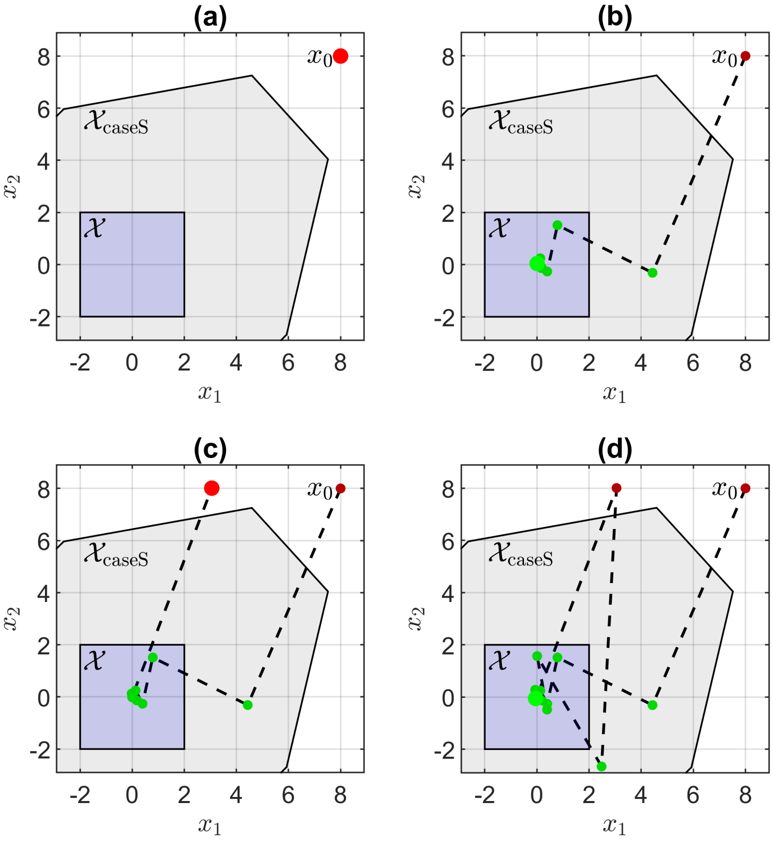


Figure 6.9: Simulation of CVPM-MPC with an unmodeled disturbance (green: Safe Case; red: Probabilistic Case; large bright marker: current state).

straint $\mathcal{X}_t$ is indicated by the blue box. The set $\mathcal{X}_{\text{caseS}}$ is marked in gray. If the system state lies within $\mathcal{X}_{\text{caseS}}$, the safe case is applicable. The initial state $\boldsymbol{x}_0 = (8,\, 8)^\top$ does not allow for zero constraint violation probability in the next step, i.e., $\boldsymbol{x}_0 \notin \mathcal{X}_{\text{caseS}}$; therefore, the probabilistic case is required, indicated by the red dot in Figure 6.9(a).

Applying the CVPM-MPC procedure for the probabilistic case moves the system state into the set $\mathcal{X}_{\text{caseS}}$, as seen in Figure 6.9(b). In $\mathcal{X}_{\text{caseS}}$ the control input is determined based on the safe case, as indicated by the green dots, since it is possible to reach $\mathcal{X}_t$ in the next step. The subsequent steps with the safe case move the system state towards the origin; however, due to the disturbance, the origin is not reached perfectly.

At time step $t = 8$, an unmodeled disturbance occurs, which moves the system state outside of $\mathcal{X}_{\text{caseS}}$, as illustrated in Figure 6.9(c). Note that input-to-state stability is not guaranteed in this step as the uncertainty bound increased. Similar to the initial simulation state, the probabilistic case is required because it is not possible to reach the constraint set $\mathcal{X}_t$ in the next step. By switching from the safe case to the probabilistic case, recursive feasibility is maintained.

Afterwards, only modeled disturbances are present, allowing the CVPM-MPC method to steer the system state back to $\mathcal{X}_{\text{caseS}}$ within one step. Input-to-state stability is again guaranteed. Subsequently, CVPM-MPC first ensures that the state constraint is satisfied and then the system state is moved towards the origin, as seen in Figure 6.9(d).

In summary, the CVPM-MPC approach yields input-to-state stability as long as modeled disturbances occur. In the presence of unmodeled disturbances, recursive feasibility is maintained and the system returns to constraint satisfaction as soon as possible.

6.3.5 Discussion

In contrast to the CVPM-MPC method presented in Section 6.2, the CVPM-MPC approach proposed in this section is more general. Here, we consider general linear constraints for a system with additive uncertainty, whereas Section 6.2 specifically focuses on two systems, one with uncertainty, and a norm-based constraint describing a relation between these two systems. The proposed approach here significantly extends the possible applications to all linear or linearized systems where constraints are linear or where constraints may be linearized. Furthermore, three cases were necessary for the CVPM-MPC method in Section 6.2, whereas this section provides a different CVPM-MPC framework, reducing the necessary cases such that only two cases need to be considered.

In Section 6.3.3, stability is discussed. As seen in the simulation example, the CVPM-MPC method is capable of remaining feasible even if unmodeled disturbances are present. The stability results may not hold in case of unmodeled disturbances, as a bounded uncertainty is assumed in the proofs (Assumption 6.14). The proposed method, however, allows us to update the assumed uncertainty bound in any step. Therefore, the stability proof becomes valid again for an increased uncertainty bound once the uncertainty bound is updated. If the uncertainty bound is not known initially, a conservative guess may be chosen and then the bound may be tightened over time, based on recorded data. The potential short loss of a stability guarantee is acceptable, however, as the main focus in this section is the minimization of constraint violation probability. Note that recursive feasibility remains guaranteed even for unmodeled disturbances.

Similar to unmodeled disturbances, the proposed CVPM-MPC method can cope with time-varying constraints. While this was not specifically shown in the simulation, changing constraints yields similar behavior compared to unmodeled disturbances, which were addressed within the simulation. Recursive feasibility remains guaranteed and stability is maintained once the adapted constraint is considered.

Considering time-variant constraints and changing uncertainty bounds is especially relevant for applications where environments change constantly, e.g., new cars appearing within the detection range of an autonomous vehicle or humans entering the workspace of a robot.

6.4 Conclusion

The proposed CVPM-MPC algorithm yields a minimal violation probability for constraints while also optimizing further objectives. The ability of CVPM-MPC to cope with time-variant constraints and uncertainty bounds provides a significant benefit for safety-critical systems. Recursive feasibility is guaranteed and, for linear constraints, stability is ensured if the uncertainty distribution is time-invariant. While the suggested method is inspired by RMPC and SMPC, it provides feasible and efficient solutions in scenarios where RMPC and SMPC encounter difficulties or are not applicable.

As norm constraints are especially useful in collision avoidance applications, the advantages of the norm-constrained CVPM-MPC method can be exploited in applications such as autonomous vehicles or robots, especially in shared environments with humans. A brief example is introduced where a controlled vehicle is overtaking a bicycle while minimizing the collision probability. Especially for collision avoidance, it is also of interest not only to focus on the collision probability but to consider the severity of collision if a collision is inevitable. Besides the previously mentioned applications, CVPM-MPC may also be useful in other areas where norm constraints are considered, such as portfolio optimization in finance.

The generalized CVPM-MPC is suitable for linear and linearized systems, enabling the use in applications such as quadcopter control or automated vehicles. Furthermore, the proposed CVPM-MPC method may be extended to consider probabilistic constraints and robust state constraints simultaneously. This extension would allow practitioners to employ robust constraints where possible and necessary, as well as probabilistic CVPM constraints if suitable.

Whereas robust methods guarantee safety for predictable events, unpredictable environment changes are not covered. This is especially tricky if ethical concerns are relevant for applications, e.g., how autonomous systems should behave if collision avoidance cannot be guaranteed [72]. CVPM-MPC provides a novel way to handle such scenarios and ethical issues.

7 Further Work

Besides the previously presented contributions, we have made further advances to MPC for automated vehicles. The first contribution, legible MPC, enables more efficient traffic flow without increasing risk by planning legible trajectories for an automated vehicle. The second contribution, Gaussian process-based SMPC for racing, demonstrates how learning-based methods can be combined with SMPC in order to improve performance, which may be exploited in autonomous racing. In this chapter, we briefly describe these two contributions. Details may be found in the respective publications.

7.1 Legible Model Predictive Control for Automated Vehicles

The following is based on the work published in [26]. Safety and efficiency are two defining factors for autonomous vehicles. While we presented multiple approaches to safely handle automated driving, approaches for higher efficiency are usually designed on an individual basis and are often accompanied by an increased risk of collision. Consider a highway scenario with two autonomous vehicles intending to overtake a leading vehicle. If both vehicles pursue a safe course, this can lead to a situation where both cars hesitate; each vehicle inhibits the other from overtaking due to safety constraints, resulting in an unresolved standoff, as algorithms of autonomous vehicles are typically conservative. As turn signals are, if at all, often only used when the lane change actually occurs, they are not sufficient for reasonable predictions. This raises the question of how to improve efficiency in such scenarios, while keeping risk at a similar level. It is assumed that vehicle-to-vehicle (V2V) communication is not a dependable solution in the near future as this would require a substantial number of vehicles communicating with each other. Furthermore, in other scenarios aiming at increasing legibility for pedestrians or bicycles, V2V communication is not applicable.

There are various definitions of legible behavior, usually related to robots. Among others, [1] and [106] state that legibility results from predictability. A further definition of legibility in human-robot interaction is provided by [63, 64]. There, it is declared that a motion is legible if it allows the spectator to confidently derive the robot's correct goal given an initial trajectory. Predictable motion is defined as the trajectory observers would expect if they knew the robot's goal prior to the execution. To illustrate this, [63] provides an example of a robot reaching out to grasp the right one of two bottles that are located next to each other. Knowing which bottle the goal is, the observer would predict the robot to follow a straight path to the bottle. However, the beginning of this trajectory would make it difficult for another spectator, who does not have knowledge of the goal, to infer which bottle the robot is aiming to grasp. Therefore, the robot should start a motion that exaggerates its movement to the right to emphasize its goal, the right bottle, which is then considered a legible motion.

In our work published in [26], in addition to focusing on the individual behavior of an autonomous vehicle, we also consider how to support other traffic participants in correctly inferring the future maneuvers of the controlled autonomous vehicle. This enables secure and efficient traffic flow. We propose a legibility-based MPC method that provides a framework to improve the readability of planned maneuvers while simultaneously optimizing factors such as comfort and energy efficiency. Legibility is introduced into the optimal control problem as part of the objective function. Safety is not affected by considering legibility in the MPC optimal control problem, as the safety constraints remain unchanged. Thus, traffic efficiency is positively influenced without increasing the risk of collision.

7.2 Gaussian Process-based SMPC for Overtaking in Autonomous Racing

The following is based on [27]. A fundamental aspect of racing is overtaking other racing cars. Whereas previous research on autonomous racing has majorly focused on lap-time optimization [92, 166, 167, 183, 184, 197], in [27] we propose a combined Gaussian Process (GP) and SMPC method to plan overtaking maneuvers in autonomous racing.

Gaussian process regression is a powerful non-parametric tool used to infer values of an unknown function given previously collected measurements. In addition to exhibiting very good generalization properties, a major advantage of GPs is that they come equipped with a measure of model uncertainty, making them particularly beneficial for safety-critical applications. These characteristics have made GPs particularly attractive for developing control algorithms [47,48,70,88]. In the context of autonomous driving, GPs have also seen a rise in interest. In [85, 92], GP regression is used to improve the model of the autonomous vehicle using collected data, which in turn leads to an improvement in control performance. GPs have also been employed to predict the behavior of cut-in maneuvers of surrounding vehicles and obtain safe autonomous vehicle control [202].

Our work [27] outlines a combined GP and SMPC approach for autonomous overtaking maneuvers in racing. The major challenge is to plan trajectories for a controlled racing car such that a leading racing car may be passed. Based on previous behavior of the leading vehicle, given the interaction between both vehicles, a GP is trained. The GP predictions for the leading vehicle are then used in an SMPC algorithm to plan efficient overtaking maneuvers. The proposed method has the potential to be a powerful method in autonomous racing. Ideally, the GP identifies weaknesses in the driving behavior of the leading vehicle while the controlled trailing vehicle is trying to overtake. The SMPC planner allows us to efficiently consider the GP output and to plan optimistic vehicle trajectories, which are fundamental for racing. Given an increased sample set of data, the controlled racing car increases its chances of finding the right spot on a race track and a suitable driving approach to successfully overtake.

In summary, the long-term goal is to develop a method for competitive racing: First, the controlled autonomous racing car learns weak spots of other (leading) vehicles while driving behind those vehicle. Then, the SMPC approach enables optimistically overtaking the leading vehicle at the right part of the race track, based on the results of the GP.

Conclusion 8

Ever increasing needs for automation require control algorithms that are suited for safety-critical applications in uncertain environments. Examples for such safety-critical applications are robots acting in a shared work space with humans or automated vehicles avoiding collisions with surrounding vehicles and pedestrians. MPC has shown promise to be a capable method for such systems. Whereas RMPC is often too conservative, SMPC makes it possible to plan optimistic control behavior with the disadvantage of allowing for a small probability of constraint violation. These issues require new solutions to make MPC a suitable and reliable method for safety-critical systems.

This thesis has focused on advancing SMPC, as well as guaranteeing safety for MPC in safety-critical applications subject to uncertainty. The most prominent application examples throughout this thesis were automated vehicles and mobile robots, as both currently are highly relevant safety-critical applications.

8.1 Summary of Contributions

After briefly introducing SMPC and providing a simple simulation example in Chapter 2, the subsequent four chapters present the contributions of this thesis, covering SMPC advances, approaches to increase efficiency in control behavior, and safety guarantees.

Chapter 3 proposes two SMPC advances, specifically focusing on the reformulation of SMPC chance constraints. First, an SMPC method is proposed that combines analytic and sampling-based chance constraint reformulation. This combination allows us to efficiently consider two vastly different types of uncertainty within applications: task uncertainty and task execution uncertainty of other agents. Second, we propose a novel grid-based approach to evaluate the chance constraint, specifically designed for two-dimensional applications. This grid-based SMPC approach allows us to directly use occupancy grid data, obtained in the sensing and perception module. Based on a risk threshold, the grid is divided into admissible and inadmissible cells. Then, an MPC optimal control problem is solved, planning trajectories only on the admissible cells, yielding a computationally efficient way to reformulate chance constraints.

In Chapter 4, the focus is on extending the MPC prediction horizon without increasing computational complexity. We first consider the undisturbed case and combine two known approaches: MPC for models of different granularity and MPC with a non-uniformly spaced optimization horizon. Then, uncertainty is introduced into the algorithm. We use robust constraints for short-term planning to ensure safety, and we employ chance constraints on the long-term horizon, together with a simplified model and larger sampling time steps. This allows us to improve long-term planning without overly conservative long-term constraints and without large computational effort. For the undisturbed case, we provide a recursive feasibility guarantee for the MPC optimal control problem.

The next two chapters address safety within MPC. Chapter 5 proposes a safety algorithm for SMPC, particularly suitable for automated vehicles. The algorithm is designed in such a way that the optimistic SMPC inputs are applied as long as a safe backup planner is still able to find a safe backup trajectory. An autonomous highway driving simulation study shows that in regular scenarios, this procedure yields similar behavior to SMPC. However, in emergency scenarios where applying SMPC inputs results in collisions, the proposed algorithm ensures safety. We prove recursive feasibility of the safe SMPC algorithm designed for vehicles and guarantee that no self-inflicted collisions occur. Additionally, we prove input-to-state stability for the general safe SMPC algorithm.

In Chapter 6, we propose a novel MPC approach that handles unexpected events. Instead of focusing on hard constraints or chance constraints, we minimize the probability of constraint violation. The proposed MPC algorithm is recursively feasible and input-to-state stability is ensured for time-invariant uncertainty bounds.

8.2 Implications

In the introduction, we outlined four key challenges to successfully design MPC for safety-critical applications. In the following, we revisit these challenges and elaborate on implications regarding the contributions of this thesis.

Challenge 1: SMPC chance constraint reformulation. SMPC chance constraints require tractable surrogates within the optimal control problem. Whereas multiple ideas for such chance constraint reformulations exist, each one is accompanied by limitations or disadvantages. We proposed two application-oriented approaches, one to formulate chance constraints and one to obtain a tractable chance constraint reformulation.

In Chapter 3, we proposed a general algorithm to formulate SMPC chance constraints based on applications with task and task execution uncertainty. However, *task* does not need to be taken literally here. This SMPC algorithm is also applicable to distantly related applications, e.g., portfolio optimization [84,153] or dynamic option hedging [11,154] where uncertainty in the stock market may be described by bullish or bearish markets with varying individual trends.

The second proposed approach, grid-based SMPC, focuses on the chance constraint reformulation. This grid-based approach enables to exploit data from occupancy grids, making it possible to further merge sensing, perception, and planning, especially for automated vehicles. As each part of the road is given a probability of occupation, this grid-based SMPC approach facilitates a comprehensible trade-off between efficiency and risk, potentially helping to simplify type approval of automated vehicles.

Challenge 2: Extending the MPC prediction horizon. The proposed modular MPC approach described in Chapter 4 allows us to increase the MPC prediction horizon without significantly increasing computational complexity. Depending on the particular application and demands, different aspects of the proposed method may be employed. A non-uniformly spaced prediction horizon, potentially with more than two different sampling time step sizes, can be combined with multiple prediction models of varying complexity. Furthermore, depending on the system uncertainty and control requirements, robust constraints or chance constraints can be employed for the short-term or long-term prediction. This flexibility makes it possible to customize controllers for a broad class of systems and application scenarios. However, the discontinuity between different prediction models and the changing sampling time step size make the proofs of theoretic guarantees challenging.

Challenge 3: Safety in SMPC. Designing SMPC for safety-critical applications subject to uncertainty is a research area that has just started to draw attention recently. Our safety algorithm for SMPC in Chapter 5 avails the domain of safety-critical applications for SMPC. Having provided a safety guarantee for general SMPC approaches therefore justifies an increased research focus on SMPC, as application opportunities are vast if the disadvantage of a non-zero constraint violation probability is canceled. Furthermore, our safety framework for SMPC may also be used to provide safety guarantees for other control methods or to provide a safety filter for human actions, e.g., as part of advanced driver assistance systems.

Challenge 4: Minimal constraint violation probability with MPC. Typically, safety is guaranteed for the case that certain assumptions hold. However, in many practical applications, assumptions either have to be overly conservative or will be violated at some point. If violated, the theoretic safety guarantees are obsolete in practice. To counter this issue, we proposed a novel MPC method in Chapter 6, minimizing constraint violation probability, which is a new way of considering uncertainty within MPC that lies between RMPC and SMPC: as long as uncertainty and constraint assumptions hold, robust safety is guaranteed; once an unexpected event occurs, the solution with minimal future constraint violation probability is found. Together with the proposed safe SMPC algorithm of Chapter 5, these two approaches cover safety-critical application scenarios where assumptions are either met or violated.

8.3 Future Research Directions

This thesis provided significant research contributions towards safely applying MPC in safety-critical applications. Nevertheless, many challenges remain unsolved and contributions made here are only the starting point for future research directions. In the following, we highlight open research questions and opportunities.

Extended prediction horizons for MPC. For the proposed methods that extend the MPC prediction horizon, theoretic properties still remain mostly unaddressed. Passivity theory may be used to show stability for our MPC methods including a non-uniformly spaced optimization horizon. Future work may also focus on investigating possibilities to relax the requirements for the link between the detailed and coarse model. This link is, to this point, very application-dependent and limits the applicability. This challenge may be addressed by considering methods of model order reduction [80].

Grid-based stochastic MPC. The proposed grid-based SMPC method is designed for two-dimensional applications and specifically for automated vehicles. Besides testing with real-world data, future research efforts may focus on further exploiting occupancy grid data obtained during sensing and perception, effectively merging the sensing and planning modules. Furthermore, the proposed idea is extendable to three-dimensional applications, enabling the use for robots, drones, and underwater vehicles.

Safe MPC for uncertain systems. The major contribution of this thesis is in the area of safety within MPC for systems subject to uncertainty. However, open challenges still remain. The presented safe SMPC approach for automated highway driving may be extended to other scenarios that automated vehicles encounter, specifically in urban environments. Furthermore, our safe SMPC approach can be used in further safety-critical systems, such as robotics.

As we first proposed MPC with minimal constraint violation probability, this method offers various opportunities for improvement. A wider system class may be considered and robust constraints as well as chance constraints can be included. Then, depending on the application and uncertainty, a constraint can be considered robustly, as a chance constraint, or the constraint violation probability is minimized.

Eventually, MPC with constraint violation probability minimization may serve as a further backup to the proposed safe SMPC approach, e.g., covering those scenarios in automated driving where other traffic participants violate traffic rules. Then, the combined method causes no self-inflicted collisions and minimizes collision probability even if other traffic participants behave illegally. This approach could not only drastically increase the applicability of MPC for vehicles, but also pave the way to tackle frequently debated ethical issues regarding automated vehicles [72] and would, therefore, improve public perception of automated driving.

Data-driven stochastic MPC. MPC uses system models for predictions; however, the behavior of real systems is often difficult to model. This challenge is addressed by data-driven control. In the case of MPC, results of behavioral systems theory are applicable, where system predictions are possible only using past system data. A combination of data-driven control with ideas of SMPC will enable efficient control of unknown systems in uncertain environments. Particular challenges within data-driven stochastic MPC are handling noisy data in the data collection phase and considering disturbances and measurement noise in the control phase.

A Probability Theory

The following sections provide details on necessary fundamentals of probability theory.

A.1 Normal Distributions

A normal distribution (also known as a Gaussian distribution) is characterized by its mean $\boldsymbol{\mu}$ and its covariance $\boldsymbol{\Sigma}$, abbreviated by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. A special case is given by the standard normal distribution where $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$. In the following, we consider a random variable W subject to a univariate normal distribution with mean μ, variance σ^2, and standard deviation σ.

The probability density function (PDF) of the random variable W is given by

$$f_W(w) = \frac{1}{\sigma}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{w-\mu}{\sigma}\right)\right) \tag{A.1}$$

where it holds that

$$\int_{-\infty}^{\infty} f_W(w)\mathrm{d}w = 1. \tag{A.2}$$

The factor $\frac{1}{\sigma}$ accounts for the adjusted variance compared to the standard normal distribution.

For a continuous distribution, the PDF at a specific point w does not yield a probability. Therefore, we introduce the cumulative distribution function (CDF) $F(w)$ with random variable W

$$\begin{aligned} F(w) = \Pr(W \leq w) = \int_{-\infty}^{w} f_W(w)\mathrm{d}w &= \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{w-\mu}{\sigma\sqrt{2}}\right)\right] && \text{(A.3a)} \\ &= \frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{w-\mu}{\sigma\sqrt{2}}\right). && \text{(A.3b)} \end{aligned}$$

For $w = \mu$, the cumulative probability is exactly 0.5.

In the CDF, only one side of the distribution is considered, i.e., $W \leq w$. If we consider both sides of the distribution, i.e., $-w \leq W \leq w$, we obtain

$$\begin{aligned}
\Pr(-w \leq W \leq w) &= \int_{-w}^{w} f_W(w)\mathrm{d}w && \text{(A.4a)}\\
&= \int_{-w}^{0} f_W(w)\mathrm{d}w + \int_{0}^{w} f_W(w)\mathrm{d}w && \text{(A.4b)}\\
&= 2\int_{0}^{w} f_W(w)\mathrm{d}w && \text{(A.4c)}\\
&= 2\left(\int_{-\infty}^{w} f_W(w)\mathrm{d}w - \int_{-\infty}^{0} f_W(w)\mathrm{d}w\right) && \text{(A.4d)}\\
&= 2\left(\left(\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{w-\mu}{\sigma\sqrt{2}}\right)\right) - \frac{1}{2}\right) && \text{(A.4e)}\\
&= \mathrm{erf}\left(\frac{w-\mu}{\sigma\sqrt{2}}\right). && \text{(A.4f)}
\end{aligned}$$

A.2 Chebyshev's Inequality

Chebyshev's inequality provides a bound for probability distributions and indicates how likely it is that a random variable differs from the mean given a certain threshold c_{th}.

For a random variable W with zero mean distribution, i.e., $\mu = 0$, and variance σ^2, it holds that

$$\Pr\left(|W| \geq c_{\text{th}}\right) \leq \frac{\sigma^2}{c_{\text{th}}^2} \tag{A.5}$$

where $c_{\text{th}} > 0$. Complementary, it holds that

$$\Pr\left(|W| < c_{\text{th}}\right) \geq 1 - \frac{\sigma^2}{c_{\text{th}}^2}. \tag{A.6}$$

The zero mean results may be extended to distributions with mean $\mu \neq 0$, yielding

$$\begin{aligned}
\Pr\left(|W-\mu| \geq c_{\text{th}}\right) &\leq \frac{\sigma^2}{c_{\text{th}}^2} && \text{(A.7a)}\\
\Pr\left(|W-\mu| < c_{\text{th}}\right) &\geq 1 - \frac{\sigma^2}{c_{\text{th}}^2}. && \text{(A.7b)}
\end{aligned}$$

Cantelli's inequality. The one-sided Chabyshev's inequality is also known as the Cantelli's inequality where only a single tail of the distribution is considered. Cantelli's inequality for zero mean and non-zero mean distributions is given by

$$\begin{aligned}
\Pr\left(W \geq c_{\text{th}}\right) &\leq \frac{\sigma^2}{\sigma^2 + c_{\text{th}}^2} && \text{(A.8a)}\\
\Pr\left(W - \mu \geq c_{\text{th}}\right) &\leq \frac{\sigma^2}{\sigma^2 + c_{\text{th}}^2}. && \text{(A.8b)}
\end{aligned}$$

from which it follows that

$$\begin{aligned}
\Pr\left(W < c_{\text{th}}\right) &= 1 - \Pr\left(W - \mu \geq c_{\text{th}}\right) \geq 1 - \frac{\sigma^2}{\sigma^2 + c_{\text{th}}^2} && \text{(A.9a)}\\
\Pr\left(W - \mu < c_{\text{th}}\right) &= 1 - \Pr\left(W - \mu \geq c_{\text{th}}\right) \geq 1 - \frac{\sigma^2}{\sigma^2 + c_{\text{th}}^2}. && \text{(A.9b)}
\end{aligned}$$

Analytic SMPC Chance Constraint Reformulation

B

In the following, a brief derivation of chance constraints in SMPC for systems with additive uncertainties is provided. Here, we omit explicitly denoting the prediction step k.

Starting from the chance constraint (2.6), Assumption 2.2, and the reformulated system state (2.7), we rewrite the chance constraint as

$$\Pr\left(\boldsymbol{g}^\top \boldsymbol{z} + \boldsymbol{g}^\top \boldsymbol{e} \leq h\right) \geq \beta. \tag{B.1}$$

The chance constraint (B.1) is not a deterministic expression. It is necessary to reformulate the chance constraint such that a tractable expression is obtained, which is then used to solve the optimal control problem. For zero uncertainty, the deterministic part of the state must satisfy the state constraint. However, if uncertainty is present, the constraint must be tightened by a tightening parameter γ. This tightening parameter is determined depending on the uncertainty and the risk parameter β. We therefore reformulate the chance constraint (B.1) into

$$\boldsymbol{g}^\top \boldsymbol{z} \leq h - \gamma \tag{B.2a}$$

$$\Pr\left(\boldsymbol{g}^\top \boldsymbol{e} \leq \gamma\right) = \beta \tag{B.2b}$$

where (B.2b) ensures that the tightening parameter is chosen in such a way, that the uncertainty only causes constraint violations as allowed by the risk parameter.

It is to note that (B.2) is still not a deterministic expression. In the following, we derive how (B.2b) is reformulated into a deterministic approximation, based on the underlying uncertainty distribution.

B.1 Normally Distributed Uncertainty

First, we analyze the chance constraint reformulation for an uncertainty $\boldsymbol{w}$ according to Assumption 2.1. Similar reformulations are also used in [16, 30, 50, 67, 150].

Given the normal distribution with zero mean and the error propagation (2.10), the error is also normally distributed according to

$$\boldsymbol{e} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}^e\right). \tag{B.3}$$

It follows that the state expression is also normally distributed according to

$$\boldsymbol{g}^\top \boldsymbol{e} \sim \mathcal{N}\left(0, \sigma^2\right) \tag{B.4}$$

where $\boldsymbol{g}^\top \boldsymbol{e}$ is scalar and we abbreviate the variance with $\sigma^2 = \boldsymbol{g}^\top \boldsymbol{\Sigma}^e \boldsymbol{g}$.

This now allows us to reformulate the chance constraint (B.2b). Based on the CDF $F(\gamma)$ of a normal distribution

$$\Pr\left(\boldsymbol{g}^\top \boldsymbol{e} \leq \gamma\right) = F(\gamma) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{\gamma}{\sigma\sqrt{2}}\right)\right) = \beta, \tag{B.5}$$

it is possible to find a deterministic expression for the tightening parameter γ with the following reformulation:

$$\frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{\gamma}{\sigma\sqrt{2}}\right)\right) = \beta \tag{B.6a}$$

$$\operatorname{erf}\left(\frac{\gamma}{\sigma\sqrt{2}}\right) = 2\beta - 1 \tag{B.6b}$$

$$\gamma = \sqrt{2}\sigma \operatorname{erf}^{-1}(2\beta - 1) \tag{B.6c}$$

Details on the CDF of a normal distribution are found in Appendix A.1. Inserting the variance $\sigma^2 = \boldsymbol{g}^\top \boldsymbol{\Sigma}^e \boldsymbol{g}$, we obtain the tightening parameter

$$\gamma = \sqrt{2}\sqrt{\sigma^2} \operatorname{erf}^{-1}(2\beta - 1) \tag{B.7a}$$

$$= \sqrt{2}\sqrt{\boldsymbol{g}^\top \boldsymbol{\Sigma}^e \boldsymbol{g}} \operatorname{erf}^{-1}(2\beta - 1) \tag{B.7b}$$

$$= \sqrt{2\boldsymbol{g}^\top \boldsymbol{\Sigma}^e \boldsymbol{g}} \operatorname{erf}^{-1}(2\beta - 1), \tag{B.7c}$$

as stated in (2.11b). The error function $\operatorname{erf}(\cdot)$ and the inverse error function $\operatorname{erf}^{-1}(\cdot)$ are displayed in Figure B.1.

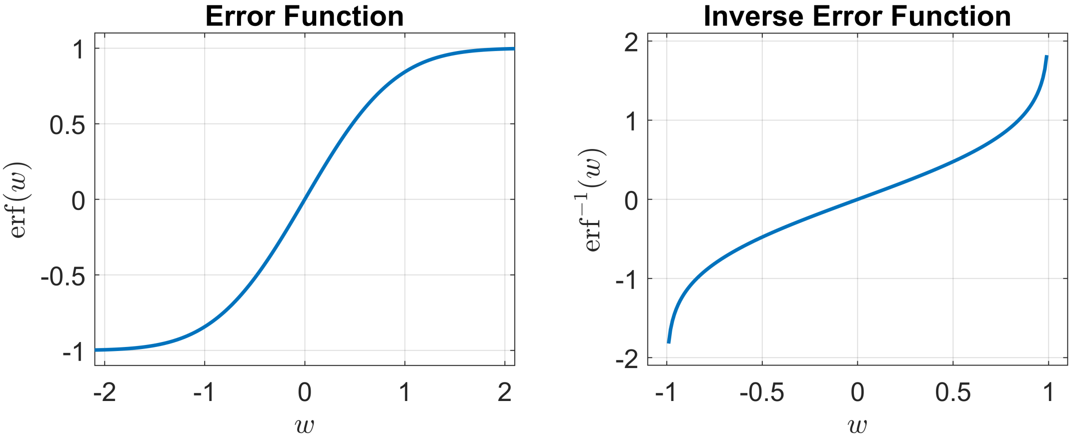


Figure B.1: Error function and inverse error function.

Remark B.1. *Note that a similar expression to [68, Section 3.2.1] is obtained by considering the risk parameter $\beta = 1 - \tilde{\beta}$ introduced in Remark 2.2. Then, with variance σ and based on the CDF $F_{\text{snd}}(\gamma)$ of the standard normal distribution, it follows that*

$$\Pr\left(\boldsymbol{g}^\top \boldsymbol{e} \leq \gamma\right) = F(\gamma) = 1 - \tilde{\beta} \tag{B.8a}$$

$$\gamma = F^{-1}\left(1 - \tilde{\beta}\right) = \sigma F_{\text{snd}}^{-1}\left(1 - \tilde{\beta}\right), \tag{B.8b}$$

which yields

$$\gamma = \sqrt{\boldsymbol{g}^\top \boldsymbol{\Sigma}^e \boldsymbol{g}}\; F_{\text{snd}}^{-1}\left(1 - \tilde{\beta}\right). \tag{B.9}$$

B.2 General Probability Distributions

A chance constraint reformulation of (B.2b) is also possible for general probability distributions.

Assumption B.1. *The uncertainty $\boldsymbol{w}$ is a univariate distribution with zero mean and variance σ^2.*

While a normal distribution allows for an exact reformulation, chance constraints for general distributions are often only approximated. An overview is found in [68].

Based on the Cantelli's inequality, see Appendix A, it is possible to determine a bound on the CDF, yielding

$$\Pr\left(\boldsymbol{g}^\top \boldsymbol{e} < \gamma\right) \geq 1 - \frac{\sigma^2}{\gamma^2 + \sigma^2} = \beta \tag{B.10}$$

where the risk parameter represents the required bound. Reformulating results in

$$\begin{aligned} 1 - \frac{\sigma^2}{\gamma^2 + \sigma^2} &= \beta && \text{(B.11a)} \\ \sigma^2 &= \gamma^2(1-\beta) + \sigma^2(1-\beta) && \text{(B.11b)} \\ \sigma^2 + \sigma^2(\beta - 1) &= \gamma^2(1-\beta) && \text{(B.11c)} \\ \gamma &= \sigma\sqrt{\frac{\beta}{1-\beta}} && \text{(B.11d)} \end{aligned}$$

representing a constraint tightening parameter for general probability distributions. We can rewrite this expression as

$$\gamma = \sigma F^{-1}(\beta) \tag{B.12}$$

where the quantile function $F^{-1}(\beta)$ is considered as the inverse CDF. Unlike the reformulation for the norm constraint, the risk parameter here may take values $0 \leq \beta < 1$.

A comparison between constraint tightening based on (B.7c), i.e,

$$F^{-1}(\beta) = \sqrt{2}\mathrm{erf}^{-1}(2\beta - 1), \tag{B.13}$$

and (B.11d), i.e., $\sqrt{\beta/(1-\beta)}$, is shown in Figure B.2. Note that the constraint tightening in Figure B.2 is not γ, as variance is not considered. It is obvious that the constraint tightening for general probability distributions according to (B.11d) is more conservative.

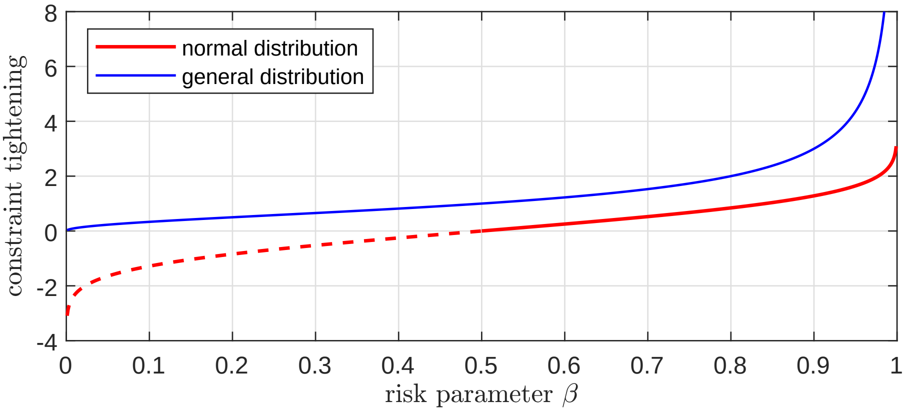


Figure B.2: Constraint tightening comparison between (B.7c) and (B.11d).

Details on Safety for Stochastic MPC

C

This appendix chapter provides additional information for Chapter 5, which addressed safety in SMPC.

C.1 Linearized and Discretized Vehicle Model

The linearized, discrete-time system matrices $\boldsymbol{A}_\mathrm{d}$ and $\boldsymbol{B}_\mathrm{d}$ in (5.6) are given by

$$\boldsymbol{A}_\mathrm{d}= \begin{bmatrix} 1 & 0 & -\Delta t v \sin z_1 & \Delta t v \cos z_1 - \frac{z_2 \sin z_1}{2 z_4} \\ 0 & 1 & \Delta t v \cos z_1 & \Delta t v \sin z_1 + \frac{z_2 \cos z_1}{2 z_4} \\ 0 & 0 & 1 & \frac{\Delta t \tan \delta_\mathrm{f}}{z_4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.1a}$$

$$\boldsymbol{B}_\mathrm{d}= \begin{bmatrix} \frac{\Delta t^2 \cos z_1}{2} & -\frac{\Delta t^2 v z_7 \sin z_1}{2} - \frac{z_8 \sin z_1}{z_9} \\ \frac{\Delta t^2 \sin z_1}{2} & \frac{\Delta t^2 v z_7 \cos z_1}{2} + \frac{z_8 \cos z_1}{z_9} \\ \frac{\Delta t^2 \tan \delta_\mathrm{f}}{2 z_4} & \Delta t z_7 \\ \Delta t & 0 \end{bmatrix} \tag{C.1b}$$

with

$$z_1= \phi + \arctan\left(\frac{l_\mathrm{r} \tan \delta_\mathrm{f}}{l_\mathrm{r} + l_\mathrm{f}}\right) \tag{C.2a}$$

$$z_2= \Delta t^2 v \tan \delta_\mathrm{f} \tag{C.2b}$$

$$z_3= \left(l_\mathrm{r} \tan \delta_\mathrm{f}\right)^2 \tag{C.2c}$$

$$z_4= \left(l_\mathrm{r} + l_\mathrm{f}\right) \left(\frac{z_3}{\left(l_\mathrm{r} + l_\mathrm{f}\right)^2} + 1\right)^{\frac{1}{2}} \tag{C.2d}$$

$$z_5= v \left(\left(\tan \delta_\mathrm{f}\right)^2 + 1\right) \tag{C.2e}$$

$$z_6= \left(l_\mathrm{r} + l_\mathrm{f}\right)^3 \left(\frac{z_3}{\left(l_\mathrm{r} + l_\mathrm{f}\right)^2} + 1\right)^{\frac{3}{2}} \tag{C.2f}$$

$$z_7= \frac{z_5}{z_4} - \frac{z_3 z_5}{z_6} \tag{C.2g}$$

$$z_8= \Delta t l_\mathrm{r} z_5 \tag{C.2h}$$

$$z_9= \left(l_\mathrm{r} + l_\mathrm{f}\right) \left(\frac{z_3}{\left(l_\mathrm{r} + l_\mathrm{f}\right)^2} + 1\right). \tag{C.2i}$$

C.2 Proof of Lemma 5.1

Proof. As longitudinal and lateral motion are uncorrelated, the covariance matrix is given in terms of a block diagonal matrix

$$\boldsymbol{\Sigma}_k^e = \begin{bmatrix} \boldsymbol{\Sigma}_{x,k}^e & 0 \\ 0 & \boldsymbol{\Sigma}_{y,k}^e \end{bmatrix}, \tag{C.3}$$

with

$$\begin{aligned} \boldsymbol{\Sigma}_{x,k}^e &= \begin{bmatrix} \sigma_{x,k}^2 & \sigma_{xv_x,k}^2 \\ \sigma_{xv_x,k}^2 & \sigma_{v_x,k}^2 \end{bmatrix}, \\ \boldsymbol{\Sigma}_{y,k}^e &= \begin{bmatrix} \sigma_{y,k}^2 & \sigma_{yv_y,k}^2 \\ \sigma_{yv_y,k}^2 & \sigma_{v_y,k}^2 \end{bmatrix}, \end{aligned} \tag{C.4}$$

and each direction is computed independently. We will show that the error for the position coordinate x is distributed with a PDF only depending on $\sigma_{x,k}^2$. Let $\boldsymbol{x}_{x,k} = (x_k,\ v_{x,k})^\top$ be the state vector projected onto its longitudinal coordinates with the estimated states $\hat{\boldsymbol{x}}_{x,k} = (\hat{x}_k,\ \hat{v}_{x,k})^\top$ and $\boldsymbol{e}_{x,k} = \hat{\boldsymbol{x}}_{x,k} - \boldsymbol{x}_{x,k}$, then $\boldsymbol{e}_{x,k}$ is Gaussian distributed with the bivariate probability function

$$f_{\boldsymbol{e}}(\boldsymbol{e}_{x,k}) = \frac{1}{2\pi\sqrt{\det\left(\boldsymbol{\Sigma}_{x,k}^e\right)}} \exp\left(-\frac{1}{2}\boldsymbol{e}_{x,k}^\top \left(\boldsymbol{\Sigma}_{x,k}^e\right)^{-1} \boldsymbol{e}_{x,k}\right). \tag{C.5}$$

We obtain the marginal PDF by integrating $f_{\boldsymbol{e}}(\boldsymbol{e}_{x,k})$ over the longitudinal velocity, i.e.,

$$f_{\boldsymbol{e},x}(\boldsymbol{e}_{x,k}) = \int_{-\infty}^{\infty} f(\boldsymbol{e}_{x,k}) \mathrm{d}v_{x,k}, \tag{C.6}$$

which yields

$$f_{\boldsymbol{e},x}(\boldsymbol{e}_{x,k}) = \frac{1}{\sqrt{2\pi\sigma_{x,k}^2}} \exp\left(-\frac{1}{2\sigma_{x,k}^2} x_k^2\right). \tag{C.7}$$

As (C.7) corresponds to the univariate Gaussian distribution of the x-coordinate with only $\sigma_{x,k}$, the x-direction is computed independently of $\sigma_{v_x,k}$ and $\sigma_{xv_x,k}$, which proves the lemma. The same proof holds for the lateral direction. □

C.3 Constraint Generation

Here, we give a complete overview of the cases considered. The cases and conditions for SMPC are found in Table C.1. The values $c_{i,x/y,k}^{\mathrm{EV/TV}}$, $i \in \{1,2,3,4\}$ indicate the corner x-position and y-position of the EV shape or TV safety rectangle, according to Figure C.1. While the main idea of the cases is already given in Section 5.2.4, some extra details were not previously mentioned for clarity. In the brief constraint generation overview, we only used the constant part ρ_{close}. However, this was a simplification; we employ a function $\rho_{\mathrm{close}}(\Delta v)$ consisting of a constant part ρ_{close} and a variable part $\max\{0, \operatorname{sign}(\Delta v)\,\Delta v N \Delta t\}$, i.e., the necessary distance between the EV and TV depends on the velocity difference. A larger velocity difference results in greater difference of distance covered by the EV and TV within the prediction time $N\Delta t$. This yields

$$\rho_{\mathrm{close}}(\Delta v) = \rho_{\mathrm{close}} + \max\{0, \operatorname{sign}(\Delta v)\,\Delta v N \Delta t\}. \tag{C.8}$$

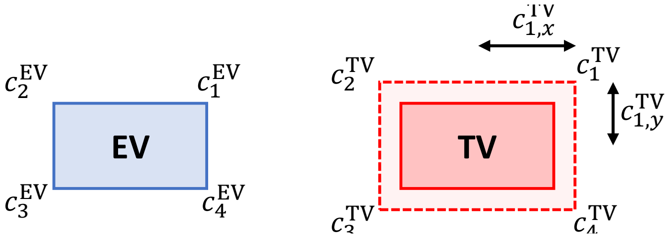


Figure C.1: Corner description for the EV shape and TV safety rectangle. © 2021 IEEE.

As the general plan for the EV is to overtake only on the left, case E is extended slightly compared to Section 5.2.4. When the EV is too close to the TV, based on a left-lane margin ρ_{llm}, a vertical constraint behind the TV replaces the inclined constraint (case E_3). It is only planned to overtake TVs on a lane left of the EV if the EV velocity is larger than the TV velocity (case E_2). For cases D and E the constraint slope is bounded such that it does not lie within with the EV shape or the TV safety rectangle.

The cases and conditions for FTP are similar to the SMPC cases. A complete description is given in Table C.2. The main idea of the FTP cases are described in Section 5.2.5. Here, we give a detailed description of the cases where three placeholder TV predictions are considered (cases J^* and C^*). The cases with a TV prediction in the same lane as the EV are denoted by $\mathrm{J}^*_{\mathrm{S}}$, $\mathrm{C}^*_{\mathrm{S}}$, while cases with a TV prediction in a lane to the left or right of the TV are denoted by $\mathrm{J}^*_{\mathrm{L}}$, $\mathrm{C}^*_{\mathrm{L}}$ and $\mathrm{J}^*_{\mathrm{R}}$, $\mathrm{C}^*_{\mathrm{R}}$, respectively. The slopes for cases $\mathrm{C}^*_{\mathrm{L}}$ and $\mathrm{C}^*_{\mathrm{R}}$ are limited to $q_{x,k} \leq 0$ and $q_{x,k} \geq 0$, respectively. If the values for $q_{x,k}$ exceed the respective limits, the cases $\mathrm{C}^*_{\mathrm{L,lim}}$ and $\mathrm{C}^*_{\mathrm{R,lim}}$ are applied. The cases F^*_2 and H^*_2 represent scenarios where the EV center is not in the TV lane, but the EV shape is already in the TV lane. In these cases, where the EV is behind the TV, vertical constraints behind the TV are built. The cases F^* and H^* are split into F^*_a, F^*_b and H^*_a, H^*_b, respectively.

case	longitudinal conditions (position and velocity)	lateral conditions (position)	$q_{x,k}$	$q_{y,k}$	$q_{t,k}$
A	$\lvert\Delta x_0^{\mathrm{EV,TV}}\rvert \geq \rho_{\mathrm{lar}}$	-	0	0	0
B	$\rho_{\mathrm{close}}(\Delta v) < -\left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	-	1	0	$-c_{2,x,k}^{\mathrm{TV}}$
C	$\rho_{\mathrm{close}}(\Delta v) < \left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	-	-1	0	$c_{1,x,k}^{\mathrm{TV}}$
D	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	$\max\left\{0, \frac{c_{4,y,0}^{\mathrm{EV}} - c_{2,y,k}^{\mathrm{TV}}}{c_{4,x,0}^{\mathrm{EV}} - c_{2,x,k}^{\mathrm{TV}}}\right\}$	-1	$c_{2,y,k}^{\mathrm{TV}} - s_{x,k} c_{4,x,0}^{\mathrm{EV}}$
E	$-\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$, $s_0 + 0.5 w_{\mathrm{veh}} + \rho_{\mathrm{llm}} \leq x_0^{\mathrm{TV}}$, $v_0 - v_{x,0}^{\mathrm{TV}} > 0$	$y_{\mathrm{lane},0}^{\mathrm{EV}} + w_{\mathrm{lane}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	$\max\left\{0, \frac{c_{4,y,0}^{\mathrm{EV}} - c_{2,y,k}^{\mathrm{TV}}}{c_{4,x,0}^{\mathrm{EV}} - c_{2,x,k}^{\mathrm{TV}}}\right\}$	-1	$c_{2,y,k}^{\mathrm{TV}} - s_{x,k} c_{4,x,0}^{\mathrm{EV}}$
E_2	$-\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$, $s_0 + 0.5 w_{\mathrm{veh}} + \rho_{\mathrm{llm}} \leq x_0^{\mathrm{TV}}$, $v_0 - v_{x,0}^{\mathrm{TV}} > 0$	$y_{\mathrm{lane},0}^{\mathrm{EV}} + w_{\mathrm{lane}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	1	0	$-c_{2,x,k}^{\mathrm{TV}}$
E_3	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$, $s_0 + 0.5 w_{\mathrm{veh}} + \rho_{\mathrm{llm}} > x_0^{\mathrm{TV}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} + w_{\mathrm{lane}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	1	$-c_{3,y,k}^{\mathrm{TV}}$
F	$\lvert\Delta x_0^{\mathrm{EV,TV}}\rvert \leq \rho_{\mathrm{close}}(\Delta v)$	$y_{\mathrm{lane},0}^{\mathrm{EV}} > y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	-1	$c_{2,y,k}^{\mathrm{TV}}$
G	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$	$y_{\mathrm{lane},0}^{\mathrm{EV}} + 2 w_{\mathrm{lane}} \leq y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	1	$-c_{3,y,k}^{\mathrm{TV}}$
H	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$	$y_{\mathrm{lane},0}^{\mathrm{EV}} \leq y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	1	$-c_{4,y,k}^{\mathrm{TV}}$
J	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}(\Delta v)$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	0	0

Table C.1: SMPC constraints.

case	longitudinal conditions (position and velocity)	lateral conditions (position)	$q_{x,k}$	$q_{y,k}$	$q_{t,k}$
A*	$\lvert\Delta x_0^{\mathrm{EV,TV}}\rvert \geq \rho_{\mathrm{lar}}$	-	0	0	0
B*	$\rho_{\mathrm{close}}^{\mathrm{FTP}} < -\left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	-	1	0	$-c_{2,x,k}^{\mathrm{TV}}$
D*	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	1	0	$-c_{2,x,k}^{\mathrm{TV}}$
F$_a^*$	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} > y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	-1	$c_{2,y,k}^{\mathrm{TV}}$
F$_b^*$	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} > y_{\mathrm{lane},0}^{\mathrm{TV}}$, $d_0 \geq y_{\mathrm{lane},0}^{\mathrm{TV}} + 0.5w_{\mathrm{lane}} + 0.5w_{\mathrm{veh}}$	0	-1	$c_{2,y,k}^{\mathrm{TV}}$
F$_2^*$	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} > y_{\mathrm{lane},0}^{\mathrm{TV}}$, $d_0 < y_{\mathrm{lane},0}^{\mathrm{TV}} + 0.5w_{\mathrm{lane}} + 0.5w_{\mathrm{veh}}$	1	0	$-c_{2,x,k}^{\mathrm{TV}}$
H$_a^*$	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} < y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	1	$-c_{3,y,k}^{\mathrm{TV}}$
H$_b^*$	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} < y_{\mathrm{lane},0}^{\mathrm{TV}}$, $d_0 \leq y_{\mathrm{lane},0}^{\mathrm{TV}} - 0.5w_{\mathrm{lane}} - 0.5w_{\mathrm{veh}}$	0	1	$-c_{3,y,k}^{\mathrm{TV}}$
H$_2^*$	$0 \leq -\left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} < y_{\mathrm{lane},0}^{\mathrm{TV}}$, $d_0 > y_{\mathrm{lane},0}^{\mathrm{TV}} - 0.5w_{\mathrm{lane}} - 0.5w_{\mathrm{veh}}$	1	0	$-c_{2,x,k}^{\mathrm{TV}}$
J$_{\mathrm{S}}^*$	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	-1	0	$c_{1,x,k}^{\mathrm{TV}}$
J$_{\mathrm{L}}^*$	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	1	$-c_{4,y,k}^{\mathrm{TV}}$
J$_{\mathrm{R}}^*$	$0 < \left(\Delta x_0^{\mathrm{EV,TV}}\right) \leq \rho_{\mathrm{close}}^{\mathrm{FTP}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	0	-1	$-c_{1,y,k}^{\mathrm{TV}}$
C$_{\mathrm{S}}^*$	$\rho_{\mathrm{close}}^{\mathrm{FTP}} < \left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	-1	0	$c_{1,x,k}^{\mathrm{TV}}$
C$_{\mathrm{L}}^*$	$\rho_{\mathrm{close}}^{\mathrm{FTP}} < \left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	$\frac{c_{3,y,0}^{\mathrm{EV}} - c_{1,y,k}^{\mathrm{TV}}}{c_{3,x,0}^{\mathrm{EV}} - c_{1,x,k}^{\mathrm{TV}}}$	-1	$c_{1,y,k}^{\mathrm{TV}} - s_{x,k} c_{3,x,0}^{\mathrm{EV}}$
C$_{\mathrm{L,lim}}^*$	$\rho_{\mathrm{close}}^{\mathrm{FTP}} < \left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	-1	0	$c_{1,x,k}^{\mathrm{TV}}$
C$_{\mathrm{R}}^*$	$\rho_{\mathrm{close}}^{\mathrm{FTP}} < \left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	$-\frac{c_{2,y,0}^{\mathrm{EV}} - c_{4,y,k}^{\mathrm{TV}}}{c_{2,x,0}^{\mathrm{EV}} - c_{4,x,k}^{\mathrm{TV}}}$	1	$-\left(c_{4,y,k}^{\mathrm{TV}} - s_{x,k} c_{2,x,0}^{\mathrm{EV}}\right)$
C$_{\mathrm{R,lim}}^*$	$\rho_{\mathrm{close}}^{\mathrm{FTP}} < \left(\Delta x_0^{\mathrm{EV,TV}}\right) < \rho_{\mathrm{lar}}$	$y_{\mathrm{lane},0}^{\mathrm{EV}} = y_{\mathrm{lane},0}^{\mathrm{TV}}$	1	0	$-c_{4,x,k}^{\mathrm{TV}}$

Table C.2: FTP constraints.

Details on Minimizing Constraint Violation Probability in MPC

D

This appendix chapter provides additional information for Chapter 6, which focuses on minimizing constraint violation probability in MPC for systems subject to uncertainty.

D.1 Proof of Lemma 6.1

Proof. According to Assumption 6.6, $f_{\boldsymbol{W}_t}$ is symmetric and unimodal, and therefore $f_{\boldsymbol{W}_t}$ is decreasing with increasing $\|\boldsymbol{w}_t\|_2$, i.e., the larger $\|\boldsymbol{w}_t\|_2$, the lower its PDF value. The uncertainty realization with the highest relative likelihood is the mode of $f_{\boldsymbol{W}_t}$ with $\boldsymbol{w}_t = \boldsymbol{0}$, yielding the most likely random output $\boldsymbol{y}^{\mathrm{DO}}_{t+1} = \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}$. It follows that

$$f_{\boldsymbol{W}_t}\left(\tilde{\boldsymbol{w}}_t\right) < f_{\boldsymbol{W}_t}\left(\boldsymbol{w}_t\right) \quad \text{for} \quad \|\tilde{\boldsymbol{w}}_t\|_2 > \|\boldsymbol{w}_t\|_2 \tag{D.1}$$

where $\tilde{\boldsymbol{y}}^{\mathrm{DO}}_{t+1} = \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1} + \tilde{\boldsymbol{w}}_t$ is less likely than $\boldsymbol{y}^{\mathrm{DO}}_{t+1} = \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1} + \boldsymbol{w}_t$ and

$$\left\|\overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1} - \tilde{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}\right\|_2 > \left\|\overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1} - \boldsymbol{y}^{\mathrm{DO}}_{t+1}\right\|_2 \tag{D.2}$$

due to $\|\tilde{\boldsymbol{w}}_t\|_2 > \|\boldsymbol{w}_t\|_2$.

It follows that the larger $\left\|\boldsymbol{y}_{t+1} - \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}\right\|_2$, the higher the PDF value of a large $\left\|\boldsymbol{y}_{t+1} - \boldsymbol{y}^{\mathrm{DO}}_{t+1}\right\|_2$ due to (D.1). Therefore, the larger the norm $\left\|\boldsymbol{y}_{t+1} - \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}\right\|_2$, the higher the PDF value of $\left\|\boldsymbol{y}_{t+1} - \boldsymbol{y}^{\mathrm{DO}}_{t+1}\right\|_2 \geq h_t$. This results in

$$\left\|\tilde{\boldsymbol{y}}_{t+1} - \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}\right\|_2 > \left\|\boldsymbol{y}_{t+1} - \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}\right\|_2 \quad \Leftrightarrow \quad p_{t+1}\left(\tilde{\boldsymbol{u}}_t, \boldsymbol{y}^{\mathrm{DO}}_t\right) \leq p_{t+1}\left(\boldsymbol{u}_t, \boldsymbol{y}^{\mathrm{DO}}_t\right) \tag{D.3}$$

with $\tilde{\boldsymbol{y}}_{t+1} \neq \boldsymbol{y}_{t+1}$ and $\tilde{\boldsymbol{y}}_{t+1} = \boldsymbol{C}\left(\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\tilde{\boldsymbol{u}}_t\right)$ according to (6.1).

The same holds for $\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}^{\mathrm{DO}}_t\right\|_2$ instead of $\left\|\boldsymbol{y}_{t+1} - \overline{\boldsymbol{y}}^{\mathrm{DO}}_{t+1}\right\|_2$, showing that p_t is decreasing with increasing $\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}^{\mathrm{DO}}_t\right\|_2$. □

D.2 Derivation of Equation (6.31)

For (6.31), in the following, it is shown that $\left\|\boldsymbol{y}_k - \boldsymbol{y}^{\mathrm{DO}}_k\right\|_2 \geq h_k$ holds if $\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}^{\mathrm{DO}}_k\right\|_2 \geq h_k + w_{\max,k-1}$. From (6.2) it follows that

$$\left\|\boldsymbol{y}_k - \boldsymbol{y}^{\mathrm{DO}}_k\right\|_2 = \left\|\boldsymbol{y}_k - \left(\overline{\boldsymbol{y}}^{\mathrm{DO}}_k + \boldsymbol{w}_{k-1}\right)\right\|_2 \geq h_k. \tag{D.4}$$

Using the reverse triangle inequality yields

$$\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}} - \boldsymbol{w}_{k-1}\right\|_2 \geq \left|\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 - \left\|\boldsymbol{w}_{k-1}\right\|_2\right| \geq h_k \tag{D.5}$$

with

$$\left|\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 - \left\|\boldsymbol{w}_{k-1}\right\|_2\right| \geq \left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 - \left\|\boldsymbol{w}_{k-1}\right\|_2 . \tag{D.6}$$

Given (6.3) it follows that

$$\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 - \left\|\boldsymbol{w}_{k-1}\right\|_2 \geq \left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 - w_{\mathrm{max},k-1} \tag{D.7}$$

for all $\left\|\boldsymbol{w}_{k-1}\right\|_2 \leq w_{\mathrm{max},k-1}$. Therefore, if

$$\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 - w_{\mathrm{max},k-1} \geq h_k, \tag{D.8}$$

is fulfilled, (D.4) holds, i.e.,

$$\left\|\boldsymbol{y}_k - \overline{\boldsymbol{y}}_k^{\mathrm{DO}}\right\|_2 \geq h_k + w_{\mathrm{max},k-1} \Rightarrow \left\|\boldsymbol{y}_k - \boldsymbol{y}_k^{\mathrm{DO}}\right\|_2 \geq h_k. \tag{D.9}$$

Equation (6.31) in Section 6.2.2 is obtained for $k = 1$.

D.3 Collision Probability Function

Here the collision probability $p_{\mathrm{col},t}$ is described in detail, which is only needed for the evaluation of the simulation but not the proposed method. The PDF $f_{\boldsymbol{W}_t}$ is chosen to be

$$f_{\boldsymbol{W}_t}\left(\tau_t\right) = \begin{cases} \frac{1}{\sigma z \sqrt{2\pi}} e^{-\frac{\tau_t^2}{2\sigma^2}} & \text{if } 0 \leq \tau_t \leq w_{\mathrm{max},t}, \\ 0 & \text{otherwise} \end{cases} \tag{D.10}$$

where τ_t is used instead of $\boldsymbol{w}_t$ and

$$\operatorname{supp}\left(f_{\boldsymbol{W}_t}\right) = \left\{\tau_t \mid 0 \leq \tau_t \leq w_{\mathrm{max},t}\right\} \tag{D.11}$$

with variance $\sigma = 1$ and

$$z = \Phi\left(w_{\mathrm{max},t}\right) - \Phi\left(0\right), \tag{D.12}$$

$$\Phi\left(\tau\right) = 0.5\left(1 + \operatorname{erf}\left(\frac{\tau}{\sqrt{2}}\right)\right), \tag{D.13}$$

such that

$$\int\limits_{\operatorname{supp}\left(f_{\boldsymbol{W}_t}\right)} f_{\boldsymbol{W}_t}\left(\tau_t\right) \mathrm{d}\tau_t = 1. \tag{D.14}$$

As the main aim of this simulation is to minimize the constraint violation probability, i.e., the collision probability, an expression for this probability is necessary in order to analyze the simulation results. The controlled vehicle and the obstacle collide if their bodies overlap, i.e., $r_{\mathrm{comb}} > \left\|\boldsymbol{y}_t - \boldsymbol{y}_t^{\mathrm{DO}}\right\|_2$ with the combined radius $r_{\mathrm{comb}} = r_{\mathrm{c}} + r_{\mathrm{DO}}$. A collision at step t is

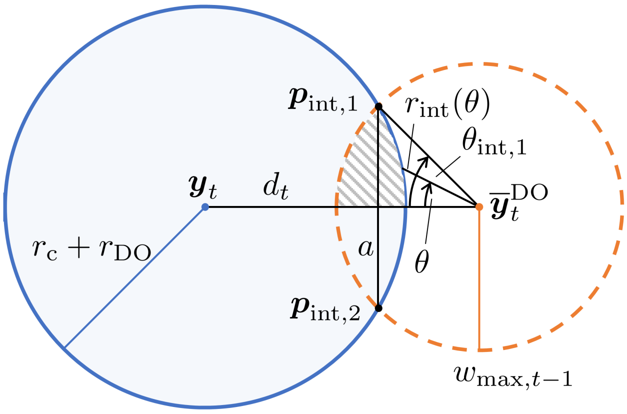

$\boldsymbol{p}_{\text{int},1}$
$r_{\text{int}}(\theta)$
$\theta_{\text{int},1}$
$\boldsymbol{y}_t$
d_t
$\overline{\boldsymbol{y}}_t^{\text{DO}}$
$r_{\text{c}} + r_{\text{DO}}$
a
θ
$\boldsymbol{p}_{\text{int},2}$
$w_{\max,t-1}$

Figure D.1: Collision probability calculation. The blue circle combines the radius of the controlled vehicle and the obstacle, the dashed orange circle represents the area potentially covered by the uncertainty. The striped area represents one half of the intersection between the two circles.

inevitable, if $\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\text{DO}}\right\|_2 + w_{\max,t-1} < r_{\text{comb}}$, i.e., even for the best-case $w_{\max,t-1}$ the objects will collide at step t. For $\left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\text{DO}}\right\|_2 - w_{\max,t-1} \geq r_{\text{comb}}$ it follows that $p_{\text{col},t} = 0$.

The collision probability is calculated according to the following procedure. We consider a circle where the radius is the required distance r_{comb} and a circle with radius τ_t. The intersection of both circles can be interpreted as the collision probability, by integrating the intersection area of both circles, weighted with $f_{\boldsymbol{W}_t}(\tau_t)$. This is illustrated in Figure D.1. In case that there is no intersection area, then $p_{\text{col},t} = 0$. If an intersection exists, there are two intersection points. The intersection area is therefore bounded on one side by the arc with radius r_{comb} and on the other side by the arc of the boundary of the uncertainty. As the intersection area is symmetric, it is sufficient to derive the calculation for one half, i.e., the area between the line connecting $\overline{\boldsymbol{y}}_t^{\text{DO}}$ and $\boldsymbol{y}_t$ and the intersection point $\boldsymbol{p}_{\text{int},1}$ as depicted by the striped area in Figure D.1. This yields an angle $\theta_{\text{int},1} \in [0; 0.5\pi]$ between the two lines connecting $\overline{\boldsymbol{y}}_t^{\text{DO}}$ and $\boldsymbol{y}_t$ as well as $\overline{\boldsymbol{y}}_t^{\text{DO}}$ and $\boldsymbol{p}_{\text{int},1}$. The distance $r_{\text{int}}(\theta)$ between $\overline{\boldsymbol{y}}_t^{\text{DO}}$ and the controlled vehicle boundary between the two intersection points follows from the law of cosines

$$r_{\text{comb}}{}^2 = r_{\text{int}}(\theta)^2 + d_t^2 - 2 d_t r_{\text{int}}(\theta)\cos(\theta) \tag{D.15}$$

where $d_t = \left\|\boldsymbol{y}_t - \overline{\boldsymbol{y}}_t^{\text{DO}}\right\|_2$ and $\theta \in [0; \theta_{\text{int},1}]$ with

$$\theta_{\text{int},1} = \sin^{-1}\left(\frac{a_\theta}{2 w_{\max,t-1}}\right), \tag{D.16}$$

$$a_\theta = \frac{\sqrt{4 d_t^2 w_{\max,t-1}{}^2 - \left(d_t^2 - r_{\text{comb}}{}^2 + w_{\max,t-1}{}^2\right)^2}}{d_t}. \tag{D.17}$$

This yields

$$r_{\text{int}}(\theta) = 0.5\left(2 d_t \cos(\theta) - \sqrt{\left(2 d_t \cos(\theta)\right)^2 - 4\left(d_t^2 - r_{\text{comb}}{}^2\right)}\right). \tag{D.18}$$

The intersection area on both sides of the line between $\boldsymbol{y}_t$ and $\overline{\boldsymbol{y}}_t^{\mathrm{DO}}$, weighted with the PDF $f_{\boldsymbol{W}_{t,\mathrm{pol}}}$, yields the collision probability

$$p_{\mathrm{col},t} = 2 \int\limits_{0}^{\theta_{\mathrm{int},1}} \frac{1}{2\pi} \int\limits_{r_{\mathrm{int}}(\theta)}^{w_{\mathrm{max},t-1}} f_{\boldsymbol{W}_t}\left(\tau_t\right) \mathrm{d}\tau_t \mathrm{d}\theta \tag{D.19}$$

for $d_t + w_{\mathrm{max},t-1} \geq r_{\mathrm{comb}}$, depending on the angle $\theta_{\mathrm{int},1}$.

This yields the overall collision probability

$$p_{\mathrm{col},t} = \begin{cases} 1 & \text{if } d_t + w_{\mathrm{max},t-1} < r_{\mathrm{comb}}, \\ 0 & \text{if } d_t - w_{\mathrm{max},t-1} \geq r_{\mathrm{comb}}, \\ \text{(D.19)} & \text{otherwise.} \end{cases} \tag{D.20}$$

For reasons of readability, the dependence on t for $\boldsymbol{p}_{\mathrm{int}}$, $\theta_{\mathrm{int},1}$, r_{int}, and a_θ is omitted.

D.4 Proof of Lemma 6.4

The following proof is based on [19].

Proof. The set $\underline{\mathcal{U}}_{\mathcal{X},N_{\mathrm{c}}-1}$ consists of all input sequences $\underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1}$ that yield constraint satisfaction of $\underline{\boldsymbol{x}}_{N_{\mathrm{c}}} \in \mathcal{X}^{N_{\mathrm{c}}}$, given the uncertainty $\underline{\boldsymbol{w}}_{N_{\mathrm{c}}-1} \in \mathcal{W}^{N_{\mathrm{c}}}$, with $\mathcal{W}^{N_{\mathrm{c}}} = \mathcal{W} \times \dots \times \mathcal{W}$. It follows that

$$\begin{aligned}
\underline{\mathcal{U}}_{\mathcal{X},N_{\mathrm{c}}-1} =& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \underline{\boldsymbol{x}}_{N_{\mathrm{c}}} \in \mathcal{X}^{N_{\mathrm{c}}} \quad \forall \underline{\boldsymbol{w}}_{N_{\mathrm{c}}-1} \in \mathcal{W}^{N_{\mathrm{c}}} \right\} && \text{(D.21a)} \\
=& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0 + \underline{\boldsymbol{B}}_{N_{\mathrm{c}}} \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} + \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \underline{\boldsymbol{w}}_{N_{\mathrm{c}}-1} \in \mathcal{X}^{N_{\mathrm{c}}} \quad \forall \underline{\boldsymbol{w}}_{N_{\mathrm{c}}-1} \in \mathcal{W}^{N_{\mathrm{c}}} \right\} && \text{(D.21b)} \\
=& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0 + \underline{\boldsymbol{B}}_{N_{\mathrm{c}}} \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \in \mathcal{X}^{N_{\mathrm{c}}} \ominus \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \circ \mathcal{W}^{N_{\mathrm{c}}} \right\} && \text{(D.21c)} \\
=& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \boldsymbol{z} = \underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0 + \underline{\boldsymbol{B}}_{N_{\mathrm{c}}} \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \boldsymbol{z} \in \mathcal{X}^{N_{\mathrm{c}}} \ominus \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \circ \mathcal{W}^{N_{\mathrm{c}}} \right\} && \text{(D.21d)} \\
=& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \underline{\boldsymbol{B}}_{N_{\mathrm{c}}} \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} = \boldsymbol{z} - \underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0 \boldsymbol{z} \in \mathcal{X}^{N_{\mathrm{c}}} \ominus \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \circ \mathcal{W}^{N_{\mathrm{c}}} \right\} && \text{(D.21e)} \\
=& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \underline{\boldsymbol{B}}_{N_{\mathrm{c}}} \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \in \mathcal{X}^{N_{\mathrm{c}}} \ominus \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \circ \mathcal{W}^{N_{\mathrm{c}}} \oplus \{-\underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0\} \right\} && \text{(D.21f)} \\
=& \left\{ \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \,\middle|\, \underline{\boldsymbol{u}}_{N_{\mathrm{c}}-1} \in \left(\mathcal{X}^{N_{\mathrm{c}}} \ominus \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \circ \mathcal{W}^{N_{\mathrm{c}}} \oplus \{-\underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0\} \right) \circ \underline{\boldsymbol{B}}_{N_{\mathrm{c}}} \right\} && \text{(D.21g)}
\end{aligned}$$

and, therefore,

$$\underline{\mathcal{U}}_{\mathcal{X},N_{\mathrm{c}}-1} = \left(\mathcal{X}^{N_{\mathrm{c}}} \ominus \underline{\boldsymbol{E}}_{N_{\mathrm{c}}} \circ \mathcal{W}^{N_{\mathrm{c}}} \oplus \{-\underline{\boldsymbol{A}}_{N_{\mathrm{c}}} \boldsymbol{x}_0\} \right) \circ \underline{\boldsymbol{B}}_{N_{\mathrm{c}}}. \tag{D.22}$$

□

D.5 Proof of Lemma 6.8

First, some preparations are introduced before starting the proof. We use system (6.84) for $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t(\boldsymbol{x}_t), \boldsymbol{w}_t)$ where the applied input $\boldsymbol{u}_t(\boldsymbol{x}_t)$ is a control law, depending on $\boldsymbol{x}_t$. Therefore, we can write $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t(\boldsymbol{x}_t), \boldsymbol{w}_t)$ as $\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}_t)$. The optimal input $\boldsymbol{u}_t^*(\boldsymbol{x}_t)$ is obtained by the MPC optimal control problem (6.83). Based on (6.66), we abbreviate the stage cost by $l(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{x}_t^\top \boldsymbol{Q} \boldsymbol{x}_t + \boldsymbol{u}_t^\top \boldsymbol{R} \boldsymbol{u}_t$ and the terminal cost by $V_{\mathrm{f}}(\boldsymbol{x}_t) = \boldsymbol{x}_t^\top \boldsymbol{Q}_{\mathrm{t}} \boldsymbol{x}_t$.

We define $\overline{\boldsymbol{x}}_{t|t} = \boldsymbol{x}_{t|t}$, yielding

$$\overline{\boldsymbol{x}}_{t+k|t} = \boldsymbol{A}\overline{\boldsymbol{x}}_{t+k-1|t} + \boldsymbol{B}\boldsymbol{u}_{t+k-1|t} \tag{D.23a}$$

$$\overline{\boldsymbol{x}}^*_{t+k|t} = \boldsymbol{A}\overline{\boldsymbol{x}}^*_{t+k-1|t} + \boldsymbol{B}\boldsymbol{u}^*_{t+k-1|t}, \quad \overline{\boldsymbol{x}}^*_{t|t} = \overline{\boldsymbol{x}}_{t|t}. \tag{D.23b}$$

We obtain $\overline{\boldsymbol{x}}_{t+1|t+1} = \boldsymbol{A}\overline{\boldsymbol{x}}_{t|t} + \boldsymbol{B}\boldsymbol{u}_{t|t} + \boldsymbol{E}\boldsymbol{w}_t$, depending on the uncertainty $\boldsymbol{w}_t$ at time step $t = t$. We then define $\overline{\boldsymbol{x}}_{t+1|t+1} = \boldsymbol{x}_{t+1|t+1}$, which allows us to write

$$\overline{\boldsymbol{x}}_{t+k|t+1} = \boldsymbol{A}^{k-1}\left(\boldsymbol{A}\overline{\boldsymbol{x}}_{t|t} + \boldsymbol{E}\boldsymbol{w}_t\right) + \sum_{j=0}^{k-1} \boldsymbol{A}^{(k-1)-j}\boldsymbol{B}\boldsymbol{u}_{t+j|t} \tag{D.24}$$

for all $k \geq 1$. It follows that for $\underline{\boldsymbol{u}}^*_{N-1|0}$ it holds that

$$\overline{\boldsymbol{x}}_{t+k|t+1} = \boldsymbol{A}^k\overline{\boldsymbol{x}}^*_{t|t} + \sum_{j=0}^{k-1} \boldsymbol{A}^{(k-1)-j}\boldsymbol{B}\boldsymbol{u}^*_{t+j|t} + \boldsymbol{A}^{k-1}\boldsymbol{E}\boldsymbol{w}_t \tag{D.25a}$$

$$= \overline{\boldsymbol{x}}^*_{t+k|t} + \boldsymbol{A}^{k-1}\boldsymbol{E}\boldsymbol{w}_t. \tag{D.25b}$$

We now prove Lemma 6.8.

Proof. For $\boldsymbol{x}_t \in \mathcal{X}_{\text{caseS}}$ with Assumption 6.14, $\mathcal{X}_{\text{caseS}}$ is robust positively invariant according to Lemma 6.5.

We consider $V(\boldsymbol{x}_t) = J^*(\boldsymbol{x}_t)$ where $J^*(\boldsymbol{x}_t)$ is the optimal cost based on (6.66) with the feedback law $\underline{\boldsymbol{u}}_{N-1|t}(\boldsymbol{x}_t)$ obtained according to (6.83) where $\mathcal{U}_{\text{cvpm}}$ follows from the the safe case. As V is continuous, positive definite, and radially unbounded based on Assumption 6.15 and 6.16, $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ exist such that $\alpha_1(|\boldsymbol{x}_t|) \leq V(\boldsymbol{x}_t) \leq \alpha_2(|\boldsymbol{x}_t|)$ is fulfilled [99]. Additionally, V is Lipschitz continuous on $\mathcal{X}_{\text{caseS}}$ as V only consists of quadratic terms and $\mathcal{X}_{\text{caseS}}$ is bounded.

Next, we prove the descent property (2.21). Based on $\underline{\boldsymbol{u}}^*_{N-1|t} = \left(\boldsymbol{u}^*_{t|t}, ..., \boldsymbol{u}^*_{t+N-1|t}\right)$ we obtain

$$J^*(\boldsymbol{x}_t) = l\left(\overline{\boldsymbol{x}}_{t|t}, \boldsymbol{u}^*_{t|t}\right) + \sum_{k=1}^{N-1} \left(l\left(\overline{\boldsymbol{x}}^*_{t+k|t}, \boldsymbol{u}^*_{t+k|t}\right)\right) + V_{\text{f}}(\overline{\boldsymbol{x}}^*_{t+N|t}) \tag{D.26a}$$

$$= l\left(\overline{\boldsymbol{x}}_{t|t}, \boldsymbol{u}^*_{t|t}\right) + q\left(\overline{\boldsymbol{x}}^*_{t+1|t}\right) \tag{D.26b}$$

where $q\left(\overline{\boldsymbol{x}}^*_{t+1|t}\right)$ summarizes the total cost starting at $\overline{\boldsymbol{x}}^*_{t+1|t}$, which is used similarly in [38]. For $t+1$ with $\underline{\boldsymbol{u}}^*_{t+N|t+1} = \left(\boldsymbol{u}^*_{t+1|t+1}, ..., \boldsymbol{u}^*_{t+N|t+1}\right)$ the optimal cost is

$$J^*(\boldsymbol{x}_{t+1}) = \sum_{k=1}^{N-1} \left(l\left(\overline{\boldsymbol{x}}_{t+k|t+1}, \boldsymbol{u}^*_{t+k|t+1}\right)\right) + l\left(\overline{\boldsymbol{x}}_{t+N|t+1}, \boldsymbol{u}^*_{t+N|t+1}\right) + V_{\text{f}}(\overline{\boldsymbol{x}}^*_{t+N+1|t+1}). \tag{D.27}$$

With the non-optimal input sequence $\left(\boldsymbol{u}^*_{t+1|t}, ..., \boldsymbol{u}^*_{t+N-1|t}\right)$, we obtain

$$J^*(\boldsymbol{x}_{t+1}) \leq \sum_{k=1}^{N-1} \left(l\left(\overline{\boldsymbol{x}}_{t+k|t+1}, \boldsymbol{u}^*_{t+k|t}\right)\right) + V_{\text{f}}(\overline{\boldsymbol{x}}_{t+N|t+1}) \tag{D.28a}$$

$$= q\left(\overline{\boldsymbol{x}}_{t+1|t+1}\right) \tag{D.28b}$$

where $\overline{\boldsymbol{x}}_{t+k|t+1} = \overline{\boldsymbol{x}}^*_{t+k|t} + \boldsymbol{A}^{k-1}\boldsymbol{G}\boldsymbol{w}_t$ according to (D.25) and $q\left(\overline{\boldsymbol{x}}_{t+1|t+1}\right) = q\left(\overline{\boldsymbol{x}}^*_{t+1|t} + \boldsymbol{E}\boldsymbol{w}_t\right)$. Note that at $t+1$ the optimal input at prediction step k is $\boldsymbol{u}^*_{t+k|t+1}$, based on $\overline{\boldsymbol{x}}_{t+k|t+1}$, whereas $\boldsymbol{u}^*_{t+k|t}$, obtained at t, is not optimal as $\boldsymbol{x}_{t+1|t+1}$ was affected by $\boldsymbol{w}_t$.

Due to optimality, it holds that $J^*(\boldsymbol{x}_{t+1}) \leq J(\boldsymbol{x}_{t+1})$. With (D.26) and (D.28) we obtain

$$J^*(\boldsymbol{x}_{t+1}) - J^*(\boldsymbol{x}_t) \leq q\left(\overline{\boldsymbol{x}}_{t+1|t+1}\right) - l\left(\overline{\boldsymbol{x}}_{t|t}, \boldsymbol{u}^*_{t|t}\right) - q\left(\overline{\boldsymbol{x}}^*_{t+1|t}\right). \tag{D.29}$$

Due to Assumptions 6.15 and 6.16, bounded sets $\mathcal{X}_{\text{caseS}}$ and $\mathcal{U}$, and since $q(\cdot)$ is the sum of quadratic terms, $q(\cdot)$ is Lipschitz continuous, i.e.,

$$\left\| q\left(\overline{\boldsymbol{x}}^*_{t+1|t} + \boldsymbol{E}\boldsymbol{w}_t\right) - q\left(\overline{\boldsymbol{x}}^*_{t+1|t}\right)\right\| \leq L_q \left\|\overline{\boldsymbol{x}}^*_{t+1|t} + \boldsymbol{E}\boldsymbol{w}_t - \overline{\boldsymbol{x}}^*_{t+1|t}\right\| = L_q ||\boldsymbol{G}\boldsymbol{w}_t|| \tag{D.30}$$

with Lipschitz constant L_q. Given (D.29) and (D.30),

$$\begin{aligned} J^*(\boldsymbol{x}_{t+1}) - J^*(\boldsymbol{x}_t) &\leq -l\left(\overline{\boldsymbol{x}}_{t|t}, \boldsymbol{u}^*_{t|t}\right) + L_q||\boldsymbol{G}\boldsymbol{w}_t|| \\ &\leq -l\left(\overline{\boldsymbol{x}}_{t|t}, \boldsymbol{0}\right) + L_q||\boldsymbol{G}\boldsymbol{w}_t|| \\ &\leq -\alpha_3(||\boldsymbol{x}_t||) + L_q||\boldsymbol{E}\boldsymbol{w}_t||. \end{aligned} \tag{D.31}$$

It is straightforward that the previous procedure, considering $J(\boldsymbol{x}_{t+1}) - J^*(\boldsymbol{x}_t)$, holds for any $t \geq 0$. Therefore, all requirements of Definition 2.6 are fulfilled, i.e., V is an ISS Lyapunov function and the origin of system (6.84) is ISS if $\boldsymbol{x}_t \in \mathcal{X}_{\text{caseS}}$. □

D.6 Proof of Lemma 6.9

Unlike the safe case, only one input is admissible according to (6.81). Therefore, in addition to $\boldsymbol{w}_t$, this input is considered a disturbance in order to show input-to-state stability again.

Proof. We consider the combined input $\boldsymbol{w}'_t = (\boldsymbol{u}_t^\top, \boldsymbol{w}_t^\top)^\top$, the system $\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t) = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{E}'\boldsymbol{w}'_t$ with $\boldsymbol{G}' = [\boldsymbol{B}, \boldsymbol{E}]$, and the candidate Lyapunov function $V'(\boldsymbol{x}_t) = J'(\boldsymbol{x}_t)$ where

$$J'(\boldsymbol{x}_t) = \sum_{k=0}^{N-1} l'\left(\overline{\boldsymbol{x}}_{t+k}\right) + V'_{\text{f}}(\overline{\boldsymbol{x}}_{t+N}) \tag{D.32}$$

with $l'\left(\overline{\boldsymbol{x}}_{t+k}\right) = ||\overline{\boldsymbol{x}}_{t+k}||_{\boldsymbol{Q}} = \left(\overline{\boldsymbol{x}}^\top_{t+k}\boldsymbol{Q}\overline{\boldsymbol{x}}_{t+k}\right)^{\frac{1}{2}}$ and $V'_{\text{f}}(\overline{\boldsymbol{x}}_{t+N}) = ||\overline{\boldsymbol{x}}_{t+N}||_{\boldsymbol{Q}_{\text{t}}} = \left(\overline{\boldsymbol{x}}^\top_{t+N}\boldsymbol{Q}_{\text{t}}\overline{\boldsymbol{x}}_{t+N}\right)^{\frac{1}{2}}$. Note that (D.32) does not consider the input sequence (compared to (6.66)) as we treat the input sequence as a disturbance. Similar to Lemma 6.8, V' is continuous, positive definite, and radially unbounded based on Assumption 6.15 and 6.16; hence, V' fulfills requirement (2.20). As V' is a sum of norms, V' is Lipschitz continuous on $\mathbb{R}^{n_{\boldsymbol{x}}}$. Additionally, it holds that $\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t) \in \mathbb{R}^{n_{\boldsymbol{x}}}$ for all $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$, i.e., $\mathbb{R}^{n_{\boldsymbol{x}}}$ is robust positively invariant for $\tilde{\boldsymbol{f}}$.

Due to Assumption 6.12, $\boldsymbol{A}$ is stable. Hence, the undisturbed system (6.62) with $\boldsymbol{w}'_t = \boldsymbol{0}$ is asymptotically stable and it holds that V' is a Lyapunov function with

$$V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0})\right) - V'(\boldsymbol{x}_t) \leq -\alpha_3(||\boldsymbol{x}_t||). \tag{D.33}$$

As $\tilde{\boldsymbol{f}}$ is a linear function, it follows that $\tilde{\boldsymbol{f}}$ is Lipschitz continuous on $\mathbb{R}^{n_{\boldsymbol{x}}} \times \mathcal{U} \times \mathcal{W}$.

As V' and $\tilde{\boldsymbol{f}}$ are Lipschitz continuous, it holds that

$$\left\|V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t)\right) - V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0})\right)\right\| \leq L'_V \left\|\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t) - \tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0})\right\| \leq L'_V L_{\boldsymbol{f}} ||\boldsymbol{w}'_t|| \tag{D.33a}$$

for all $\boldsymbol{w}_t \in \mathcal{W}$ with respective Lipschitz constants L'_V and $L_{\boldsymbol{f}}$.

We obtain

$$\begin{aligned}
V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t)\right) - V'(\boldsymbol{x}_t) &= V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t)\right) - V'(\boldsymbol{x}_t) + \left(V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0})\right) - V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0})\right)\right) && \text{(D.34a)} \\
&= V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0})\right) - V'(\boldsymbol{x}_t) \;\; + V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{w}'_t)\right) - V'\left(\tilde{\boldsymbol{f}}(\boldsymbol{x}_t, \boldsymbol{0}))\right) && \text{(D.34b)} \\
&\leq -\alpha_3(||\boldsymbol{x}_t||) \qquad\qquad\quad + L'_V L_{\boldsymbol{f}} ||\boldsymbol{w}'_t||. && \text{(D.34c)}
\end{aligned}$$

It follows that for $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$, which includes $\boldsymbol{x}_t \notin \mathcal{X}_{\text{caseS}}$, the origin of (6.84) is ISS. □

Acronyms

BG	binary grid.
CA	controlled agent.
CDF	cumulative distribution function.
CoG	center of gravity.
CVPM	constraint violation probability minimization.
DO	dynamic obstacle.
EV	ego vehicle.
FTP	failsafe trajectory planning.
i.i.d.	independent and identically distributed.
ISS	input-to-state stable.
LC	lane changing.
LK	lane keeping.
MPC	Model Predictive Control.
NUSH	non-uniformly spaced horizon.
OCP	optimal control problem.
OG	occupancy grid.
PDF	probability density function.
PG	probabilistic grid.
RMPC	robust Model Predictive Control.
S+SC MPC	stochastic + scenario Model Predictive Control.
SCMPC	scenario Model Predictive Control.
SMPC	stochastic Model Predictive Control.
SMPC+FTP	stochastic Model Predictive Control + failsafe trajectory planning.
TV	target vehicle.
V2V	vehicle-to-vehicle.

Notation

Mathematical Conventions and Operators

Sets, Spaces, and Function Classes

$\emptyset$	empty set
$\mathbb{N}$	set of all natural numbers
$\mathbb{N}_{\geq 0}$	set of all non-negative natural numbers
$\mathbb{R}$	set of all real numbers
$\mathbb{R}_{\geq 0}$	set of all non-negative real numbers
$\mathbb{R}_{>0}$	set of all positive real numbers
$\mathbb{R}^n$	n-dimensional Euclidean space (n omitted for $n = 1$)
$\mathbb{R}^{n \times m}$	set of real-valued $n \times m$ matrices
$\{a, b, c\}$	set of elements (order irrelevant)
(a, b, c)	sequence of elements (order relevant)
$[a, b]$	closed (real) interval between a and b
$]a, b[$	open (real) interval between a and b
$\mathcal{K}$	class of positive definite and strictly increasing functions
$\mathcal{K}_\infty$	class of positive definite, strictly increasing, and unbounded functions

Scalars, Vectors, and Matrices

Scalars are indicated by standard letters (a, A), lowercase bold letters denote vectors ($\boldsymbol{a}$), and uppercase bold letters are used for matrices (A). Sets are typically written as calligraphic letters ($\mathcal{A}$). Zero vectors and zero matrices are denotes by $\mathbf{0}$, the identity matrix is denoted by $\boldsymbol{I}$, all with appropriate dimensions. Dimensions may be explicitly denoted, e.g., $\boldsymbol{I}_{m \times n} \in \mathbb{R}^{m \times n}$. The i-th element of a vector is denoted by $\boldsymbol{a}_i$. The element in the i-th row and j-th column of a matrix $\boldsymbol{A}$ is indicated by a_{ij} and the i-th row of a matrix $\boldsymbol{A}$ is denoted by $[\boldsymbol{A}]_i$. The i-th entry of a vector $\boldsymbol{a}$ may similarly be expressed by $[\boldsymbol{a}]_i$ to improve clarity.

Operators

$\mathbb{I}_{a,b}$	integer interval between (and including) a and b
$\lvert a\rvert$	absolute value of a
$\lvert\mathcal{A}\rvert$	cardinality of set $\mathcal{A}$
$\lvert\lvert\boldsymbol{a}\rvert\rvert$	norm of vector $\boldsymbol{a}$
$\lvert\lvert\boldsymbol{a}\rvert\rvert_2$	Euclidean norm of vector $\boldsymbol{a}$
$\lvert\lvert\boldsymbol{a}\rvert\rvert_p$	p-norm of vector $\boldsymbol{a}$
$\lvert\lvert\boldsymbol{a}\rvert\rvert_{\boldsymbol{A}}$	weighted norm of vector $\boldsymbol{a}$, i.e., $\lvert\lvert\boldsymbol{a}\rvert\rvert_{\boldsymbol{A}} = \sqrt{\boldsymbol{a}^\top \boldsymbol{A}\boldsymbol{a}}$
$\boldsymbol{A}^\top$	transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	inverse of a square matrix $\boldsymbol{A}$
$\boldsymbol{A} \succ 0$	symmetric, positive definite matrix $\boldsymbol{A}$
$\boldsymbol{A} \succeq 0$	symmetric, positive semi-definite matrix $\boldsymbol{A}$
$\boldsymbol{A} \prec 0$	symmetric, negative definite matrix $\boldsymbol{A}$
$\boldsymbol{A} \preceq 0$	symmetric, negative semi-definite matrix $\boldsymbol{A}$
$\mathrm{diag}(a_1, \cdots, a_n)$	diagonal $n \times n$ matrix where the i-th diagonal matrix element is equal to a_i
$\mathrm{blkdiag}(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_n)$	block-diagonal matrix where the i-th matrix block is equal to $\boldsymbol{A}_i$
$\partial\mathcal{A}$	boundary of set $\mathcal{A}$
$\mathrm{int}(\mathcal{A})$	interior of set $\mathcal{A}$
$\mathrm{conv}(\cdot)$	convex hull
$\mathrm{cov}(\cdot)$	covariance matrix
$\mathrm{proj}(\cdot)$	projection
Δ	difference, depending on respective variable and context
∇	gradient operator
$\mathcal{A} \oplus \mathcal{B}$	set addition, $\mathcal{A} \oplus \mathcal{B} := \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B}\}$
$\mathcal{A} \ominus \mathcal{B}$	set subtraction $\mathcal{A} \ominus \mathcal{B} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \{\boldsymbol{x}\} \oplus \mathcal{B} \subseteq \mathcal{A}\}$
$(\cdot)^{\leftarrow}$	sequence shift operator
$\mathrm{supp}\,(\cdot)$	support
$\exp(\cdot)$	exponential function

Stochastic

$\Pr(A)$	probability of event A
p	probability (value)
P	probability distribution (of a random variable)
$\boldsymbol{\mu}$	mean
$\boldsymbol{\sigma}$	standard deviation
σ^2	variance
$\boldsymbol{\Sigma}$	covariance matrix
f_P, f_W	probability density function (or probability mass function) related to probability distribution $\mathcal{P}$ or related to random variable W
F_P, F_W	cumulative distribution function related to probability distribution P or related to random variable W
F	cumulative distribution function
F^{-1}	quantile function
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathrm{E}(\cdot)$	expected value
χ^2_n	chi-square distribution with n degrees of freedom

Accents, Subscripts, and Superscripts

$\hat{(\cdot)}$	estimated values
$(\cdot)^*$	optimal value or solution
$\overline{(\cdot)}$	mean value
$\underline{(\cdot)}$	augmented vector or system description
$\tilde{(\cdot)}$	adapted value
$(\cdot)_{t+k\|t}$	value at prediction step $t+k$, evaluated at time step t; abbreviated to $(\cdot)_{t+k}$ if clear from context
$\boldsymbol{a}_{i,t}$	i-th element of vector $\boldsymbol{a}$ at time step t
$\boldsymbol{a}_{i,t+k\|t}$	i-th element of vector $\boldsymbol{a}$ at prediction step $t+k$, evaluated at time step t

$(\cdot)^{\mathrm{G}}$	related to a grid representation
$(\cdot)^{\mathrm{DO}}$	related to a dynamic obstacle
$(\cdot)^{\mathrm{EV}}$	related to the ego vehicle
$(\cdot)^{\mathrm{TV}}$	related to a target vehicle
$(\cdot)_{\mathrm{c}}$	related to a controlled system
$(\cdot)_{\mathrm{col}}$	collision
$(\cdot)_{\mathrm{cv}}$	constraint violation
$(\cdot)_{\mathrm{cvpm}}$	constraint violation probability minimization
$(\cdot)_{\mathrm{lb}}$	lower bound
$(\cdot)_{\mathrm{lt}}$	related to the long-term system model
$(\cdot)_{\max}$	maximal value
$(\cdot)_{\min}$	minimal value
$(\cdot)_{\mathrm{ref}}$	reference
$(\cdot)_{\mathrm{st}}$	related to the short-term system model
$(\cdot)_{\mathrm{ub}}$	upper bound

Variables and Constants

Systems, Control Theory, and Optimization

$\boldsymbol{A}$	system matrix
$\boldsymbol{A_K}$	stabilized system matrix $\boldsymbol{A_K} = \boldsymbol{A} + \boldsymbol{BK}$
$\boldsymbol{B}$	input matrix
$\boldsymbol{c}$	new input decision variable ($\boldsymbol{u} = \boldsymbol{Kx} + \boldsymbol{c}$)
$\boldsymbol{C}$	output matrix
$\boldsymbol{d}(\cdot)$	constraint function
$\boldsymbol{e}$	error
$\boldsymbol{E}$	uncertainty input matrix
$\boldsymbol{E}_{\mathrm{lt}}$	uncertainty input matrix of long-term system model

$\boldsymbol{f}(\cdot)$	system dynamics
$\boldsymbol{g}$	constraint vector
$\boldsymbol{G}$	constraint matrix
h	constraint value
$\boldsymbol{h}$	constraint vector
$\boldsymbol{H}$	constraint matrix
J	cost function
k	prediction step (within optimal control problem)
K	scenario MPC samples
$\boldsymbol{K}$	feedback matrix with individual elements k_{ij}
l	stage cost (within objective function)
n	dimension, e.g., state dimension $n_{\boldsymbol{x}}$ or input dimension $n_{\boldsymbol{u}}$
N	prediction horizon
$\mathbb{P}$	optimal control problem
$\boldsymbol{Q}$	state weighting matrix
$\boldsymbol{Q}_\mathrm{f}$	terminal state weighting matrix
$\boldsymbol{R}$	input weighting matrix
$\mathcal{R}$	reachset
$\boldsymbol{S}$	input rate weighting matrix
$\mathcal{S}_\infty$	minimal disturbance invariant set
t	time or (current) time step
Δt	sampling time
$\boldsymbol{u}$	control input
$\boldsymbol{U}$	sequence of control inputs
$\mathcal{U}$	set of admissible control inputs
$\mathcal{U}_{\boldsymbol{\nu}}$	set of admissible control inputs of long-term system model
V	value function
V_f	terminal cost function

$\boldsymbol{w}$	system uncertainty
$\boldsymbol{w}^{\mathrm{meas}}$	measurement uncertainty, noise
$\boldsymbol{w}^{\mathrm{ex}}$	(task) execution uncertainty
$\boldsymbol{w}^{\mathrm{ta}}$	task uncertainty
$\mathcal{W}$	set of possible uncertainty realizations
$\boldsymbol{x}$	system state
$\boldsymbol{X}$	sequence of states
$\mathcal{X}$	set of admissible system states
$\mathcal{X}_{\mathrm{f}}$	set of admissible terminal system states
$\mathcal{X}^{\mathrm{safe}}$	set of safe states
$\mathcal{X}^{\mathrm{adm}}$	set of admissible states
$\boldsymbol{y}$	system output
$\boldsymbol{z}$	deterministic part of system state ($\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{e}$)
β	chance constraint risk parameter
$\tilde{\beta}$	alternative chance constraint risk parameter ($\tilde{\beta} = 1 - \beta$)
$\boldsymbol{\gamma}$	constraint tightening
$\boldsymbol{\Gamma}$	set of safe feasible trajectories
$\boldsymbol{\delta}$	system uncertainty of long-term system model
$\boldsymbol{\epsilon}$	error of long-term system model
$\boldsymbol{\zeta}$	deterministic part of system state of long-term system model ($\boldsymbol{\xi} = \boldsymbol{\zeta} + \boldsymbol{\epsilon}$)
$\boldsymbol{\eta}$	new input decision variable of long-term system model ($\boldsymbol{\nu} = \boldsymbol{K}_{\mathrm{c}}\boldsymbol{\xi} + \boldsymbol{\eta}$)
$\varkappa$	tolerance level
λ	slack variable
Λ	horizon length for multi-step CVPM-MPC method
$\boldsymbol{\nu}$	control input of long-term system model
$\boldsymbol{\xi}$	system state of long-term system model
$\boldsymbol{\Xi}$	set of admissible system states of long-term system model

Ξ_f	set of admissible terminal system states for the long-term system model
$\boldsymbol{\pi}$	policies
$\varrho(\cdot)$	substitute probability function
$\boldsymbol{\chi}^{\boldsymbol{U}}_{\boldsymbol{x}_t}$	state sequence resulting from the initial state $\boldsymbol{x}_t$ and the input sequence $\boldsymbol{U}$
ω	samples
$\boldsymbol{\Omega}$	set of samples

Vehicles and Robotics

a	acceleration
a_x	acceleration in x-direction
a_x	acceleration in y-direction
a_r	rectangle length or ellipse semi-major axis
b_r	rectangle width or ellipse semi-minor axis
$\mathcal{B}^\mathrm{G}$	binary grid
c^G	cell in a grid
$\mathcal{C}^\mathrm{G}$	grid
d	distance or distance constraint
d_safe	safety distance
d^ra	lateral displacement in road-aligned frame
F_x	force in x-direction
F_y	force in y-direction
l_f	distances from the vehicle CoG to the front axle
l_r	distances from the vehicle CoG to the rear axle
l_x	cell length
l_y	cell width
l_veh	vehicle length

m	mass
M	maneuver
$\mathcal{M}$	maneuver set
N_m	number of maneuvers
N_v	number of vehicles
p_th	probability threshold value
$\mathcal{P}^\mathrm{G}$	probability grid
r	radius
s^ra	longitudinal position in road-aligned frame
T	task
$\mathcal{T}$	task set
v	velocity
v_x	velocity in x-direction
v_y	velocity in y-direction
w_lane	lane width
w_veh	vehicle width
x	x-position
y	y-position
α	steering variable
δ_f	steering angle (front)
θ	angle
ι_safe	safety margin
κ	road curvature
ρ	distance measure
ψ	heading angle
ψ^ra	orientation offset in road-aligned frame

List of Figures

List of Tables

Bibliography

[1] R. Alami, A. Clodic, V. Montreuil, E. Sisbot, and R. Chatila, "Toward Human-Aware Robot Task Planning." in *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, Beijing, China, Jan. 2006, pp. 39–46.

[2] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency model predictive control for automated vehicles," in *2019 American Control Conference (ACC)*, Philadelphia, PA, USA, 2019, pp. 717–722.

[3] J. P. Alsterda and C. Gerdes, "Contingency model predictive control for linear time-varying systems," 2021, arXiv: 2102.12045.

[4] M. Althoff, D. Heß, and F. Gambert, "Road occupancy prediction of traffic participants," in *2013 IEEE 16th International Conference on Intelligent Transportation Systems (ITSC)*, The Hague, Netherlands, 2013, pp. 99–105.

[5] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, June 2017, pp. 719–726.

[6] M. Bahram, A. Wolf, M. Aeberhard, and D. Wollherr, "A prediction-based reactive driving strategy for highly automated driving function on freeways," in *2014 IEEE Intelligent Vehicles Symposium (IV)*, Dearborn, MI, USA, 2014, pp. 400–406.

[7] T. Bäthge, S. Lucia, and R. Findeisen, "Exploiting models of different granularity in robust predictive control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, USA, Dec 2016, pp. 2763–2768.

[8] H. Bauer, "Minimalstellen von funktionen und extremalpunkte," *Archiv der Mathematik*, vol. 9, no. 4, pp. 389–393, Nov 1958.

[9] A. Bemporad and M. Morari, *Robust model predictive control: A survey.* London: Springer London, 1999, pp. 207–226.

[10] A. Bemporad and D. Muñoz de la Peña, "Multiobjective model predictive control," *Automatica*, vol. 45, no. 12, pp. 2823–2830, 2009.

[11] A. Bemporad, L. Puglia, and T. Gabbriellini, "A stochastic model predictive control approach to dynamic option hedging with transaction costs," in *2011 American Control Conference (ACC)*, San Francisco, CA, USA, June 2011, pp. 3862–3867.

[12] T. Benciolini, T. Brüdigam, and M. Leibold, "Multistage stochastic model predictive control for urban automated driving," in *2021 IEEE 24th International Conference on Intelligent Transportation Systems (ITSC)*, Indianapolis, IN, USA, 2021, pp. 417–423.

[13] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, June 2012.

[14] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 6077–6082.

[15] L. Blackmore, M. Ono, A. Bektassov, and B. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, Jun. 2010.

[16] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[17] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[18] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016, arXiv: 1604.07316.

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

[20] F. Bounini, D. Gingras, H. Pollart, and D. Gruyer, "Modified artificial potential field method for online path planning applications," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 180–185.

[21] S. Boyd and L. Vandenberghe, *Convex optimization*, 1st ed. Cambridge: Cambridge University Press, 2004.

[22] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, pp. 25–30, 1965.

[23] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[24] C. Brüdigam, "Intelligente Fahrmanöver sehender autonomer Fahrzeuge in autobahnähnlicher Umgebung," Dissertation, Universität der Bundeswehr München, 1994.

[25] T. Brüdigam, "(Stochastic) model predictive control - a simulation example," 2021, arXiv: 2101.12020.

[26] T. Brüdigam, K. Ahmic, M. Leibold, and D. Wollherr, "Legible model predictive control for autonomous driving on highways," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 215–221, 2018, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018.

[27] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian process-based stochastic model predictive control for overtaking in autonomous racing," in *2021 IEEE International Conference on Robotics and Automation (ICRA) Workshop on Opportunities and Challenges with Autonomous Racing*, Xian, China, 2021.

[28] T. Brüdigam, F. di Luzio, L. Pallottino, D. Wollherr, and M. Leibold, "Grid-based stochastic model predictive control for trajectory planning in uncertain environments," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1–8.

[29] T. Brüdigam, V. Gaßmann, D. Wollherr, and M. Leibold, "Minimization of constraint violation probability in model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 14, pp. 6740–6772, 2021.

[30] T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr, "Combining stochastic and scenario model predictive control to handle target vehicle uncertainty in autonomous driving," in *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 2018, pp. 1317–1324.

[31] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving: Extended version," 2020, arXiv: 2009.09381.

[32] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2021.

[33] T. Brüdigam, D. Prader, D. Wollherr, and M. Leibold, "Model predictive control with models of different granularity and a non-uniformly spaced prediction horizon," in *2021 American Control Conference (ACC)*, New Orleans, LA, USA, 2021, pp. 3876–3881.

[34] T. Brüdigam, J. Teutsch, D. Wollherr, and M. Leibold, "Combined robust and stochastic model predictive control for models of different granularity," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7123–7129, 2020, 21st IFAC World Congress.

[35] T. Brüdigam, J. Teutsch, D. Wollherr, M. Leibold, and M. Buss, "Probabilistic model predictive control for extended prediction horizons," *at - Automatisierungstechnik*, vol. 69, no. 9, pp. 759–770, 2021.

[36] T. Brüdigam, J. Zhan, D. Wollherr, and M. Leibold, "Collision avoidance with stochastic model predictive control for systems with a twofold uncertainty structure," in *2021 IEEE 24th International Conference on Intelligent Transportation Systems (ITSC)*, Indianapolis, IN, USA, 2021, pp. 432–438.

[37] E. A. Buehler, J. A. Paulson, and A. Mesbah, "Lyapunov-based stochastic nonlinear model predictive control: Shaping the state probability distribution functions," in *2016 American Control Conference (ACC)*, 2016, pp. 5389–5394.

[38] M. Bujarbaruah, X. Zhang, M. Tanaskovic, and F. Borrelli, "Adaptive stochastic mpc under time-varying uncertainty," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2840–2845, 2021.

[39] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, “Move blocking strategies in receding horizon control,” in *2004 43rd IEEE Conference on Decision and Control (CDC)*, Nassau, Bahamas, Dec 2004, pp. 2023–2028.

[40] ——, “Move blocking strategies in receding horizon control,” *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[41] G. C. Calafiore and M. C. Campi, “The scenario approach to robust control design,” *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[42] G. Calafiore and L. Fagiano, “Robust model predictive control via scenario optimization,” *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, Jan 2013.

[43] M. C. Campi and S. Garatti, “A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality,” *Journal of optimization theory and applications*, vol. 148, no. 2, pp. 257–280, 2011.

[44] M. Cannon, B. Kouvaritakis, S. V. Rakovic, and Q. Cheng, “Stochastic tubes in model predictive control with probabilistic constraints,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 194–200, Jan 2011.

[45] M. Cannon, B. Kouvaritakis, and X. Wu, “Model predictive control for systems with stochastic multiplicative uncertainty and probabilistic constraints,” *Automatica*, vol. 45, no. 1, pp. 167–172, 2009.

[46] ——, “Probabilistic constrained mpc for multiplicative and additive stochastic uncertainty,” *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1626–1632, July 2009.

[47] A. Capone and S. Hirche, “Backstepping for partially unknown nonlinear systems using gaussian processes,” *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 416–421, 2019.

[48] ——, “Anticipating the long-term effect of online learning in control,” in *2020 American Control Conference (ACC)*, Denver, CO, USA, 2020, pp. 3865–3872.

[49] A. Carvalho, “Predictive control under uncertainty for safe autonomous driving: Integrating data-driven forecasts with control design,” Dissertation, University of California, Berkeley, 2016.

[50] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, “Stochastic predictive control of autonomous vehicles in uncertain environments,” in *12th International Symposium on Advanced Vehicle Control*, Tokyo, Japan, 2014.

[51] A. Carvalho, S. Lefèvre, G. Schildbach, J. Kong, and F. Borrelli, “Automated driving: The role of forecasts and uncertainty: A control perspective,” *European Journal of Control*, vol. 24, pp. 14–32, 2015.

[52] V. Causevic, Y. Fanger, T. Brüdigam, and S. Hirche, “Information-constrained model predictive control with application to vehicle platooning,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3124–3130, 2020, 21st IFAC World Congress.

[53] 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, Fall 2017.

[54] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, “A review of motion planning for highway autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2020.

[55] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, “Bayesian occupancy filtering for multi-target tracking: an automotive application,” *International Journal of Robotic Research*, vol. 25, 2006.

[56] C. R. Cutler and B. L. Ramaker, “Dynamic matrix control - a computer control algorithm,” *IEEE Transactions on Automatic Control*, vol. 17, p. 72, 1979.

[57] S. di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, “Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2014.

[58] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen, “The seeing passenger car ‘vamors-p’,” in *Proceedings of the Intelligent Vehicles ’94 Symposium*, 1994, pp. 68–73.

[59] E. D. Dickmanns and A. Zapp, “Autonomous high speed road vehicle guidance by computer vision,” *IFAC Proceedings Volumes*, vol. 20, no. 5, Part 4, pp. 221–226, 1987, 10th Triennial IFAC Congress on Automatic Control - 1987 Volume IV, Munich, Germany, 27-31 July.

[60] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, p. 269–271, Dec 1959.

[61] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah, “Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2310–2323, 2020.

[62] A. Domahidi and J. Jerez, “Forces professional,” Embotech AG, url=https://embotech.com/FORCES-Pro, 2014-2019.

[63] A. Dragan and S. Srinivasa, “Generating legible motion,” in *Proceedings of the Robotics: Science and Systems IX*, Berlin, Germany, June 2013.

[64] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa, “Legibility and predictability of robot motion,” in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Pittsburg, PA, USA, March 2013, pp. 301–308.

[65] L. Fagiano and M. Khammash, “Nonlinear stochastic model predictive control via regularized polynomial chaos expansions,” in *2012 IEEE 51st Conference on Decision and Control (CDC)*, Maui, HI, USA, 2012, pp. 142–147.

[66] M. Farina, L. Giulioni, L. Magni, and R. Scattolini, “A probabilistic approach to model predictive control,” in *2013 52nd IEEE Conference on Decision and Control (CDC)*, Firenze, Italy, 2013, pp. 7734–7739.

[67] ——, “An approach to output-feedback mpc of stochastic linear discrete-time systems,” *Automatica*, vol. 55, pp. 140–149, 2015.

[68] M. Farina, L. Giulioni, and R. Scattolini, “Stochastic linear model predictive control with chance constraints – a review,” *Journal of Process Control*, vol. 44, pp. 53–67, 2016.

[69] M. Farina, X. Zhang, and R. Scattolini, “A hierarchical multi-rate mpc scheme for interconnected systems,” *Automatica*, vol. 90, pp. 38–46, 2018.

[70] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.

[71] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, “A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles,” *Vehicle System Dynamics*, vol. 52, no. 6, pp. 802–823, 2014.

[72] M. Geisslinger, F. Poszler, J. Betz, C. Lütge, and M. Lienkamp, “Autonomous driving ethics: from trolley problem to ethicsof risk,” *Philosophy & Technology*, pp. 1–23, 2021.

[73] M. Giftthaler, M. Neunert, M. Stäuble, and J. Buchli, “The Control Toolbox - an open-source C++ library for robotics, optimal and model predictive control,” in *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Brisbane, QLD, Australia, May 2018, pp. 123–129.

[74] J. H. Gillula, “Guaranteeing safe online machine learning via reachability analysis,” Dissertation, Stanford University, 2013.

[75] R. Gondhalekar and J. Imura, “Least-restrictive move-blocking model predictive control,” *Automatica*, vol. 46, no. 7, pp. 1234–1240, 2010.

[76] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, “Optimization over state feedback policies for robust control with constraints,” *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.

[77] A. Grandia, A. Taylor, A. D. Ames, and M. Hutter, “Multi-layered safety for legged robots via control barrier functions and model predictive control,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China, 2021, pp. 8352–8358.

[78] L. Grüne, “Nmpc without terminal constraints,” *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 1–13, 2012, 4th IFAC Conference on Nonlinear Model Predictive Control.

[79] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*, 2nd ed. Cham: Springer-Verlag, 2017.

[80] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[81] B. Gutjahr, L. Gröll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying mpc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, June 2017.

[82] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[83] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *2013 European control conference (ECC)*, Zürich, Switzerland, Jul. 2013, pp. 502–510.

[84] F. Herzog, G. Dondi, and H. P. Geering, "Stochastic model predictive control and portfolio optimization," *International Journal of Theoretical and Applied Finance*, vol. 10, no. 02, pp. 203–233, 2007.

[85] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for autonomous miniature race cars," in *2018 European Control Conference (ECC)*, Limassol, Cyprus, 2018, pp. 1341–1348.

[86] L. Hewing, K. P. Wabersich, and M. N. Zeilinger, "Recursively feasible stochastic model predictive control using indirect feedback," *Automatica*, vol. 119, p. 109095, 2020.

[87] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 5–17, 2018.

[88] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using gaussian processes," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Jeju, South Korea, 2020, pp. 3699–3704.

[89] C. Jewison, R. S. Erwin, and A. Saenz-Otero, "Model predictive control with ellipsoid obstacle constraints for spacecraft rendezvous," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 257–262, 2015, 1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles ACNAAV'15.

[90] Y. Jiang, C. Wan, J. Wang, Y. Song, and Z. Y. Dong, "Stochastic receding horizon control of active distribution networks with distributed renewables," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1325–1341, March 2019.

[91] I. Jurado, P. Millán, D. Quevedo, and F. R. Rubio, "Stochastic mpc with applications to process control," *International Journal of Control*, vol. 88, no. 4, pp. 792–800, 2015.

[92] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[93] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, “A survey of path planning algorithms for mobile robots,” *Vehicles*, vol. 3, no. 3, pp. 448–468, 2021.

[94] C. Katrakazas, M. Quddus, W. Chen, and L. Deka, “Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions,” *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.

[95] H. Kazemi, H. N. Mahjoub, A. Tahmasbi-Sarvestani, and Y. P. Fallah, “A learning-based stochastic mpc design for cooperative adaptive cruise control to handle interfering vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 266–275, 2018.

[96] S. Keerthi and E. G. Gilbert, “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations,” *Journal of optimization theory and applications*, vol. 57, no. 2, pp. 265–293, 1988.

[97] M. Keller, C. Hass, A. Seewald, and T. Bertram, “Driving simulator study on an emergency steering assist,” in *International IEEE Conference on Systems, Man, and Cybernetics*, San Diego, CA, USA, 2014, pp. 3008–3013.

[98] S. Kerz, J. Teutsch, T. Brüdigam, D. Wollherr, and M. Leibold, “Data-driven stochastic model predictive control,” 2021, arXiv: 2112.04439.

[99] H. K. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[100] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *1985 IEEE International Conference on Robotics and Automation (ICRA)*, St. Louis, MO, USA, 1985, pp. 500–505.

[101] O. Khatib and J. Le Maitre, “Dynamic controlof manipulators operating in a complex environment,” in *On Theory and Practice of Robots and Manipulators, 3rd CISM-IFToMM Symp*, vol. 267, 1978.

[102] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, “A computationally efficient robust model predictive control framework for uncertain nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2021.

[103] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015, pp. 1094–1099.

[104] M. Korda, R. Gondhalekar, J. Cigler, and F. Oldewurtel, “Strongly feasible stochastic model predictive control,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, Dec 2011, pp. 1245–1251.

[105] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng, “Explicit use of probabilistic distributions in linear predictive control,” *Automatica*, vol. 46, no. 10, pp. 1719–1724, 2010.

[106] T. Kruse, P. Basili, S. Glasauer, and A. Kirsch, “Legible robot navigation in the proximity of moving humans,” in *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, Munich, Germany, May 2012, pp. 83–88.

[107] R. Kumar, M. J. Wenzel, M. J. Ellis, M. N. ElBsat, K. H. Drees, and V. M. Zavala, “A stochastic model predictive control framework for stationary battery systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4397–4406, July 2018.

[108] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, “Robust model predictive control using tubes,” *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[109] S. M. Lavalle, “Rapidly-exploring random trees: A new tool for path planning,” Computer Science Deptartment, Iowa State University, USA, Tech. Rep., 1998.

[110] M. Lazar, W. P. M. H. Heemels, B. J. P. Roset, H. Nijmeijer, and P. P. J. van den Bosch, “Input-to-state stabilizing sub-optimal nmpc with an application to dc–dc converters,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 890–904, 2008.

[111] D. Lenz, T. Kessler, and A. Knoll, “Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015, pp. 292–297.

[112] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, “Towards fully autonomous driving: Systems and algorithms,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011, pp. 163–168.

[113] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, “Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems,” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1782–1797, 2018.

[114] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, “Racing miniature cars: Enhancing performance using stochastic mpc and disturbance feedback,” in *2017 American Control Conference (ACC)*, Seattle, WA, USA, 2017, pp. 5642–5647.

[115] J. Löfberg, “Yalmip : A toolbox for modeling and optimization in matlab,” in *2004 IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2004, pp. 284–289.

[116] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgoewer, “Constraint-tightening and stability in stochastic model predictive control,” *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3165–3177, July 2017.

[117] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, “Stochastic mpc with offline uncertainty sampling,” *Automatica*, vol. 81, pp. 176–183, 2017.

[118] M. Lorenzen, M. A. Müller, and F. Allgöwer, “Stochastic model predictive control without terminal constraints,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 15, pp. 4987–5001, 2019.

[119] J. Lu and D. Xi, Y.and Li, “Stochastic model predictive control for probabilistically constrained markovian jump linear systems with additive disturbance,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 15, pp. 5002–5016, 2019.

[120] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, “Efficient sampling-based motion planning for on-road autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1961–1976, 2015.

[121] Y. Ma, J. Matuško, and F. Borrelli, “Stochastic model predictive control for building hvac systems: Complexity and conservatism,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 101–116, 2015.

[122] J. M. Maciejowski, A. Lecchini-Visintini, and J. Lygeros, “Nmpc for complex stochastic systems using a markov chain monte carlo approach,” in *Assessment and Future Directions of Nonlinear Model Predictive Control*, F. R., A. F., and B. L.T., Eds. Berlin, Heidelberg: Springer, 2007, pp. 269–281.

[123] S. Magdici and M. Althoff, “Fail-safe motion planning of autonomous vehicles,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 452–458.

[124] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer, “Robust model predictive control for nonlinear discrete-time systems,” *International Journal of Robust and Nonlinear Control*, vol. 13, no. 3-4, pp. 229–246, 2003.

[125] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, and F. Allgöwer, “An offline-sampling smpc framework with application to autonomous space maneuvers,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 388–402, 2020.

[126] S. Manzinger, C. Pek, and M. Althoff, “Using reachable sets for trajectory planning of automated vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 232–248, 2021.

[127] C. Massera Filho, M. H. Terra, and D. F. Wolf, “Safe optimization of highway traffic with robust model predictive control-based cooperative adaptive cruise control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3193–3203, 2017.

[128] D. Mayne and W. Langson, “Robustifying model predictive control of constrained linear systems,” *Electronics Letters*, vol. 37, pp. 1422–1423, 2001.

[129] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[130] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[131] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," in *1988 27th IEEE Conference on Decision and Control (CDC)*, Austin, TX, USA, 1988, pp. 464–465.

[132] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[133] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[134] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems*, vol. 36, no. 6, pp. 30–44, Dec 2016.

[135] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," in *2014 American Control Conference (ACC)*, June 2014, pp. 2413–2419.

[136] H. Michalska and D. Q. Mayne, "Receding horizon control of nonlinear systems," in *1989 28th IEEE Conference on Decision and Control (CDC)*, Tampa, FL, USA, 1989, pp. 107–108.

[137] ——, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, 1993.

[138] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 2018, pp. 2156–2162.

[139] A. Muraleedharan, A. T. Tran, H. Okuda, and T. Suzuki, "Scenario-based model predictive speed controller considering probabilistic constraint for driving scene with pedestrian," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1–7.

[140] A. Musa, M. Pipicelli, M. Spano, F. Tufano, F. De Nola, G. Di Blasio, A. Gimelli, D. A. Misul, and G. Toscano, "A review of model predictive controls applied to advanced driver-assistance systems," *Energies*, vol. 14, no. 23, 2021.

[141] K. R. Muske and J. B. Rawlings, "Model predictive control with linear models," *AIChE Journal*, vol. 39, no. 2, pp. 262–287, 1993.

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

[143] D. C. K. Ngai and N. H. C. Yung, “A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 509–522, 2011.

[144] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, “Safe and fast tracking on a robot manipulator: Robust mpc and neural network control,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.

[145] K. Okamoto, M. Goldshtein, and P. Tsiotras, “Optimal covariance control for stochastic systems under chance constraints,” *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 266–271, 2018.

[146] K. Okamoto and P. Tsiotras, “Stochastic model predictive control for constrained linear systems using optimal covariance steering,” 2019, arXiv: 1905.13296.

[147] F. Oldewurtel, C. N. Jones, and M. Morari, “A tractable approximation of chance constrained stochastic mpc based on affine disturbance feedback,” in *2008 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico, 2008, pp. 4731–4736.

[148] F. Oldewurtel, C. N. Jones, A. Parisio, and M. Morari, “Stochastic model predictive control for building climate control,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1198–1205, May 2014.

[149] F. Oldewurtel, D. Sturzenegger, P. M. Esfahani, G. Andersson, M. Morari, and J. Lygeros, “Adaptively constrained stochastic model predictive control for closed-loop constraint satisfaction,” in *2013 American Control Conference (ACC)*, Washington, DC, USA, 2013, pp. 4674–4681.

[150] M. Ono and B. Williams, “Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint,” in *2008 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico, 2008, pp. 3427–3432.

[151] G. Pan, R. Ou, and T. Faulwasser, “On a stochastic fundamental lemma and its use for data-driven mpc,” 2021, arXiv: 2111.13636.

[152] 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)*, Maui, HI, USA, Nov 2018, pp. 1447–1454.

[153] M. G. Plessen and A. Bemporad, “Parallel investments in multiple call and put options for the tracking of desired profit profiles,” in *2017 American Control Conference (ACC)*, Seattle, WA, USA, May 2017, pp. 1091–1096.

[154] M. G. Plessen, L. Puglia, T. Gabbriellini, and A. Bemporad, “Dynamic option hedging with transaction costs: A stochastic model predictive control approach,” *International Journal of Robust and Nonlinear Control*, vol. 29, no. 15, pp. 5058–5077, 2019.

[155] J. A. Primbs and V. Nevistić, “Feasibility and stability of constrained finite receding horizon control,” *Automatica*, vol. 36, no. 7, pp. 965–971, 2000.

[156] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho, “Min-max model predictive control of nonlinear systems: A unifying overview on stability,” *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.

[157] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer US, 2005.

[158] A. V. Raković, E. Kerrigan, K. Kouramas, and D. Q. Mayne, “Invariant approximations of robustly positively invariant sets for constrained linear discrete-time systems subject to bounded disturbances,” Department of Engineering, University of Cambridge, UK, Tech. Rep., 2004.

[159] Y. Rasekhipour, A. Khajepour, S. Chen, and B. Litkouhi, “A potential field-based model predictive path-planning controller for autonomous road vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2017.

[160] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, Wisconsin: Nob Hill Publishing, 2017.

[161] E. Rehder, J. Quehl, and C. Stiller, “Driving like a human: Imitation learning for path planning using convolutional neural networks,” in *2017 IEEE International Conference on Robotics and Automation (ICRA) Workshop*, Singapore, 2017.

[162] J. Richalet, A. Rault, J. L. Testud, and J. Papon, “Model algorithmic control of industrial processes,” *IFAC Proceedings Volumes*, vol. 10, no. 16, pp. 103–120, 1977, 5th IFAC/IFIP International Conference on Digital Computer Applications to Process Control.

[163] ——, “Model predictive heuristic control,” *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[164] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. V. Kolmanovsky, “A stochastic model predictive control approach for series hybrid electric vehicle power management,” in *2010 American Control Conference (ACC)*, Baltimore, MD, USA, June 2010, pp. 5844–5849.

[165] U. Rosolia and A. D. Ames, “Multi-rate control design leveraging control barrier functions and model predictive control policies,” *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1007–1012, 2021.

[166] U. Rosolia and F. Borrelli, “Learning how to autonomously race a car: A predictive control approach,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2020.

[167] U. Rosolia, A. Carvalho, and F. Borrelli, “Autonomous racing using learning model predictive control,” in *2017 American Control Conference (ACC)*, Seattle, WA, USA, 2017, pp. 5115–5120.

[168] M. Saval-Calvo, L. Valdés, J. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez, and J. Villagra, “A review of the bayesian occupancy filter,” *Sensors*, vol. 17, Mar. 2017.

[169] R. Scattolini, “Architectures for distributed and hierarchical model predictive control – a review,” *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[170] G. Schildbach and F. Borrelli, “Scenario model predictive control for lane change assistance on highways,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, June 2015, pp. 611–616.

[171] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, “The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations,” *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[172] A. Schimpe and F. Diermeyer, “Steer with me: A predictive, potential field-based control approach for semi-autonomous, teleoperated road vehicles,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, 2020, pp. 1–6.

[173] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, “Motion planning with sequential convex optimization and convex collision checking,” *International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[174] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, “Finding locally optimal, collision-free trajectories with sequential convex optimization,” in *Robotics: Science and Systems*, Berlin, Germany, 2013, pp. 1–10.

[175] B. Schürmann, N. Kochdumper, and M. Althoff, “Reachset model predictive control for disturbed nonlinear systems,” in *2018 IEEE Conference on Decision and Control (CDC)*, Miami, FL, USA, 2018, pp. 3463–3470.

[176] A. T. Schwarm and M. Nikolaou, “Chance-constrained model predictive control,” *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999.

[177] T. Schwickart, H. Voos, M. Darouach, and S. Bezzaoucha, “A flexible move blocking strategy to speed up model-predictive control while retaining a high tracking performance,” in *2016 European Control Conference (ECC)*, Aalborg, Denmark, June 2016, pp. 764–769.

[178] M. M. Seron, G. C. Goodwin, and D. S. Carrasco, “Stochastic model predictive control: Insights and performance comparisons for linear systems,” *International Journal of Robust and Nonlinear Control*, pp. 1–20, 2018.

[179] X. Shang, J. Chen, S. Zhuang, and Y. Shi, “Scenario-based model predictive control for path planning and obstacle avoidance,” in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 446–451.

[180] R. C. Shekhar and C. Manzie, “Optimal move blocking strategies for model predictive control,” *Automatica*, vol. 61, pp. 27–34, 2015.

[181] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller, “Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control,” in *2019 European Control Conference (ECC)*, Naples, Italy, 2019, pp. 811–817.

[182] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1855–1866, June 2018.

[183] T. Stahl and F. Diermeyer, "Online verification enabling approval of driving functions—implementation for a planner of an autonomous race vehicle," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 97–110, 2021.

[184] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 3149–3154.

[185] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr, "Grid-based object tracking with nonlinear dynamic state and shape estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2874–2893, 2020.

[186] S. Steyer, G. Tanzmeister, and D. Wollherr, "Grid-based environment estimation using evidential mapping and particle tracking," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 384–396, Sep. 2018.

[187] J. Suh, H. Chae, and K. Yi, "Stochastic model-predictive control for lane change decision of automated driving vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4771–4782, June 2018.

[188] C. K. Tan, M. J. Tippett, and J. Bao, "Model predictive control with non-uniformly spaced optimization horizon for multi-timescale processes," *Computers & Chemical Engineering*, vol. 84, pp. 162–170, 2016.

[189] M. K. Tay, K. Mekhnacha, M. Yguel, C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. The Bayesian Occupation Filter, pp. 77–98.

[190] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics.* Cambridge, Mass.: MIT Press, 2005.

[191] M. J. Tippett, C. K. Tan, and J. Bao, "Non-constant prediction-step mpc for processes with multi-scale dynamics," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3068–3073, 2014, 19th IFAC World Congress.

[192] D. H. van Hessem and O. H. Bosgra, "Stochastic closed-loop model predictive control of continuous nonlinear chemical processes," *Journal of Process Control*, vol. 16, pp. 225–241, 2006.

[193] S. Vaskov, R. Quirynen, and K. Berntorp, "Cornering stiffness adaptive, stochastic, nonlinear model predictive control for vehicles," in *2021 American Control Conference (ACC)*, New Orleans, LA, USA, 2021, pp. 154–159.

[194] K. P. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 176–188, 2022.

[195] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[196] P. Wang, C. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, 2018, pp. 1379–1384.

[197] A. Wischnewski, J. Betz, and B. Lohmann, "A model-free algorithm to safely approach the handling limit of an autonomous racecar," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Graz, Austria, 2019, pp. 1–6.

[198] A. Wischnewski, M. Euler, S. Gümüs, and B. Lohmann, "Tube model predictive control for an autonomous race car," *Vehicle System Dynamics*, 2021.

[199] W. Wojsznis, A. Mehta, P. Wojsznis, D. Thiele, and T. Blevins, "Multi-objective optimization for model predictive control," *ISA Transactions*, vol. 46, no. 3, pp. 351–361, 2007.

[200] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008, pp. 3731–3736.

[201] B. Yi, P. Bender, F. Bonarens, and C. Stiller, "Model predictive trajectory planning for automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 24–38, 2019.

[202] Y. Yoon, C. Kim, J. Lee, and K. Yi, "Interaction-aware probabilistic trajectory prediction of cut-in vehicles using gaussian process for proactive control of autonomous vehicles," *IEEE Access*, vol. 9, pp. 63 440–63 455, 2021.

[203] S. Yu, M. Hirche, Y. Huang, H. Chen, and F. Allgöwer, "Model predictive control for autonomous ground vehicles: a review," *Autonomous Intelligent Systems*, vol. 1, no. 1, pp. 1–17, 2021.

[204] M. N. Zafar and J. C. Mohanta, "Methodology for path planning and optimization of mobile robots: A review," *Procedia Computer Science*, vol. 133, pp. 141–152, 2018, International Conference on Robotics and Smart Manufacturing (RoSMa2018).

[205] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, April 2016.

[206] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, no. 1, pp. 13–29, 2020.

[207] M. Zanon and S. Gros, “Safe reinforcement learning using robust mpc,” *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2021.

[208] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[209] X. Zeng and J. Wang, “A parallel hybrid electric vehicle energy management strategy using stochastic model predictive control with road grade preview,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2416–2423, 2015.

[210] X. Zhang, A. Georghiou, and J. Lygeros, “Convex approximation of chance-constrained mpc through piecewise affine policies using randomized and robust optimization,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, 2015, pp. 3038–3043.

[211] X. Zhang, A. Liniger, and F. Borrelli, “Optimization-based collision avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2021.

[212] H. Zhu and J. Alonso-Mora, “Chance-constrained collision avoidance for mavs in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.