Technische Universität München
TUM School of Computation, Information and Technology

Safe and Non-Conservative Handling of Multi-Modal and Unmodeled Uncertainty in Autonomous Driving with Stochastic Model Predictive Control

Tommaso Benciolini

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Gabriele Schrag

Prüfende der Dissertation:

1. Priv.-Doz. Dr.-Ing. habil. Marion Leibold
2. Prof. Bo Wahlberg

Die Dissertation wurde am 13.06.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 26.09.2024 angenommen.

Preface

This thesis summarizes my research developed at the Chair of Automatic Control Engineering at the Technical University of Munich. I consider myself incredibly fortunate to have received huge support from numerous people, whom I would like to thank in the following.

I am very grateful to my advisor, Dr. Marion Leibold, for her invaluable support throughout my Ph.D. journey, during which I learned so much. I really appreciated the opportunity to develop my research with scientific freedom, while being able to count on fast, precise, and always constructive feedback. I would also like to thank Dr. Dirk Wollherr for his scientific support and numerous exchanges over the years, that I sincerely appreciated. I am also grateful to Prof. Martin Buss for establishing such an enjoyable research group and to all the non-scientific staff that contribute to the lab and supported my work, in particular to Larissa Schmid.

I would like to thank Prof. Bo Wahlberg for the feedback on my thesis and for being part of the committee, and Prof. Gabriele Schrag for chairing the committee.

I am grateful to Prof. Masayoshi Tomizuka and Dr. Wei Zhan for giving me the opportunity to visit the MSC lab at UC Berkeley, which has broadened my horizons both on a research and personal level. I would like to thank Dr. Chen Tang for his guidance, from which I learned so much during my stay abroad, and Catherine Weaver for her support and fruitful collaboration.

I had the pleasure to share my time at LSR with a great group of colleagues and friends. I owe deepest thanks to Johannes Teutsch, who has not only been a great office mate, but also a constant source of support and fun over the years, cheering me up during stressful times and always being patient as I rushed from one project to another. I would also like to thank Michael Fink and Annalena Daniels, who, together with Johannes, have been the backbone of my time at LSR, sharing fruitful collaboration, engaging in enriching discussions, providing me with interesting perspectives, and making the work more enjoyable. If I had such a pleasant time at LSR, it is largely because of them. Furthermore, I am indebted to Tim Brüdigam and Volker Gabler, who offered me their always welcoming support and experience countless times, and helped me find my way. I would also like to thank all other colleagues, in particular Sebastian Kerz, Salman Bari, Ni Dang, Yuhong Chen, Robert Jacumet, Jonas Marxen, Nehir Güzelkaya, and Christopher Narr, for the support and for making LSR a funny and stimulating place. I am also grateful to the ITR colleagues and my students, in particular Yuntian Yan and Xuhui Zhang, who greatly contributed to my research. Furthermore, a big thank you to Maya Carson, Andrea Rodriguez Blanco, Shivani Sharma, Young Seo Park Kim, and Alexander Naumann for making my time in Berkeley so pleasant.

Finally, I am thankful for the support of my family and friends. In particular, I owe a great deal of gratitude to my parents, to my brother, and to my sister, for always being so encouraging and genuinely interested in my projects throughout these years, as well as for giving me good advice. Finally, my gratitude goes to my wife Francesca, the biggest supporter of my PhD, who relentlessly stood by me, motivated me, and pushed me to dream big and achieve high goals.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) with the project number 490649198. The work discussed in Chapter 5 is part of a project of the University of California Berkely in collaboration with Sony AI, and Polyphony Digital Inc. The research stay of the author at the University of California Berkeley was supported by a fellowship within the "Research Grants for Doctoral Students" program of the German Academic Exchange Service (DAAD) and the Bavaria California Technology Center (BaCaTeC) grant 12-[2022-2].

Abstract

In recent years, control systems have been used in an ever-widening range of applications to increase efficiency and relieve the human operator of a number of tasks. As a result, control systems are increasingly used in highly dynamic scenarios where the environment is greatly uncertain, in part because of the thinner separation between automation and human tasks, such as in autonomous driving, where human drivers, pedestrians, and cyclists share the road with automated vehicles. In particular, interaction and cooperation with humans poses complex challenges.

Not only is it fundamental to ensure that controlled systems operate safely and do not harm humans, but achieving good performance in the presence of high uncertainty is complex for algorithms. In fact, the future behavior of the other agents, humans especially, is not known and must be predicted. However, a large number of different behaviors is possible, causing multi-modal uncertainty. It is extremely challenging to guarantee safety accounting for several different future movements of other agents, while at the same time preventing overly conservative and cautious behaviors that would severely limit efficiency.

This thesis focuses on increasing efficiency in safe schemes to handle high and multi-modal uncertainty, with a specific focus on the trajectory planning problem for autonomous vehicles. We delve into the description of the uncertainty from online data. We discuss how information collected from multiple sources to estimate the probability of uncertainty modes is combined by providing a quantitative estimate of the reliability of the information so that inconsistencies are revealed. We also discuss the problem of simultaneously identifying models representing different modes of uncertainty and determining the most likely mode at any given time. Such a description of multi-modal uncertainty is well suited for trajectory planning, considering multiple possible behaviors of other agents while limiting conservatism.

Secondly, we discuss the planning problem under multi-modal uncertainty and provide several contributions to plan the trajectory of the controlled agent, for example to include long-term reasoning with limited computation effort adopting a hierarchical structure. Further, we develop a framework for considering multiple modes of the uncertainty by prioritizing the focus depending on the mode probability. This allows to consider multiple possible behaviors when the situation is unclear, while avoiding excessive conservatism that would result from paying too much attention to those modes that are currently unlikely. We also discuss further constraint tightening mechanisms to avoid making risky decisions when the information is not reliable. Besides, we develop a framework to verify that the planned efficient trajectory is safe, and to provide a more conservative but safe trajectory if it is not. If such a safe trajectory cannot be obtained due to unanticipated behavior of other agents, we deliver a trajectory that systematically minimizes the probability of constraint violation.

Finally, we address the limitation of model-based approaches in some applications by integrating a machine learning component. We provide a novel framework to purposely steer the controlled system with a view to collect information about the regions of the state space where the information is less reliable and enrich the dataset. We discuss how the framework can be implemented in two application scenarios of autonomous racing, where it is crucial to use little time to collect information, since the goal is to win the race.

We evaluate the novel algorithms in simulation studies in several simulation environments, with focus on automated vehicle examples, and show that our proposed methods significantly contribute to an efficient and safe handling of multi-modal uncertainty in complex scenarios.

Zusammenfassung

In den letzten Jahren wurden Steuerungssysteme in immer mehr Anwendungen eingesetzt, um die Effizienz zu steigern und den menschlichen Bediener von einer Reihe von Aufgaben zu entlasten. Dies hat zur Folge, dass Steuerungssysteme zunehmend in hochdynamischen Szenarien eingesetzt werden, in denen die Umwelt sehr unsicher ist, auch weil die Trennung zwischen Automatisierung und menschlichen Aufgaben immer mehr verschwimmt, wie z.B. beim autonomen Fahren, wo sich menschliche Fahrer, Fußgänger und Radfahrer die Straße mit automatisierten Fahrzeugen teilen. Insbesondere die Interaktion und Kooperation mit dem Menschen stellt eine komplexe Herausforderung dar.

Es ist nicht nur von grundlegender Bedeutung, dass die gesteuerten Systeme sicher funktionieren und den Menschen nicht schädigen, sondern auch, dass die Algorithmen eine gute Leistung unter großer Unsicherheit erbringen. Das zukünftige Verhalten der anderen Agenten, insbesondere der Menschen, ist nämlich nicht bekannt und muss vorhergesagt werden. Es ist jedoch eine Vielzahl unterschiedlicher Verhaltensweisen möglich, was zu multimodaler Unsicherheit führt. Es ist eine große Herausforderung, die Sicherheit unter Berücksichtigung so vieler verschiedener zukünftiger Bewegungen anderer Agenten zu gewährleisten und gleichzeitig ein zu konservatives und vorsichtiges Verhalten zu vermeiden, das die Effizienz stark einschränken würde.

Diese Arbeit konzentriert sich auf die Verbesserung der Effizienz von sicheren Systemen, die mit hoher und multimodaler Unsicherheit umgehen, mit besonderem Augenmerk auf das Problem der Trajektorienplanung für autonome Fahrzeuge. Wir arbeiten an der Beschreibung von Unsicherheit aus Online-Daten. Wir untersuchen, wie Informationen aus verschiedenen Quellen kombiniert werden können, um die Wahrscheinlichkeit von Unsicherheitsmodi abzuschätzen, indem wir eine quantitative Abschätzung der Zuverlässigkeit der Informationen durchführen, um Inkonsistenzen aufzudecken. Außerdem wird das Problem der gleichzeitigen Identifizierung von Modellen, die verschiedene Arten von Unsicherheit repräsentieren, und der Bestimmung der wahrscheinlichsten Art zu einem bestimmten Zeitpunkt behandelt. Eine solche Beschreibung multimodaler Unsicherheit ist für die Trajektorienplanung gut geeignet, da sie mehrere mögliche Verhaltensweisen anderer Agenten berücksichtigt und gleichzeitig den Konservatismus begrenzt.

In einem zweiten Schritt wird das Planungsproblem unter multimodaler Unsicherheit untersucht und es werden verschiedene Beiträge zur Planung der Trajektorie des gesteuerten Agenten geleistet, wie z.B. die Einbeziehung langfristiger Überlegungen mit begrenztem Rechenaufwand durch die Annahme einer hierarchischen Struktur. Darüber hinaus entwickeln wir einen Rahmen für die Berücksichtigung mehrerer Unsicherheitsmodi, indem wir uns auf die Wahrscheinlichkeit des Modus konzentrieren. Dies ermöglicht die Berücksichtigung mehrerer möglicher Verhaltensweisen, wenn die Situation unklar ist, während gleichzeitig ein übertriebener Konservatismus vermieden wird, der sich aus einer zu großen Aufmerksamkeit für derzeit unwahrscheinliche Modi ergeben würde. Wir entwickeln auch andere Mechanismen, um Beschreibungen zu verifizieren, um riskante Entscheidungen zu vermeiden, wenn die Informationen nicht verlässlich sind. Wir entwickeln auch ein Rahmenwerk, um zu überprüfen, ob die geplante effiziente Trajektorie sicher ist, und um eine konservativere, aber sichere Trajektorie zu finden, wenn dies nicht der Fall ist. Wenn eine solche sichere Trajektorie aufgrund des unvorhergesehenen Verhaltens anderer Agenten nicht erreicht werden kann, liefern wir eine Trajektorie, die die Wahrscheinlichkeit einer Beschränkungsverletzung

systematisch minimiert.

Abschließend werden die Grenzen der modellbasierten Ansätze in einigen Anwendungen durch die Integration einer Komponente für maschinelles Lernen aufgezeigt. Wir stellen ein neuartiges Rahmenwerk zur Verfügung, um das kontrollierte System gezielt zu steuern, Informationen über Regionen des Zustandsraums zu sammeln, in denen die Informationen weniger zuverlässig sind, und den Datensatz anzureichern. Wir untersuchen, wie das Rahmenwerk in zwei Anwendungsszenarien des autonomen Rennsports implementiert werden kann, in denen es entscheidend ist, wenig Zeit für das Sammeln von Informationen aufzuwenden, da das Ziel darin besteht, das Rennen zu gewinnen.

Wir evaluieren die neuen Algorithmen in Simulationsstudien in verschiedenen Simulationsumgebungen mit Schwerpunkt auf automatisierten Fahrzeugbeispielen und zeigen, dass die von uns vorgeschlagenen Methoden wesentlich zu einem effizienten und sicheren Umgang mit multimodaler Unsicherheit in komplexen Szenarien beitragen.

Contents

1. Introduction

Recent progress in research has greatly broadened the application domain of automation technology. Nowadays, controllers and algorithms for automatic decision making routinely supervise and govern complex environments, from highly automated manufacturing plants and distribution centers, to agricultural farms. However, vast application also to safety-critical fields, such as autonomous driving, makes it crucial to enhance handling of the uncertainty. Uncertainty is an ineradicable component of using control systems for real world applications, in particular when interaction with humans comes into play.

In fact, uncertainty stemming from modeling errors or unmodeled dynamics can often be quantified within finite bounds, and in some application it does not severely affect the control performance. Conversely, uncertainty resulting from interaction with humans or other autonomous and non-communicating systems, referred to as *agents* in the following, is more difficult to quantify, and plays a bigger role. In the example of an autonomous vehicle driving at low speed in a crowded urban road, modeling errors in the vehicle dynamics are generally less troublesome than the uncertainty about the prediction of the future motion of pedestrian, cyclists and other vehicles. In particular, the uncertainty in motion planning scenarios involving interaction with humans has a twofold nature. On one hand, the high-level *intention* that determines the behavior of the human is unknown. For example, an intention could represent the desire to reach a target position, the plan to follow a particular gait, or vehicle maneuvers like turns at intersections or lane changes. On the other hand, the behavior of agents realizing an intention might still be partially non-deterministic and therefore unpredictable. Thus, an additional lower-level uncertainty must be considered, in the form of uncertainty about the nominal trajectory performing a particular intention. Therefore, a multitude of uncertain possible trajectories is obtained for each Dynamic Obstacle (DO), making the uncertainty caused by the unknown future behavior of DOs multi-modal.

In addition to the general framework of human-robot interaction, where predicting human behavior involves multi-modal uncertainty [1], applications where dealing with multi-modal uncertainty is relevant can be found in other domains. In smart grids and energy systems [2], in order to maintain a balance between demand and consumption, it is necessary to forecast both the production of the various energy sources, where several scenarios must be analyzed to take into account the variability of renewable sources, and the consumption patterns, where consumer behavior is influenced by the time of day, the weather and possible social reasons. Another example is the need for financial institutions to evaluate multiple economics scenarios to decide on conditions for lending and insurances, in order to maximize profit and minimize risk [3]. Also the diagnosis of health conditions might require considering multiple possible scenarios and treatment plans based on observed symptoms, and recent works highlighted the importance of the diagnostic uncertainty [4]. Finally, predictive maintenance involves uncertainty about the type of failure and when it will occur, so it is beneficial to

consider different maintenance activities accordingly [5].

In this thesis, we focus on the application of motion planning for multi-wheeled robots, especially vehicles. The aim is to determine the sequence of control actions, for example acceleration and steering, to drive the mobile robot in presence of large and multi-modal uncertainty caused by the unknown future motion of pedestrian, cyclists and other vehicles, generally referred to as DOs. From a control perspective, the challenge is to provide a safe and yet efficient control in presence of such large uncertainty.

Safety can be ensured with very conservative control strategies, based on the worst-case realization of the uncertainty. Nevertheless, these approaches result in very hesitant behaviors of the controlled agents. In the previous example of an autonomous vehicle driving in an urban environment, if the vehicle must be ready to react to every possible sudden maneuver of, e.g., a pedestrian walking on the sidewalk, and guarantee safety nonetheless, the vehicle will end up maintaining an unreasonably large safety lateral distance from the pedestrian, occupying the left-most part of the road. In presence of upcoming traffic in the other direction, the combination of both uncertainties will likely result in a standstill of the vehicle. Such conservative behavior severely limits the benefit of autonomous driving technology. Therefore, to fully take advantage of the potential of such application, it is crucial not only to provide safe approaches, but also to find a balance with performance optimization that is not too conservative and risk-averse. With a view at improving the control efficiency while still guaranteeing safety, it is beneficial to address both the description of the uncertainty and the balance between safety and limiting conservatism in the control problem.

Several works have discussed how to model the uncertainty about the prediction for the autonomous driving scenario. Expecting communications between all traffic participants will still be unrealistic for a significant amount of time, therefore research developed strategies that do not assume this feature and rather treat the future motion of traffic participants as unknown. Information concerning positions and velocities of other vehicles, cyclists, or pedestrians is collected from onboard sensors, cameras, radars along the road and intelligent infrastructure. Then, prediction models of the future motion of DOs are designed, considering not only the dynamics of DOs, but also the high-level pursued intention driving their motion. Since the intention of other agents is not communicated, different candidate behaviors, i.e., candidate nominal future trajectories, are possible for each DO, each resulting in a different *uncertainty mode*. The modes might not be known in advance and the currently most likely mode must be estimated by data. First, candidate trajectories representing the motion resulting from specific maneuvers are designed, then the measurements collected are compared to the candidate trajectories and candidate behaviors are ranked by evaluating a measure of similarity [6]–[13]. In particular, it is beneficial to consider all the available information about the uncertainty and to develop intention-based trajectory estimation algorithms. Understanding the uncertainty about the future motion of DO as deviation from one of several uncertainty modes, rather than from a single nominal mode, is beneficial to reduce the overall level of uncertainty and allow for efficient yet safe control.

The multi-modal description of the uncertainty is then used to plan the control action of the controlled vehicle. In particular, it is crucial to consider multiple possible realizations of the uncertainty, while still avoiding too cautious planning. Among the existing approaches to plan a trajectory, taking into account the dynamics of the vehicle and the constraints on its actuation, and to avoid collisions with surrounding DOs, predictive controllers and in particular Model Predictive Control (MPC) proved to be well suited, since they consider the

future evolution of the traffic over a finite horizon and iteratively update the planned motion. At each sampling time, the currently available information is used to plan the sequence of inputs to be applied on the ego vehicle within the considered finite prediction horizon. The sequence of inputs is obtained by minimizing a cost function, in which the control goals are encoded. In MPC, only the first element of the minimizing input sequence is applied and at the following sampling time a new optimal sequence of inputs for a shifted horizon is computed, based on the newly collected information. This allows to quickly accommodate changes in the surrounding environment.

The optimal control problem can also include constraints that deal with uncertainty. In the autonomous driving example, these are collision avoidance constraints. The future positions of other traffic participants are predicted, typically propagating the associated uncertainty, and forbidden areas are designed, preventing the automated vehicle from entering the expected future positions of traffic participants. In Robust Model Predictive Control (RMPC) approaches [14], constraint satisfaction is guaranteed for the worst possible realization of the uncertainty. Yet, in automated driving in uncertain environments [15], planning for the worst possible scenario induces an excess of conservatism. A second approach adopts Stochastic Model Predictive Control (SMPC) [16], in which safety constraints are required to be satisfied up to a pre-defined probability level, allowing a more optimistic and efficient planning [17], but introducing a non-zero probability of collision, which must be considered.

This thesis focuses on developing new strategies to balance the trade-off between ensuring safety and limiting conservatism in control of systems subject to multi-modal uncertainty. We focus both on yielding a reliable and consistent description of the uncertainty which does not change too frequently even in presence of highly unclear scenarios, as well as on the main planning of the control action. We consider several automated driving scenarios, which serve as background and motivation for the analysis and proposed methodologies. While most of the work leverages model-based strategies, we also advance the usage of machine learning techniques, providing a framework to collect representative data that allow for an accurate prediction of the behavior of traffic participants. We evaluate our novel algorithms in several simulation environments, showcasing their benefit with respect to state-of-the-art approaches.

1.1. Challenges

When planning control strategies for systems subject to large and multi-modal uncertainty, balancing between the need for certainty and the need to limit conservatism is challenging. In the following, we subdivide this task into distinct challenges, which are necessary for practical application in safety-critical scenarios, and performance maximization, that is, avoiding unnecessarily conservative behaviors.

Challenge 1. *How may environmental information collected online be processed in a way to develop estimates and prediction models of uncertainty modes that are suitable to balance safety and limit conservatism in motion planning?*
In applications in which multiple uncertainty modes are considered, online data are used to assess the currently most likely mode, representing, for example, the pursued intention of a DO. The data are usually collected from a number of sources, which may be inconsistent with each other. In highly uncertain environments, the mode probability estimation might

also change significantly and repetitively between consecutive time steps. Therefore, such estimation is not suitable to be used as environment model for motion planning algorithms, which need a reliable description of the environment. Furthermore, the data collected online must be used to identify models of the environment, tuning the free parameters to maximize the prediction accuracy. However, it is important to avoid overfitting issues arising when conclusions are drawn from incomplete information, that is, tuning the free parameters exclusively considering short and not representative data sequences, resulting in models with poor prediction capabilities.

Challenge 2. *How may the motion of an autonomous agent be planned balancing the trade-off between safety and performance maximization in presence of multi-modal uncertainty?*
If information about multi-modal environmental uncertainty is available the motion of the controlled agent must be planned with a view at balancing the trade-off between safety and efficiency. On the one hand, multiple possible future realizations of the uncertainty should be considered, and safety must be guaranteed with respect to all sufficiently likely realizations. On the other hand, excessive caution resulting from robust approaches must be avoided. The degree of reliability in the description of the environment itself must also be taken into account to avoid making risky decisions when the information is not reliable. Finally, if the dynamic environment allows, that is, if worst-case assumptions on the uncertainty are satisfied, robust approaches give safety guarantees. However, frameworks must be developed to address those situations in which no formal safety guarantees can be given because the realization of the uncertainty violates the assumed worst-case bounds. A systematic procedure to plan a trajectory that handles these circumstances should be provided.

Challenge 3. *How to best exploit machine learning techniques to compensate for limitations of model-based predictions?*
In some scenarios, deriving prediction models of the uncertainty is itself difficult. Thus, model-based strategies are compounded by machine learning techniques, which address the unmodeled components. However, machine learning approaches are only effective when a large amount of descriptive data is available. In some applications, collecting such data is not trivial, since data collected during normal operation, with the goal of maximizing performance, are not descriptive enough. Further, time spent on collecting data might be costly. Systematic procedures to collect rich and diverse data for such applications are needed.

In the following, we summarize how this thesis addresses the three challenges mentioned above and how the novel algorithms developed are consequently organized in chapters.

1.2. Contribution and Outline

This thesis proposes novel control strategies to handle systems subject to safety constraints in presence of multi-modal uncertainty, addressing the previously mentioned challenges. The aim of the work is to balance between the requirements for safety and limiting conservatism to a large extent. This thesis primarily focuses on automated vehicles and mobile robots as key examples of safety-critical applications. These instances require both security assurance and efficient functionality to realize their potential in real-world scenarios. The main chapters of this thesis are summarized as follows.

Chapter 2 - Uncertainty Handling in MPC. We discuss the problem of handling uncertainty in motion planning, especially in MPC-based schemes, then we give an overview of methods from literature that are related to our novel algorithms. Finally, we revise existing algorithms that are used to develop our novel approaches.

Chapter 3 - Effective Description of Multi-Modal Uncertainty. This chapter addresses Challenge 1, by providing novel algorithms to process information collected online and derive a description of the surrounding dynamic environment that allows to balance between safety and limit conservatism in the design of the control action.

The first problem addressed is to combine information collected from a variety of sources into a coherent estimate of the probability of uncertainty modes, focusing on the intention of DOs. We propose a novel algorithm based on the Belief Function Theory (BFT) framework, which provides a quantitative measure of the reliability of the estimated probabilities. Our algorithm allows to analyze possible conflicts between information collected from different sources by increasing the quantitative measure of the reliability of the estimated probability. As a result, estimates of intention probabilities do not change too suddenly over time.

Then, we tackle the problem of designing prediction models for the uncertainty modes and to update them to optimally fit the data collected online, considering a robot moving in an unstructured environment. We propose a novel method to simultaneously estimate the free parameters of models for multiple modes and the probability of each mode using online data. Our algorithm resolves overfitting problems that often arise in parameter estimation when data are too sparse and is computationally undemanding.

The results presented in this chapter have been published in [18], [19].

Chapter 4 - Safe and Non-Conservative SMPC-Based Planning with Multi-modal Uncertainty. This chapter addresses Challenge 2 and focuses on motion planning algorithms to balance safety and efficiency. Here, efficiency is understood as avoiding overly conservative behaviors, that are particularly frequent if robust approaches are applied in presence of multi-modal uncertainty. We focus on the autonomous driving application.

First, we address the challenge of optimizing over a long prediction horizon in SMPC while limiting the computational complexity and accounting for the increased uncertainty for further prediction steps. We propose a multistage SMPC framework in which the action of the control agent is decided by taking advantage of long-term reasoning by explicitly distinguishing between short-term and long-term decisions. Our novel approach takes advantage of a hierarchical structure to reveal superior decisions that would prove beneficial in the long run, and is real-time capable.

We then address the problem of determining a safe trajectory for the controlled vehicle that accounts for multiple modes of uncertainty, i.e., multiple possible realizations of the future motion of DOs, while avoiding overly conservative motions. We introduce a novel method to handle multi-modal uncertainty: the probability of the modes of the uncertainty is estimated online; then, safety constraints are designed assigning to each mode a priority depending on the estimated probability in an SMPC fashion. Our novel approach allows multiple modes of uncertainty to be considered simultaneously without being overly constrained by those that are currently unlikely.

Subsequently, we discuss the problem of computing the trajectory by prioritizing focus for different uncertainty modes in trajectory planning in presence of highly uncertain estimates

of the mode probabilities. We propose a novel framework in which safety constraints for multiple modes are generated in SMPC fashion by varying the degree of conservatism depending on the reliability of the estimated probability, which are provided with a BFT-based description. Our novel approach prevents risky decisions by inducing a more conservative planning when the information about the mode probability is significantly unclear, while still allowing an efficient motion when the DO intention is clear.

Finally, we address the problem of providing a safe trajectory while taking advantage of the optimistic SMPC planning, which introduces a non-zero probability of collision. We propose a novel algorithm that applies the SMPC trajectory only if it is a priori verified to be safe, otherwise a robust planner is used. If the robust planner does not yield a solution, e.g., because the realization of the uncertainty violated the worst-case assumptions and no provably safe trajectories exist, the framework yields a trajectory for the controlled vehicle that minimizes the probability of constraint violation. Our novel approach significantly advances the applicability of SMPC approaches to practical scenarios by addressing such cases where unmodeled components of uncertainty lead to failure of safety guarantees. Furthermore, the computational complexity is kept low.

The results presented in this chapter have been published in [20]–[23].

Chapter 5 - Addressing Unmodeled Uncertainty through Active Exploration. Here Challenge 3 is addressed, which tackles the problem of how to systematically collect representative measurements of the system that allow to effectively derive new prediction models or improve existing ones using machine learning techniques. We develop a novel framework for active exploration in Gaussian Process (GP) regression that is capable of compensating for both the modeling errors due to uncertainty in the system dynamics and the environment in an application-specific environment, i.e., autonomous racing. In our framework the system is purposefully encouraged to visit those regions of the state space where the prediction performance is especially uncertain. For example, the autonomous vehicle systematically tests the reaction of the opponent to specific overtaking attempts. As a result, rather than collecting a large number of measurements during normal operation, a small but rich dataset of measurements leads to a significant improvement in the prediction accuracy of the GP model. The content of this chapter has been published in [24].

Furthermore, in Appendix A.1 we present an efficient constraint generation mechanism, which is well suited for complex frameworks and was presented in [25]. Finally, conclusive remarks and an outlook to future research directions are given in Chapter 6.

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. Benciolini, X. Zhang, D. Wollherr, and M. Leibold, "Information Fusion for Online Estimation of the Behavior of Traffic Participants using Belief Function Theory," in *Frontiers in Future Transportation, section Connected Mobility and Automation*, Volume 4, 2023.

  - A. Daniels[1], T. Benciolini, D. Wollherr, and M. Leibold, "Adaptive Multi-Model Fault Diagnosis of Dynamic Systems for Motion Tracking," in *IEEE Access*, 2024 (accepted).

- Chapter 4 is based on:
  - T. Benciolini, T. Brüdigam and M. Leibold, "Multistage Stochastic Model Predictive Control for Urban Automated Driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, IN, USA, 2021, pp. 417-423.
  - T. Benciolini, D. Wollherr and M. Leibold, "Non-Conservative Trajectory Planning for Automated Vehicles by Estimating Intentions of Dynamic Obstacles," in *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2463-2481, 2023.
  - T. Benciolini, Y. Yan, D. Wollherr, and M. Leibold, "Combining Belief Function Theory and Stochastic Model Predictive Control for Multi-Modal Uncertainty in Autonomous Driving," in *2024 American Control Conference (ACC)*, Toronto, ON, Canada, 2024, pp. 5042–5048.
  - T. Benciolini, M. Fink[2], N. Güzelkaya, D. Wollherr, and M. Leibold, "Safe and Non-Conservative Trajectory Planning for Autonomous Driving Handling Unanticipated Behaviors of Traffic Participants," in *2024 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Edmonton, AB, Canada, 2024.

- Chapter 5 is based on:
  - T. Benciolini, C. Tang, M. Leibold, C. Weaver, M. Tomizuka, and W. Zhan, "Active Exploration in Iterative Gaussian Process Regression for Uncertainty Modeling in Autonomous Racing," in *IEEE Transactions on Control Systems Technology*, 2024.

- Furthermore, the content of Appendix A.1 is based on:
  - T. Benciolini, T. Brüdigam, D. Wollherr, and M. Leibold, "Linear Safety Constraints for Urban Automated Driving: A Case Differentiation," in *International Symposium on Advanced Vehicle Control (AVEC)*, Kanagawa, Japan, 2022.

[1] AD and TB initiated the idea for the paper and designed the algorithm, discussed the properties, and drafted the manuscript. Furthermore, AD is the main responsible person for the implementation of the simulations and the analysis of the results.

[2] TB and MF equally contributed to this work, initiating the idea for the paper and designing of the algorithm, discussing the properties, implementing the simulations, analyzing the results, and drafting of the manuscript.

2. Uncertainty Handling in MPC

In this chapter, we discuss the problem of handling uncertainty in motion planning, especially in MPC-based schemes, in Section 2.1. Then, in Section 2.2 we review existing work from literature that is related to the novel methods presented in this thesis. Finally, in Section 2.3 we revise models and algorithms from the literature that are used to develop the novel algorithms in Chapter 3, Chapter 4, and Chapter 5.

2.1. MPC-based Schemes for Uncertainty Handling

Here, we discuss common approaches to plan the trajectory of the Ego Vehicle (EV) based on MPC. The aim is to deliver a safe trajectory, that is, prevent collisions with static and moving objects, considering the limitations of the vehicle dynamics. In MPC, the control input is determined iteratively by taking the evolution of the system over the prediction horizon into account. Specifically, at each sampling time $t = 0, 1, \dots$ the control action $\boldsymbol{U}_k = [\boldsymbol{u}_t^\top, \dots, \boldsymbol{u}_{t+N-1}^\top]^\top$ over a prediction horizon of length N is obtained minimizing cost function J, i.e., by solving

$$\min_{\boldsymbol{U}_t} J(\boldsymbol{\xi}_t, \boldsymbol{U}_t) \tag{2.1a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k), \qquad \forall k = t, \dots, k+N-1 \tag{2.1b}$$

$$\boldsymbol{u}_k \in \mathcal{U}_k, \qquad \forall k = t, \dots, t+N-1 \tag{2.1c}$$

$$\boldsymbol{\xi}_k \in \mathcal{X}_k, \qquad \forall k = t+1, \dots, t+N \tag{2.1d}$$

$$\boldsymbol{\xi}_k \in \mathcal{S}_k(\boldsymbol{\xi}_k^{\mathrm{DO}}), \qquad \forall k = t+1, \dots, t+N \tag{2.1e}$$

$$\boldsymbol{\xi}_{k+1}^{\mathrm{DO}} = \boldsymbol{f}^{\mathrm{DO}}(\boldsymbol{\xi}_k^{\mathrm{DO}}), \qquad \forall k = t, \dots, t+N-1, \tag{2.1f}$$

where $\boldsymbol{\xi}$ is the state of the controlled system and (2.1b) is its nominal discrete-time dynamics, used as prediction model. The cost function used in this thesis are of the form

$$J(\boldsymbol{\xi}_t, \boldsymbol{U}_t) = \|\Delta\boldsymbol{\xi}_{t+N}\|_{\boldsymbol{P}}^2 + \sum_{k=t}^{t+N-1} \left(\|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2 \right). \tag{2.2}$$

J is designed to penalize large inputs, where $\boldsymbol{R} > 0$. Moreover, large deviations $\Delta\boldsymbol{\xi} = \boldsymbol{\xi} - \boldsymbol{\xi}^*$ of the EV from the desired reference $\boldsymbol{\xi}^*$ are penalized, with weighting matrices $\boldsymbol{Q}, \boldsymbol{P} \geq 0$. The reference is in general assumed to be time-invariant.

Constraints on the input and state of the controlled system are included in the optimal control problem in (2.1c) and (2.1d), for example including physical limitations. Some state constraints, represented by $\mathcal{S}(\cdot)$ in 2.1e, may depend on the predicted state $\boldsymbol{\xi}^{\mathrm{DO}}$ of the dynamic elements of the surrounding environment, e.g., a Dynamic Obstacle (DO). For this

reason, we use the superscript "DO", consistently with the rest of this thesis. The future state of the environment is obtained through a prediction model (2.1f).

In MPC, the optimal control problem (2.1) is solved at each sampling time t, obtaining an optimal sequence of inputs $\boldsymbol{u}_t^*, \dots, \boldsymbol{u}_{t+N-1}^*$. The first element of the optimal sequence, $\boldsymbol{u}_t^*$, is applied to the system, and the process is repeated at the next sampling time, considering the shifted prediction horizon $k = t+1, \dots, t+N+1$.

In practice, uncertainty affects the system. On the one hand, mapping (2.1b) typically represents only the approximated dynamics, either because the (known) real dynamics is too complex to be handled numerically, or because the dynamics is not fully known, or because other uncontrolled forces, e.g., disturbances, influence the state evolution. Such unmodeled effects can be summarized in an uncertainty term $\boldsymbol{w}$, yielding

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{f}^{\text{real}}(\boldsymbol{\xi}_k, \boldsymbol{u}_k, \boldsymbol{w}_k), \ \forall k = 0, \dots, N-1, \tag{2.3}$$

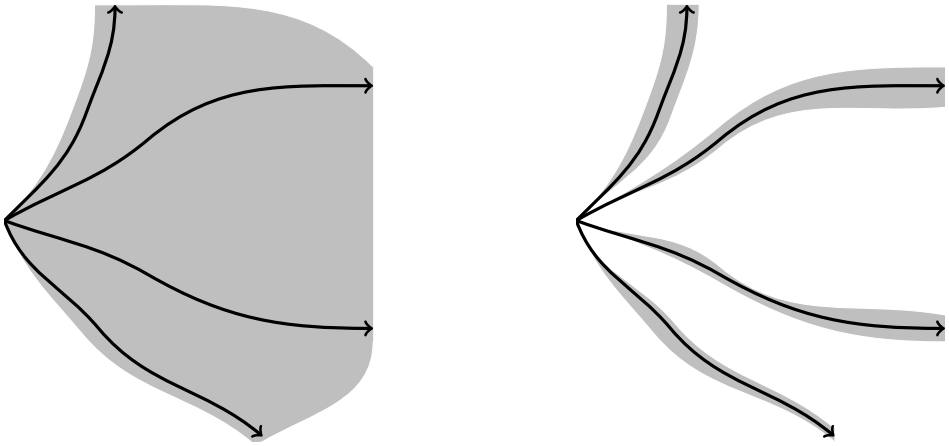
and resulting in deviations of the real system trajectory from the nominal prediction obtained in (2.1b). On the other hand, the real future evolution of the surrounding environment, e.g., the future trajectory of DOs, is not known to the EV, which can at most measure the current position of DOs. Prediction models (2.1f) deliver possible future trajectories of DOs, but in general the real future trajectory differs from the nominal prediction obtained. Typically, the set $\mathcal{S}(\cdot)$ of allowed states for the controlled system is a *deterministic* function of the predicted environment state $\boldsymbol{\xi}^{\text{DO}}$. Nevertheless, the predicted environment state $\boldsymbol{\xi}^{\text{DO}}$ is itself uncertain. Therefore, some state constraints of the controlled vehicle (2.1e) are uncertain.

In particular the uncertainty about the future trajectory of DOs causes multi-modal uncertainty. In fact, the behavior could be driven by different *intentions*, for example maintaining a certain gait, making a turn, or performing a lane change. As a consequence, the uncertainty about the future trajectory of DOs can be understood as stochastic deviations from one of the multiple possible nominal trajectories, each associated with one of high-level intention. Therefore, the uncertainty $\boldsymbol{w}^{\text{DO}}$ about the predicted state $\boldsymbol{\xi}^{\text{DO}}$ of a DO is multi-modal and it holds that

$$\boldsymbol{w}^{\text{DO}} \in \bigcup_{i=1}^{n_{\text{I}}} \mathbb{W}_i, \tag{2.4}$$

that is, $\boldsymbol{w}^{\text{DO}}$ can take any value in the union of the support of the uncertainty $\mathbb{W}_i$ associated with intention $i = 1, \dots n_{\text{I}}$. Understanding the uncertainty as the deviation from one of the multiple possible modes is beneficial to reduce the overall level of uncertainty, as displayed in Figure 2.1. However, it is important to derive an effective description of the multi-modal uncertainty. On the one hand, the probability of each mode must be dynamically estimated. Recognizing the currently most likely mode of the uncertainty, i.e., the candidate nominal trajectory most closely resembling the observed motion of the DO, allows to deduce the intention of the DO and therefore predict the likely future trajectory with smaller uncertainty. On the other hand, prediction models that describe the motion following a high-level intention must be dynamically fit. This allows to dynamically adjust the set of candidate modes, e.g., nominal trajectories, considering the current data.

Even though MPC is a recursive scheme, in which the control action is re-planned at each sampling time accounting for the latest measurement of the system state and of the environment and possibly adopting new local approximation of the dynamics, this does not suffice to handle the uncertainty. In fact, planning the new control sequence from the new state of the system, which differs from the prediction because of the realization of the



(a) Uni-modal uncertainty. (b) Multi-modal uncertainty.

Figure 2.1.: Four possible trajectories interpreted as deviations from a single nominal mode with a large set of feasible values or as realizations of one of multiple distinct modes each with a small (time varying) set of feasible values.

uncertainty, might not be possible without violating some constraints. This might result in dangerous situations for the systems. For example, in autonomous driving, it may be too late to avoid a collision by the time the controlled vehicle realizes that the actual behavior of other road users differs from the prediction based on previous sampling times. It is necessary to explicitly account for the uncertain terms in the design of the MPC scheme.

Literature offers several approaches to account for the uncertainty in the MPC design. If the worst-case realization of the uncertainty is known and bounded, RMPC [26], [27] can be implemented. The control action is planned accounting for the worst-case scenario, thus constraint satisfaction is ensured for every realization of the uncertainty by adding safety margins to the original constraint, through a procedure known as *constraint tightening*. RMPC is a very intuitive approach to handle uncertainty when the aim is to give safety guarantees. Furthermore, minimal information about the uncertainty is required, i.e., only the worst-case bounds. However, RMPC-controlled systems proved to be unnecessarily cautious in particularly uncertain applications such as autonomous driving [15], [28].

The inefficiency of RMPC comes from considering all possible values of the uncertainty independently of their probability. In view of this, SMPC implements a different approach [16], [29]. Constraints are formulated in a probabilistic sense, that is, the probability distribution of the involved quantities is considered and constraints are required to hold only up to a user-defined probability level. This approach results in significant more efficiency, since allowing even a very small probability of constraint violation permits to considerably reduce the effect of the constraint tightening and considerably increase the feasible region in the state space. In doing so, an efficient or "optimistic" control is achieved. Nevertheless, a non-zero probability of collision is introduced, which must be addressed to still give safety guarantees, as proposed in [30], [31] and in Section 4.5 of this thesis. SMPC can be applied even in presence of unbounded uncertainties, since constraint satisfaction is ensured only up to a probability $\beta < 1$ and therefore only a (bounded) subset of the support of the uncertainty is considered.

However, SMPC requires more knowledge of the uncertainty compared to RMPC. In fact,

to evaluate the probability of the probabilistic constraints, information about the probability distribution of the uncertainty is necessary. There are several ways to handle probabilistic constraints in a numerical solver, in particular analytic reformulations and sampling-based schemes. In an analytic reformulation, the distribution of the system states or of the right-hand side of the constraint in (2.1e) at each prediction step in the horizon is computed. Then, safety margins are determined to ensure that the constraint is satisfied at least with the required level of probability [32]–[35]. Because the probability distribution must be propagated through the system dynamics for multiple prediction steps and the safety margin must be analytically computed, this approach is only possible for simple system dynamics and uncertainty distributions, mainly for linear systems and Gaussian disturbances. If such conditions are not met, the sampling-based approach can be used. Rather than analytically determining the distribution of the predicted states, a generative model is used to draw a large number of samples of the uncertainty. Then, the (deterministic) trajectory of the state associated with each uncertainty sample is evaluated and the probability level determines the number of scenarios for which violations are tolerated [36]–[38]. The disadvantage of sampling-based methods is that they are computationally demanding for high-dimensional state spaces, since a high number of samples is required. Furthermore, obtaining formal guarantees of constraint satisfaction from sample mechanisms is challenging.

2.2. Related Work

Here we discuss previous work which is related to the methods we develop in this thesis. Firstly, we discuss work that addresses the problem of modelling the uncertainty about the future motion of DOs, and of estimating the intention pursued, as well as the FDD approaches that have been used to derive multiple prediction models. These works are pertinent to the problem of using online data to derive an effective description of the uncertainty surrounding the future motion of DOs, which we address in Chapter 3. Then, we discuss work that considers the problem of delivering a safe and non-conservative trajectory for the ego vehicle using SMPC, which we address in Chapter 4. We review MPC-based trajectory planning schemes that account for the uncertainty. Subsequently, we examine approaches that permit to extend the prediction horizon in MPC, as well as methodologies for considering multiple potential future trajectories of DO, and works that address the uncertainty about the estimated probabilities of candidate DO trajectories. Furthermore, we review works that discuss how safety can be guaranteed while taking advantage of the SMPC optimistic planning. Finally, we consider previous work related to exploiting machine learning techniques to compensate for limitations of model-based predictions in autonomous racing, that we address in Chapter 5. We discuss approaches to model the vehicle dynamics near handling limits, as well as the opponent behavior in reacting to the EV's own decision. At last, we discuss existing active exploration schemes.

Modeling the Uncertainty about the Future Motion of Dynamic Obstacles Modeling and predicting the future trajectory of DOs, quantifying the uncertainty about such prediction, is a fundamental requirement to enforce safety collision avoidance constraints in autonomous driving, and was addressed in several works, as summarized in the survey [39]. Prediction models are classified into three categories [40]: physics-based, maneuver-based, and interaction aware. Physics-based models predict the future trajectory of DOs purely

considering the evolution of the vehicle dynamics subject to constant inputs, for example constant accelerations, and are not reliable in the long run. Conversely, interaction-aware models [41] also include the reaction of the other vehicles to the future motion of the EV, but including them in the optimization is still challenging because of the increased computational complexity. Maneuver-based models assume that the DO input is not constant, but rather corresponding to an internal *intention* to be executed independently from other drivers. Maneuver-based models are a reasonable trade-off and current state of the art [6], [42], allowing to take the possible intentions driving the behavior of the other traffic participants into account, without excessively increasing the complexity. However, being the intention unknown (especially for pedestrians and cyclists), considering a single model for prediction is limiting. Thus multiple intentions should be considered in the prediction framework and in generating safety constraints. Thus, the uncertainty about the future motion of DOs is twofold [43], [44], i.e., dependent on both the high-level intention being pursued and on the uncertainty in the motion realizing a given intention. [45] used a Partially Observable Markov Decision Process (POMDP) regarding the (unknown) intention of other traffic participants as a hidden variable. This approach accounts for the change in the prediction accuracy directly in the computation of the policy. However, rather simple transition models for other traffic participants must be adopted and the method is computationally demanding because it operates in a continuous state space. Alternatively, Inverse Optimal Control (IOC) and especially Inverse Model Predictive Control (IMPC) [46] are adopted to model and predict the future motion of human-driven vehicles, exhibiting notable accuracy. However, a prior on the features to be included in the candidate cost functions must be available and handling multi-modal motions representing different intentions could be challenging. By contrast, the Interacting Multiple-Model (IMM) algorithm, recalled in Section 2.3.1 and used in several novel algorithms throughout this thesis, allows joint estimation of the uncertainty about estimates and assessment of the probability of each mode with low computational effort and ease of implementation.

Belief Function Theory to Estimate of the Intention of Dynamic Obstacles The problem of estimating the intention driving the behavior of DOs has been studied in several works, for example detecting the intention to perform a lane change in the highway scenario [7]. However, such approach relies on precise a-priori information about the lane-changing trajectory of DOs, and does not generalize well. Especially in the less structured urban environment, it is unrealistic to have very precise information about the execution of the intention. In [17], the IMM algorithm is used to estimate the intended trajectory of target vehicles. However, the variability of the estimated probability of each intention is not considered and the estimation might suddenly and repetitively change if none of the considered models perfectly matches the dynamics, making the estimate unreliable.

By contrast, BFT, which is revised in Section 2.3.5, is well suited to express the uncertainty about the estimated probability. BFT was applied to other fields in which the reliability of the information must be taken into account, such as mapping and tracking of dynamic environments for autonomous navigation [47], [48] and safe reinforcement learning in robotics [49], [50]. BFT was also applied for sensor fusion in human-computer interaction [51]. However, conflict among observations is not dealt with and thus additional observations always render the result more certain, even when they do not agree with each other. Several conflict detection mechanisms were proposed in [52], which, however, do not

consider the uncertainty of the sources, which is to some extent unreasonable. A conflict handling mechanism called *uncertainty maximization* has been presented in [53], transferring part of the belief masses to the quantitative measure of the uncertainty independently of the degree of conflict, which, depending on the application, is also unreasonable for small conflicts. In the work presented in Section 3.2, we propose a framework for estimating the intention of DOs using BFT, in which conflicts between different sources of information are analyzed and the degree of reliability of the combined estimation is consequently decided.

Fault Detection Techniques to Derive Multiple Prediction Models In Section 3.3, we propose to use Fault Detection and Diagnosis (FDD) techniques to derive prediction models of the future trajectory of DOs, by considering a new intention driving the motion of DOs as a new "fault". FDD techniques have been widely used to analyze systems, detect anomalous behaviors, and identify causes of failures. While the data-driven and machine learning-based approaches often still struggle under the influence of noise and uncertainty [54], [55], several classical approaches [56], [57] have in common that the faults to be identified must be known beforehand. In most cases, representing all potential failure behaviors with a single model is impossible. Thus, Multiple Model (MM) techniques have been employed for FDD in previous works: several filters, each designed for a specific system mode, are used in parallel to describe or detect a possible behavior or malfunction of the system. However, if the structure of the fault is not known, a rather large number of different hypotheses must be considered simultaneously, which is impractical. A key aspect enabling FDD in practical scenarios is the ability to consider unseen faults without resulting in a large number of test models [58].

Identifying suitable models for general MM estimation is challenging as, especially for faults, these models are hard to quantify, e.g., due to varying magnitudes or multiple concurrent failures, or unknown at all. Increasing the number of models and filters does not guarantee performance improvement and leads to poorer performance, which is why [59] proposed to use models with variable structure. Over the years, different variable-structure approaches have been proposed. For example, Model-Group Switching [60] and Likely Mode Set [61] independently select subsets of the most relevant models depending on the strategy and system mode. Conversely, Expected-Mode Augmentation [62] spans a continuous mode space using a minimal number of fixed models and adding new models based on combinations of these fixed models based on their probabilities. More recently, a combination of the IMM algorithm and Maximum Likelihood Estimation (MLE) has been presented for a single sensor [63] and the fusion of multiple sensors [64]. However, all these approaches assume an initial set of models as given and only estimate the best-fitting subset or estimate parameters of the determined most likely model. If no information about the structure of the model is available, deep learning can be used to directly infer it from data, showing promising results [65], [66]. However, deep learning models lack explainability, which makes it difficult to interpret, validate, and analyze. In physics-informed machine learning [67], system knowledge is implemented into neural networks to ensure that some model knowledge and, thus, constraints are enforced. Still, training the model takes long time and is only possible if a large enough dataset is available. In addition, the required training time is unpredictable, thus it is not suitable to quickly detect changes in the model. Rather, a simple adaptive approach using a fixed macro-structure combined with MLE suffices. In Section 3.3, we propose a novel algorithm to derive multiple prediction models of the future

trajectory of DOs, which rely on candidate structures to yield explainability, and are tuned with a short sequence of measurements to allow rapid detection of changes in the motion.

Safe and Non-Conservative Trajectory Planning with Model Predictive Control Planning a safe and non-conservative motion for the EV is challenging in presence of the uncertainty about the future trajectory of DOs, which is caused by the unknown intention being pursued. Several machine learning approaches have been proposed [68]–[73], where efficient and human-like trajectories are directly learned from recorded data and replicated nearing human efficiency. However, giving formal guarantees of collision avoidance with machine learning methods is challenging. In Chapter 4, we focus on MPC-based approaches to plan the trajectory of the EV, that have been widely used [17], [74], [75]. RMPC was introduced to address uncertainty in the MPC scheme through a worst-case analysis [14] and has been applied to the autonomous driving problem [15], [28], [76]–[78]. Exact constraint satisfaction is guaranteed for every possible realization of the uncertainty, thus intrinsically yielding a conservative motion for the EV. In the urban framework, where a multitude of DOs without a clear future trajectory surrounds the EV, a robust approach is likely to result in standstills more often than necessary. More recently, SMPC [16], [29] has been applied to trajectory planning for automated driving [17], [32], [79], [80], with a view at handling the uncertainty more efficiently, by allowing a small probability of constraint violation for sufficiently unlikely scenarios in open-loop predictions. An extensive summary of advances in MPC-based methods is presented in [81]. We revise RMPC- and SMPC-based trajectory planning in Section 2.3.7.

Techniques to Allow a Long Prediction Horizon in Model Predictive Control In MPC-based planning a long prediction horizon is beneficial, in particular to optimize the trajectory taking into account possible interaction with DOs well before possible collisions occur. However, a long prediction horizon poses two main challenges. First, the prediction of the future trajectory of the DOs becomes increasingly uncertain for further time steps. Previous works [82], [83] have addressed this problem by utilizing a detailed model for short-term predictions, and a simplified one for long-term predictions. Second, a long prediction horizon poses computational challenges, which have been addressed with move blocking approaches [84] and non-uniformly spaced horizon techniques [85]. Nevertheless, these existing approaches optimize the full trajectory at once, updating simultaneously short- and long-term decisions, hence we propose a hierarchical structure, in which a large optimal control problem is divided into smaller problems solved on different layers.

In Section 4.2, we take advantage of hierarchical structures, which are extensively employed to handle systems with multiple time scales [86], to extend the prediction horizon in MPC keeping the computational complexity low. The usage of hierarchical structures in the framework of autonomous driving has already been proposed in [87]–[89], where the higher level selects the maneuver to be executed by the lower level through pre-defined switching rules. However, these works take advantage of the high level of the hierarchy to decide on the maneuver planning in the short term. By contrast, in our proposal presented in Section 4.2 we take advantage of a hierarchical structure by employing the upper layer for high-level reasoning on a longer horizon without excessively increasing the computational complexity.

Trajectory Planning considering Multiple Trajectories of Dynamic Obstacles In Section 4.3 we present a novel framework to include in the EV trajectory planning problem multiple possible future trajectories of DOs, each based on a different intention, with a view at avoiding an unnecessarily conservative planning. This problem has recently gained increasing attention in literature [90], [91]. In [17], the probability of several maneuver-based models to represent intention uncertainty is dynamically estimated by the IMM algorithm. However, only the most likely model is taken into account in the trajectory planning of the EV. Although the prediction is based on the recent behavior of the vehicles and not only on a-priori assumptions, still the most likely model can change repeatedly, particularly if none of the models considered perfectly matches the real dynamics. The possible sudden and frequent change in the dominant model makes the optimization framework and the resulting planned motion unreliable. A different approach is presented in [92], in which the relative confidence of multiple candidate models for the other traffic participants is estimated using POMDP. Then, the estimated probabilities are utilized as weights to obtain a mixture Gaussian distribution resulting from the combination of the multiple models, and a probabilistic collision avoidance constraint is determined based on the mixture distribution. This approach allows to effectively consider multiple future behaviors, but relying on a single collision avoidance constraint that accounts for all candidate future motions at once requires to significantly over-approximate the forbidden areas for the EV, and thus induces unnecessary conservatism. Furthermore, the recent work [93] considers several candidate maneuvers of DOs separately, avoiding the just-mentioned over-approximation. Still, all obtained predictions are regarded as equally likely, irrespective of the estimated probabilities. Giving equal importance to all candidate predictions also renders the planned motion of the EV unnecessarily conservative. Recent work [94] proposes a different approach, in which the level of constraint violation allowed for each mode is also an optimization variables, thus is determined with a view at minimizing the cost. However, the probability of modes is still not considered. In [95], multi-modal predictions of human behavior are derived and then collision avoidance is enforced as cost, rather than as constraint. Bayesian approaches can be used to identify the movement of dynamic obstacles [96], [97], although they are computationally demanding. In turn, considering a fixed number of models allows to easily integrate a-priori information about candidate movements. Therefore, in this thesis we focus on such approaches. In our approach presented in Section 4.3, we take advantage of the estimated probabilities estimated by the IMM algorithm to explicitly prioritize the multiple possible future trajectories of the other traffic participants depending on their probability. The proposed approach allows to consider multiple possible intentions of DOs in a non-conservative fashion.

Considering Uncertain Estimated Probabilities of the Intention of Dynamic Obstacles In Section 4.4, we address the problem of motion planning considering multiple candidate trajectories of dynamic obstacles when the estimated probabilities of the intentions of dynamic obstacles are highly uncertain. If the motion of the DO does not fit precisely any of the candidate trajectories being tested, contradictory estimated probabilities of the DO intention are produced. As mentioned previously, BFT [98] is a suitable framework to handle the uncertainty about the estimated probabilities, since it provides a quantitative measure of the epistemic uncertainty about the estimation by means of the "uncertainty" parameter [53]. In Section 3.2, BFT is used to combine information collected by different sensors in a reliable

estimate of the probability of candidate behaviors of DOs. When the collected measurements are too contradictory to draw trustworthy conclusions on the probability of candidate DO trajectories, the uncertainty parameter increases. In turn, the assessment of the intended DO behavior does not change too rapidly. Although previous works have discussed the derivation of probabilities from BFT estimates [99], [100], it was not addressed how the BFT estimates can be used to design safety constraints for SMPC, leveraging the measure of reliability of the estimation provided by BFT. We address this problem in Section 4.4.

Safety in Trajectory Planning using Stochastic Model Predictive Control Even though SMPC-based trajectory planning is very popular, relatively few works have discussed how the residual non-zero probability of collision can be accounted for, with a view at yielding a safe motion while taking advantage of the optimistic SMPC planning. In [101] a backup trajectory is planned and triggered in case the computed trajectory proves to lead to a constraint violation in future time times steps. However, it is important to evaluate in advance if the SMPC-based trajectory will lead to collisions at some time in the future, before applying even the first element of the input sequence. Previous work [30] proposed a scheme that allows to use SMPC to provide safety guarantees while taking advantage of the optimistic planning of SMPC. At each sampling time, two optimal control problems are solved, SMPC and a fail-safe trajectory planning problem. The latter is such that collision is avoided for the worst case of the future trajectory of DOs, that is, it is designed following a robust approach. The SMPC solution is only applied if it is proven that after the first step it is still possible to guarantee safety applying a pre-computed backup. However, the structure of the scheme requires to solve two optimizations problems sequentially, which is impractical in real-world applications with small sampling times. Also contingency MPC [31] addresses the problem of non-zero probability of collision allowed by SMPC, combining the SMPC and RMPC optimization in one. The algorithm computes both a nominal and a safe input sequence that produce the same first-step prediction. The nominal input sequence yields non-conservative planning. However, it is ensured that the applied input does not prevent a safe reaction to unlikely but possible events in the next time step. Since only the first element of the control input sequence is applied, the system never chooses between the optimistic mode and the safe mode. Rather, it is guaranteed that a safe input sequence for the remaining steps of the horizon exists. However, the algorithm does not allow to use the full SMPC trajectory even when it is safe to do so, since the first element of the input sequence is always optimized jointly with the robust trajectory. Furthermore, although the solution of two cascaded optimization problems is not required, computing the two input sequences at once does significantly increase the dimension of the optimization problem. Therefore, existing frameworks to guarantee safety using SMPC are computationally inefficient.

Furthermore, another significant limitation of existing approaches guaranteeing safety through a an additional robust planner [30], [31] is that the emergency or fail-safe (robust) trajectory is considered safe relying on some worst-case assumptions on the uncertainty. In practice, even considering multiple possible future motions for each DO, the EV cannot anticipate all possible maneuvers of DOs. Therefore, the design assumptions of the robust approach, used to certify safety, might be violated in practice. This means, for example, that situations in which a DO violates the traffic rules remain unaddressed. In such cases in which the collision cannot be completely excluded, the EV should behave in a way that minimizes the probability of collision. Although several works in the literature present approaches

that guarantee small collision probabilities, less attention has been paid to approaches that plan trajectories that minimize the probability of collisions. In [102], the authors propose a global planner for mobile robots minimizing collision risk, combination the Voronoi graph and the A* algorithm. Nevertheless, the uncertainty arising from the unknown future motion of dynamic obstacles is not considered. In [103], a density-based method to plan safe trajectories by minimizing the collision risk is proposed. Although the approach is promising and results in short online computation times, offline planning is required. Furthermore, the work [104] proposed a mitigation strategy for a vehicle to adapt its position and pose to minimize the severity of the impact in the event of unavoidable collisions. Nevertheless, the work offers purely a mitigation strategy and does not yield a trajectory planned to minimize the probability of collision.

In our approach presented in Section 4.5, we rely on Constraint Violation Probability Minimization (CVPM) techniques [105]–[107], revised in Section 2.3.7, which compute the trajectory and the control inputs yielding the minimum probability of constraint violation. A trajectory with zero probability of constraint violation for any realization of the uncertainty is obtained, if it exists. If not, the probability distribution of the uncertainty is used to determine a trajectory that minimizes the probability of constraint violation. As a result, our algorithm benefits from the non-conservative planning of SMPC, but prevents collisions with DOs as in previous work [30]. However, the computation is more efficient compared to [30]. Furthermore, addressing those situations in which a DO does not behave as anticipated greatly enhance the applicability of the algorithm.

Modeling Vehicle Dynamics near Handling Limits In Chapter 5, we consider autonomous racing scenarios, in which the model of the vehicle dynamics of fundamental importance to maximize the performance. Various physics-based vehicle models have been proposed depending on what assumptions are valid for the application [108], [109]. The dynamic bicycle model, which is revised in Secion 2.3.3, is common in control algorithms and models the dynamics of a single-track vehicle with two wheels [108]. The lateral tire forces may be assumed to be linear with respect to the slip angle of the tires [108], which is valid for the low slip angles encountered during autonomous urban and highway driving. However, racing vehicles operate in the nonlinear, saturated regions of the tire dynamics, and phenomena such as drifting and weight transfer have a significant effect on planning and control [110]. While parameterized tire models like Pacejka's Magic Formula [111] are more descriptive, it can be challenging to identify all of the parameters of the Magic Formula. Furthermore, modeling errors may persist due to weight transfer, suspension dynamics, or the lumped tire dynamics of the bicycle model. Therefore, rather than spending significant engineering efforts attempting to model every detailed aspect of the vehicle dynamics, learning-based approaches could leverage system data to improve model accuracy and control performance in a more efficient way.

Recent efforts have explored how GP, which are revised in Section 2.3.6, can compensate for modeling errors in real-time control [71], [112]. The GP adds to a nominal vehicle model and is trained to improve the model's accuracy on data collected from the actual system [113]. Standard neural networks often lack a measure of prediction uncertainty and, although this aspect has gained attention lately [114], GPs are a common choice because they provide an estimate of the posterior covariance that can be used to predict model uncertainty during control [115]. Retraining the GP after collecting more data can increase model accuracy

and ultimately improve the performance of planning or control [112], [113]. However, data collected during normal EV operation might not be sufficiently representative of the EV dynamics in all situations. Thus, the GP compensation model can be further improved if trained on a more diverse dataset. For example, in vehicle racing, acceleration limits can be progressively updated from collected data to safely expand safe operation as the vehicle improves [116]. In Section 5.2, we propose a novel active exploration framework to specifically target underrepresented regions to collect additional data and reduce the GP's model uncertainty.

Modeling Opponent Behavior in Autonomous Racing The autonomous racing scenario, that we consider in Chapter 5, requires to develop models of the opponent behavior, in order to optimize the trajectory of the EV and to successfully perform overtaking maneuvers. In particular, it is important to account for the interaction with the opponent, modeling the reaction to the EV's own decisions. This task has been considered in [117] using a game-theoretic framework, in which the policy of the EV is chosen as a Nash equilibrium, following well-established approaches for urban and highway autonomous driving [118]–[121]. However, the solution of a dynamic game is generally a computationally expensive task. Moreover, accurate knowledge of the opponent's own reward function and constraints is required to implement this approach, which is limiting in practice. Alternatively, machine learning methods have been used to directly learn the policy or the closed-loop trajectory of the opponent from data. In [122], GPs, which are revised in Section 2.3.6, are used to learn a mapping from the current EV and opponent state to the future opponent state and the posterior covariance of the GP is used to tighten safety collision avoidance constraints. The approach is relatively computationally inexpensive at run time, however implements a passive interaction approach, in which the reaction of the opponent to the current EV's own decisions is not considered. In [123], in the context of urban autonomous driving, a neural network is used to approximate the closed-loop behavior of other agents in a game-theoretic fashion. Instead of solving an optimization problem to predict the future trajectory of other agents, their reaction to the EV's own decision is predicted by a neural network that takes as input the future state of the EV. However, a neural network does not provide a measure of the uncertainty about the prediction. In [124], a GP is trained in a similar fashion, conditioning on the future plan of the EV as well. However, the model is trained on a dataset of measurements collected during normal operation in several previous runs. As a result, the GP prediction of the opponent is not accurate for all possible overtaking strategies that the EV can attempt. To improve the prediction accuracy, an active exploration mechanism is needed, explicitly targeting more regions of the feature space. We develop such active exploration mechanism in Section 5.2.

Active Exploration Schemes In Section 5.2, we develop a novel active exploration scheme for autonomous racing. To the best of our knowledge, this problem has not been addressed in previous works in literature, but active exploration schemes have been presented in other fields. In fact, for learning-based approaches, the choice of the training set plays a major role in determining the performance of the learned model and in the generalization capability. In particular, in iterative approaches, in which the data used are measurements collected during the previous iterations while maximizing the performance of the EV, the dataset might not be sufficiently expressive to significantly improve the model

in the whole feature space. Active learning [125] has been extensively investigated in several fields, such as coverage control [126] and autonomous navigation [127]. Recent work has dedicated attention to active learning approaches based on Koopman operators [128] and Bayesian Optimization [129], whose most recent advances are discussed in a recent survey [130]. Further approaches to active learning are also mentioned in the surveys on learning-based MPC [131], active learning in robotics [132], and deep active learning [133]. In particular, approaches for active exploration in GP regression, revised in Section 2.3.6, have been proposed for control of wind farms [134], airborne wind energy system [135], [136], or UAV delivery control [137]. In such applications, an accurate and updated estimate of the wind field is fundamental, therefore the referenced works proposed approaches to trade-off between maximizing the performance of the system and controlling it in a way to collect measurements to improve the wind field estimation. In Section 5.2, we present a novel active exploration scheme for autonomous racing, which accounts for the fact that, in this application, compromising performance objectives is acceptable only in the early stages of the competition, while eventually the focus must be on maximizing EV performance. Therefore, we dynamically tune the trade-off between exploration and performance goals. Furthermore, the decision about which regions to explore is made in real time.

2.3. Preliminaries

Here we recall dynamic models and algorithms from the literature that are used in the following chapters, where the contribution of this thesis is presented. First, Section 2.3.1 recalls the IMM algorithm, which is used to estimate model probabilities, and Section 2.3.2 the MLE principle, which is used to estimate the free parameters of a model. Then, in Section 2.3.3, we present the dynamic models used for the EV, and, in Section 2.3.4, the models for the dynamic obstacles. Section 2.3.5 summarizes BFT, and Section 2.3.6 GP regression. Finally, in Section 2.3.7 we outline approaches to account for the uncertainty in MPC-based trajectory planning.

2.3.1. Interacting Multiple-Model Algorithm

Here we recall in details the IMM algorithm, that is used in several works of this thesis, namely in Section 3.2, in Section 3.3, and in Section 4.3. In essence, the IMM algorithm is used to estimate the state of the system and infer the probability of a set of given candidate models from the latest output measurement vector $\boldsymbol{\gamma}^{\text{real}}$. We give the original formulation for different dynamic models [138]. However, in this thesis the IMM is used to consider multiple candidate trajectories of DOs.

The IMM algorithm can be understood as an extension of the standard Kalman Filter [139], adapted to obtain a better estimate of the state of target objects whose dynamics is described by different models over time, and is used for example in Aerospace for tracking maneuvering targets [140], as also in application related to object motion detection in ballistic [141] or automated driving frameworks [142], [143]. Instead of using just one model to dynamically estimate the state, multiple candidate models are considered. At each sampling time, the newly-collected output measurement is used to produce not only an estimate of the state and associated covariance matrix, but also the probability that each of the considered model

is currently used to update the dynamics of the object being tracked. Compared to [138], here a known and deterministic input is straightforwardly taken into account.

The IMM is well suited to track an object with *switching linear dynamics*, i.e., an object that at every time the state is updated according to one of n_{I} possible models. Each model is of the form

$$m^{(j)} := \begin{cases} \boldsymbol{z}^{+,(j)} = \boldsymbol{F}^{(j)}\boldsymbol{z}^{(j)} + \boldsymbol{G}^{(j)}\boldsymbol{u}^{(j)} + \boldsymbol{w}^{(j)} \\ \quad \boldsymbol{\gamma} = \boldsymbol{H}^{(j)}\boldsymbol{z}^{(j)} + \boldsymbol{v}, \end{cases} \tag{2.5}$$

with $\boldsymbol{z}^{(j)}$ being the state and $\boldsymbol{\gamma}$ the measurement vector at sampling time t. Matrices $\boldsymbol{F}^{(j)}$, $\boldsymbol{G}^{(j)}$ and $\boldsymbol{H}^{(j)}$, as well as the input vector $\boldsymbol{u}^{(j)}$ are model-specific and, in general, dependent on the parameter vector $\boldsymbol{\theta}^{(j)}$. Disturbance $\boldsymbol{w}^{(j)}$ is also model-specific. All models share the same output vector $\boldsymbol{\gamma}$ and measurement noise $\boldsymbol{v}$.

Further, it is assumed that a-priori probabilities of switching between models are given in the form of a row-stochastic matrix $\boldsymbol{\Pi} \in \mathbb{R}^{n_{\mathrm{I}} \times n_{\mathrm{I}}}$, where $\pi_{i,j}$ defines the probability to switch from model $m^{(i)}$ to $m^{(j)}$ in the next step. In general, application-specific considerations are used to design at least a nominal guess of such a-priori switching probabilities by incorporating all available information.

At each iteration of the IMM, the algorithm receives as input the estimated probability for each model from the previous iteration $\{\hat{\mu}^{(j)}\}_{j=1}^{n_{\mathrm{I}}}$ and the new measurement $\boldsymbol{\gamma}^{\mathrm{real}}$ of the real system, and returns as output the updated model probabilities $\{\hat{\mu}^{(j)}\}_{j=1}^{n_{\mathrm{I}}}$, and the accurate combined state estimate $\hat{\boldsymbol{z}}$ and associated covariance matrix $\hat{\boldsymbol{P}}$. The algorithm also considers $\hat{\boldsymbol{z}}^{(j)}, \hat{\boldsymbol{P}}^{(j)}$, i.e., estimated state and covariance matrix associated with each model $j = 1, \ldots, n_{\mathrm{I}}$. However, these variables are only internal quantities, whereas the output of the algorithm consists of the combined state and covariance estimates $\hat{\boldsymbol{z}}, \hat{\boldsymbol{P}}$. The algorithm consists of the following three steps [138]:

1. *State Interaction*: the estimates of the individual filters from the past iteration $\hat{\boldsymbol{z}}^{(j)}, \hat{\boldsymbol{P}}^{(j)}$ are combined, accounting for a possible switch in the dynamics. Precisely, the conditional probabilities $\tilde{\mu}^{(j|i)}$ of reaching model j starting from model i are computed as follows, for each $i, j = 1, \ldots n_{\mathrm{I}}$

$$\tilde{\mu}^{(j|i)} = [\boldsymbol{\Pi}]_{ij} \frac{\hat{\mu}^{(i)}}{\bar{c}^j}, \; \bar{c}^j = \sum_{i=1}^{n_{\mathrm{I}}} [\boldsymbol{\Pi}]_{ij} \hat{\mu}^{(i)}. \tag{2.6}$$

   Then, the initial state and covariance estimates of each model j are computed as:

$$\hat{\boldsymbol{z}}^{(0\mathrm{j})} = \sum_{i=1}^{n_{\mathrm{I}}} \tilde{\mu}^{(j|i)} \hat{\boldsymbol{z}}^{(i)}, \tag{2.7a}$$

$$\hat{\boldsymbol{P}}^{0j} = \sum_{i=1}^{n_{\mathrm{I}}} \tilde{\mu}^{(j|i)} \left[\hat{\boldsymbol{P}}^{(i)} + (\hat{\boldsymbol{z}}^{(i)} - \hat{\boldsymbol{z}}^{(0j)})(\hat{\boldsymbol{z}}^{(i)} - \hat{\boldsymbol{z}}^{(0j)})^{\top} \right]. \tag{2.7b}$$

2. *Kalman Filtering and Probability Update*: n_{I} Kalman Filters (one for every model) are fed with the combined estimates. The model probabilities are updated based on the prediction error with respect to the newly-collected measurement $\boldsymbol{\gamma}^{\mathrm{real}}$. Precisely, the updated estimate of each filter are independently obtained for each filter following a

standard Kalman Filter approach [139]:

$$\tilde{\boldsymbol{z}}^{(j)} = \boldsymbol{F}^{(j)}\hat{\boldsymbol{z}}^{(0\mathrm{j})} + \boldsymbol{G}^{(j)}\boldsymbol{u}^{(j)}, \tag{2.8a}$$

$$\tilde{\boldsymbol{P}}^{(j)} = \boldsymbol{F}^{(j)}\tilde{\boldsymbol{P}}^{(0j)}{\boldsymbol{F}^{(j)}}^{\top} + \boldsymbol{\Sigma}^{\boldsymbol{w}(j)}, \tag{2.8b}$$

$$\boldsymbol{\epsilon}^{(j)} = \boldsymbol{\gamma}^{\mathrm{real}} - \boldsymbol{H}^{(j)}\tilde{\boldsymbol{z}}^{(j)}, \tag{2.8c}$$

$$\tilde{\boldsymbol{S}}^{(j)} = \boldsymbol{H}^{(j)}\tilde{\boldsymbol{P}}^{(j)}\boldsymbol{H}^{\top} + \boldsymbol{\Sigma}^{\boldsymbol{\nu}}, \tag{2.8d}$$

$$\boldsymbol{L}^{(j)} = \tilde{\boldsymbol{P}}^{(j)}(\boldsymbol{H}^{(j)})^{\top}(\tilde{\boldsymbol{S}}^{(j)})^{-1}, \tag{2.8e}$$

$$\hat{\boldsymbol{z}}^{(j)} = \tilde{\boldsymbol{z}}^{(j)} + \boldsymbol{L}^{(j)}\boldsymbol{\epsilon}^{(j)}, \tag{2.8f}$$

$$\hat{\boldsymbol{P}}^{(j)} = (\boldsymbol{I} - \boldsymbol{L}^{(j)}\boldsymbol{H}^{(j)})\tilde{\boldsymbol{P}}^{(j)}. \tag{2.8g}$$

Then, the model probabilities are updated as follows:

$$\hat{\mu}^{(j)} = \frac{\Lambda^{(j)}\bar{c}^{(j)}}{c},\ c = \sum_{i=1}^{n_{\mathrm{I}}}\Lambda^{(i)}\bar{c}^{i},\ \Lambda^{(j)} = \frac{\exp(-0.5{\boldsymbol{\epsilon}^{(j)}}^{\top}(\tilde{\boldsymbol{S}}^{j})^{-1}\boldsymbol{\epsilon}^{(j)})}{\sqrt{\det(2\pi\tilde{\boldsymbol{S}}^{(j)})}}. \tag{2.9}$$

3. *State Estimate Combination*: a combined state estimate and covariance matrix are obtained, weighing the individual estimates of the n_{I} filters by their probability

$$\hat{\boldsymbol{z}} = \sum_{i=1}^{n_{\mathrm{I}}}\hat{\boldsymbol{z}}^{(i)}\hat{\mu}^{(i)}, \tag{2.10a}$$

$$\hat{\boldsymbol{P}} = \sum_{i=1}^{n_{\mathrm{I}}}\hat{\mu}^{(i)}\left[\hat{\boldsymbol{P}}^{(i)} + (\hat{\boldsymbol{z}}^{(i)} - \hat{\boldsymbol{z}})(\hat{\boldsymbol{z}}^{(i)} - \hat{\boldsymbol{z}})^{\top}\right]. \tag{2.10b}$$

Since the update of the estimates is also based on results from the previous iteration, past data are considered in the computation of the estimate with relatively small effort. Furthermore, compared to a bank of n_{I} independent Kalman Filters each considering only one of the n_{I} different possible models, here the initial combination of model estimates is beneficial. In case of a switch, the IMM quickly corrects the estimate, with a rate depending on the a-priori switching probabilities specified in matrix $\boldsymbol{\Pi}$. Moreover, at steady state, the combined estimate $\hat{\boldsymbol{z}}$ of the IMM algorithm performs nearly as good as a Kalman Filter based on the (currently) real model [138].

2.3.2. Maximum Likelihood Estimation Principle

Here we recall MLE for models $m^{(j)}$ in form (2.5), in which it is assumed that the matrices $\boldsymbol{F}^{(j)}$, $\boldsymbol{G}^{(j)}$ and $\boldsymbol{H}^{(j)}$ and the input vector $\boldsymbol{u}^{(j)}$ dependent on the parameter vector $\boldsymbol{\theta}^{(j)}$. These free parameters are constantly re-estimated, to adapt the models to optimally fit the system trajectory that is currently observed. If several models are considered, the estimate for the parameters is updated for each nominal candidate model independently and in parallel.

At each sampling time t, a new measurement vector of the real system $\boldsymbol{\gamma}_t^{\mathrm{real}}$ is collected, and the estimate of the parameters $\boldsymbol{\theta}^{(j)}$ of each nominal model is updated using MLE [144]. A window of the last n_{W} measurements $\underline{\boldsymbol{\gamma}}_t^{\mathrm{real}} = [(\boldsymbol{\gamma}_1^{\mathrm{real}})^{\top}, \ldots, (\boldsymbol{\gamma}_{n_{\mathrm{W}}}^{\mathrm{real}})^{\top}]^{\top}$ is considered, where, for notation convenience, the index $k = 1$ refers to a sample $\boldsymbol{\gamma}_{t-n_{\mathrm{W}}+1}^{\mathrm{real}}$, that is, collected $n_{\mathrm{W}}-1$ steps before, and the newly-collected output sample $\boldsymbol{\gamma}_t^{\mathrm{real}}$ corresponds to index $k = n_{\mathrm{W}}$ in the

measurement vector $\underline{\boldsymbol{\gamma}}_t^{\text{real}}$. The parameter vector $\boldsymbol{\theta}^{(j)}$ for each model is estimated at every time step in an MLE fashion solving

$$\hat{\boldsymbol{\theta}}^{\text{ML},(j)} = \underset{\boldsymbol{\theta}^{(j)} \in \Theta^{(j)}}{\arg\max} \Pr(\underline{\boldsymbol{\gamma}}_t^{(j)} = \underline{\boldsymbol{\gamma}}_t^{\text{real}}), \tag{2.11}$$

that is, choosing $\hat{\boldsymbol{\theta}}^{\text{ML},(j)}$ by maximizing the probability that the output of the system matches the last n_{W} measured values. The set $\Theta^{(j)}$ generally contains appropriate bounds for the parameters to be estimated. Details on implementing MLE are described in, e.g., [145]. For linear systems the parameters to be estimated enter the system dynamics linearly [146].

2.3.3. Ego Vehicle Models

The kinematic bicycle model [108] is a simplified representation of the dynamics of the EV often used in the MPC optimal control problem. In this thesis, it is used in Section 4.2, in Section 4.3, and in Section 4.5. Such model allows to take into account the coupling between the longitudinal and lateral movements for a sufficiently realistic planning, without adding unnecessary details for the scope of the analysis proposed in most works of this thesis, whose ultimate goal is to deliver a feasible trajectory. We assume the presence of a low-level module in charge for tracking the generated trajectory, where further details about the vehicle dynamics can be more appropriately addressed.

The nonlinear kinematic bicycle model is expressed in the road-aligned (Frenet) frame, that is with respect to a possibly curved reference path, represented in Figure 2.2. The EV state $\boldsymbol{\xi} = [s, d, \phi, v]^\top$ consists of the longitudinal and lateral position of the center of mass of the vehicle expressed in road-aligned coordinates, (s, d), of the yaw angle ϕ of the vehicle with respect to the reference path, and of the linear velocity v. The input is a two-dimensional vector $\boldsymbol{u} = [a, \delta]^\top$ including the linear acceleration a and the front steering angle δ. l_{f} and l_{r} are parameters of the model, representing the distance of the center of gravity from the front and rear axle, respectively.

Assuming that the reference path and its features are known to the EV, let $\kappa(s)$ be the local curvature of the reference path expressed as a function of the curvilinear coordinate s. The differential equations of the kinematic bicycle model are obtained along the lines of [147], although adapted to represent the vehicle with respect to the center of mass (and not with respect to the center of the rear axles). The nonlinear model is

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dfrac{v\cos(\alpha+\phi)}{1-\kappa(s)d} \\ v\sin(\alpha+\phi) \\ v\left(\dfrac{\sin\alpha}{l_{\text{r}}} - \dfrac{\kappa(s)\cos(\alpha+\phi)}{1-\kappa(s)d}\right) \\ a \end{bmatrix} =: \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u}), \tag{2.12}$$

where $\alpha = \arctan\left(\dfrac{l_{\text{r}}}{l_{\text{f}}+l_{\text{r}}}\tan\delta\right)$.

Assumption 2.1. *During the motion, the position (s, d) of the EV is such that $d \neq \frac{1}{k(s)}$.*

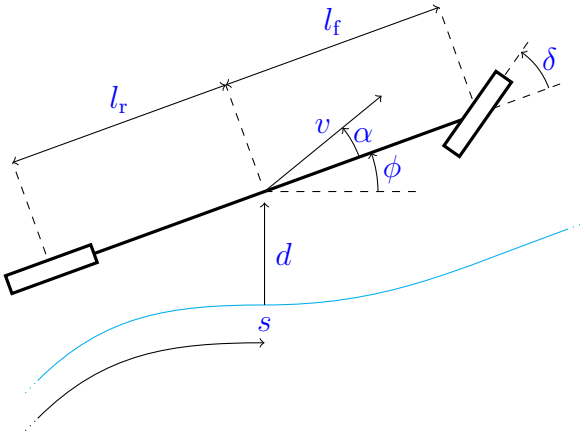


Figure 2.2.: Scheme and notation of the kinematic bicycle model in road-aligned coordinates.

Assumption 2.1 is meant to avoid singularities in the model dynamics (2.12) and is not restrictive, since standard curvature values for vehicle roads are small and the controller guarantees that the lateral displacement d from the reference is limited, so that typically it holds that $d \ll \frac{1}{k(s)}$.

To predict the state of the EV within the prediction horizon, a linear and discrete-time equivalent sufficiently reliable in predicting the future evolution of the EV dynamics in the proximity of the current state is obtained, see [30]. At first, the nonlinear dynamics (2.12) is linearized about the state at the current sampling time $t = 0$, $\boldsymbol{\xi}^* = [s_0, d_0, \phi_0, v_0]^\top$, and zero input $\boldsymbol{u}^* = [0, 0]^\top$, yielding

$$\dot{\boldsymbol{\xi}} \approx \boldsymbol{f}(\boldsymbol{\xi}^*, \boldsymbol{u}^*) + \boldsymbol{A}_\mathrm{l}(\boldsymbol{\xi} - \boldsymbol{\xi}^*) + \boldsymbol{B}_\mathrm{l}\boldsymbol{u}, \tag{2.13}$$

where the Jacobian matrices are

$$\boldsymbol{A}_\mathrm{l} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}}\right|_{(\boldsymbol{\xi}^*, \boldsymbol{u}^*)}, \quad \boldsymbol{B}_\mathrm{l} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{\xi}^*, \boldsymbol{u}^*)}. \tag{2.14}$$

Obtaining the couple of matrices $(\boldsymbol{A}_\mathrm{d}, \boldsymbol{B}_\mathrm{d})$ from $(\boldsymbol{A}_\mathrm{l}, \boldsymbol{B}_\mathrm{l})$ with zero-order hold of sampling time T, the linearized model (2.13) is discretized as

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}^* + T\boldsymbol{f}(\boldsymbol{\xi}^*, \boldsymbol{u}^*) + \boldsymbol{A}_\mathrm{d}(\boldsymbol{\xi}_k - \boldsymbol{\xi}^*) + \boldsymbol{B}_\mathrm{d}\boldsymbol{u}_k. \tag{2.15}$$

To make the computation tractable, the derivation of matrices of the discrete-time system is performed under the approximation that $\kappa'(s_0) = 0$, consistently with the assumption that the reference path was designed sufficiently smooth. The explicit formulation of $\boldsymbol{A}_\mathrm{d}$ and $\boldsymbol{B}_\mathrm{d}$ is given in Appendix A.2. Model (2.15) is used to predict the future states at step $k = 1, \ldots, N$, where N is the prediction horizon, based on the current state $\boldsymbol{\xi}_0$ and on the input sequence over the prediction horizon, $\boldsymbol{u}_k$, $k = 0, \ldots, N-1$.

Even in absence of DOs, the EV must obey the following constraints, related to static traffic rules or to physical limitations of the vehicle. The set of constraints is

$$d_\mathrm{min} + w_\mathrm{EV} \leq d \leq d_\mathrm{max} - w_\mathrm{EV}, \tag{2.16a}$$
$$0 \leq v \leq v_\mathrm{max}, \tag{2.16b}$$
$$\boldsymbol{u}_\mathrm{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_\mathrm{max}, \tag{2.16c}$$
$$\Delta\boldsymbol{u}_\mathrm{min} \leq \Delta\boldsymbol{u} \leq \Delta\boldsymbol{u}_\mathrm{max}, \tag{2.16d}$$

where $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$ is the input rate. (2.16a) is designed to ensure that the shape of the EV, of width w_{EV}, never exceeds the allowed driving area, whose limits (in the road-aligned frame) are $d_{\min}$ and $d_{\max}$. (2.16b) prevents the EV from driving backward or faster than the maximum allowed speed, $v_{\max}$. (2.16c) and (2.16d) make the trajectory planning algorithm aware of the of the actuators limits of the EV in terms of minimum and maximum acceleration and steering angle and of the minimum and maximum rate between two consecutive steps.

The kinematic bicycle model is not sufficiently accurate for some application in which a precise control of the vehicle dynamics is of primary concern, such as autonomous racing, which is discussed in Section 5.2. In this case, the dynamic bicycle model [109] can be used. The state of the vehicle is $\boldsymbol{\xi} = [v_s, v_d, \dot{\phi}, \phi, d, s]^\top$, where v_s and v_d are the vehicle's longitudinal and lateral velocity, respectively, in the vehicle's body frame, $\dot{\phi}$ is the yaw angular velocity, ϕ and d are the yaw angle and lateral displacement of the center of gravity of the vehicle with respect to the reference path, and s represents the traveled distance along the reference path. The relative yaw angle ϕ and lateral distance d in Frenet coordinates are defined with respect to the closest point of the reference path. The input $\boldsymbol{u} = [\delta, a]^\top$ consists of the front steering angle and of the longitudinal acceleration from the powertrain on the rear wheel.

The dynamics is derived from force-mass and inertia-moment balance, for the first two components v_s and v_d, and then from the kinematics for the other states. The dynamics is

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} a - \dfrac{F_{y\text{f}} \sin(\delta) + R_x + F_{x\text{w}}}{m} - g \sin(\varphi) + \dot{\phi} v_d \\ \dfrac{F_{y\text{f}} \cos(\delta) + F_{y\text{r}}}{m} - \dot{\phi} v_s \\ \dfrac{l_\text{f} F_{y\text{f}} \cos(\delta) - l_\text{r} F_{y\text{r}}}{I_{zz}} \\ \dot{\phi} - \dfrac{v_s \cos(\phi) - v_d \sin(\phi)}{1 - \kappa(s) d} \kappa(s) \\ v_s \sin(\phi) + v_d \cos(\phi) \\ \dfrac{v_s \cos(\phi) - v_d \sin(\phi)}{1 - \kappa(s) d} \end{bmatrix}, \tag{2.17}$$

where m is the mass of the vehicle, I_{zz} is the moment of inertia, and l_f and l_r represent the distance of the center of gravity from the front and rear axle, respectively. R_x is the tire rolling resistance, and $F_{x\text{w}}$ is the wind drag force applied on the vehicle body. $F_{y\text{f}}$ and $F_{y\text{r}}$ are the lateral tire forces of the front and rear tires, which are nonlinear and vary as the tire slips along the road surface. Furthermore, gravity is acting on the vehicle with acceleration g and φ is the inclination of the road. $\kappa(s)$ is the curvature of the reference path at position s. A more thorough discussion of the model is reported in [110].

The model is then discretized via forward Euler:

$$\boldsymbol{\xi}^+ = \boldsymbol{\xi} + \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u}) T, \tag{2.18}$$

where T is the sampling time and $\boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})$ is a compact representation of (2.17).

2.3.4. Dynamic Obstacle Models

Here we introduce closed-loop dynamic models used to describe possible behaviors resulting from the unknown intention of the DOs. Each intention is considered to be an approximation of the real behavior of the DOs, therefore trajectory uncertainty about each candidate intention is also accounted for.

The DO is represented by its state $\boldsymbol{z} = [x, v_x, y, v_y]^\top$, where x and y are the coordinates of the position of the DO, and v_x and v_y the longitudinal and lateral velocity, respectively. Considering input $\boldsymbol{u} \in \mathbb{R}^2$ and disturbance $\boldsymbol{w} \in \mathbb{R}^2$, the motion is assumed to be that of a simple point-mass model with decoupled longitudinal and lateral dynamics [43]

$$\boldsymbol{z}^+ = \boldsymbol{A}^{\mathrm{o}}\boldsymbol{z} + \boldsymbol{B}^{\mathrm{o}}\boldsymbol{u} + \boldsymbol{w}, \tag{2.19}$$

where, defining T the sampling time, we have

$$\boldsymbol{A}^{\mathrm{o}} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B}^{\mathrm{o}} = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix}. \tag{2.20}$$

On top of that, the DO is assumed to behave in a way which is modeled through an LQR controller tracking one of a range of n_{I} possible intentions. With the term *intentions*, we refer to specific goal behaviors, like tracking a given lateral or vertical position, or maintaining a precise speed in a direction.

Definition 1. *An intention $\boldsymbol{z}^*$ is defined as a goal target state for a DO.*

Remark 2.1. *Observe that intentions are not time-varying reference trajectories, but rather desired steady-state values for the states, fixed over time. In the example of a car, a lane change and acceleration maneuver would be formulated as a target lateral position (to steer the DO at the center of the desired lane) and a desired speed.*

Thus, the DO applies an input $\boldsymbol{u}$ chosen as

$$\boldsymbol{u} = \boldsymbol{K}^{(j)}(\boldsymbol{z} - \boldsymbol{z}^{(\mathrm{r},j)}), \tag{2.21}$$

that is, as a feedback term to steer the state $\boldsymbol{z}$ of the DO according to one of the intentions $\boldsymbol{z}^{(\mathrm{r},j)}$, $j = 1, \ldots, n_{\mathrm{I}}$. The DO is supposed to plan its trajectory depending on some internal optimality criterion, therefore the feedback gain $\boldsymbol{K}^{(j)}$ is selected as the solution of an LQR problem with weighing matrices $\boldsymbol{Q}^{(j)} \in \mathbb{R}^{4\times4}$, $\boldsymbol{Q}^{(j)} = \boldsymbol{Q}^{(j)^\top} \geq 0$ and $\boldsymbol{R} \in \mathbb{R}^{2\times2}$, $\boldsymbol{R} = \boldsymbol{R}^\top > 0$, i.e., minimizing the cost function

$$J^{\mathrm{o},j} = \sum_{k=0}^{\infty} (\boldsymbol{z} - \boldsymbol{z}^{(\mathrm{r},j)})^\top \boldsymbol{Q}^{(j)} (\boldsymbol{z} - \boldsymbol{z}^{(\mathrm{r},j)}) + \boldsymbol{u}^\top \boldsymbol{R}\boldsymbol{u}. \tag{2.22}$$

The feedback gain $\boldsymbol{K}^{(j)}$ minimizing (2.22) is obtained as

$$\boldsymbol{K}^{(j)} = -(\boldsymbol{B}^{\mathrm{o}\top}\bar{\boldsymbol{P}}\boldsymbol{B}^{\mathrm{o}} + \boldsymbol{R})^{-1}\boldsymbol{B}^{\mathrm{o}\top}\bar{\boldsymbol{P}}\boldsymbol{A}^{\mathrm{o}}, \tag{2.23}$$

where $\bar{\boldsymbol{P}}$ is the only positive semi-definite solution to the Algebraic Riccati Equation [27]

$$\bar{\boldsymbol{P}} = \boldsymbol{A}^{\mathrm{o}\top}\bar{\boldsymbol{P}}\boldsymbol{A}^{\mathrm{o}} + \boldsymbol{Q}^{(j)} - \boldsymbol{A}^{\mathrm{o}\top}\bar{\boldsymbol{P}}\boldsymbol{B}^{\mathrm{o}}(\boldsymbol{B}^{\mathrm{o}\top}\bar{\boldsymbol{P}}\boldsymbol{B}^{\mathrm{o}} + \boldsymbol{R})^{-1}\boldsymbol{B}^{\mathrm{o}\top}\bar{\boldsymbol{P}}\boldsymbol{A}^{\mathrm{o}}. \tag{2.24}$$

Remark 2.2. *For a given intention $\boldsymbol{z}^{(r,j)}$, matrices $\boldsymbol{Q}^{(j)}$ and $\boldsymbol{R}$ specify the penalty to be paid for the error in each state component with respect to the others or with respect to the penalty for large inputs. Therefore, $\boldsymbol{Q}^{(j)}$ and $\boldsymbol{R}$ are used to tune the relative importance in the speed of convergence of each component to the desired state. The future behavior of a DO is fully specified only if the tuple $(\boldsymbol{z}^{(r,j)}, \boldsymbol{Q}^{(j)}, \boldsymbol{R})$ is given, i.e., if both the intention and the relative weighing matrices are known.*

Not only is it possible to tune the relative importance in the speed of convergence, but the evolution of a specific state variable can also be completely ignored by assigning zero to the associated entries in $\boldsymbol{Q}^{(j)}$. This allows, for example, to only specify a cruise speed in x-direction. A target x-position would necessarily be time-varying or imply that the ultimate goal of the controller is to lead the DO to a fixed position, possibly contradicting the commanded cruise speed.

As the focus is on intention changes, all models share the same matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, as well as the statistical description of the disturbances $\boldsymbol{w}, \boldsymbol{v}$. However, different LQR controller gains $\boldsymbol{K}^{(j)}$ are obtained by specifying different weight matrices. Furthermore, each model tracks a different reference state $\boldsymbol{z}^{\mathrm{r},\,(j)}$.

Substituting (2.21) into (2.19), gives

$$\boldsymbol{z}^{(j)^+} = (\boldsymbol{A}^{\mathrm{o}} + \boldsymbol{B}^{\mathrm{o}}\boldsymbol{K}^{(j)})\boldsymbol{z}^{(j)} - \boldsymbol{B}^{\mathrm{o}}\boldsymbol{K}^{(j)}\boldsymbol{z}^{(\mathrm{r},j)} + \boldsymbol{w}, \tag{2.25}$$

where $\boldsymbol{z}^{(j)}$ represents the predicted state of the DO assuming it is following trajectory j. Then, by defining the closed-loop state matrix $\boldsymbol{F}^{(j)} := \boldsymbol{A}^{\mathrm{o}} + \boldsymbol{B}^{\mathrm{o}}\boldsymbol{K}^{(j)}$ and input matrix $\boldsymbol{G}^{(j)} = \boldsymbol{B}^{\mathrm{o}}$, the evolution of each DO is formulated as in (2.5), where $\boldsymbol{u}^{(j)} = -\boldsymbol{K}^{(j)}\boldsymbol{z}^{(\mathrm{r},j)}$ is the input applied to the closed-loop representation of the DO assuming it is pursuing intention j. Since intentions $\boldsymbol{z}^{(\mathrm{r},j)}$ are constant, $\boldsymbol{u}^{(j)}$ is also constant over the prediction horizon. The closed-loop models (2.25) are driven by the reference $\boldsymbol{z}^{(\mathrm{r},j)}(\boldsymbol{\theta}^{(j)})$, which is supposed to be constant over time.

Furthermore, the EV receives a measurement vector $\boldsymbol{\gamma}$ of the position of the DO obtained as in (2.5), where

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.26}$$

which is the same for every model, and $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\nu})$ represents the measurement noise. (2.26) implies that the estimation algorithms only rely on position measurements of the DO. This is an advantage over other approaches, since an estimate of the full state is produced by the IMM algorithm.

2.3.5. Belief Function Theory

Here we recall the main concepts of BFT as presented in [98], and discuss how it is used within the scope of this thesis, specifically in Section 3.2 and in Section 4.4. In this thesis, the BFT framework is used to represent the estimated probabilities of n_{I} candidate future trajectories of DOs, giving a quantification of the reliability of the estimates themselves. Therefore, we finally outline a short example explaining how the approach can be implemented using system measurements for this application.

BFT, also known as Dempster-Shafer Theory [98], is a framework used to combine the information content provided by different sources, taking the reliability of the sources into

account. For each given event, BFT defines a belief, which represents a lower bound on the probability of this event. Thus, BFT is well suited to represent the epistemic uncertainty of the information, which refers to the uncertainty of the model, e.g., due to a lack of data.

The set of all hypotheses tested, called *frame of discernment* $\Theta = \{\theta_1, \dots, \theta_{n_\mathrm{I}}\}$, consists of n_I mutually exclusive candidate outcomes. Formally, in BFT a mapping $m : 2^\Theta \to [0, 1]$ is introduced, assigning a probability mass to every element of the power set 2^Θ, where 2^Θ contains every possible subset of Θ, satisfying

$$\sum_{S \in 2^\Theta} m(S) = 1. \tag{2.27}$$

The quantity $b_S = m(S)$ assigned to subsets $S \subseteq \Theta$ is called *belief mass* of S and indicates how likely it is that one of the outcomes in S will occur. We adopt the BFT approach because it allows probability assignment not only to singletons $b_{\{\theta_1\}} = m(\{\theta_1\})$, but also to unions of events of the frame of discernment, e.g., $b_{\{\theta_1,\theta_2,\theta_3\}} = m(\{\theta_1, \theta_2, \theta_3\})$ [148]. This proves beneficial when $m(\cdot)$ reflects a probability estimation mechanism for which some of the individual events from the frame of discernment Θ are indistinguishable for some sources of information [149], [150]. In such cases, it is possible to assign a belief mass to the union of such indistinguishable events, rather than equally dividing the probability between the involved events.

Among beliefs assigned to subsets of the frame of discernment, $b_\Theta = m(\Theta)$ is especially relevant [53]. b_Θ is the probability that any of the considered outcomes $\theta_1, \dots, \theta_{n_\mathrm{I}}$ occurs and representing the gap between 1 and the sum of the belief masses of any other subset of Θ. Thus, b_Θ is understood as a measure of the epistemic uncertainty about the estimation, representing belief mass that cannot be allocated and anyhow further specified given the available evidence. For this reason, b_Θ is called *uncertainty* and is denoted by μ. The BFT estimation is summarized as the opinion vector.

Definition 2. *An opinion is the vector*

$$\boldsymbol{\omega} = \left[b_{\theta_1}, \dots, b_{\theta_{n_I}}, b_{\{\theta_1,\theta_2\}}, \dots, \mu\right]^\top, \tag{2.28}$$

where μ is inversely proportional to the subjective confidence in the opinion. Because of property (2.27)*, opinions are 1-norm unit vectors.*

Remark 2.3. *Belief masses $b_{\theta_1}, \dots, b_{\theta_{n_I}}, b_{\{\theta_1,\theta_2\}}, \dots$ are not probabilities, since they do not add up to one and are referred also to unions of singletons. Nevertheless, standard probabilities can be obtained for example by equally dividing probabilities of unions between all considered events. In doing so, also the belief mass of uncertainty must be equally split among all events.*

Finally, we recall the concept of *plausibility* of a set, which represents an upper bound to the probability of events contained in the set itself. Precisely, for a subset $S \subseteq \Theta$, the plausibility of S is

$$\mathrm{Pl}(S) = \sum_{\tilde{S} \cap S \neq \varnothing} m(\tilde{S}), \tag{2.29}$$

i.e., the sum of belief masses of propositions $\tilde{S}$ that do not exclude S. $\mathrm{Pl}(S) \geq 0$ characterizes the non-negation degree of S, thus is an upper bound of the probability of S [100]. Coherently with such interpretation, it holds that $\mathrm{Pl}(\Theta) = 1$.

Example: Opinion Generation for Intention Estimation in Autonomous Driving

We outline an example of opinion generation for a DO motion estimation application, to clarify how belief masses and uncertainty can be estimated from system measurements. The opinion generation takes place upstream of our method presented in Section 3.2 and is not part of it, thus this section serves as a pure explanatory example. Any other opinion generation mechanism resulting in opinions in the form (2.28) is suitable.

We consider a set of n_I candidate intentions or intentions, each determining a (nominal) future trajectory, and propose an opinion generation mechanism based on the lateral y-position of a DO measured by a noisy sensor. For each considered candidate intention, let $y_t^i,\ i = 1, \dots, n_\mathrm{I}$ be the y-position at time step t of the nominal trajectory realizing the i-th intention. Furthermore, for every candidate intention, we assume that the variance $(\sigma_y^i)^2$ of the y-position is available, quantifying the expected deviation from the nominal trajectory when realizing that intention. The deviation results both from the process noise, as the trajectory corresponding to each intention is just the nominal motion, and from sensor noise. For example, three intentions could be turning right, proceeding straight, and turning left at an intersection. At first, the probability of each intention at step t, p_t^i, is estimated comparing the noisy measurements y_t, collected at the current step t, to the nominal trajectory y_t^i for intention i. Precisely, measures of similarity can be obtained from a Gaussian kernel, i.e.,

$$p_t^i = \frac{1}{\sqrt{2\pi}\sigma_y^i} \exp\left\{-\frac{(y_t - y_t^i)^2}{2(\sigma_y^i)^2}\right\}, \tag{2.30}$$

then re-scaling masses p_t^i so that they sum to one.

Then, to assess the uncertainty of the probabilities obtained from (2.30), we consider the variation of p_t^i over a window of n_W steps, i.e.,

$$\mu_t = \frac{1}{2(n_\mathrm{W} - 1)} \sum_{h=t-n_\mathrm{W}+2}^{t} \|\boldsymbol{p}_h - \boldsymbol{p}_{h-1}\|_1, \tag{2.31}$$

where $\boldsymbol{p} = [p^1, \dots, p^{n_\mathrm{I}}]^\top$. Expression (2.31) gives a measure of the variability of the masses distribution over the time window. The summation is normalized with respect to the largest theoretical possible value of variation, ensuring that $\mu_t \in [0, 1]$. Observe that (2.31) increases in presence of considerable variations in the estimated probability of each considered intention albeit the relative order between intentions remains the same. Indeed, since a large variation of the mass distributions reflect little reliability of the source in general, (2.31) serves as a measure of uncertainty, because it quantifies the variability of the information across multiple time steps, and therefore how unreliable the estimate is. However, since the uncertainty is a subjective measure, there is no unique method to quantify it and other methods to quantitatively measure the variability of the estimations could be adopted.

Finally, the opinion is obtained as

$$\boldsymbol{\omega}_t = \left[(1 - \mu_t)\boldsymbol{p}_t^\top, \mu_t\right]^\top, \tag{2.32}$$

that is, re-scaling the probabilities from (2.30) given the quantified uncertainty. Depending on the source of information, a similar approach can be used to generate other opinions.

The same procedure applied to the longitudinal velocity of a DO approaching an intersection would not be able to discern between a right and a left turn. Indeed, in both cases, the

expected longitudinal speed profile would be the same, as it is reasonable to imagine that a vehicle needs to slow down in a similar way when approaching the intersection. In this case, it is more reasonable to estimate the belief mass of proceeding straight and of a turn, without further specifying the direction of the turn. This is the reason why the algorithm proposed in Section 3.2 admits sources that specify the belief masses of unions of singletons, allowing more flexibility.

Furthermore, also non-sensor sources can be included, for example using statistics of the recorded traffic. Belief masses can be set proportionally to the statistical frequency of each intention, and the uncertainty can be determined based on the variance of the recorded data. In this case, the resulting opinion is independent of the current intention of a DO, but rather constitutes a bias, which can be included in the information fusion.

Independently of the opinion generation mechanism, the choice of the set of candidate intentions plays a crucial role, as it is discussed in more detail in Section 4.3. However, should a DO execute a intention that does not belong to the set of considered intentions, the uncertainty component would increase, signalizing that none of the considered modes can be reliably trusted.

Remark 2.4. *Here a discrete set of candidate intentions is considered. Nevertheless, in practice the number of possible future trajectories of DOs is infinite. Although this is undoubtedly a limitation of the approach, the n_I candidate trajectories can be seen as nominal trajectories, each representing a cluster of similar future motions. Once the high-level intention has been recognized (and fixed), an appropriate model can be fit to estimate the precise parameters describing the motion of the DO, thus yielding a refined prediction of the future trajectory, as discussed in Section 3.3. However, for the problem of intention recognition, a discrete set of candidate nominal trajectories is sufficient.*

2.3.6. Gaussian Process Regression

Here we revise GP regression, that is a machine learning method used to infer the value of an unknown function given a dataset of M measurements $\mathcal{D} = \{\boldsymbol{z}_i, \boldsymbol{y}_i\}_{i=1}^{M}$, with $\boldsymbol{z}_i \in \mathbb{R}^{n_z}$ the input features and $\boldsymbol{y}_i \in \mathbb{R}^{n_y}$ the output features. A GP is defined as a collection of random variables, each subset of which is jointly normally distributed, and is fully specified by the prior mean and the kernel used as prior covariance [151]. It is assumed that the underlying unknown function $\boldsymbol{g}(\cdot)$ relates the input and the output features as follows

$$\boldsymbol{y}_i = \boldsymbol{g}(\boldsymbol{z}_i) + \boldsymbol{w}_i, \tag{2.33}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{n_y}, \boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{w}})$ is i.i.d. Gaussian noise with diagonal covariance matrix $\boldsymbol{\Sigma}^{\boldsymbol{w}} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{n_y}^2)$. The unknown function is specified through its mean, which we assume zero without loss of generality, and a kernel function $k^a(\boldsymbol{z}, \boldsymbol{z}')$, where $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}$ are two input GP input feature vectors. The scalar function $k^a(\boldsymbol{z}, \boldsymbol{z}')$ is chosen to encode the prior assumptions and the function properties. In Section 5.2, to approximate the modeling error in the vehicle dynamics, we use the squared exponential kernel [151]

$$k^a(\boldsymbol{z}, \boldsymbol{z}') = \sigma_{k^a}^2 \exp\left(-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}')^\top \boldsymbol{L}_{k^a}^{-2}(\boldsymbol{z} - \boldsymbol{z}')\right), \tag{2.34}$$

with parameter L_{k^a} defining the characteristic length-scale and $\sigma_{k^a}^2$ the squared signal variance, whereas to infer the future trajectory of the opponent, we employ the Matérn kernel

with parameter $\nu = 1.5$ [151]

$$k^a(\boldsymbol{z}, \boldsymbol{z}') = \left(1 + \frac{\sqrt{3}\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{l_{k^a}}\right) \exp\left(-\frac{\sqrt{3}\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{l_{k^a}}\right), \tag{2.35}$$

where l_{k^a} is a length scale parameter. Both kernels are widely used and have been chosen consistently with [113] and [124], respectively, to allow for a comparison in which the effect of our active exploration mechanism can be thoroughly discussed. The parameters of the kernels are optimized by maximizing the marginal likelihood of the observations [151].

The posterior mean and covariance of d-th entry $g_d(\boldsymbol{z})$ of the underlying unknown function $\boldsymbol{g}(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{z}), \boldsymbol{\Sigma}(\boldsymbol{z}))$ at the arbitrary point $\boldsymbol{z}^*$ conditioned on the training set $\mathcal{D}$ are obtained as

$$\mu_d(\boldsymbol{z}^*) = (\boldsymbol{k}^a)^\top \boldsymbol{K}^{-1} \boldsymbol{\gamma}_d, \tag{2.36a}$$

$$\sigma_d(\boldsymbol{z}^*) = k^{a*} - (\boldsymbol{k}^a)^\top \boldsymbol{K}^{-1} \boldsymbol{k}^a, \tag{2.36b}$$

where $\boldsymbol{k}^a = [k^a(\boldsymbol{z}_1, \boldsymbol{z}^*), \ldots, k^a(\boldsymbol{z}_M, \boldsymbol{z}^*)]^\top$, the entries of matrix $\boldsymbol{K}$ are $K_{ij} = k^a(\boldsymbol{z}_i, \boldsymbol{z}_j)$, $k^{a*} = k^a(\boldsymbol{z}^*, \boldsymbol{z}^*)$, and $\boldsymbol{\gamma}_d = [y_{1,d}, \ldots, y_{M,d}]^\top$ contains the training outputs corresponding to the d-th entry.

2.3.7. Uncertainty-Aware MPC-Based Trajectory Planning Schemes

In the following, we recall two approaches to account for the uncertainty in MPC-based trajectory planning schemes, that are adopted in several works presented in Chapter 4. The first, SMPC, handles the uncertainty optimistically, allowing a non-zero probability of constraint violation to deliver an efficient trajectory. The second, CVPM, delivers a collision-free trajectory when possible, otherwise minimizes the probability of collision.

The schemes presented here differ only in the formulation of safety constraints. For simplicity of notation, it is assumed that $t = 0$ is the current time step.

Stochastic Model Predictive Control

In the SMPC approach, hard constraints for collision avoidance are relaxed and must be satisfied only up to a user-specified level of probability, $0 \leq \beta < 1$. In the following we recall the analytical approach, in which he probability distribution of the uncertainty is used to derive regions that contain the future position of the DO at least with probability equal to the risk parameter β, and constraints are designed to prevent the future EV trajectory from entering such regions.

The optimal control problem considered at each sampling time is

$$\text{SMPC} \quad \min_{\boldsymbol{U}_0} \; J(\boldsymbol{\xi}_0, \boldsymbol{U}_0) \tag{2.37a}$$

$$\text{s.t.} \quad \boldsymbol{\xi}_{k+1} = \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k), \qquad \forall k = 0, \ldots, N-1 \tag{2.37b}$$

$$\boldsymbol{\xi}_k \in \mathcal{X}_k, \qquad \forall k = 1, \ldots, N \tag{2.37c}$$

$$\boldsymbol{u}_k \in \mathcal{U}_k, \qquad \forall k = 0, \ldots, N-1 \tag{2.37d}$$

$$\Pr(\boldsymbol{\xi}_k \in \mathcal{S}(\boldsymbol{\xi}_k^{\text{DO}})) \geq \beta, \; \forall k = 1, \ldots, N, \tag{2.37e}$$

where N is the prediction horizon, $\boldsymbol{U}_0 = \left[\boldsymbol{u}_0^\top, \ldots, \boldsymbol{u}_{N-1}^\top\right]^\top$, $\boldsymbol{f}$ in (2.37b) is a compact representation of the vehicle dynamics (2.15), and $\mathcal{X}_k$ and $\mathcal{U}_k$ in (2.37c) and (2.37d), respectively, are state and input hard constraints due to the physical limitations of the EV and the traffic rules. The cost function (2.37a) is as in (2.2).

Constraints (2.37e) are probabilistic chance constraints: the EV must meet the safety requirements $\mathcal{S}(\boldsymbol{\xi}_k^{\mathrm{DO}})$ with respect to the future positions of the DOs. The set of safe positions $\mathcal{S}(\boldsymbol{\xi}_k^{\mathrm{DO}})$ is obtained considering the probability distribution of the uncertainties $\boldsymbol{w}^{\mathrm{DO}}$, i.e., the unknown future behavior of DOs, and $\boldsymbol{v}^{\mathrm{DO}}$, i.e., the measurement noise. To handle chance constraints numerically, we utilize unbounded disturbances and reformulate (2.37e) analytically using the assumptions on the probability distribution of the uncertainty. Observe that the SMPC approach allows a $1-\beta$ probability of violation in open loop for each prediction step k in the prediction horizon, that is, in the predictions a collision is possible.

Deterministic forbidden areas for the EV, based on the probabilistic distribution of the uncertainty, are derived as follows. The procedure is run for every DO and for every prediction model independently. The last current measurement or estimate of the state of the DO is used as initial condition and the dynamics (2.25) is iterated for N steps, delivering a sequence of predicted states. The linear dynamics (2.25) yields the following iterative formula to propagate the covariance

$$\boldsymbol{\Sigma}_{k+1}^{(j)} = \boldsymbol{F}^{(j)} \boldsymbol{\Sigma}_k^{(j)} (\boldsymbol{F}^{(j)})^\top + \boldsymbol{\Sigma}^{\boldsymbol{w}}, \tag{2.38}$$

where covariance matrix $\boldsymbol{\Sigma}_0^{(j)}$ is initialized to zero or to the covariance matrix of the current estimate of the DO state. Observe that the input $\boldsymbol{u}^{(j)}$ in (2.25) is deterministic and therefore does not contribute to the propagation of the uncertainty.

Then, safety constraints are obtained. User-defined (deterministic) conditions, such as maintaining a minimum distance from the DO, result in a set $\mathcal{S}$ of allowed (safe) states for the EV. Such constraints are chance constraints because the future position of the DOs is not known deterministically, and therefore also the set $\mathcal{S}$ of safe states for the EV is uncertain.

As safety condition we consider an ellipsoidal region around the position of the DO, that the EV must not enter, as in [30]. Such a requirement needs the future position of the DO to be assessed, which is stochastic. Thus, we derive regions around each of the n_{I} possible nominal trajectories of the DO designed to contain the (real) future position of the DO with probability β_j [17]. We are only interested in the position of the DO, therefore the reduced state $\check{\boldsymbol{z}}_k^j = \left[x_k^j, y_k^j\right]^\top$ is considered, which is modeled by a bivariate Gaussian distribution $\check{\boldsymbol{z}}_k^j \sim \mathcal{N}(\check{\boldsymbol{\mu}}_k^j, \check{\boldsymbol{\Sigma}}_k^j)$, with mean $\check{\boldsymbol{\mu}}_k^j = \left[\tilde{x}_k^j, \tilde{y}_k^j\right]^\top$ and covariance $\check{\boldsymbol{\Sigma}}_k^j = \mathrm{diag}(\sigma_{x,k,j}^2, \sigma_{y,k,j}^2)$ extracted from the full nominal prediction $\tilde{\boldsymbol{z}}_k^{(j)}$ and full covariance matrix $\boldsymbol{\Sigma}_k^{(j)}$, respectively. The ellipsoidal regions designed to contain the position of the future state of the DO with probability β_j consist of all the positions $\check{\boldsymbol{z}}_k^j$ satisfying

$$\left(\check{\boldsymbol{z}}_k^j - \check{\boldsymbol{\mu}}_k^j\right)^\top \left(\check{\boldsymbol{\Sigma}}_k^j\right)^{-1} \left(\check{\boldsymbol{z}}_k^j - \check{\boldsymbol{\mu}}_k^j\right) \leq \zeta(\beta_j). \tag{2.39}$$

$\zeta(\beta_j) = -2\ln(1-\beta_j)$ is the tolerance level and depends on the required level of probability [30]. The regions described by (2.39) increase for larger uncertainty ($\sigma_{x,k,j}^2$ and $\sigma_{y,k,j}^2$ in $\check{\boldsymbol{\Sigma}}_k^j$) and for a higher required probability of constraint satisfaction β_j; furthermore, for $\beta_j \to 0$ these regions collapse to points, i.e., the constraints are neglected in practice.

Constraint Violation Probability Minimization

In the following, we recall the concept of CVPM which was introduced in [105]–[107]. CVPM switches between two optimization problems to enable a robust solution as long as possible by solving the problem CVPM Robust, denoted as *robust case*. If the robust problem is not feasible, the *probabilistic case* is applied, where CVPM Prob is solved and minimizes the probability of constraint violation.

Differently from SMPC, in the robust case the uncertainty about the DO trajectory and the measurement noise are modeled as *truncated* Gaussian variables, $\boldsymbol{w}^{\mathrm{DO}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}^{\boldsymbol{w},\mathrm{DO}}, \mathbb{W}^{\mathrm{DO}}\right)$, $\boldsymbol{v}^{\mathrm{DO}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}^{v,\mathrm{DO}}, \mathbb{V}^{\mathrm{DO}}\right)$.

CVPM considers the set of safe EV state sequences $\hat{\mathcal{S}}\left(\boldsymbol{\Xi}_1^{\mathrm{DO}}\right) = \prod_{k=1}^{N} \mathcal{S}(\boldsymbol{\xi}_k^{\mathrm{DO}})$, for a given DO state sequence, represented as $\boldsymbol{\Xi}_1^{\mathrm{DO}} = [\boldsymbol{\xi}_1^{\mathrm{DO}\top}, \ldots, \boldsymbol{\xi}_1^{\mathrm{DO}\top}]^\top$. The set $\hat{\mathcal{S}}\left(\boldsymbol{\Xi}_1^{\mathrm{DO}}\right)$ is implemented as in [30, Section V-B]. The optimal control problem of the robust case is

$$\begin{aligned}
\text{CVPM Robust}: \ \min_{\boldsymbol{U}_0} \ & J(\boldsymbol{\xi}_0, \boldsymbol{U}_0) && && \text{(2.40a)}\\
\text{s.t. } & \boldsymbol{\xi}_{k+1} = \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k), && \forall k = 0, \ldots, N-1 && \text{(2.40b)}\\
& \boldsymbol{\xi}_k \in \mathcal{X}_k, && \forall k = 1, \ldots, N && \text{(2.40c)}\\
& \boldsymbol{u}_k \in \mathcal{U}_k, && \forall k = 0, \ldots, N-1 && \text{(2.40d)}\\
& \Pr\left(\boldsymbol{\Xi}_k \notin \hat{\mathcal{S}}(\hat{\boldsymbol{\Xi}}_k^{\mathrm{DO}})\right) = 0, \ \forall k = 1, \ldots, N, && && \text{(2.40e)}
\end{aligned}$$

where the cost function in (2.40a) is as in (2.2). The constraint (2.40e) guarantees that the EV state sequence $\boldsymbol{\Xi}_1 = [\boldsymbol{\xi}_1^\top, \ldots, \boldsymbol{\xi}_N^\top]^\top$ satisfies $\boldsymbol{\Xi}_1 \in \hat{\mathcal{S}}(\hat{\boldsymbol{\Xi}}_k^{\mathrm{DO}})$, $\forall \boldsymbol{w}_k^{\mathrm{DO}} \in \mathbb{W}^{\mathrm{DO}}$ and $\forall \boldsymbol{v}_k^{\mathrm{DO}} \in \mathbb{V}^{\mathrm{DO}}$, i.e., the collision avoidance constraint is satisfied robustly.

The *probabilistic case* is used if a solution with zero probability of constraint violation does not exist. Therefore, as in [105], we minimize the probability of constraint violation over the whole prediction horizon by solving the following optimal control problem

$$\begin{aligned}
\text{CVPM Prob}: \ \min_{\boldsymbol{U}_0} \ & \Pr\left(\boldsymbol{\Xi}_k \notin \hat{\mathcal{S}}(\hat{\boldsymbol{\Xi}}_k^{\mathrm{DO}})\right) && && \text{(2.41a)}\\
\text{s.t. } & \boldsymbol{\xi}_{k+1} = f(\boldsymbol{\xi}_k, \boldsymbol{u}_k), && \forall k = 0, \ldots, N-1 && \text{(2.41b)}\\
& \boldsymbol{\xi}_k \in \mathcal{X}_k, && \forall k = 1, \ldots, N && \text{(2.41c)}\\
& \boldsymbol{u}_k \in \mathcal{U}_k, && \forall k = 0, \ldots, N-1. && \text{(2.41d)}
\end{aligned}$$

Remark 2.5. *For bounded disturbances, the probability of violating the collision-avoidance constraint $\boldsymbol{\Xi}_1 in \hat{\mathcal{S}}\left(\boldsymbol{\Xi}_N^{DO}\right)$ far away from the boundaries of the set is constantly 1, which makes the numerical optimization difficult since the gradient vanishes. Therefore, for the probabilistic case an untruncated Gaussian distribution of the input disturbance and of the noise is utilized, i.e., $\boldsymbol{w}_k^{DO} \sim \mathcal{N}\left(0, \Sigma^{\boldsymbol{w},DO}\right)$ and $\boldsymbol{v}_k^{DO} \sim \mathcal{N}\left(0, \Sigma^{v,DO}\right)$, respectively. This allows to extend the basin of attraction due to the infinite support of the Gaussian distribution.*

The *robust case*, where a constraint violation probability of zero is feasible, is applied if [106]

$$\exists \boldsymbol{U}_0 : \Pr\left(\boldsymbol{\Xi}_1 \notin \hat{\mathcal{S}}\left(\boldsymbol{\Xi}_1^{\mathrm{DO}}\right)\right) = 0. \tag{2.42}$$

To evaluate this condition in practice, we determine the set of feasible input trajectories for the robust case

$$\mathcal{U}_{\text{robust}} = \left\{ \boldsymbol{U}_0 \;\middle|\; \boldsymbol{\Xi}_1 \in \hat{\mathcal{S}}\left(\boldsymbol{\Xi}_1^{\text{DO}}\right), \forall \boldsymbol{w}_k^{\text{DO}} \in \mathbb{W}^{\text{DO}}, \forall \boldsymbol{v}_k^{\text{DO}} \in \mathbb{V}^{\text{DO}} \right\}. \tag{2.43}$$

Methods of the set algebra [152] are used to investigate whether $\mathcal{U}_{\text{robust}}$ is non-empty, leading to the application of the robust case. The probabilistic case is used if $\mathcal{U}_{\text{robust}} = \emptyset$. The solutions of the problems CVPM Robust and CVPM Prob yield the optimal input sequence $\boldsymbol{U}_0^*$, where the first element is applied to the system.

Remark 2.6. *In contrast to the chance constraints* (2.37e) *in SMPC, which are defined separately for each time step, the probability distribution in CVPM is considered over the sequence of states. In SMPC, considering each time step separately allows the chance constraints to be treated as tightened constraints. In CVPM, the joint probability distribution is used for minimization, because reducing too much the probability of constraint violation in the first steps might in turn disproportionately increase the probability of constraint violation for further steps. In considering the joint distribution, we account for these effects.*

Effective Description of Multi-Modal Uncertainty

3.

In this chapter, the problem of intention estimation for multi-modal uncertainty is addressed. The uncertainty is not understood as stochastic deviation from a single nominal trajectory, but rather as deviation from several possible nominal trajectories, called *uncertainty modes*, which allows for a more efficient and yet safe control. The dynamic description of the uncertainty must be constantly updated from data, refraining from drawing conclusions when the information is sufficiently clear, in order to avoid misinterpretation issues. We consider the example of intention estimation of moving objects, named here DOs, with unclear behavior and propose two algorithms to merge information available from different sensors and to fit appropriate models representing high-level intentions, in both cases addressing issues that arise when conclusions are drawn from insufficient data. The content of this chapter was first presented in [18], [19].

3.1. Introduction

The motion planning field of mobile robots, in particular autonomous driving, has recently seen incredible progress, especially in terms of dealing with complex situations. Handling the interaction with DOs is complicated, as it requires predicting their future motion. Connected and Automated Vehicles (CAVs) partially alleviate the task, as information is shared, making other CAVs aware of their intention, for example a planned lane change and in which direction, and possibly of the planned trajectory, for example a full prediction of the state sequence while realizing the lane change. However, although the share of CAVs on roads is expected to quickly increase, still for a significant amount of time traffic will mainly be characterized by human-driven vehicles, whose future motion is unknown because of the lack of communication. Moreover, also in the long term, automated vehicles will continue to have to deal with the presence of non-communicating DOs, such as cyclists and pedestrians, who should be given special attention as vulnerable road users.

In model-based control schemes as MPC, the future behavior of the automated vehicle is optimized, accounting for the future moves of DOs, whose future motion must, therefore, be predicted. Several common approaches use the current position and velocity of other DOs, estimated using information collected from onboard sensors, cameras and radars along the road and from intelligent infrastructures communicating with automated vehicles, to infer their expected future motion, relying on prediction models. The design of such prediction models is a challenging task, as they have to consider both the dynamics of DOs and their internal objective, depending, for example, on the driving style and on the maneuver being executed [40], which we call here *intention*. Tuning the prediction model depending on the intention or even the selection of different prediction models depending on the traffic

situation requires an online assessment of the intention of DOs. The intention driving the behavior of a given DO can also be estimated using environmental information. The past trajectory of each DO is analyzed and candidate intentions ranked evaluating a measure of similarity with respect to properly designed models representing different situations [6].

In this chapter, we address the problem of deriving an effective description of the multi-modal uncertainty from data. On the one hand, we use data from several sources to infer probability of uncertainty modes. In the autonomous driving application, the goal is to assign a probability to candidate nominal DO trajectories, quantifying the uncertainty about the estimated probabilities. This is equivalent to deducing the intention of DO, since each candidate nominal trajectory represents the motion resulting from a specific intention. On the other hand, we aim to derive models to predict the future evolution of the uncertainty modes. With reference to the DO motion, we aim to derive prediction models of the DO future trajectory, so that the recent measurements are used to update the prediction.

After the intention of DOs has been recognized, the future motion of vehicles and vulnerable road users can be predicted. Both the predicted trajectories and their estimated probabilities can be passed on to the motion planner accounting for the multiple possible future motions of DOs depending on their probability, as, e.g., in the approach presented in Section 4.3. However, recognizing the nominal trajectory most closely resembling the current motion of DOs and therefore deducing the intention driving the motion of DOs, presents several challenges.

3.1.1. Challenges

Several approaches have been presented to analyze the DO motion using online data, mostly relying on different models depending on the maneuver being executed. However, such models produce inconsistent results if the real observed behavior differs significantly from either of the hypotheses being tested, i.e., from the prediction of either prediction model. Moreover, typically different sensors are used to retrieve information, which must then be combined. Further, heuristics and other non-vision or radar based sources, e.g., traffic statistics, can provide a bias useful to categorize the behavior of DOs. When different sources or multiple motion tracking algorithms are used, a multitude of estimates of the probabilities of DO candidate intentions are produced, which might differ. One main challenge in combining different intention estimations lies in handling the inherent uncertainty about the information provided by each source and in their possible discord. Furthermore, the resulting overall estimation of the intention of DOs should be made as stable and reliable as possible and the reliability of the provided information should be explicitly quantified. Assessing the reliability of the provided estimation is especially important to allow cascaded decision making blocks to subsequently take action to address the uncertainty and to avoid aggressive decisions until the estimation is reliable enough, particularly when interacting with vulnerable users. Finally, combining individual estimations from different sources is challenging if they do not consider the same set of candidate intentions. For example, a source might not discern between two of the candidate intentions to estimate their probability separately, but rather only assign probability to their unions, that is, that either of them will occur.

Furthermore, the choice of the models used to represent each intention plays an important role in the intention estimation process. A variety of FDD approaches analyze systems, detect a change in the intention and identify the cause of failures. Such approaches could in

principle be applied to the problem of intention estimation, by considering each candidate intention as a possible *fault* or hypothesis to be tested. However, a common limitation is that the structure of the fault must be known beforehand. Otherwise, a large number of different hypotheses must be considered and tested simultaneously, which is impractical. To overcome the inconvenience, parametric models must be considered, by dynamically re-identifying the models using the observed data. However, jointly estimating the parameters of candidate models and estimating the probability of candidate models raises overfitting issues, in which unlikely models appear reasonable to explain the current short data sequence, although they generalize poorly.

3.1.2. Contribution

In this chapter, the mentioned challenges are addressed proposing two algorithms. In Section 3.2, we consider the problem of deducing the intention of DOs, i.e., assigning probability to candidate nominal trajectories by combining the probabilities individually estimated by different sources. We propose a new approach based on BFT [98], in which the estimation from different sources are combined through two steps. Compared to previous approaches, we explicitly consider the reliability of each source through the "uncertainty" parameter of the BFT framework. The first step of the algorithm allows to explicitly analyze possible inconsistencies between the individual estimations of the sources. Rather than just being compensated in an average, inconsistency in the individual estimates decrease the quantitative measure of reliability of the information, which is provided together with the estimated probabilities. The second combination step of our novel algorithm considers the temporal variability of the estimates is considered, with a view to make the estimation steady over time. Therefore, we greatly reduce the large variability of probability estimates within consecutive time steps typical of previous approaches such as the IMM. Our new approach leads to improved safety when automated vehicles need to interact with non-communicating DOs.

Then, in Section 3.3, we address the problem of simultaneously tuning prediction models of the uncertainty modes and estimating the probability of each mode, relying on the recent measurements. We propose the Simultaneous Parameter and Model Estimation (SPME) algorithm, a novel variable-structure multi-model algorithm that takes advantage of FDD techniques in the presence of unknown faults to address the problem of intention detection in unstructured environments. Changes of intention are classified as "faults" from the perspective of the last intention. Our novel approach iteratively determines the parameters of each model and the probability of the models, each representing specific dynamics of the object while pursuing a particular high-level intention. The free parameters allow to further differentiate between behaviors of similar forms, constantly re-tuning the models to fit recent measurements. By combining the IMM algorithm and MLE, we jointly estimate the probability of each candidate nominal trajectory with limited computation effort. In our approach, changes in the motion are rapidly detected, since just a short sequence of recent measurements is needed. We avoid overfitting issues typical of repeated parameter estimation by analyzing inconsistencies in the prediction. Our novel algorithm yields models well suited to predict the nominal modes of the uncertainty, e.g., nominal trajectories of DOs.

Although both approaches are evaluated considering the problem of intention estimation of DOs, their application is not limited to this scenario and the developed methods are well suited for other situations in which modes of uncertainty can be identified.

3.2. Accounting for Uncertainty in Mode Estimation using Belief Function Theory

In this section, we present a BFT-based framework to combine information collected by different sources accounting for the uncertainty, in order to obtain estimates of the probability of the uncertainty modes, i.e., the intention of the DO. The estimated probability do not vary too suddenly even in presence of large ambiguity in the behavior of the observed system and, rather, an explicit quantification of the uncertainty about the estimate is provided. We compare our approach with the IMM algorithm [138] through numerical simulations in SUMO [153]. This section is based on the work presented in [18].

3.2.1. Problem Formulation

We assume that n_{S} sources independently collect information about the motion of a DO using the BFT framework, which is recalled in Section 2.3.5. In this work, the set of all hypotheses tested, called *frame of discernment* $\Theta = \{\theta_1, \ldots, \theta_{n_{\mathrm{I}}}\}$, consists of n_{I} mutually exclusive candidate (nominal) DO trajectories.

At each time step t, each source, e.g., a sensor, generates a new opinion $\boldsymbol{\omega}_t$ of the form (2.28). The opinion consists not only of an estimate of the probability of each considered outcome, but include an assessment of the reliability of the information provided, taking advantage of the "uncertainty" concept from BFT [154], that represents the epistemic uncertainty regarding the reliability of information, as formally stated in Section 2.3.5. Depending on the sensor or detection mechanism, some individual and separate outcomes might not be distinguishable for a given source of information. In those cases, we allow the source to provide an estimation assigning probability to the union of singletons, rather than, for example, redistributing equally the probability within the singletons involved.

Information from several sources must be combined in coherent joint estimates, in which possible differences in the estimated probabilities are analyzed. In the combined estimation, the intention probabilities should not vary too suddenly, and an assessment of the reliability of the estimation should be provided. Therefore, the following objective is obtained.

Objective 3.1. *The aim is to combine the opinions $\boldsymbol{\omega}_t^1, \ldots, \boldsymbol{\omega}_t^{n_S}$ in a joint opinion $\boldsymbol{\omega}_t$ of the intention of the DO, so that $\boldsymbol{\omega}_t$ does not change too suddenly over time in case the individual opinions $\boldsymbol{\omega}_t^1, \ldots, \boldsymbol{\omega}_t^{n_S}$ produced by the sources vary significantly and provides a quantitative measure of the reliability of the estimation.*

Our focus is combining information with a view at assessing the intention of DOs, but the method is not limited to this specific application. In the following, we present our method. First, we give an overview of the approach. Then, we outline the two steps composing the proposed belief processing algorithm to obtain reliable estimates of the probability of candidate intentions of the DOs, i.e., the combination of the current estimations from multiple sources analyzing conflicts and the temporal propagation.

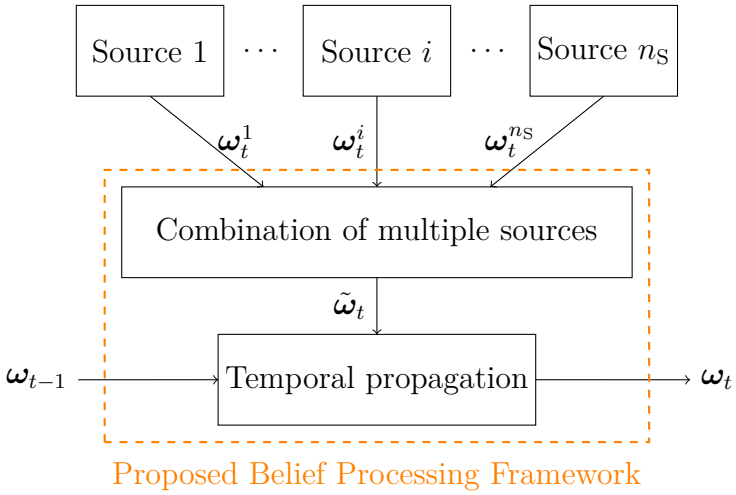


Figure 3.1.: Scheme of the approach. © 2023 Frontiers.

3.2.2. Overview of the Proposed BFT-Based Approach for Information Fusion

The information fusion framework presented here must be repeated in parallel for each DO separately, gathering and processing the full information related to one DO at a time. The procedure can be divided into two parts, namely multi-source information fusion and temporal belief distribution propagation. The scheme is represented in Figure 3.1.

First, independently generated opinions are combined, gathering the information resulting from the current time step. An agreement between individual estimations provided by different sources is found, based on the last samples available. In particular, possible changes in the intention of the DO are reported in this phase. The processing of individual estimations is independent of how opinions are generated. Observe that the combined opinion is not just an average of the estimated probabilities from every separated source. Rather, opinions are merged considering the uncertainty of each of them, weighing in more sources that provide more reliability. Furthermore, an assessment on the uncertainty of the combined estimate is also provided. Several independent but coherent opinions reduce the overall level of uncertainty, whereas possible disagreement is treated through a mechanism outlined in Section 3.2.3. The combined opinion is labelled $\tilde{\boldsymbol{\omega}}_t$.

Secondly, to propagate the estimation over time, the combined opinion gathering information from the current time step $\tilde{\boldsymbol{\omega}}_t$ is fused with the information obtained up to the previous time step, $\boldsymbol{\omega}_{t-1}$, resulting in $\boldsymbol{\omega}_t$, the output of the algorithm. Differently from the previous combination, here the goal is to give steadiness to the estimate, so that too sudden fluctuations in the estimate, that would make the information unreliable, are attenuated. Rather, fluctuations within the estimates provided by the sources result in an increase in the explicit quantification of the uncertainty.

In the approach presented here, singletons and individual probabilities are referred to different intentions of DOs, resulting in different expected future trajectories. As a possible purpose of providing such an estimate, we refer to the motion planning problem using predictive controllers, that will be discussed in detail in Chapter 4. Therein, an estimation on the probabilities of several candidate future trajectories of DOs allows to optimize with

respect to different outcomes depending on their probabilities. Nevertheless, the algorithm for information handling is potentially general and applicable for other purposes.

Remark 3.1. *Our algorithm provides a reliable combined estimate of the probability of the current intention of traffic participants, which is suitable to be used to predict future intentions or future trajectories (given the intention). However, such (potential) prediction module is not part of our algorithm, whose output is the combined estimate of the probabilities of candidate intentions.*

3.2.3. Combination of Individual Estimates from Multiple Sources

Here we introduce the first opinion combination mechanism. We omit the time dependence, since all quantities are related to the current time step. The aim is to gather the whole information collected at the current iteration, that is, merge opinions $\boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^{n_\mathrm{S}}$, obtained from each source independently, in the combination $\tilde{\boldsymbol{\omega}} = \left[\tilde{b}_1, \ldots, \tilde{b}_{n_\mathrm{I}}, \tilde{\mu}\right]^\top$. Bearing in mind that the scope of such information fusion is feeding an accommodation algorithm to account for several outcomes depending on their probabilities, for example as discussed in Section 4.3, the combined opinion $\tilde{\boldsymbol{\omega}}$ only comprises n_I beliefs for mutually-exclusive singleton events and assessment of the overall uncertainty.

Remark 3.2. *In the following formulation of the algorithm, we assume independence of the n_S sources. When some of the considered sources are dependent, they should be combined into a single opinion upstream of our algorithm, so that the input to the algorithm consists of n_S independently generated beliefs. Alternatively, the algorithm can be adapted to handle dependent sources along the lines of [155].*

In order to combine the most recent opinions generated by different sources of information, we use a revised version of the Dempster's rule of combination [148]. We modify the rule from the original formulation so that the combined opinion consists only of singletons and the union of all singletons. This design is advantageous for the application in question, where the probability of each individual candidate intention must be obtained, rather than the probability of unions of intentions. However, if of interest, straightforward adaptions allow to maintain belief masses of unions also in the remaining steps of the algorithm.

Opinions are iteratively combined two at a time. We assume possibly heterogeneous sources of information, thus opinions might, in general, consider different unions of singletons. Given two opinions $\boldsymbol{\omega}^\mathrm{A}$ and $\boldsymbol{\omega}^\mathrm{B}$, the new belief masses for all singletons $i = 1, \ldots, n_\mathrm{I}$ and the overall uncertainty are obtained as:

$$\overline{b}_i = \frac{\sum\limits_{j \cap h = i} b_j^\mathrm{A} b_h^\mathrm{B}}{1 - \sum\limits_{(j \cap h) \notin \mathbb{I}} b_j^\mathrm{A} b_h^\mathrm{B}}, \ \forall i = 1, \ldots, n_\mathrm{I} \tag{3.1a}$$

$$\overline{\mu} = \frac{\mu^\mathrm{A} \mu^\mathrm{B}}{1 - \sum\limits_{(j \cap h) \notin \mathbb{I}} b_j^\mathrm{A} b_h^\mathrm{B}}, \tag{3.1b}$$

where $\mathbb{I} = \{1, \ldots, n_\mathrm{I}, \cup_{i=1}^{n_\mathrm{I}} i\}$ is the set comprising each singleton and the union of all singletons. The belief mass of each singleton is obtained summing contributions from every

combination that makes the singleton possible, and then belief masses are normalized leaving out the belief mass of combinations that do not uniquely identify a singleton. For some unrealistic degenerate cases, the denominator in (3.1) could be zero: in such cases, the combined opinion $\overline{\boldsymbol{\omega}}$ is set as the completely uncertain opinion, i.e., $\overline{\mu} = 1$ and all other belief masses equal to zero, since further specifications of the belief masses are not possible.

The procedure is commutative with respect to the order of the opinions and satisfies the following property.

Theorem 3.1. *The combined uncertainty $\overline{\mu}$ obtained from* (3.1) *cannot increase with respect to the two individual uncertainties μ^A and μ^B.*

Proof. We show that $\overline{\mu} \leq \mu^{\mathrm{B}}$, likewise it is obtained that $\overline{\mu} \leq \mu^{\mathrm{A}}$. Since $b_1^{\mathrm{B}}, \ldots, b_M^{\mathrm{B}} \geq 0$ and $\sum_{i=1}^{M} b_i^{\mathrm{B}} \leq 1$, it holds that

$$\sum_{(j \cap h) \not\subset \mathbb{I}} b_j^{\mathrm{A}} b_h^{\mathrm{B}} = \sum_{j=1}^{M} b_j^{\mathrm{A}} \sum_{\substack{h=1 \\ (j \cap h) \not\subset \mathbb{I}}}^{M} b_h^{\mathrm{B}} \leq \sum_{j=1}^{M} b_j^{\mathrm{A}} \sum_{h=1}^{M} b_h^{\mathrm{B}} \leq \sum_{j=1}^{M} b_j^{\mathrm{A}}. \tag{3.2}$$

Then, since from (2.27) $\mu^{\mathrm{A}} = 1 - \sum_{j=1}^{M} b_j^{\mathrm{A}}$, (3.1b) yields

$$\overline{\mu} = \frac{\mu^{\mathrm{A}} \mu^{\mathrm{B}}}{1 - \sum\limits_{(j \cap h) \not\subset \mathbb{I}} b_j^{\mathrm{A}} b_h^{\mathrm{B}}} = \frac{1 - \sum\limits_{j=1}^{M} b_j^{\mathrm{A}}}{1 - \sum\limits_{(j \cap h) \not\subset \mathbb{I}} b_j^{\mathrm{A}} b_h^{\mathrm{B}}} \mu^{\mathrm{B}} \leq \mu^{\mathrm{B}}, \tag{3.3}$$

that is, the uncertainty of the combined opinion is upper bounded by the uncertainty of each individual opinion. □

As a result, the greater the number of opinions considered, the smaller the uncertainty of the combination, following the idea that more independent sources make the information more reliable. However, this effect might also be undesirable, if the sources of information contradict one another. For this reason, we add a conflict detection mechanism, that reassigns part of the belief mass to the uncertainty of the combined opinion $\overline{\boldsymbol{\omega}}$ depending on the possible conflict among opinions. We define the conflict between two opinions as

$$C^{\mathrm{A,B}} = \frac{1}{2} \left\| \frac{\boldsymbol{b}^{\mathrm{A}}}{\|\boldsymbol{b}^{\mathrm{A}}\|_1} - \frac{\boldsymbol{b}^{\mathrm{B}}}{\|\boldsymbol{b}^{\mathrm{B}}\|_1} \right\|_1 \sqrt{(1 - \mu^{\mathrm{A}})(1 - \mu^{\mathrm{B}})}, \tag{3.4}$$

where $\boldsymbol{b}^{\mathrm{A}}$ and $\boldsymbol{b}^{\mathrm{B}}$ contain all belief masses of the opinions but the uncertainty component, i.e., $\boldsymbol{\omega} = \left[\boldsymbol{b}^{\top}, \mu\right]^{\top}$. $C^{\mathrm{A,B}}$ is designed to increase if the considered opinions assign differently the belief masses while having high confidence (small uncertainty). Consistency in the ratio of belief masses rather than values matters in the comparison of belief distributions, whereas uncertainty does not play a role, since it only scales belief masses. This is why belief masses are normalized before taking the difference in (3.4). The conflict is a non-negative quantity and $C^{\mathrm{A,B}} = 1$ for the extreme case of two sources completely certain ($\mu^{\mathrm{A}} = \mu^{\mathrm{B}} = 0$) and yet assigning the belief masses in completely contradictory ways, making the belief distributions orthogonal, for example $\boldsymbol{b}^{\mathrm{A}} = [1, 0, 0]^{\top}$ and $\boldsymbol{b}^{\mathrm{B}} = [0, 0.2, 0.8]^{\top}$.

Then, drawing from [52], [53], belief masses from $\overline{\boldsymbol{\omega}} = \left[\overline{\boldsymbol{b}}^\top, \overline{\mu}\right]^\top$ (3.1) are redistributed as

$$\tilde{\boldsymbol{b}} = \left(\prod_{i \neq j} \left(1 - C^{i,j}\right)\right)^{\frac{1}{n_\mathrm{S}}} \overline{\boldsymbol{b}}, \tag{3.5a}$$

$$\tilde{\mu} = 1 - \|\tilde{\boldsymbol{b}}\|_1. \tag{3.5b}$$

As for any two given sources i, j the conflict is $0 \leq C^{i,j} \leq 1$, and, thus, so it is also the coefficient in brackets in (3.5a). As a result, the adjusted belief masses $\tilde{\boldsymbol{b}}$ are non-larger than the previous $\overline{\boldsymbol{b}}$, conversely the uncertainty is $\tilde{\mu} \geq \overline{\mu}$. If the opinions of all sources completely match, i.e., $C^{i,j} = 0\ \forall i, j$, then there is no redistribution of belief mass to the uncertainty, that is $\tilde{\boldsymbol{b}} = \overline{\boldsymbol{b}}$, $\tilde{\mu} = \overline{\mu}$.

The resulting opinion $\tilde{\boldsymbol{\omega}} = \left[\tilde{\boldsymbol{b}}^\top, \tilde{\mu}\right]^\top$ gathers the whole information collected by different sources in the last sampling time.

3.2.4. Propagation of the Estimation over Consecutive Time Steps

After combining the information collected through different sources in the current time step, the resulting opinion is further combined with the information collected up to the previous step. The aim of this second combination is the temporal propagation of the information, allowing the detection of patterns that are revealed over multiple steps and providing a stable and reliable estimate preventing large fluctuations between consecutive time steps, which is of primary concern. Therefore, we adopt the Weighted Belief Fusion (WBF) operator [53].

Opinion $\tilde{\boldsymbol{\omega}}_t$ gathering information from the current time step and obtained through the procedure in Section 3.2.3 is combined with the overall estimate $\boldsymbol{\omega}_{t-1}$ from the previous time step, originating the overall estimate for the current time step $\boldsymbol{\omega}_t = \left[\boldsymbol{b}_t^\top, \mu_t\right]^\top$, with

$$\boldsymbol{b}_t = \frac{\tilde{\boldsymbol{b}}_t(1 - \tilde{\mu}_t)\mu_{t-1} + \boldsymbol{b}_{t-1}(1 - \mu_{t-1})\tilde{\mu}_t}{\tilde{\mu}_t + \mu_{t-1} - 2\tilde{\mu}_t\mu_{t-1}}, \tag{3.6a}$$

$$\mu_t = \frac{(2 - \tilde{\mu}_t - \mu_{t-1})\tilde{\mu}_t\mu_{t-1}}{\tilde{\mu}_t + \mu_{t-1} - 2\tilde{\mu}_t\mu_{t-1}}. \tag{3.6b}$$

Observe that from (3.6), if one of the two opinions is certain ($\mu = 0$), then the result coincides with that one, that is, the opinion not fully certain is neglected and the uncertainty of the combined opinion is also set to zero. If both opinions are certain but indicating different beliefs, then $\boldsymbol{\omega}_t$ is set to the completely uncertain opinion, to signal that the information is completely unreliable.

In general, the resulting opinion $\boldsymbol{\omega}_t$ is obtained as a weighted average of the two opinions $\tilde{\boldsymbol{\omega}}_t$, representing the combination of current estimations from the different sources, and $\tilde{\boldsymbol{\omega}}_{t-1}$, representing the result from the previous time step. In the weighted average, each of the two opinions $\tilde{\boldsymbol{\omega}}_t$ and $\tilde{\boldsymbol{\omega}}_{t-1}$ is accounted for depending on its uncertainty. Moreover, in this case the uncertainty does not necessarily decrease and for example the combination of two identical opinions results in the same opinion as an output, leaving the level of uncertainty unchanged. Indeed, the WBF operator [53] can be adopted under assumption of dependent sources, thus new opinions do not necessarily result in additional evidence [53].

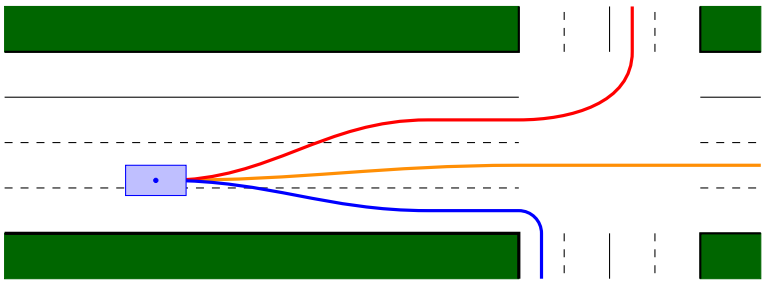


Figure 3.2.: Candidate intentions of the vehicle. © 2023 Frontiers.

3.2.5. Simulations of BFT-Based Intention Recognition in Autonomous Driving

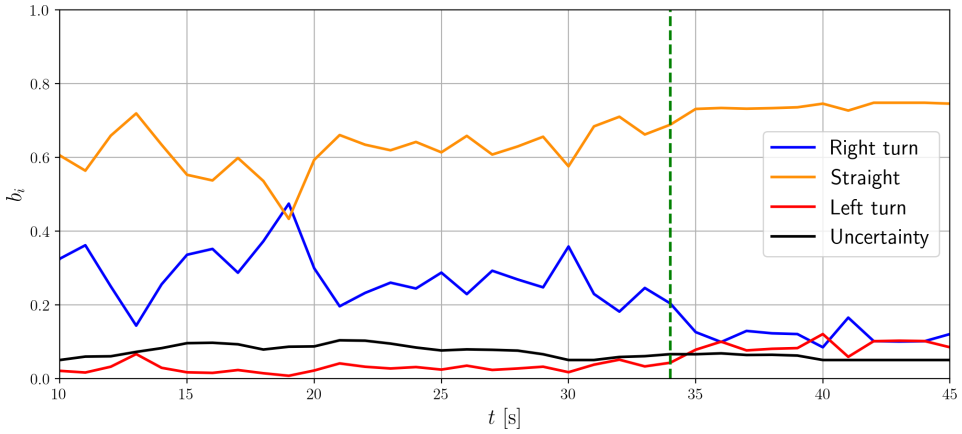
In this section we showcase the belief processing algorithm through numerical simulations in SUMO [153]. SUMO is a widely used open-source traffic simulation environment to validate and compare algorithms in urban and highway environments, providing a microscopic simulation platform that includes models of different means of transportation, including pedestrians, bicycles, and vehicles, providing a realistic representation of traffic dynamics.

We consider two scenarios in which the information provided by different sources is processed to estimate the intention executed by a vehicle approaching an intersection and a pedestrian. We consider both the case in which sources provide coherent individual estimations and the case with conflict among sources. We compare our BFT-based algorithm with the IMM algorithm. The IMM algorithm, which is recalled in Section 2.3.1, consists of one Kalman Filter for every candidate intention, and estimates from each filter are combined depending on the estimated probability, accounting for possible switches between intentions occurring between consecutive time steps.

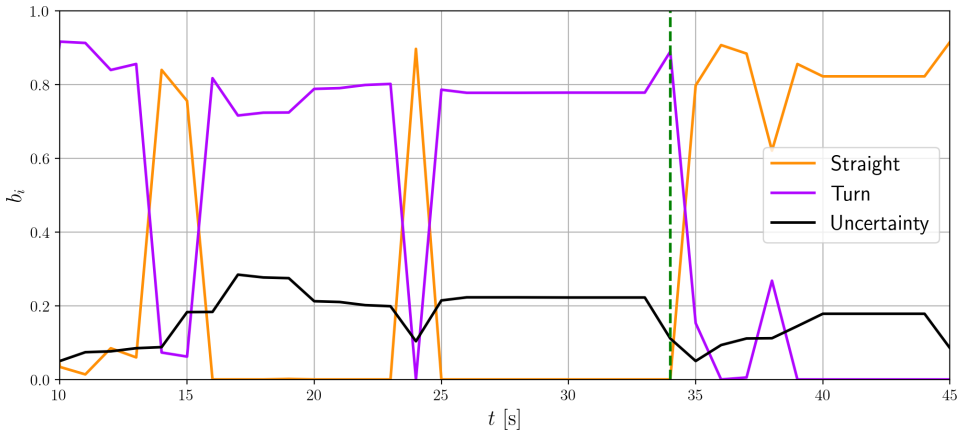
In the simulations, the BFT-based algorithm combines opinions generated by three sources: a sensor for the lateral position applying the procedure presented in the example on opinion generation in Section 2.3.5; a sensor for the longitudinal velocity also applying the procedure from the example in Section 2.3.5; a constant opinion representing traffic statistics. The IMM algorithm is implemented considering as measurements the lateral position and the longitudinal velocity. The motion of both the vehicle and the pedestrian is realized using the *TraCI* library in SUMO and noisy measurements are used in both estimation frameworks.

Uncertain Behavior of a Vehicle

In the first simulation, a vehicle approaches an intersection on a three-lane road and is detected by sensors of an automated vehicle position on the same road, behind. Alternatively, the lateral position and the longitudinal speed of the vehicle could be measured by sensors placed on the infrastructure at the intersection and communicated to connected vehicles. Finally, the traffic statistics representing the third opinion, as mentioned above, are assumed to be available from an online server. The candidate intentions are: A) right turn, B) proceed straight, C) left turn, see Figure 3.2. In reality, a larger number of candidate intentions could be included, so as to differentiate among, e.g., multiple rates for the lane change and the turn. However, a few considerations concerning the choice of multiple candidate intentions representing relatively similar future motions arise, as discussed in Section 4.3. We consider three candidate intentions, each intended as representative of a cluster of possible trajectories.



(a) Opinion from lateral position measurements. © 2023 Frontiers.



(b) Opinion from longitudinal velocity measurements. © 2023 Frontiers.

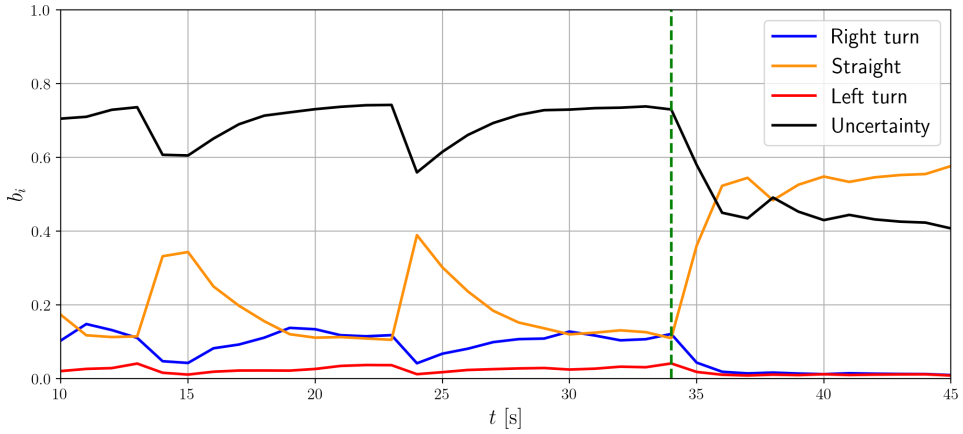
Figure 3.3.: Opinions generated by different sources. The vertical green line represents the moment when the vehicle proceeds straight at the intersection.

The vehicle exhibits an ambiguous behavior as it approaches the intersection, possibly simulating a distracted driver. The vehicle stays in the center lane, as to proceed straight, but not at the center, rather proceeds irregularly and slightly to the right of the lane, close to the edge. Thus, the opinion based on the lateral position regards going straight as the most likely intention, although the belief mass of a right turn maneuver is non-negligible, as depicted in Figure 3.3a. Furthermore, the vehicle slows down, therefore the opinion based on the longitudinal velocity tends to give a higher belief mass to the turn maneuver, as in Figure 3.3b. This intention is the union of the left and of the right turn maneuvers, that are not distinguishable considering the longitudinal velocity alone. Finally, the BFT algorithm includes a constant bias $\boldsymbol{\omega} = [b_{\text{right}}, b_{\text{straight}}, b_{\text{left}}, \mu]^\top = [0.18, 0.32, 0.17, 0.33]^\top$.

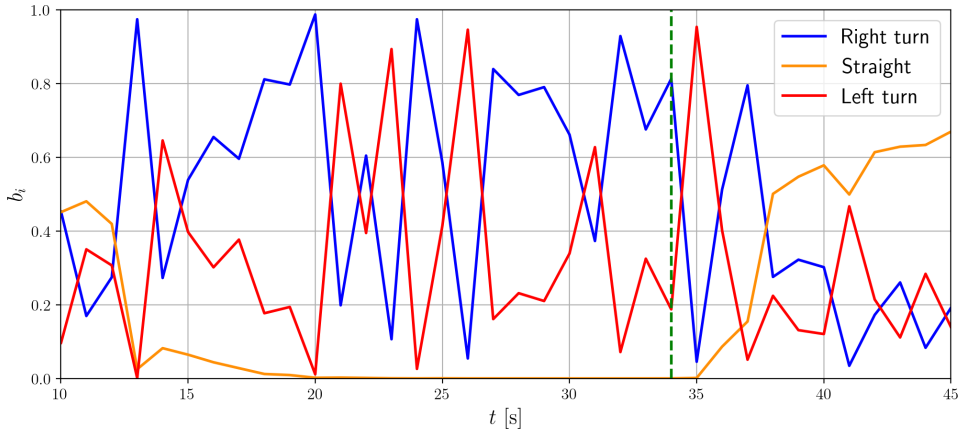
The result of the estimation for the two frameworks is presented in Figure 3.4. While the vehicle is still approaching the intersection and showing an ambiguous behavior, the opinions produced by the three sources are conflicting. The combination provided by our BFT-based algorithm generally assigns the relative highest belief to the *proceed straight* intention, which is altogether to be considered the most likely. However, the uncertainty is very high, quantitatively expressing that a reliable estimate cannot be extracted from the

current data. Therefore, a motion planner based on the BFT estimate might make use of the uncertainty information to use caution while the intention of this DO is not clear and the prediction could not be reliable. However, once the vehicle actually reaches the intersection, the conflict between sources is resolved and the belief mass of the proceed straight intention gradually increases and the uncertainty decreases.

Conversely, the IMM estimation is extremely noisy and shows large and repetitive fluctuations even between consecutive time steps, since none of the models can consistently explain the data. Moreover, confidence oscillates between right turn and left turn, whereas the proceed straight intention only emerges as dominant when the vehicle reaches the intersection. As a matter of fact, the IMM considers the dynamics of the model, thus the irregular behavior is repeatedly interpreted as the beginning of a turning intention. The IMM estimation is not reliable enough to be used for prediction of DO future trajectories.



(a) Combined opinion from the BFT-based algorithm. © 2023 Frontiers.



(b) Probabilities estimated by the IMM. © 2023 Frontiers.

Figure 3.4.: Probability estimate produced by the two frameworks. The vertical green line represents the moment when the vehicle proceeds straight at the intersection.

Clear Behavior of a Pedestrian

In the second simulation, the two frameworks are compared in estimating the intention of a pedestrian, initially located on the grass in the vicinity of the road. Also this scenario assumes

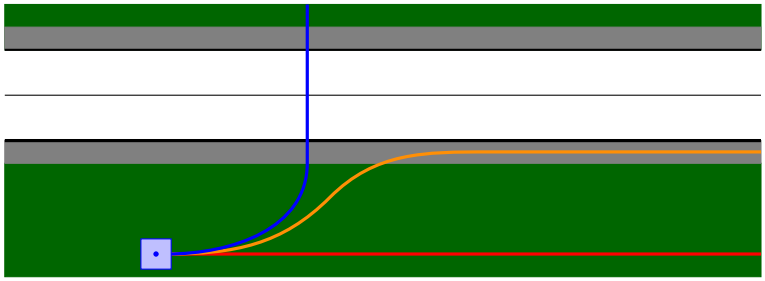


Figure 3.5.: Candidate intentions of the pedestrian. © 2023 Frontiers.

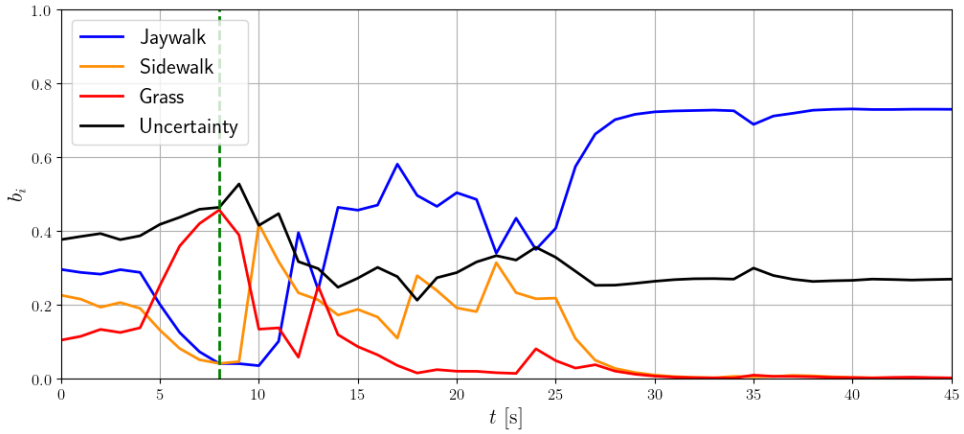
that the position and speed of the pedestrian are detected by sensors of the intelligent infrastructure on the side of the road or by sensors of an automated vehicle on the street, moving from left to right, that may potentially cross the path of the pedestrian. The three candidate intentions are: A) remain on the grass, B) move to the sidewalk, C) jaywalk, as represented in Figure 3.5. For this simulation, both the BFT-based algorithm and the IMM collect measurements of the lateral velocity, rather than of the lateral position. Furthermore, the bias is set to $\boldsymbol{\omega} = [b_{\text{grass}}, b_{\text{sidewalk}}, b_{\text{jaywalk}}, \mu]^\top = [0.36, 0.22, 0.1, 0.32]^\top$, to discourage the combined BFT opinion from getting to the conclusion that the pedestrian is indeed attempting to cross the road outside of the crosswalk, as normally is not the case.

The results of the estimation for the two frameworks are presented in Figure 3.6. Despite discouraged by the bias opinion, the BFT algorithm quickly recognizes that the pedestrian wants to jaywalk. The two sensor-based opinions generate coherent estimates, thus the uncertainty of the combination is mainly due to partially contradicting measurements, because of the noise, and due to conflicts with the bias opinion. The latter, however, has a large uncertainty, being based on statistics. Therefore, the conflict handling mechanism introduced in Section 3.2.3 does not severely affect the estimation and the uncertainty of the combined estimate is considerably smaller than in the previous simulation. Finally, the IMM also promptly recognizes that the pedestrian will jaywalk. However, due to the lack of the uncertainty component in the estimate, model mismatches and noisy measurements result in considerably large fluctuations in the estimated probabilities, requiring a post-processing of the estimate to smooth the results.

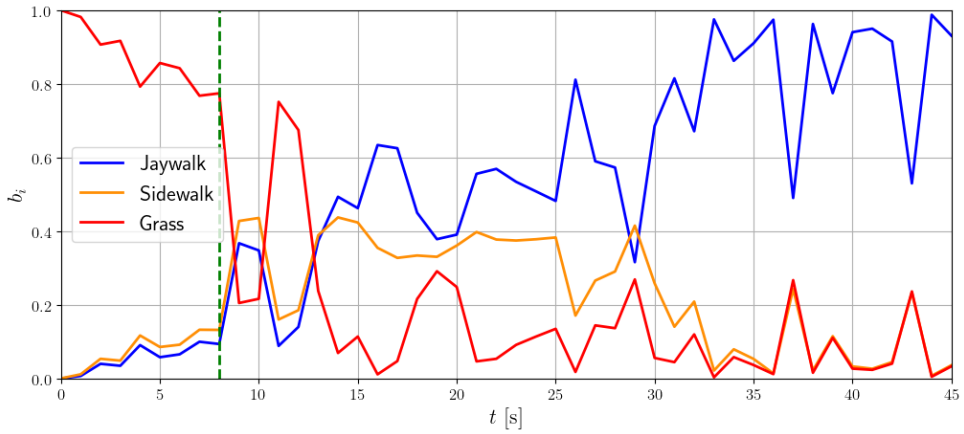
3.2.6. Discussion

Classical estimation mechanism such as the IMM do not explicitly quantify probability uncertainty, rather model mismatches result in large estimation variance for the state estimate, which, depending on the application, might be impractical. Moreover, different sensors are included in the IMM algorithm as different outputs of the model. Thus, the handling of the degree of uncertainty of different sources in the IMM is restricted to specifying different statistical properties of the disturbances in the underlying models. However, such statistical properties cannot be straightforwardly adapted online depending on the data collected over time, thus the IMM algorithm is less suited to cope with sources with different reliability. Furthermore, conflict between information provided by each sensor is not explicitly addressed and reflects in large fluctuations in the estimate yielded at consecutive time steps.

In practice, a discrete set of mutually exclusive trends of realizations of the uncertainty might be difficult to identify. For example, the pedestrian might cross the road at different



(a) Combined opinion from the BFT-based algorithm. © 2023 Frontiers.



(b) Probabilities estimated by the IMM. © 2023 Frontiers.

Figure 3.6.: Probability estimate produced by the two frameworks. The vertical green line represents the moment when the pedestrian starts moving toward the road.

locations. To account for this, further candidate trajectories could be included, each considering a different crossing location. Furthermore, once the most likely nominal crossing location has been recognized, an ad hoc model could be fit, so that the exact crossing location over a continuous range can be identified, instead of relying only on a few nominal candidate locations. These ideas are developed in Section 3.3.

An accurate estimation of the probability of outcomes to be addressed plays a major role in decision making. In the example of collision avoidance constraints in motion planning of automated vehicles, estimating the probability of the intention of DOs allows to focus on the most likely future DO trajectories, as it is discussed in Section 4.3. For such a scope, the BFT framework provides a quantification of the uncertainty, which can be used to tighten or relax constraints depending on whether the estimated probabilities are reliable, i.e., preventing aggressive maneuvers from being taken if the motion of the DOs is not clearly recognizable yet. This approach is discussed in Section 4.4.

3.3. Deriving Multi-Structure Models to Describe Multiple Modes of the Uncertainty

This section addresses the problem of estimating the most likely model describing the observed data, while also estimating the free parameters of each candidate model. Our SPME method combines MLE and the IMM algorithm. Further, we demonstrate how it is applied to the context of intention estimation of DO. This section is based on the work presented in [19].

3.3.1. Problem Formulation

We consider a set of n_I linear, discrete-time candidate models $m^{(j)}$, $j \in [1, \ldots, n_\mathrm{I}]$, each of the form (2.5), that is, suitable for the IMM algorithm, which is recalled in Section 2.3.1. The state of each model is $\boldsymbol{z}^{(j)}$ and each model is described by matrices $\boldsymbol{F}^{(j)}$, $\boldsymbol{G}^{(j)}$ and $\boldsymbol{H}^{(j)}$, as well as the input vector $\boldsymbol{u}^{(j)}$. Each model dynamics includes a disturbance $\boldsymbol{w}^{(j)}$. All models share the same output vector $\boldsymbol{\gamma}$ and measurement noise $\boldsymbol{v}$.

These models, sometimes referred to as hypotheses, represent possible system *structures* due to different behaviors or failures. On the one hand, models can switch within time steps. On the other hand, models include free parameters included in vector $\boldsymbol{\theta}^{(j)}$, that may change over time. Changes occur in case of a fault, when a new model with a different parameter vector becomes more accurate, or when the parameter vector of the same model adapts to the new behavior. Matrices $\boldsymbol{F}^{(j)}$, $\boldsymbol{G}^{(j)}$ and $\boldsymbol{H}^{(j)}$ and the input vector $\boldsymbol{u}^{(j)}$ are model-specific and, in general, dependent on the parameter vector $\boldsymbol{\theta}^{(j)}$. This makes it possible to further differentiate between behaviors of similar forms. For example, given a model representing a specific fault, the free parameters $\boldsymbol{\theta}^{(j)}$ describe the magnitude of the fault, for a fixed type of fault. These free parameters of each model are constantly re-estimated, to adapt the models when the observed behavior changes. For notation simplicity, the time dependence of parameters $\boldsymbol{\theta}^{(j)}$ is omitted. However, parameters are always understood as the "current" parameters, subject to frequent changes

Assumption 3.1. *The system disturbance and the measurement noise are Gaussian distributed,* $\boldsymbol{w}^{(j)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{w}^{(j)}})$ *and* $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{\nu}})$*. They are white and mutually independent for all times, therefore* $Cov(\boldsymbol{w}_h, \boldsymbol{w}_k) = Cov(\boldsymbol{v}_h, \boldsymbol{v}_k) = \boldsymbol{0}\ \forall h \neq k$*, and* $Cov(\boldsymbol{w}_h, \boldsymbol{v}_k) = \boldsymbol{0}\ \forall h, k$*.*

Further, it is assumed that a-priori probabilities of switching between models are given in the form of a row-stochastic matrix $\boldsymbol{\Pi} \in \mathbb{R}^{n_\mathrm{I} \times n_\mathrm{I}}$, where $\pi_{i,j} = \Pr(m_{t+1}^{(j)} | m_t^{(i)})$ defines the probability to switch from model $m^{(i)}$ to $m^{(j)}$ in the next time step. In general, application-specific considerations are used to design at least a nominal guess of such a-priori switching probabilities by incorporating all available information.

We assume that output measurements from the past n_w time steps are available and stored. Such data must be used both to identify the parameters and to assess which model is considered more likely to predict the trajectory for further prediction steps, yielding the following objective.

Objective 3.2. *The goal is iteratively identify the free parameters* $\boldsymbol{\theta}^{(j)}$ *of each model and, simultaneously, estimate the probability of each model for the current time step, relying on the output measurements from the past* n_w *time steps, and obtaining models that generalize for further prediction steps.*

3.3.2. Overview of the Proposed SPME Scheme

This section discusses the SPME algorithm and two modifications to address overfitting. The approach involves employing MLE and IMM to determine the best-fitting models and parameters based on a measured noisy trajectory. First, we outline the proposed algorithm and discuss potential issues with the approach. Subsequently, two modifications to this method are introduced to address the mentioned issues, by introducing a regularization term and by reversing parts of the algorithm. The IMM algorithm and MLE are recalled in Section 2.3.1 and in Section 2.3.2, respectively.

At each sampling time, the SPME algorithm consists of the following five steps:

1. **Model Parameter Estimation (MPE)**: In the initial step, MLE of the unknown parameters is conducted. This process takes place in all models in parallel, leveraging the current data window and an estimated vector for the initial state, $\hat{\boldsymbol{z}}_0$, obtained from the previous run of the IMM. It provides best-fitting parameters for all models, resulting in a set of models that best explain the observed trajectory within their defined structures.

2. **Model Fusion**: Under certain conditions, multiple models describe a very similar behavior, potentially hindering IMM performance. Hence, the similarity of these models is assessed, based on predictions of the state given the current state and the current models. If the Euclidean distance between the predictions over n_{pred} prediction steps is smaller than a threshold r_{pred}, the models are treated as identical, and their probabilities are aggregated.

   Example 3.1 (Similar Models in Target Tracking)**.** *If the goal is to walk along a line in x-direction at a specific y-position, and the target has already reached the correct y-position, the behavior is either described by tracking the desired y-position or by setting the desired velocity in y-direction to 0. At steady state, these two models are indistinguishable, making it irrelevant to look for switches among them. Thus, logically indistinguishable models are merged.*

3. **IMM**: Subsequently, the IMM algorithm is initiated using the determined model set with $\boldsymbol{\theta}^{\text{ML},(j)}$. The state estimation $\hat{\boldsymbol{z}}$ along with a model probability distribution $\boldsymbol{\mu}$ are provided. If two or more models were identified as very similar in Step 2), the individual probabilities of different models are replaced with their sum. Nevertheless, models are always handled as distinct in the IMM algorithm.

4. **FDD**: If a switch to another model occurs, a fault is detected, and since the model describes the fault, the fault is also isolated. Furthermore, if no switching to another model occurs, the magnitude of the change in the parameters $\hat{\boldsymbol{\theta}}^{\text{ML},(j)}$ is investigated, and a certain threshold could be defined to trigger fault detection.

5. **Data Shift**: At the next time step, the MLE data window is shifted forward, and the algorithm is repeated.

Figure 3.7 shows a simplified overview of the algorithm, in which for simplicity the model fusion is included in the IMM block. The proposed algorithm is shown in orange. The only input is the measurement of the most recent data points, depicted in green. The output is marked in blue. The SPME algorithm consists of two parts: the model parameter

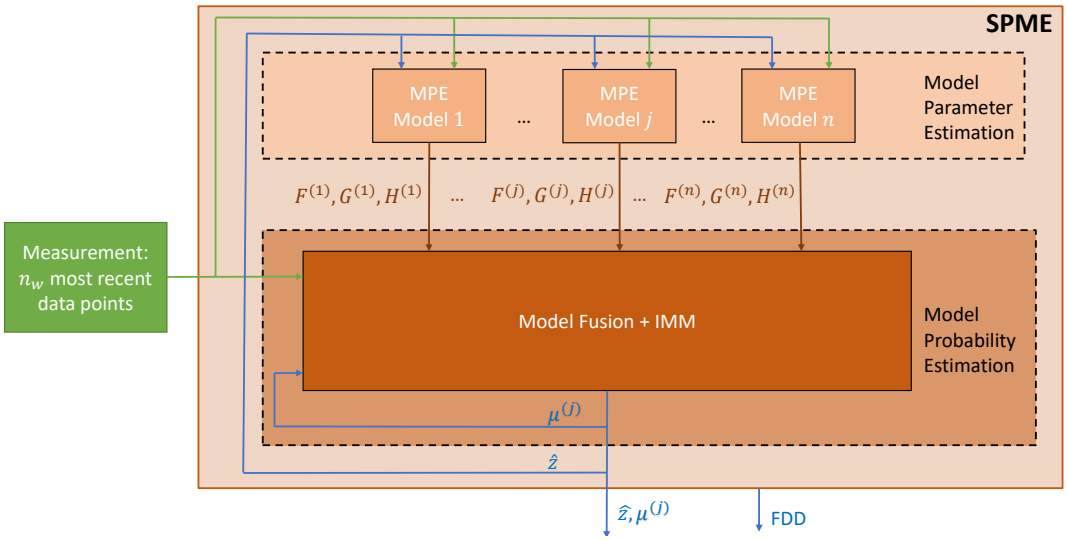


Figure 3.7.: Components of the SPME algorithm: model parameter estimation and model probability estimation, both using as dataset the last n_w data points.

estimation for each defined model and the model probability estimation using the IMM. This combination provides an accurate estimation of the current state and a probability distribution for the identified best-fit models. It also enables the detection and isolation of unknown faults or other structural changes in a system.

However, as previously mentioned, employing time-varying models with new estimations for $\boldsymbol{\theta}^{(j)}$ in the IMM at every sampling time introduces *overfitting*. A short data window allows different models to fit the data sufficiently well. Yet, checking the prediction over a long horizon reveals that the models are different, and model fusion does not apply. This behavior arises from the frequent update of models at every time step and is challenging, especially for integrator states.

Example 3.2 (Issues of SPME in Target Tracking)**.** *At a measured x-position and a sampling time of* $1\,\mathrm{s}$*, for a true velocity of the target at* $3\,\mathrm{m/s}$ *in x-direction two different candidate models are available, namely model M1 with reference* $v_x^r = 3\,\mathrm{m/s}$ *and model M2 with a constant position reference x^r. In a short data window, the MLE fits parameters so that both models are accurate, and the IMM assigns a high probability to both. When the models are updated, the reference parameter of M1, v_x^r, remains around* $3\,\mathrm{m/s}$*, while the reference parameter of M2, x^r, grows at a rate of* $3\,\mathrm{m/s}$*, and both model probabilities stay high.*

This simple example is illustrated in Figure 3.8. In the upper part, the boxes represent the position of the object for the initial condition ($k = 0$) and the prediction obtained using M1 ($k = 1, 2, 3$). We consider two consecutive sampling times $t = 0\,\mathrm{s}$ *and* $t = 1\,\mathrm{s}$*. M1 reflects a motion with constant target speed, which is estimated to be* $v_x^r = 3\,\mathrm{m/s}$ *both at time step* $t = 0\,\mathrm{s}$ *and* $t = 1\,\mathrm{s}$*. Then, in both cases, the boxes are roughly equally spaced every* $3\,\mathrm{m}$*, i.e., the distance traveled in a sampling time. Similarly, the lower part of the figure shows the predicted positions obtained using M2. M2 represents a motion with a target fixed position x^r. Therefore, the boxes are not equally spaced within the prediction horizon since the object is expected to slow down as it approaches the target position. However, at sampling time* $t = 0\,\mathrm{s}$ *the target position is estimated to be* $x^r = 10\,\mathrm{m}$*, whereas at sampling time* $t = 1\,\mathrm{s}$ *it is estimated to be* $x^r = 13\,\mathrm{m}$*.*

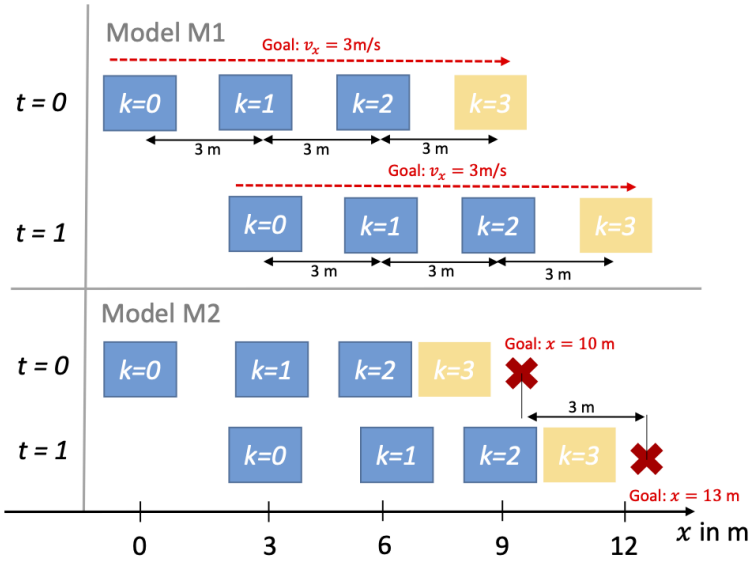


Figure 3.8.: Example of the overfitting problem of SPME in target tracking. For a short observed horizon $k = 0, 1, 2$, different models, M1 and M2, can describe a similar behavior by appropriately changing the model parameters at each time step, e.g., the target x in this example. Thus, it is impossible to identify the correct model by just focusing on a short data sequence.

By comparing the prediction obtained by M1 and M2, it is observed that for the first prediction steps the two trajectories are roughly the same (blue boxes, $k = 0, 1, 2$), whereas, at the end of the considered horizon, the prediction differs significantly (yellow boxes, $k = 3$). This reflects the fact that the two models can be tuned in a way to fairly reproduce the first short sequence of collected data, generating overfitting. Nevertheless, the prediction of the two models greatly differs for further prediction steps.

In summary, problems arise only when incorrect estimates appear plausible due to the short time considered. We assume that faults do not occur in quick succession, thus the parameters in the "correct" model do not change often. Rather, repeated change of the parameters in a model is understood as consequence of overfitting, and the probability for this model should be reduced. Another option to address overfitting is to use different datasets in the MLE and IMM such that inconsistencies lead to low probabilities for the non-fitting models. In the following, two adjustments based on the mentioned principles are formulated. It is worth noting that extending the MLE time horizon would also solve the overfitting problem in some cases, but we focus on short time windows due to the computationally expensive nature of the MLE. Therefore, this option is not considered.

First, we use a regularization approach within the IMM that considers the rate of change in the MLE. The IMM is designed to recognize significant changes in the MLE and assign lower probabilities when substantial changes occur in the estimates over a defined horizon. The modified algorithm, called Regularized Simultaneous Parameter and Model Estimation (RegSPME), is summarized as:

1. **MPE**:
   a) Identical to SPME Step 1)

b) **Rate of MPE change**: The rate of change in the estimations $\Delta\boldsymbol{\theta}_t^{(j)}$ for each model j at time step t is defined as the sum of the difference in the estimated parameters over n_Δ time steps

$$\Delta\boldsymbol{\theta}_t^{(j)} = \sum_{k=t-n_\Delta}^{t} |\hat{\boldsymbol{\theta}}_k^{\mathrm{ML},(j)} - \hat{\boldsymbol{\theta}}_{k-1}^{\mathrm{ML},(j)}|, \tag{3.7}$$

where the absolute value applies component-wise.

2. **Model Fusion** - identical to SPME Step 2)

3. **IMM**: For regularization, the standard IMM as outlined in SPME Step 3) is adapted. The factors $\Lambda^{(j)}$ from (2.9) now include the rate of change $\Delta\boldsymbol{\theta}^{(j)}$

$$\Lambda^{(j)} = \frac{\exp\left(-0.5\left(\|\boldsymbol{\epsilon}^{(j)}\|^2_{\tilde{\boldsymbol{S}}^{(j)})^{-1}} + \|\Delta\boldsymbol{\theta}^{(j)})\|^2_{\boldsymbol{D}}\right)\right)}{\sqrt{\det(2\pi\tilde{\boldsymbol{S}}^{(j)})\det(\boldsymbol{D}^{-1})}}, \tag{3.8}$$

with weighting matrix $\boldsymbol{D}$. This formulation achieves a balanced trade-off for the probability distribution over the models, considering both the adjustments required for the models at each time step and their accuracy in describing the observed behavior.

4. **FDD**: identical to SPME Step 4)

5. **Data Shift**: identical to SPME Step 5)

The second method, which is alternative to RegSPME, involves using a different and time-shifted dataset in the IMM than what is used in the MLE. This also eliminates the high probability of a wrong model being valid for a long time. The MLE estimate is based on a dataset that is shifted backward, while IMM uses the most recent data. This approach, Reverse Simultaneous Parameter and Model Estimation (RevSPME), effectively reverses the algorithm, with IMM estimation being performed first, followed by MLE.

In order to enhance the impact of an inaccurate estimate in the MLE, the IMM algorithm is run over multiple prediction steps. This involves using an initial state that is n_{rev} before the current time, predicting the current expected state with the models, and then comparing it with the current measurement. The state prediction in the IMM algorithm now reads

$$\tilde{\boldsymbol{z}}_k^{(j)} = (\boldsymbol{F}^{(j)})^{n_{\mathrm{rev}}}\boldsymbol{z}_{k-n_{\mathrm{rev}}}^{(j)} + \sum_{h=0}^{n_{\mathrm{rev}}-1}(\boldsymbol{F}^{(j)})^{n_{\mathrm{rev}}-1-h}\boldsymbol{G}^{(j)}\boldsymbol{z}_r. \tag{3.9}$$

The prediction now involves multiple steps and $\boldsymbol{z}_{k-n_{\mathrm{rev}}}^{(j)}$ represents the initial state $k - n_{\mathrm{rev}}$ steps ago. The estimated parameters $\hat{\boldsymbol{\theta}}_k^{\mathrm{ML},\,(j)}$ in the models are kept constant for these internal predictions. The output prediction error is then computed as in (2.8c)

$$\boldsymbol{\epsilon}_k^{(j)} = \boldsymbol{\gamma}_k^{\mathrm{real}} - \boldsymbol{H}^{(j)}\tilde{\boldsymbol{z}}_k^{(j)}. \tag{3.10}$$

In summary, the updated RevSPME algorithm consists of:

0. **Initialization**:

a) Initial MPE - identical to SPME Step 1)

b) Model Fusion - identical to SPME Step 2)

1. **IMM**: At time step t, compute the model probabilities by running the IMM for all data points in the selected window, starting from $\boldsymbol{z}_{t-n_{\text{rev}}}^{(j)}$ and keeping the estimated model parameters $\hat{\boldsymbol{\theta}}_t^{\text{ML},\,(j)}$ constant.
2. **FDD** - identical to SPME Step 4)
3. **MPE** - identical to SPME Step 1) but now again with the *old* data up to step t.
4. **Model Fusion** - identical to SPME Step 2)
5. **Data Shift**: identical to SPME Step 5)

In summary, the updated algorithms enable precise estimation of the current state and an early detection and isolation of faults. Restricting the number of parameters to be estimated in each model limits the computational cost and maintaining multiple models with specific structures, each representing a distinct fault, ensures results remain interpretable.

3.3.3. SPME for Motion Tracking

We apply our new algorithms to the example of a moving object in an unstructured environment, such as a warehouse. In this example, abrupt changes in the intention of the object (e.g., a robot or a human) are common. However, depending on the environment the object is interacting with, e.g., other robots and humans, such changes are critical to detect and identify early to avoid dangerous collisions and to enable efficient path planning. This example is noteworthy because the nominal model is unknown, a scenario not addressed by other MM algorithms.

As candidate intention-based prediction models of the future motion of DOs, we use the closed-loop dynamical models introduced in Section 2.3.4. Such models are in a suitable form to be included in the IMM algorithm. We account for trajectory uncertainty about each candidate intention, to consider that the real motion of the DOs might not precisely match any of the candidate nominal trajectories representing intentions.

The DO is represented by its state $\boldsymbol{z} = [x, v_x, y, v_y]^\top$, where x and y are the coordinates of the position of the DO, and v_x and v_y the longitudinal and lateral velocity, respectively. The motion is modeled as a simple point-mass model with decoupled longitudinal and lateral dynamics [43]. The input $\boldsymbol{u} \in \mathbb{R}^2$ from (2.21) depends on the difference between the current DO state $\boldsymbol{z}$ and the goal target state $\boldsymbol{z}^{(\text{r},j)}$ for the assumed intention.

The objective is the identification of the unknown intentions by using SPME, which includes estimating the unknown parameters $\boldsymbol{\theta}^{(j)}$ contained in the reference state $\boldsymbol{z}^{\text{r},(j)}$. This means $\boldsymbol{z}^{\text{r},(j)}$ becomes $\boldsymbol{z}^{\text{r}}(\boldsymbol{\theta}^{(j)})$. In general, only components of $\boldsymbol{z}^{\text{r},\,(j)}$ are "free" parameters.

Example 3.3 ("Free" Parameters in Target Tracking)**.** *If the moving object tracks a constant position in y-direction while maintaining a constant velocity in x-direction, the resulting parameter vector is* $\boldsymbol{\theta} = [v_x^r, y^r]^\top$*. This implies that* $v_y^r = 0$ *and that* x^r *is irrelevant for the controller. As a result, these variables are excluded from the estimation process, reducing the size of the estimation problem.*

Measurements from the closed-loop models (2.25) are used to estimate the reference state $\boldsymbol{z}^{(\mathrm{r},j)}(\boldsymbol{\theta}^{(j)})$, which is supposed to be constant over time. If faults or changes in the behavior of the system happen, the parameters $\boldsymbol{\theta}^{(j)}$, however, switch to a new constant value.

In order to solve the MPE problem (2.11), the probability distribution of the output $\boldsymbol{\gamma}^{(j)}$ needs to be known. The model dynamics is linear, and the disturbances $\boldsymbol{w}$ and $\boldsymbol{v}$ are Gaussian. Thus, the output $\boldsymbol{\gamma}$ is an affine function of Gaussian random vectors and is itself Gaussian. The Gaussian probability density function is used to obtain the log-likelihood of the parameters $\boldsymbol{\theta}^{(j)}$. For this purpose, it is necessary to determine the mean and covariance of $\boldsymbol{\gamma}^{(j)}$. The following derivation holds for each model. Therefore, the notation is simplified in the following by omitting the model index j.

Integrating (2.25) over time yields, for $k = 1, \ldots, n_\mathrm{W}$,

$$\boldsymbol{z}_k = \boldsymbol{F}^{n_\mathrm{W}} \boldsymbol{z}_0 + \sum_{\tau=0}^{k-1} \boldsymbol{F}^{k-1-\tau} \left[\boldsymbol{G} \boldsymbol{z}^\mathrm{r}(\boldsymbol{\theta}) + \bar{\boldsymbol{w}}_\tau \right], \tag{3.11}$$

where $\boldsymbol{z}^\mathrm{r}(\boldsymbol{\theta})$ is assumed to be constant and deterministic. The state $\boldsymbol{z}_0$ at the beginning of the samples window is not known. It is modeled as a random vector with mean and covariance equal to the mean $\hat{\boldsymbol{z}}_0$ and covariance $\hat{\boldsymbol{P}}_0$ of the combined estimates provided by the IMM at that sampling time. The same $\hat{\boldsymbol{z}}_0$ and $\hat{\boldsymbol{P}}_0$ are used for all models. Notably, the random vector $\boldsymbol{z}_0$ depends on measurements collected up to sampling time $t - n_\mathrm{W}$. Thus, it is not correlated to $\boldsymbol{w}_\tau$ and $\boldsymbol{v}_\tau$ for $\tau = t - n_\mathrm{W} + 1, \ldots, t$.

From the linearity of the expected value, it follows that

$$\mathbb{E}[\boldsymbol{\gamma}_k] = \boldsymbol{H} \boldsymbol{F}^{n_\mathrm{W}-1} \hat{\boldsymbol{z}}_0 + \boldsymbol{H} \tilde{\boldsymbol{F}} \boldsymbol{G} \boldsymbol{z}^\mathrm{r}(\boldsymbol{\theta}), \tag{3.12a}$$

$$\tilde{\boldsymbol{F}} = \sum_{\tau=k-n_\mathrm{W}+1}^{k-1} \boldsymbol{F}^{k-1-\tau}. \tag{3.12b}$$

In order to compute the covariance of two output vectors $\boldsymbol{\gamma}_{k_1}$ and $\boldsymbol{\gamma}_{k_2}$, it is worth observing that they are not uncorrelated. In fact, although disturbances $\boldsymbol{w}$ and $\boldsymbol{v}$ are uncorrelated for all times, the output vectors $\boldsymbol{\gamma}_{k-1}$ and $\boldsymbol{\gamma}_k$ both depend on $\boldsymbol{w}_h$, $h = 0, \ldots, k-2$. Assumption 3.1 implies $\mathrm{Cov}(\boldsymbol{w}_{h_1}, \boldsymbol{v}_{h_2}) = \boldsymbol{0}\ \forall h_1 \neq h_2$. Thus, (3.11) and the properties of the covariance yield, for $1 \leq k_1 \leq k_2 \leq n_\mathrm{W}$,

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{\gamma}_{k_1}, \boldsymbol{\gamma}_{k_2}) = {} & \boldsymbol{H} \boldsymbol{F}^{k_1} \hat{\boldsymbol{P}}_0 (\boldsymbol{F}^{k_2})^\top \boldsymbol{H}^\top + \boldsymbol{R} \delta(k_1 - k_2) \\ & + \sum_{\tau=0}^{k_1-1} \boldsymbol{H} \boldsymbol{F}^{k_1-1-\tau} \boldsymbol{\Sigma}^{\boldsymbol{w}(j)} \left(\boldsymbol{F}^{k_2-1-\tau} \right)^\top \boldsymbol{H}^\top, \end{aligned} \tag{3.13}$$

where $\delta(k)$ is the Dirac delta function, i.e., it is 1 if $k = 0$ and 0 otherwise, that is, the term $\boldsymbol{R}$ in (3.13) appears only for the covariance of a given output vector with itself, i.e., $k_1 = k_2$.

With (3.12a) and (3.13), the joint probability of the last n_W outputs $\underline{\boldsymbol{\gamma}}$ is defined, where the only missing parameter is vector $\boldsymbol{\theta}$. Then, $\boldsymbol{\theta}$ is chosen such that it maximizes the probability of the actually measured samples $\underline{\boldsymbol{\gamma}}^\mathrm{real}$ as in (2.11) [144].

Remark 3.3. *The methods presented do not require linear models and Gaussian disturbances. Yet, computing of the required probabilities without such assumptions is challenging.*

3.3.4. Simulations of Model and Intention Recognition for a Mobile Robot in an Unstructured Environment

First, the simulation setup in the context of motion tracking is explained in order to then show the results of these simulations regarding the proposed algorithms. Then the performance of the proposed algorithms against each other is compared.

Simulation Setup

In the following, all quantities are given in SI units. The four candidate models for the intended motion, each with different target state references, are:

a) target x-position and target y-velocity (follow a straight line in y-direction),

b) target x-position and target y-position (reach a goal point),

c) target x-velocity and target y-position (follow a straight line in x-direction),

d) target x-velocity and target y-velocity (follow a straight line in $x-y$-direction).

In order to test all use cases and to show the adaptability, detection, and isolation speed of the algorithms, the following trajectory for time steps $t = 0, \ldots, 220$ is used, which alternates between the intentions every 50 seconds:

(i) $0 \leq t < 50$: model d) with $v_x^{\mathrm{r}} = v_y^{\mathrm{r}} = 2$.

(ii) $50 \leq t < 100$: model a) with $x^{\mathrm{r}} = 100,\ v_y^{\mathrm{r}} = 2$. This model becomes equal to d) with $v_x^{\mathrm{r}} = 0,\ v_y^{\mathrm{r}} = 2$ when the correct x-position is reached.

(iii) $100 \leq t < 150$: model a) with $x^{\mathrm{r}} = 130,\ v_y^{\mathrm{r}} = 2$. This model becomes equal to d) with $v_x^{\mathrm{r}} = 0,\ v_y^{\mathrm{r}} = 2$ when the correct x-position is reached.

(iv) $150 \leq t < 200$: model c) with $v_x^{\mathrm{r}} = 2,\ y^{\mathrm{r}} = 200$. This model becomes equal to d) with $v_x^{\mathrm{r}} = 2,\ v_y^{\mathrm{r}} = 0$ when the correct y-position is reached.

(v) $200 \leq t < 220$: model b) with $x^{\mathrm{r}} = 300,\ y^{\mathrm{r}} = 300$. When the reference is reached, all models are equal, as all model parameters are then estimated to be zero.

Not only does the trajectory contain all four different behaviors, but it also shows a switch where the model shape remains the same, but the reference values change (from segment (ii) to (iii), both model a) behavior).

The sampling time is set to $T = 1$, and a controller of the form (2.21) is used. Additionally, Gaussian noise is applied to the state with $\boldsymbol{w} = \mathrm{diag}(0.1, 0.1)$ and to the measurement with $\boldsymbol{v} = \mathrm{diag}(0.1,\ 0.1)$. The simulated *real* tracked trajectory of these references is shown in Figure 3.9. The goal is to detect intention changes and identify the underlying models.

For the three proposed algorithms, SPME, RegSPME, and RevSPME, the target parameters in each model a) - d) are iteratively estimated using MLE based on the most recent n_{W} measurements, where n_{W} is varied between 2 and 11. In the MPE step of the algorithms, the velocities v_x and v_y are bounded to $[-200, 200]$. Matrix $\boldsymbol{\Pi}$ in the IMM algorithm is designed by fixing a certain probability of model persistence, $\pi_{i,i} = 0.7$, and then setting

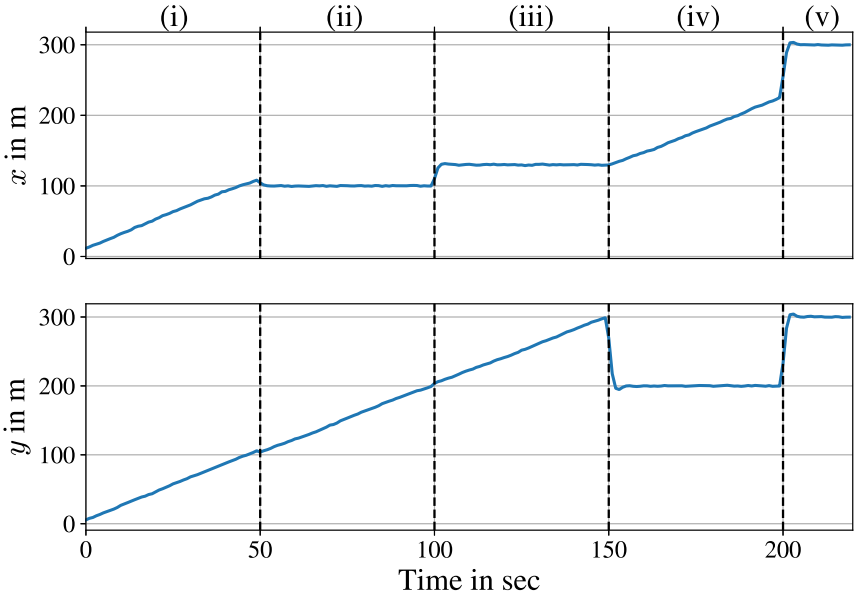


Figure 3.9.: *Real* trajectory with a change of intention/reference in every segment (i)-(v).

the probabilities of switching from model i to other models $j \neq i$ to $\pi_{i,j} = \frac{1}{n_\mathrm{I} - 1}(1 - \pi_{i,i})$, i.e., by equally scaling the probability of switching to any other model. The regularization (3.7, 3.8) in the RegSPME algorithm is applied with $\boldsymbol{D} = \mathrm{diag}(10, 1, 10, 1)$ and an n_Δ varied between 1 and 2. The reverse algorithm runs the IMM algorithm over a horizon of $n_\mathrm{rev} = 2$. Models are similar and merged according to Section 3.3.2, Step 2) if the difference in the predictions over $n_\mathrm{pred} = 10$ time steps is smaller than a threshold $r_\mathrm{pred} = 5$.

In the following, the performance of the three algorithms in detecting and identifying the correct intention is compared.

Motion Tracking Results

In Figure 3.10, the probabilities estimated by the three proposed methods are shown: A) SPME, B) RegSPME, and C) RevSPME. The different colors represent different candidate models (a - d). Different colors and dashed lines are used in time steps where two models describe the same behavior (see Model Fusion, Section 3.3.2), and their probabilities are thus summed. The case where three models describe similar behavior does not occur in this example. The top bar provides an overview of the currently correct models and their corresponding colors for each segment, where the top line represents the primary model that is valid for the entire segment. The bottom line represents the result of the model fusion evaluation. It indicates the color that represents two or more correct models when they are too similar to distinguish. This becomes valid only when a certain configuration is reached, typically after about 10 seconds. The vertical black dashed lines in each plot indicate the switching times between different intentions of the real model. The plot shows two cases of the SPME where only the length of the considered horizon n_w is varied (A.1 and A.2). Plots B.1 and B.2 show two different cases of the RegSPME algorithm where the horizon is varied for the rate of change of the estimated parameters n_Δ, and plot C) the probabilities of the models for the RevSPME algorithm.

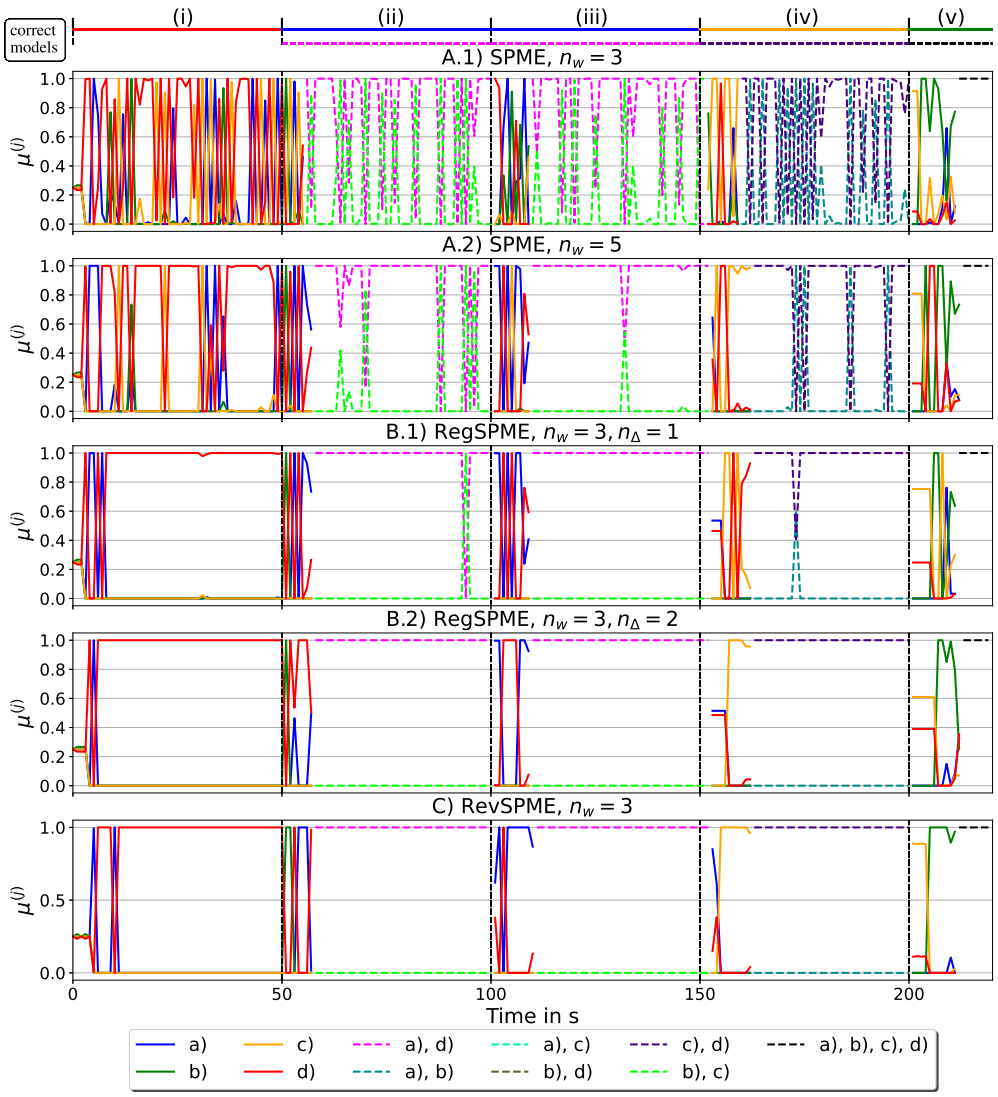


Figure 3.10.: Estimated model probabilities $\mu^{(j)}$ ($j = \mathrm{a}\ldots\mathrm{d}$) for the three proposed algorithms for a trajectory with changing intentions every 50 time steps.

Initially, when the object moves diagonally (phase (i), $0 \leq t < 50$), the SPME approach (A.1) struggles to identify the correct intention, especially for a short horizon. It alternates between all four models, even though they describe very different behaviors. Even tracking a fixed point, model b), seems plausible for some time since the target point constantly shifts as the parameters are re-estimated at each sampling time. Yet, such a model performs poorly in predicting future motion, underscoring the importance of distinguishing between them. This effect is due to the *overfitting*. Plot A.2) shows that extending the observation horizon clearly helps, but even $n_w = 5$ is insufficient to reliably identify the correct intention. In turn, longer observation horizons lead to slower reaction times and a significant increase in computation time. The RegSPME algorithm (B.1) performs much better even with a short horizon for the rate of change factor $\Delta\boldsymbol{\theta}_t^{(j)}$ ($n_\Delta = 1$) and correctly identifies model d) after a few seconds. Extending the horizon $n_\Delta = 2$ (B.2) eliminates any switches back to wrong models. However, it is not recommended to increase the horizon n_Δ any further, as this again leads to detection and identification delays. The RevSPME algorithm (C)

shows results similar to those of the RegSPME algorithm, although it takes slightly longer to identify the correct model. In summary, both the RegSPME and the RevSPME algorithm solve the *overfitting* problem by correctly identifying the model structure d) with accurately estimated parameter values.

During phases (ii), $50 \leq t < 100$, and (iii), $100 \leq t < 150$, the algorithms take a few seconds to adjust to the change in parameters when following the form of model a). Initially, all algorithms correctly identify model a) as correct. Yet, it is eventually recognized that models a) and d) describe the same behavior when the correct x-position is reached if the parameter estimation in model d) considers a zero target velocity in the x-direction (see Figure 3.11). During the transition from one x-position to another ((ii) to (iii) at $t = 100$), all algorithms detect the switch. These "confusion zones" reveal that models a) and d) do not, in fact, describe the same behavior during the switch. This fact is used to detect a "fault" or, here, a significant change in intention, even though the general shape of the correct model remains the same.

During phase (iv), $150 \leq t < 200$, the object maintains a fixed y-position while tracking a velocity in the x-direction. Initially, only model c) is correct, although models a) and c) describe the same behavior when the target y-position is reached. The proposed algorithms exhibit a behavior as in (ii) and (iii), with B.2) and C) once again yielding the best performance. The final phase (v), $200 \leq t < 220$, in which the object reaches a specific goal point (target x-position and target y-position), is the most challenging. All algorithms take the longest to correctly identify model b). For some of the algorithms, by the time of a clear identification, the object has potentially already reached its goal point and stopped moving. Thus, most estimated parameters in the tracking models are 0, and all models describe the same behavior.

When considering possible markers for fault detection, examining the estimated parameters is important, see Figure 3.11. The left-hand side displays the evolution of the estimated parameters in comparison to the real trajectory, while the right-hand side displays the parameters in comparison to the reference velocities. Clearly, the estimates for x and y for the relevant models closely follow the trajectory, with only a short initial oscillation due to incorrect initial parameter guesses. The velocity plots, in turn, exhibit a more dynamic behavior throughout the trajectory. After each change of intention, at least one velocity exhibits a peak and oscillations before reaching the correct values. These peaks may serve as markers for detecting faults or changes in intent and result from the parameter adjustment required after each change. This jump to the correct intent model from an incorrect position and velocity estimate leads to overshoots. In general, all algorithms can detect changes of intention in less than 5 time steps.

Performance Comparison

Two additional studies were conducted to confirm the performance results indicated in Figure 3.10. The data were obtained from 10 algorithm runs of the same setup as before. Figure 3.12 shows the computation time of a computer with 8 AMD Ryzen 5 PRO 3500U cores for different horizon lengths in logarithmic scale, indicating that all three algorithms have similar computation times. The black dashed line marks the sampling time of 10 ms, typically required for robotics and autonomous driving. Evidently, horizon lengths beyond $n_w = 3$ are unlikely to be feasible, especially if a trajectory planning algorithm is added in the same time step. Thus, most plots in Figure 3.10 use $n_w = 3$.

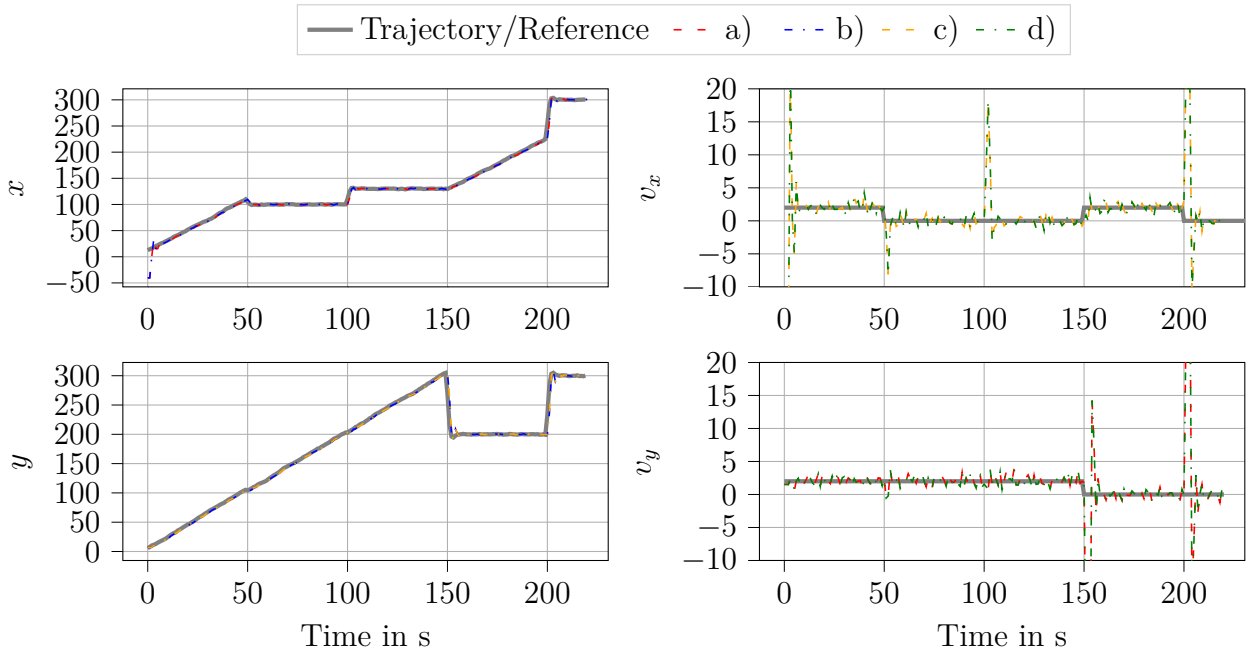


Figure 3.11.: Evolution of the estimated parameters compared to the real trajectories and references shown for the SPME algorithm (Method A).

The second study demonstrates the overall detection capabilities of the algorithms and is shown in Figure 3.13. When considering different horizon lengths, it is evident that the RevSPME algorithm outperforms the other algorithms, particularly for short horizons. Even with an increase in the horizon length, the SPME cannot achieve the same detection accuracy, as the increase leads to longer delays in detection.

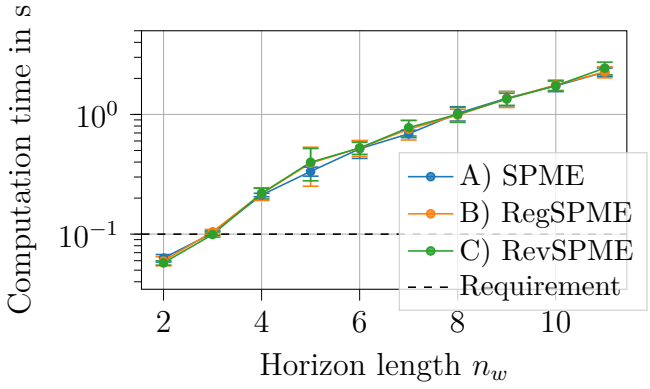


Figure 3.12.: Computation time per time step over different horizon lengths n_w, which stands for the number of considered data points.

3.3.5. Discussion

This study aims to fill a research gap by detecting faults or model switches and simultaneously identifying unknown models while adhering to a predefined structure. This approach

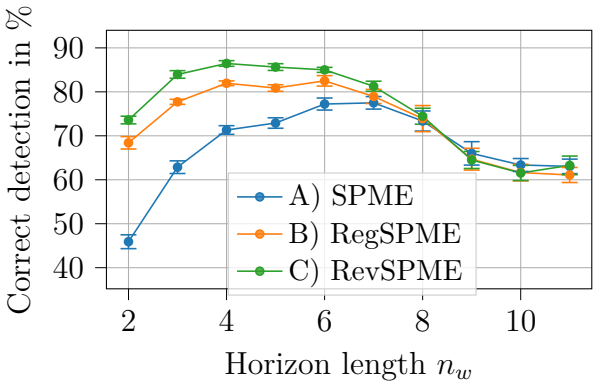


Figure 3.13.: Rate of detecting the correct model over the horizon length n_w, which is the number of considered data points in the algorithms.

aids interpretability which is important for posterior tasks using the results of this method such as path prediction. Most existing machine learning approaches lack this interpretability, while parameter estimation methods in the field of fault detection require prior knowledge of a nominal model. The proposed algorithms effectively bridge this gap by using a set of models with a pre-defined structure and a self-adapting approach to a set of unknown parameters. While it is possible to identify or "learn" only one model, as is common in system identification approaches, there are several reasons why we did not use this approach here. First, if only one model is to be identified, all parameters must be kept as variables, and it is computationally expensive to identify them all. In addition, if MLE is used, a high-dimensional minimum search identifies multiple solutions (minima) where different parameter combinations describe the limited data equally well, and overfitting can occur. Further, using only one model leaves the problem of fault detection completely open and the identified model also lacks interpretability if it does not follow a specific shape.

Conversely, using a large number of fixed (non-parametric) candidate models is not beneficial either, as discussed in Section 3.2 and later in Section 4.3. On the one hand, if models are fixed, a large number of models must be considered so that the non-updating parameters approximate all reasonable values sufficiently well. On the other hand, this approach introduces the risk that models are not sufficiently "differentiated", leading to persistent indecision in the estimated probabilities in the IMM in the presence of noisy data. Rather, it is convenient to design only one candidate model for each macro behavior or fault type and use free parameters to tune the model to the exact fault magnitude.

In the results, in Section 3.3.4, we show that this MM approach can indeed be used to prevent these problems. Using a parameter estimation approach first and then feeding the identified models into an IMM algorithm (SPME) shows good results for the motion tracking example, but also that this combination overfits in certain model structures, such as those in the shown motion tracking scenario. To address this issue, the proposed regularization techniques of the RegSPME algorithm and the RevSPME approach both mitigate overfitting. While the regularization method offers more tuning parameters, making it more flexible, the RevSPME approach is simpler and easier to tune. The proposed algorithms accurately identify unknown models, making them suitable for other FDD applications.

Additionally, it was demonstrated that short horizon lengths remain the preferred solution. This is not only due to long computation times on standard computers but also as the

algorithm's accuracy in detecting the correct model decreases as the observation horizon becomes longer, resulting in reduced sensitivity to new data points and slower detection speeds. It is shown that for short horizons, the regularized RegSPME and inverse RevSPME methods have an even greater advantage over the standard SPME approach.

While the study demonstrates the usefulness of these algorithms in model detection applications, the focus remains on the identification part, especially for motion tracking. For fault detection, we recommend implementing a well-defined marker and a thresholding mechanism to flag model changes, i.e., a possible fault. This approach also allows for a more detailed investigation of reaction times to model changes. The algorithms and results showed that the approach provides enough possibilities to define such a threshold, where, e.g., the 3-Sigma rule [156] and the Hampel identifier [157] can be used to combine the identified peaks in the estimated velocities with the observed alternating high probabilities between very different models into a marker for new onset of a fault or model change. Additionally, we showed the known limitation of the IMM algorithm, which always assigns probabilities totaling 100 %. In complex environments where the correct model shapes are not fully known, this design flaw can lead to incorrect models being assigned erroneously high probabilities, undermining the effectiveness of the algorithm.

3.4. Conclusion

In this chapter, we addressed the problem of estimating probabilities of uncertainty modes and of deriving prediction models of the nominal trajectories of uncertainty modes using recent measurements. Firstly, a novel BFT-based information fusion algorithm was presented, which solves the problem of combining information resulting from multiple sources in a coherent and steady estimation of the probability of uncertainty modes, considering the candidate nominal trajectories of DO driven by an unknown intention as uncertainty modes. An explicit quantification of the uncertainty of the estimates is provided, representing the reliability of the information. Our novel algorithm is superior to existing approaches in that it is well suited for combining information from potentially inconsistent sources. When the different sources provide individual estimates that do not match, the degree of conflict is evaluated and part of the belief mass is transferred to the uncertainty parameter. Other estimation mechanisms that do not address possible contradictions among information provided by different sources, like the IMM, tend to produce inconsistent results over consecutive sampling times, making the overall combined estimate unreliable and therefore unfit for practical application. Furthermore, in our approach sudden variations in the estimate probabilities are attenuated, which is especially important when humans are considered, as sudden contradictory movements for a short time are typical due to distraction. The estimate provided by our algorithm incorporates a quantitative, though subjective, measure of reliability of the information contained, which motion planners can take advantage from.

Secondly, we presented a novel variable-structure multi-model approach (SPME), solves the problem of deriving prediction models of the uncertainty modes utilizing multiple models with a predetermined structure but unknown and free parameters that are estimated. This approach was applied to a motion tracking problem, where several candidate models were used to describe possible macro-behaviors of DOs, while the exact parameters were estimated using a MLE approach. The probability of each model is estimated using the IMM algorithm. Overfitting problems caused by the combination of MLE and IMM were discussed, and two solutions were introduced: regularization with respect to the change in the MLE parameters (RegSPME) and a reverse-order algorithm (RevSPME). It was also shown that short data sequences are computationally necessary. This is highly beneficial for the computational effort and it is a strong advantage for prompt fault detection. This approach is suitable for unknown faults where the system behavior can be described in a macroscopic form.

Safe and Non-Conservative SMPC-Based Planning with Multi-modal Uncertainty

4.

This chapter addresses the problem of motion planning in presence of multi-modal uncertainty. From a control perspective, uncertainty poses the challenge of balancing between safety and efficiency, where efficiency is understood as the necessity to avoid overly "cautious" approaches. We propose several approaches that account for the intrinsic multi-modal nature of the uncertainty caused by the unknown intention of a traffic participant, named here Dynamic Obstacle (DO), and to enhance safety by addressing the unmodeled uncertainty that might result in a violation of the worst-case assumptions. The content of this chapter was published in [20]–[23].

4.1. Introduction

Motion planning is a research field which addresses the problem of generating a feasible trajectory for the controlled agent. A very popular and rapidly expanding application field of motion planning is autonomous driving, in which a feasible trajectory for the autonomous vehicle, here Ego Vehicle (EV), is computed considering the vehicle dynamics, the traffic rules, as well as avoiding collisions with DOs. Despite the considerable advances in highway scenarios, urban environments remain a challenging planning framework, especially due to the interaction with complex traffic populated by pedestrians and cyclists. These show extremely different behaviours, and generate large uncertainty. In particular, the uncertainty about future motion of a DO is twofold: first, it depends on the high-level intention pursued, for example following a particular gait or making a turn; secondly, the precise trajectory of the DO when following a specific intention is stochastic, since humans do not behave following a purely deterministic strategy and the dynamics of DOs is itself only approximately known. As a result, the uncertainty about the future motion of DOs is multi-modal. Complexity also arises from the need to adapt the driving environment to different traffic conditions, rules, and road geometry. Furthermore, in contrast to the highway framework, maneuver switches appear more frequently in urban environments, to adapt the motion to the changing conditions. With regard to motion planning, a reasonable trade-off between safety and efficiency must be found, and the interplay with DOs must be handled in a safe and yet non-conservative way, in a human-like manner.

Learning-based methods [158] use recorded data to produce a trajectory for the EV. They allow for a flexible motion, able to handle several possible configurations with no need for rigidly defined switching rules, which is extremely convenient to face complex situations typical for the European urban road environment [159]. Yet, safety is hardly provable, thus

additional structures such as time-consuming safety checks are required, with the need to rely on alternative solutions such as backup trajectories, increasing the complexity [160].

MPC is a popular model-based algorithm for trajectory planning and control [161]. In MPC the trajectory of the automated vehicle is computed by iteratively solving an optimal control problem over fixed-length receding horizons. Desired control objectives are given as a cost function and safety conditions and traffic rules, together with physical limitations of the vehicle, result in a set of constraints. At each sampling time the optimal sequence of actions of the EV for the full prediction horizon is obtained. Only the first input of the optimal sequence is applied and the process is repeated when new measurements of the environment are collected, adapting the future trajectory to the changing traffic configuration.

To avoid collisions between the EV and static obstacles, fixed forbidden areas for the EV are included as constraints in the MPC optimal control problem. The same approach cannot directly be used to avoid collisions with another traffic participant, since its future motion is not known and only probabilistic models of DOs are available. Therefore, in order to consider safety requirements to prevent collisions with DOs, the MPC scheme must be able to consider uncertainty.

The uncertain future motion of DOs can be described in terms of worst-case movements underlying a given intention and considering the traffic rules. Then, the EV trajectory is *safe* if it is collision free for all realizations of the uncertain DO future motion within the worst-case assumptions, realizing a robust approach [15], [28], [162]. However, this approach is likely to result in an excessively cautious controller producing unnaturally conservative behaviors, up to the point that under particular conditions the automated vehicle cannot take action at all, as briefly discussed in [30].

To reduce conservatism, SMPC [16], [29] has been proposed. Differently from RMPC, in SMPC safety constraints are designed in a probabilistic fashion. A probability distribution for movements within the worst-case outcomes is assumed and collision avoidance constraints are reformulated in chance (probabilistic) constraints, that must only be satisfied up to a user-specified risk level. As a result, SMPC yields a non-conservative EV trajectory [17], [79] and therefore SMPC is particularly well suited to deal with a highly uncertain urban framework. However, SMPC-based planning in presence of large and multi-modal uncertainty introduces a spectrum of new challenges.

4.1.1. Challenges

Here we discuss the challenges of SMPC-based planning under large and multi-modal uncertainty, considering both those that intrinsically pertain to the SMPC approach, and those who result from the multi-modal nature of the uncertainty, as well as possibly incomplete description of the uncertainty itself.

SMPC produces open-loop predictions with non-zero probability of collision that may consequently result in collisions in closed loop. Therefore, SMPC does not provide safety guarantees typical of robust approaches, although approaches have been presented to address the remaining probability of constraint violation [30], [31]. SMPC is widely used for autonomous driving in highway-like scenarios [17], [163], [164], but little attention is dedicated to the typical urban framework, where stops are required to avoid collisions with crossing pedestrians or upcoming vehicles at intersections.

Moreover, especially in urban settings, it is beneficial to take a long prediction horizon

into account, allowing to reveal superior maneuver choices, which would prove to be efficient in the long run. Yet, a detailed model is not necessarily useful to optimally plan long-term maneuvers, and a simplified one, considering the dominant dynamics only, may suffice. Additionally, due to the accumulated action of the uncertainty, it is not reasonable to rely on detailed prediction models for long-term predictions. A long prediction horizon also causes a computational burden difficult to cope with in a quick real-time computation. Moreover, in autonomous urban driving there is a clear difference between low-level control actions (instant acceleration, steering angle) to drive the vehicle in the short term and high-level actions (cruise speed planning, lane changing) to be determined considering a longer horizon.

Other important challenges come from the multi-modal nature of the uncertainty. As discussed previously, for every DO multiple candidate trajectories are available, each resulting from a different high-level intention pursued by the DO. Then, the challenge lies in generating safety constraints non-conservatively by taking multiple possible trajectories into account. In order to reduce conservatism while still accounting for the multi-modal nature of the uncertainty, safety constraints should be prioritized considering the probability of candidate DO trajectories. In fact, both enforcing collision-avoidance only for the most likely trajectory [17] and for all candidate trajectories with equal required risk parameter [93] is not a good trade-off. An approach prioritizing the focus depending on the expected behavior of DOs, however, relies on accurate estimation of the probabilities of the considered candidate trajectories, otherwise the EV might misjudge the traffic scenario. The estimated probabilities of the candidate trajectories are obtained comparing recorded data and expected behaviors from nominal models with the measurements collected online by onboard EV sensors and possibly intelligent infrastructure. When the motion of the DO is ambiguous, the estimated probabilities change quickly over time and are not reliable. In presence of especially ambiguous behaviors of DO, or if the observed behavior of the DO does not match any of the considered candidate trajectories, the estimated probabilities are rather unreliable and change very often. As a result, the motion planned by the EV would also change very often and be inconsistent within multiple time steps, either with risky consequences or at least leading to suboptimal performances. Nevertheless, including information about the reliability of the estimated probabilities of candidate DO trajectories in the EV planning is challenging with existing approaches.

It is also important to give formal safety guarantees about collision avoidance despite the SMPC-based planning. Few approaches have dedicated attention on enforcing a-priori safety checks on the planned SMPC trajectory before it is applied, to evaluate whether it would lead to actual collision in closed-loop if the realization of the uncertainty reaches the worst-case bounds. However, such schemes with a-priori safety check must be implemented with a view to maintaining the computational complexity as close as possible to that of an SMPC planner, requiring the solution of a single optimal control problem, otherwise the computation time significantly increases, preventing practical application.

Finally, safe planners should also consider unmodeled uncertainty, i.e., possible realizations that are not included in the description of the uncertainty provided, e.g., that are not attributable to any of the modes considered, even taking into account the uncertainty about the modes themselves. In fact, candidate modes are derived heuristically from previous knowledge about the context. Similarly, in practice worst-case bounds are derived from the physical limitations of the dynamics of DOs, which however are not exactly known, and from other heuristic considerations. Therefore, it is necessary to consider also the role of

unmodeled uncertainty, which might result in violation of the assumed worst-case bounds. Although this prevents the formulation of safety guarantees, a criterion must be formulated that specifies how such situations are to be systematically addressed.

4.1.2. Contribution

In this chapter, the challenges from Section 4.1.1 are addressed proposing several novel motion planning algorithms. In Section 4.2, we apply SMPC to autonomous urban driving to balance between safety and efficiency in presence of uncertainty. Starting from a stochastic description of the future motion of the other traffic participants (DOs), safety conditions for the ego vehicle are derived, resulting in constraints that ensure that the probability of collision remains below the user-defined acceptable risk level. Moreover, our approach is superior to existing ones because it explicitly allows to separate the optimization of maneuver and trajectory planning, given their different characteristic time scales and required update frequency, leveraging a hierarchical MPC structure. Although hierarchical structures have already been employed in autonomous driving, the focus of the high-level layer was mainly directed to the maneuver planning in the short term through a set of pre-defined switching rules. Conversely, our novel approach exploits the hierarchical structure to allow for high-level reasoning for a long-horizon optimization, by means of a higher layer, in which a simple model and a large sampling time are considered. By taking advantage of the hierarchical structure, our planning algorithm allows to benefit from a long-term view, without increasing the computational effort of the trajectory planner.

After discussing how the high-level decisions of the EV can be optimized, we focus on accounting for the high-level decisions of DOs. In Section 4.3, we propose a novel framework to greatly reduce conservatism by explicitly prioritizing the most likely possible future trajectories of each DO. We estimate the probability of multiple candidate future DO, each representing a high-level intention, using the IMM algorithm [138] combined with several LQR-based approximations of DO trajectories. Then, we propose a new approach to plan the trajectory of the EV in an SMPC fashion. Probabilistic safety constraints are generated for all the predicted trajectories of the DOs in the vicinity of the EV and are required to hold with a probability which depends on the estimated confidence in each candidate trajectory. Therefore, our approach advances the state of the art by allowing to consider multiple candidate DO trajectories giving higher priority to those most likely to be executed, without being excessively restricted by those that are considered unlikely.

However, prioritizing the focus of the EV for the most likely future DOs intentions requires accurate probabilities of the candidate intentions, which might be difficult to estimate in practice. Therefore, in Section 4.4 we extend the approach from Section 4.3 by explicitly considering that the estimated intention (mode) probabilities are themselves uncertain. We take advantage of the BFT framework presented in Section 3.2, where the measure of the reliability of the estimate is provided quantitatively by the uncertainty parameter. We propose two new approaches to explore how the design of SMPC safety constraints can take advantage from the assessment on the reliability of the information, rather than from standard probabilities of intentions. Firstly, we introduce a novel transformation of the BFT estimates into probabilities with a view at not underestimating the probability of seemingly unlikely future trajectories, if the estimation is very uncertain. Such probabilities can be used as risk parameter in the SMPC formulation. Further, we propose a second approach,

in which the uncertainty measure provided by the BFT estimation is used to tighten constraints, restricting the motion of the EV if the intention of DOs is not clear. However, the combination of BFT and SMPC does not result in overly conservative EV trajectories, if the intention of DOs is clear. We show that our approach significantly advances the state of the art by discussing how overconfident and dangerous EV behaviors are produced neglecting the reliability of the estimation.

Finally, we address the challenge of providing a safe motion despite the non-zero probability of constraint violation allowed by SMPC. In Section 4.5, we propose a novel control scheme for efficient and safe trajectory planning for automated driving that leverages the optimistic SMPC planning still yielding a collision-free motion. At first, the SMPC optimal control problem is solved, yielding an optimistic non-conservative EV trajectory. Then, the first predicted state of the SMPC trajectory is used as initial condition for a further planning problem, *robust* with respect to the uncertain DO motion. The solution to this problem is not computed, the problem only serves as certification of safety. If such further optimal control problem admits a solution, the first SMPC input is guaranteed not to lead to unavoidable collisions. Therefore, in this case, the SMPC input is considered safe and is applied. By contrast, if a solution to the further robust control problem does not exist, i.e., the robust control problem is infeasible, the SMPC solution is disregarded, since it could lead to collisions that cannot be prevented in the next time steps. If the SMPC solution fails the safety certification, we plan a robust trajectory from the current initial condition, which exists because of the safety certification of the previous iteration. Thus, the EV does not cause collisions with DOs.

Furthermore, we address the unmodeled uncertainty, i.e., situations in which the actual motion of DOs violates the worst-case assumptions. We refer to these DO motions as *unanticipated* behaviors, including also illegal behaviors where a DO violates the traffic rules. In presence of unanticipated behaviors of DOs the safety guarantee from combining SMPC with a robust backup does not hold, and there might be instants in which also the robust planner is not able to yield a trajectory guaranteed to be collision-free. In such situations, we plan the EV trajectory by minimizing the probability of collision over the prediction horizon. We implement the robust planner and the probability-minimization planner using the CVPM approach [105]: when feasible, a control action yielding zero probability of collision is applied (*robust case*); otherwise, the probability of collision is minimized (*probabilistic case*). Our novel scheme, named SMPC+CVPM, is computationally efficient since it is not necessary to compute the robust backup after the solution of the SMPC problem and, rather, the scheme can benefit from parallelization.

Whereas all proposed methods are evaluated in several automated driving situations settings, a general application to other scenarios characterized by multi-modal uncertainty is possible, for example in dynamic risk allocation by accounting for multiple financial scenarios [3], or dynamic optimization of the balance in energy systems, considering multiple possible consumption patterns [2].

4.2. Multistage SMPC for Long-Term Planning

In the following, we present an SMPC-based framework for trajectory planning in urban autonomous driving. Taking advantage of a hierarchical approach, we benefit from long-term reasoning without increasing the computational effort. This section is based on the work presented in [20].

4.2.1. Problem Formulation

We consider the linearized and discretized kinematic bicycle model (2.15) as prediction model for the EV, which is outlined in Section 2.3.3. The EV state is defined as $\boldsymbol{\xi} = [s, d, \phi, v]^\top$, where (s, d) are the longitudinal and lateral position of the center of mass expressed in the road-aligned reference frame of a (possibly curved) reference path, ϕ is the orientation of the vehicle with respect to the road itself, and v is the linear velocity of the EV. The input vector is $\boldsymbol{u} = [a, \delta]^\top$, with acceleration a and front steering angle δ. Furthermore, the EV is subject to constraints (2.16) on the velocity, to prevent it from going backwards or violating the maximum allowed speed, on the lateral position, to prevent the vehicle from leaving the road, and on the maximum values for the acceleration and steering angle, as well as on the maximum rate of change thereof.

As prediction model for DOs, we distinguish between Target Vehicle (TV)s and pedestrians. For TVs we use prediction models outlined in Section 2.3.4. The dynamics of TVs is modeled as a discrete-time point mass with state $\boldsymbol{\xi}^{\mathrm{TV}} = [x^{\mathrm{TV}}, v_x^{\mathrm{TV}}, y^{\mathrm{TV}}, v_y^{\mathrm{TV}}]^\top$, composed of longitudinal position x^{TV} and velocity v_x^{TV}, and lateral position y^{TV} and velocity v_y^{TV}, expressed in the inertial world frame. The system input $\boldsymbol{u}^{\mathrm{TV}}$, consisting of the longitudinal and lateral accelerations, is modeled as the sum of a feedback term and a perturbation term $\boldsymbol{w}^{\mathrm{TV}}$. The feedback term is designed to correct possible deviations from the desired trajectory $\boldsymbol{\xi}_{\mathrm{ref}}^{\mathrm{TV}}$. Then, the perturbation term is modeled as Gaussian noise and represents the uncertainty of the future TV motion.

In order to compute the prediction of the future states of the pedestrians, we refer to the model in [165], which reflects the model used for the TV but without the feedback term of the input. This increases the uncertainty about the future motion of pedestrians, with respect to the TV model. The EV has access to a noisy measurement or estimate of the state of all DOs.

It is convenient to include long-term reasoning in the planning of the EV, while maintaining the computational complexity low and without relying excessively on inaccurate predictions of the motion of DOs for further steps in the future, obtaining the following objective.

Objective 4.1. *The goal is to compute the trajectory of the EV considering a long prediction horizon to benefit from long-term reasoning, while maintaining the computational complexity manageable and accounting for increased uncertainty for further prediction steps.*

4.2.2. Overview of the Proposed Multi-Stage Predictive Planning Approach

Here an overview of the control hierarchy is given, highlighting the role of each layer, and the connections between them. The proposed control algorithm is composed of two layers. Both layers work in a receding horizon setting, selecting the optimal sequence of control actions

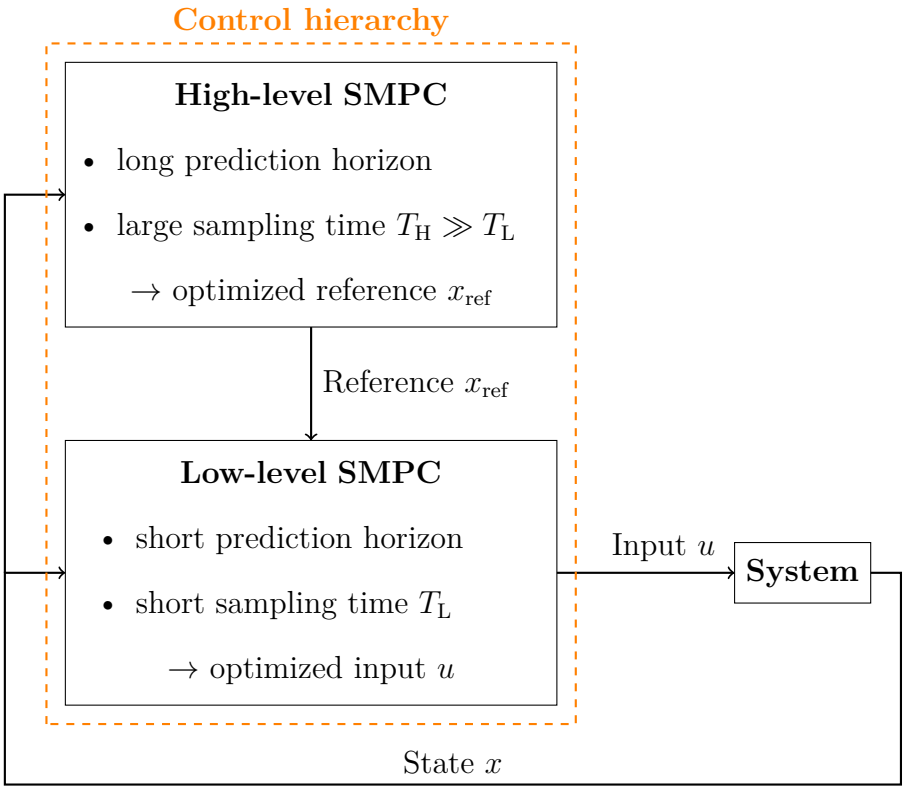


Figure 4.1.: Scheme of the control hierarchy. © 2021 IEEE.

by minimizing a properly designed cost function, while meeting the requirements formulated as a set of constraints. A scheme of the control hierarchy is represented in Fig. 4.1.

The lower layer determines the immediate control actions to be applied on the system. A detailed model is used to compute the new optimal control input every time a new measurement of the state is available, and the constraints guarantee safety up to risk parameter.

The high-level layer is an additional structure, used to benefit from a long-term view in the optimization. The goal of this layer is not to determine the control input to be directly applied on the system, but rather to benefit from a long prediction horizon and provide this information to the low-level controller. For this reason, the optimization variable of this level is the reference trajectory used by the low-level controller, which is determined considering a longer prediction horizon. Moreover, since the reference trajectory must not be updated too often, a longer sampling time is used. The model used by the high-level controller is chosen to be an abstraction of the low-level one, including only the information that can be predicted in a sufficiently reliable way even in the long term. Furthermore, not necessarily all the state features (e.g. the lateral displacement) are relevant to decide the maneuver. Finally, the cost function and the constraints are based on the low-level ones, but adapted to be consistent with the high-level model.

4.2.3. Combined Trajectory and Maneuver Planning on Different Levels

In this section we outline our method discussing the low-level trajectory planner and of the high-level maneuver planner.

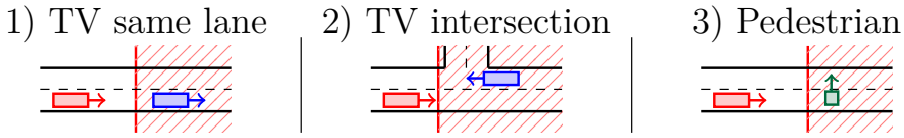


Figure 4.2.: Constraints used in the low-level trajectory planner. © 2021 IEEE.

Trajectory Planning

In the following, the details of the low-level SMPC algorithm used for the trajectory planning are discussed. First, we discuss about the determination of the deterministic safety distance based on the stochastic description of the uncertainty about the prediction. Then, the generation of safety constraints depending on the type of DO and on the relative configuration with the EV is discussed and we formulate the trajectory planning optimal control problem.

1) Safety Distance In this section, a safety distance to be observed from each traffic participant is derived, so that the collision is avoided at least with the required probability β. Considering the DO, along the lines of [30], the safety distance can be obtained as

$$a_k^{\mathrm{DO}} = \frac{l^{\mathrm{DO}}}{2} + \Delta s_{\mathrm{stop}}^{\mathrm{DO}} + e_{s,k}^{\mathrm{DO}} + \epsilon_{\mathrm{safe}}^{\mathrm{DO}}, \tag{4.1}$$

with l^{DO} the DO length, and $\Delta s_{\mathrm{stop}}^{\mathrm{DO}}$ the distance needed to stop if the EV is driving faster. $e_{s,k}^{\mathrm{DO}}$ is a deterministic margin to take the uncertainty on the DO position into account, and $\epsilon_{\mathrm{safe}}^{\mathrm{DO}}$ is an extra safety parameter. a_k^{DO} is computed for each prediction step k. In the following, the superscript DO is omitted for readability.

The term $e_{s,k}$ deterministically increases the safety distance such that the considered region contains the DO position with probability β. Given the linear model, and the Gaussian distribution of the noise $\boldsymbol{w}_k$, the prediction error covariance matrix $\boldsymbol{\Sigma}_k^e$ is obtained propagating the uncertainty as in Section 2.3.7 starting from the initialization $\boldsymbol{\Sigma}_0^e = \mathbf{0}$ (the current DO position is assumed to be perfectly known). From $\boldsymbol{\Sigma}_k^e$, we obtain the standard deviation of the longitudinal position prediction error $\sigma_{s,k}$, and, using the standard approximation explained in detail in [30], the term $e_{s,k} = \sigma_{s,k}\sqrt{\gamma}$ is derived, where $\gamma = -2\ln(1-\beta)$.

Even though a_k^{DO} has been derived in a probabilistic framework, they determine a deterministic *safety area* about the predicted position of each other traffic participant. As a result, the chance constraints can be reformulated as deterministic constraints preventing the EV from entering the safety areas, as explained in Section 4.2.3.

2) Safety Constraints In this section, we derive the deterministic constraints to ensure the EV does not enter the safety areas of DOs. Each possible scenario is considered separately, and the constraints are represented in Fig. 4.2. The constraints are derived considering a simple urban environment composed of roads with a single lane (per direction) and an intersection. Moreover, it is assumed the EV can measure the state of DO in road-aligned coordinates. One new constraint is generated for each traffic participant, with different rules depending on the type of DO, i.e., TV or pedestrian, and on the relative configuration with respect to the EV.

First, we consider the case of a TV in the same lane in front. In this case, the EV must stay behind. Thus, the following constraint based on the length l^{EV} of the EV is generated,

ensuring the safety distance a_k^{DO} is always kept

$$s_k + \frac{l^{\mathrm{EV}}}{2} \leq s_k^{\mathrm{DO}} - a_k^{\mathrm{DO}}. \tag{4.2}$$

Second, if the EV is approaching the intersection, planning a left turn, and a TV is predicted to cross the intersection before the EV can leave the intersection free, then the EV must wait before entering the intersection, giving the right of way. As a result, a constraint is generated before the intersection area

$$s_k \leq s_{\mathrm{int}}. \tag{4.3}$$

When a pedestrian is predicted to cross the road, the EV must safely stop before. Thus, we have

$$s_k + \frac{l^{\mathrm{EV}}}{2} \leq s_k^{\mathrm{DO}} - a_k^{\mathrm{DO}}. \tag{4.4}$$

3) Optimal Control Problem Given the EV model (2.15), the state and input constraints from (2.16d), and the positional constraints derived in Section 4.2.3, the full optimal control problem for the low-level trajectory planner is formulated, considering for simplicity the sampling time $t = 0$:

$$\min_{\boldsymbol{u}_k} \sum_{k=0}^{N-1} \|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2 + \|\Delta\boldsymbol{u}_k\|_{\boldsymbol{S}}^2 + \|\Delta\boldsymbol{\xi}_N\|_{\boldsymbol{P}}^2 \tag{4.5a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}^{\mathrm{d}}(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k, \boldsymbol{u}_k), \qquad \forall k = 0, \ldots, N-1 \tag{4.5b}$$

$$\boldsymbol{\xi}_{k+1}^{\mathrm{DO}} = \boldsymbol{A}^{\mathrm{DO}}\boldsymbol{\xi}_k^{\mathrm{DO}} + \boldsymbol{B}^{\mathrm{DO}}\boldsymbol{u}_k^{\mathrm{DO}}, \qquad \forall k = 0, \ldots, N-1 \tag{4.5c}$$

$$\boldsymbol{\xi}_k \in \mathcal{X}, \qquad \forall k = 1, \ldots, N \tag{4.5d}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \qquad \forall k = 0, \ldots, N-1 \tag{4.5e}$$

$$0 \geq \boldsymbol{q}_s(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k)s_k + \boldsymbol{q}_d(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k)d_k + \boldsymbol{q}_t(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k), \ \forall k = 1, \ldots, N, \tag{4.5f}$$

where N is the prediction horizon, $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{S}$, and $\boldsymbol{P}$ are positive semi-definite weighting matrices, and, for the input difference, $\boldsymbol{u}_{-1}$ is the last applied input. Prediction model (4.5c) is run for each DO independently.

Note that the positional constraint coefficients $\boldsymbol{q}_s$, $\boldsymbol{q}_d$, and $\boldsymbol{q}_t$ in (4.5f), which are a reformulation of the constraints (4.2), (4.3), and (4.4), can be computed from the current EV state and the predicted TV and pedestrian states, before the optimization starts. As a result, the optimal control problem (4.5) is a Quadratic Program (QP) with linear constraints.

Maneuver Planning

In this section, the high-level maneuver planner is introduced, specifying the motivation, the details of the model being used, the constraints generation, and the cost function.

1) High-Level Models The scope of the maneuver planner is to determine the optimal sequence of high-level actions, considering the future behavior of the DOs over a long prediction horizon. However, planning maneuvers in the long run requires fewer details with respect to the immediate trajectory planning for the short term. Moreover, due to the uncertainty of the behavior of the other traffic participants, a detailed prediction of other vehicles

or pedestrians for farther steps would not be reliable. For this reason, here we present simplified models, which are used at the higher level to predict the only information which can be predicted reliably enough even in the long term, and which is needed to plan the maneuver.

Given the simple structure of the road, the main maneuver the EV can plan is to accelerate or decelerate, for example in order to reach the intersection when it is free, avoiding a full stop that might otherwise be required. In order to determine the optimal speed, we only consider the position of the EV along the (fixed) reference path, which evolves according to a point-mass model dynamics with piece-wise constant speed, without considering lateral displacements. As a result, the high-level dynamics is

$$s_{h+1} = s_h + \nu_h T_{\mathrm{H}}, \tag{4.6}$$

where the input ν is the speed, and h is the prediction step of the high-level maneuver planner. T_{H} and N_{H} are the sampling time and the prediction horizon of the high-level model, respectively, and we assume $N_{\mathrm{H}} T_{\mathrm{H}} \geq NT$. The simplified model is subject to input constraints in the form

$$0 \leq \nu_h \leq \nu_{\max}(s_h). \tag{4.7}$$

$\nu_{\max}(s_h)$ is the maximum speed allowed depending on the position s_h, and might be lower than $v_{\max}$ in (2.16b), e.g., to take into account that it is unrealistic to drive at the maximum speed in highly curved stretches. This constraint makes the prediction more reliable, since the simplified model does not take the path's curvature into account.

Concerning the DOs, it is enough to predict their longitudinal position along the path, in order to formulate the constraints of the low-level trajectory planning. This can be obtained by computing the equivalent of the low models from (2.3.4) with a time-step T_{H}, and by projecting the position of the TV and the pedestrian on the reference path, obtaining s_h^{TV} and s_h^{P}. Firstly, the point-mass linear model matrices (2.20) are recomputed using T_{H} in place of T, obtaining $\boldsymbol{A}_{\mathrm{H}}$ and $\boldsymbol{B}_{\mathrm{H}}$. The feedback gain $\boldsymbol{K}$ from (2.23) is also substituted by $\boldsymbol{K}_{\mathrm{H}}$, which is computed so that the closed-loop matrix $\boldsymbol{A}_{\mathrm{H}} + \boldsymbol{B}_{\mathrm{H}}\boldsymbol{K}_{\mathrm{H}}$ is the equivalent of $\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$ with a sampling time T_{H} instead of T. This implies, for example, that the eigenvalues of $\boldsymbol{A}_{\mathrm{H}} + \boldsymbol{B}_{\mathrm{H}}\boldsymbol{K}_{\mathrm{H}}$ correspond to the eigenvalues of $\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$ to the power of $\frac{T_{\mathrm{H}}}{T}$. Then, the statistical properties of the DO system noise must also be scaled. We consider the average of the $\bar{k} = \lfloor \frac{T_{\mathrm{H}}}{T} \rfloor$ disturbances $\boldsymbol{w}_k^{\mathrm{DO}}, \ldots, \boldsymbol{w}_{k+\bar{k}-1}^{\mathrm{DO}}$ affecting the DO within a high-level sampling time T_{H}. The average noise $\boldsymbol{w}_h^{\mathrm{DO}}$ is still a zero-mean Gaussian, and, being $\boldsymbol{w}_k^{\mathrm{DO}}$ independent, its variance is

$$\mathrm{Var}\left(\sum_{j=0}^{\bar{k}-1} \frac{\boldsymbol{w}_{k+j}^{\mathrm{DO}}}{\bar{k}} \right) = \sum_{j=0}^{\bar{k}-1} \mathrm{Var}\left(\frac{\boldsymbol{w}_{k+j}^{\mathrm{DO}}}{\bar{k}} \right) = \frac{\boldsymbol{\Sigma}_{\boldsymbol{w}}^{\mathrm{DO}}}{\bar{k}}. \tag{4.8}$$

2) High-Level Constraints Then, the constraints of the high-level optimal control problem are formulated. Repeating the procedure of Section 4.2.3, the safety distance ρ_h^{DO} is derived, based on the statistic description of the high-level noises $\boldsymbol{w}_h^{\mathrm{DO}}$ and on the risk parameter β_{H}. Similarly to the low-level case in Section 4.2.3, each case is considered separately and a proper constraint is generated. A scheme of the constraints is given in Fig. 4.3.

If a TV is predicted to occupy the same lane in front, the EV must remain behind and a constraint is generated behind the TV

$$s_h + \frac{l^{\mathrm{EV}}}{2} \leq s_h^{\mathrm{DO}} - \rho_h^{\mathrm{DO}}. \tag{4.9}$$

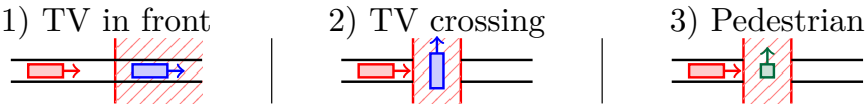


Figure 4.3.: Constraints used in the high-level maneuver planner. Coherently with the abstract model (4.6), no lateral displacement (or lane) is considered. © 2021 IEEE.

Note that (4.9) is not needed for safety, guaranteed by the analogous condition (4.2) in the low level, but rather to make the maneuver planner aware of the future position of the TV, allowing to plan a smooth deceleration when needed.

When a TV is predicted to temporarily cross the path of the EV (e.g., at an intersection), the maneuver planner should take this into consideration and plan the speed accordingly. If possible, the EV could reach the intersection point before, when it is still free, or otherwise it could plan to decelerate, in order to smooth the motion and minimize the stop time. The constraint is designed as

$$\begin{cases} s_h + \dfrac{l^{\mathrm{EV}}}{2} \leq s_h^{\mathrm{DO}} - \rho_h^{\mathrm{DO}} & \text{if } s_h \leq s_h^{\mathrm{DO}} \\ s_h - \dfrac{l^{\mathrm{EV}}}{2} \geq s_h^{\mathrm{DO}} + \rho_h^{\mathrm{DO}} & \text{if } s_h > s_h^{\mathrm{DO}}, \end{cases} \tag{4.10}$$

which can be reformulated in a quadratic expression

$$(s_h - s_h^{\mathrm{DO}} + \Delta_1)(-s_h + s_h^{\mathrm{DO}} + \Delta_2) \leq 0, \tag{4.11}$$

where Δ_1 and Δ_2 collect the terms from (4.10), plus possible additional safety parameters. Note that not only constraint (4.11) is nonlinear, but it is also position dependent, and generates a non-connected admissible set.

Similar to the previous case, if a pedestrian is predicted to cross the street, the EV should plan to get over the crossing point before the pedestrian starts to cross, or to decelerate to minimize the stop time potentially required.

3) Optimal Control Problem From the simplified high-level models and constraints, we can formulate the optimal control problem for the high-level maneuver planner, considering again $t = 0$:

$$\min_{\nu_h} \sum_{h=0}^{N_{\mathrm{H}}-1} (\nu_h - \nu_{h-1})^2 + r_H(\nu_h - v_{\mathrm{ref}})^2 \tag{4.12a}$$

$$\text{s.t. } s_{h+1} = s_h + \nu_h T_{\mathrm{H}}, \qquad \forall h = 0, \dots, N_{\mathrm{H}} - 1 \tag{4.12b}$$

$$\boldsymbol{\xi}_{h+1}^{\mathrm{DO}} = \boldsymbol{A}_{\mathrm{H}} \boldsymbol{\xi}_h^{\mathrm{DO}} + \boldsymbol{B}_{\mathrm{H}} \boldsymbol{u}_h^{\mathrm{DO}}, \quad \forall h = 0, \dots, N_{\mathrm{H}} - 1 \tag{4.12c}$$

$$(s_{h+1}, \nu_h) \in \mathcal{Z}, \qquad \forall h = 0, \dots, N_{\mathrm{H}} - 1, \tag{4.12d}$$

where ν_{-1} is set equal to the current speed v_0, and set $\mathcal{Z}$ in (4.12d) compactly represents constraints (4.7), (4.9), and (4.11). Prediction model (4.5c) is run for each DO independently. The cost function (4.12a) is designed to penalize large accelerations, for sake of comfort and coherently with (4.5a), and to penalize deviations from a desired cruise speed. This second term is needed to avoid that the optimal control problem (4.12) admits the trivial optimal

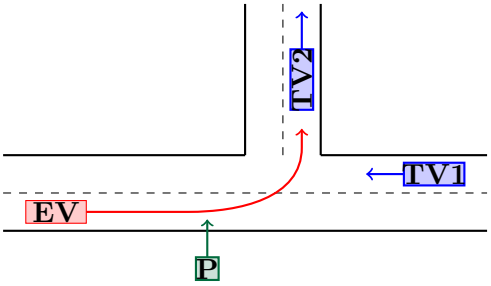


Figure 4.4.: Scheme of the urban framework considered in the simulations. © 2021 IEEE.

solution $\nu_h = 0\ \forall h$ when the current EV speed is zero. Note that the optimal reference speed ν_h must be updated solving (4.12) once every T_H seconds, and not at each iteration of the low-level controller.

4.2.4. Simulations of Trajectory Planning for Autonomous Vehicles considering Long-Term Reasoning

We discuss two numerical simulations developed in Matlab and based on the NMPC toolbox [166], showing the potential of the proposed control algorithm. We consider an urban intersection, see Fig. 4.4. Each road has one lane per direction, and the intersection is in the origin of the coordinate system. Units are omitted, as all quantities are given in SI units. The reference path of the EV is the center of the lane, and a Bézier curve is used at the intersection. The lane width is $w_\mathrm{lane} = 3$, all vehicles have a rectangular shape with $l_\mathrm{veh} = 5$ and $w_\mathrm{veh} = 2$, and $l_\mathrm{f} = l_\mathrm{r} = 2$, and the pedestrian has a squared shape $l_\mathrm{P} = w_\mathrm{P} = 1$. The maximum speed allowed is $v_\mathrm{max} = 13$. The EV input bounds are $\boldsymbol{u}_\mathrm{max} = [5, 0.52]^\top$, $\boldsymbol{u}_\mathrm{min} = [-9, -0.52]^\top$, $\Delta\boldsymbol{u}_\mathrm{max} = [9, 0.4]^\top$, and $\Delta\boldsymbol{u}_\mathrm{min} = -\Delta\boldsymbol{u}_\mathrm{max}$. For the TVs, we have $\boldsymbol{u}^\mathrm{TV}_\mathrm{max} = [5, 0.4]^\top$ and $\boldsymbol{u}^\mathrm{TV}_\mathrm{min} = [-9, -0.4]^\top$. We have

$$\boldsymbol{K} = \begin{bmatrix} 0 & k_{12} & 0 & 0 \\ 0 & 0 & k_{21} & k_{22} \end{bmatrix}, \tag{4.13}$$

with $k_{12} = -0.55$, $k_{21} = -0.63$, and $k_{22} = -1.15$. The high-level feedback gain $\boldsymbol{K}_\mathrm{H}$ has the same structure, with $k^\mathrm{H}_{12} = -0.34$, $k^\mathrm{H}_{21} = -0.21$, and $k^\mathrm{H}_{22} = -0.67$. For TVs, the noise covariance matrix is $\boldsymbol{\Sigma}^\mathrm{DO}_{\boldsymbol{w}} = \mathrm{diag}(0.15, 0.03)$, the safety margin $\epsilon^\mathrm{DO}_\mathrm{safe} = 4$, and the risk parameter is $\beta^\mathrm{DO} = 0.8$ for the trajectory planner and $\beta^\mathrm{TV}_\mathrm{H} = 0.4$ for the maneuver planner. For pedestrians, we use $\boldsymbol{\Sigma}^\mathrm{DO}_{\boldsymbol{w}} = \mathrm{diag}(0.05, 0.2)$, $\epsilon^\mathrm{DO}_\mathrm{safe}$=1, and $\beta^\mathrm{DO} = 0.9$ for the trajectory planner and $\beta^\mathrm{P}_\mathrm{H} = 0.5$ for the maneuver planner.

In (4.5a), $\boldsymbol{Q} = \boldsymbol{P} = \mathrm{diag}(0, 1, 1, 1)$, $\boldsymbol{R} = \mathrm{diag}(0.33, 5)$, $\boldsymbol{S} = \mathrm{diag}(0.33, 15)$, and $r_\mathrm{H} = 0.5$ in (4.12a). The reference speed for the EV is $v_\mathrm{ref} = 10$. In all the simulations, $d_\mathrm{ref} = \phi_\mathrm{ref} = 0$, the sampling time and prediction horizon are $T = 0.2$ and $N = 10$ for the trajectory planner, and $T_\mathrm{H} = NT = 2$ and $N_\mathrm{H} = 8$ for the maneuver planner.

For performance comparison, the following cost function is used, resulting from the computation of the stage cost of (4.5a) for all the N_sim steps of the simulation:

$$J_\mathrm{sim} = \sum_{\tau=1}^{N_\mathrm{sim}} \|\Delta\boldsymbol{\xi}_\tau\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}_\tau\|^2_{\boldsymbol{R}} + \|\Delta\boldsymbol{u}_\tau\|^2_{\boldsymbol{S}}. \tag{4.14}$$

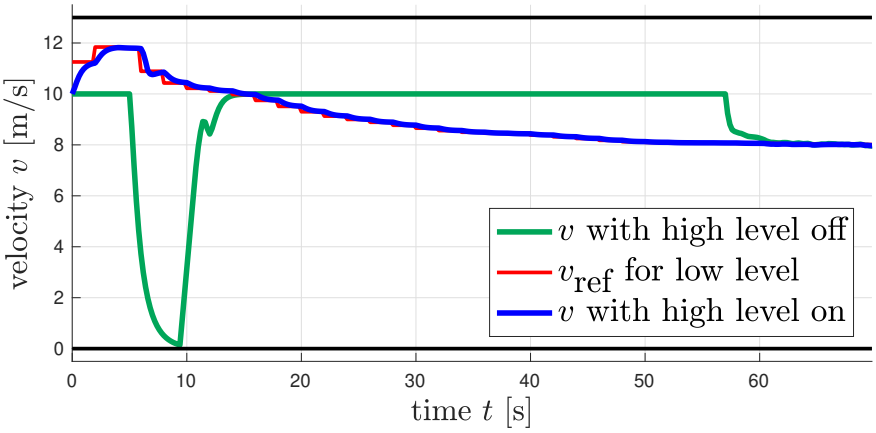


Figure 4.5.: EV speed profile in the anticipating TV scenario. © 2021 IEEE.

The last component of deviation of the state with respect to the reference trajectory $\Delta\boldsymbol{\xi}_{[4]}$ in (4.14) is computed considering the desired cruise speed $v_{\text{ref}} = 10$, independently of the optimal reference ν_0 computed by the maneuver planner.

Anticipating TV Scenario

In the first simulation, the EV initial state is $\boldsymbol{\xi}_0 = [-70, 0, 0, 10]^\top$, and two TVs are presentsee Fig. 4.4: TV1 approaches the EV from the right, driving in the opposite direction ($\boldsymbol{\xi}_0^{\text{TV1}} = [60, -7.5, 1.5, 0]^\top$), TV2 drives upward in the vertical road ($\boldsymbol{\xi}_0^{\text{TV2}} = [1.5, 0, 10, 8]^\top$). No pedestrian is considered in this scenario.

When the high-level maneuver planner is not used, the EV drives at cruise speed $v = 10$ up to the intersection. Then, TV1 is predicted to occupy the intersection before the EV can safely leave it free, thus the EV stops. Afterwards, the EV turns and proceeds at cruise speed until it reaches TV2, which is driving slower. Eventually, the EV decelerates, in order to keep the safety distance from TV2, based on the risk parameter β^{TV}. By decreasing the risk parameter β^{TV}, the trajectory planner can execute a more aggressive and possibly efficient motion. Conversely, by requiring more safety, the safety distances derived in Section 4.2.3 are enlarged, and this ultimately comes at the prices of a higher cost. SMPC permits to balance this trade-off, proving to be notably useful when the disturbances considered are unbounded, and cannot be handled by robust control approaches.

If the maneuver planner is active, the high-level predicts that a stop is necessary at the intersection and acts to prevent it from happening. Hence, a high reference speed is set at first, and the EV reaches the intersection while is still free, before being occupied by TV1. Then, the EV decelerates, reaching the desired cruise speed $v_{\text{ref}} = 10$. Finally, as soon as TV2 is visible, a smooth deceleration is planned.

Fig. 4.5 shows the speed profile of the EV if the high-level SMPC is inactive (green line) or active (blue), and the optimal reference speed v_{ref} (red) dynamically computed by the maneuver planner solving (4.12). The maneuver planner allows to optimize the speed, avoiding the stop at the intersection, and making the deceleration smoother when needed. The benefit is also visible in terms of the cost function, since we have $J_{\text{sim,1a}} = 2215.8$ for the first case, and $J_{\text{sim,1b}} = 781.2$ if the maneuver planner is used.

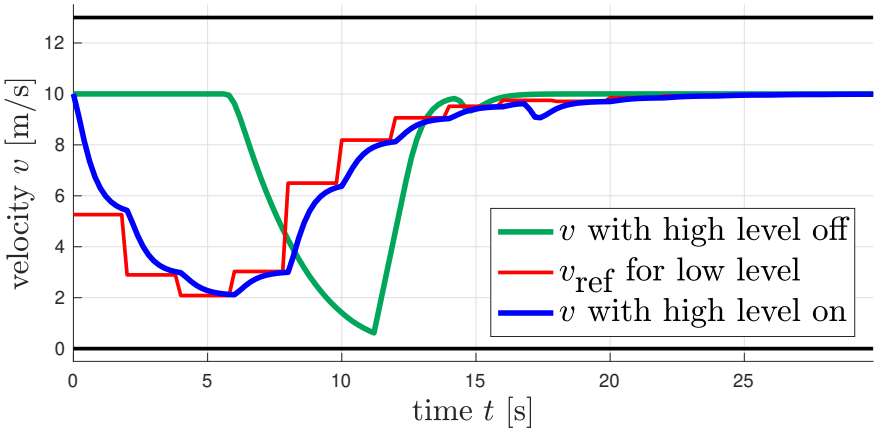


Figure 4.6.: EV speed profile in the pedestrian crossing scenario. © 2021 IEEE.

Pedestrian Crossing Scenario

In the second simulation, the EV initial state is $\boldsymbol{\xi}_0 = [-100, 0, 0, 10]^\top$, and a pedestrian intends to cross the street in front of the intersection ($\boldsymbol{\xi}_0^{\text{P}} = [-15, 0, -11, 1.2]^\top$), see Fig. 4.4. No TV is considered in this scenario.

Without the maneuver planner, at first the EV proceeds at cruise speed. Then, it rapidly decelerates, to keep the desired distance from the pedestrian, according to the risk parameter β^{P}, until the pedestrian completes crossing. Then, the EV accelerates again to reach the desired cruise speed $v_{\text{ref}} = 10$.

Conversely, if the maneuver planner is used, the EV stop caused by the crossing pedestrian is predicted. Yet, now the maneuver planner cannot set the speed in order to safely pass before the pedestrian crosses, and therefore an initial deceleration is planned. In such a way, the EV can safely pass the crossing point without stopping, since the pedestrian is already on the other side of the street.

The speed profile of the EV for this scenario is shown in Fig. 4.6, if the maneuver planner is inactive (green line) or active (blue). The optimal reference speed v_{ref} is represented in red. Even though the maneuver planner effectively avoids the full stop, the speed is significantly decreased in the first stretch, and the overall cost is $J_{\text{sim,2b}} = 2049.2$, which is slightly higher than in the cost in the first case, $J_{\text{sim,2a}} = 1992.0$, if the high-level maneuver planner is inactive.

4.3. Non-Conservative SMPC-Based Planning with Multi-Modal Uncertainty

In this section, we present a framework to control the EV accounting for multiple intentions of DOs while still limiting excessive caution. The framework is based on a combination of IMM, used to dynamically estimate the probability of candidate DO trajectories, and SMPC, in which the EV trajectory is planned by prioritizing the focus to DO trajectories based on their probability. This section is based on the work presented in [21].

4.3.1. Problem Formulation

As EV model we consider again the linearized and discretized kinematic bicycle model is expressed in the road-aligned frame outlined in Section 2.3.3. The EV state $\boldsymbol{\xi} = [s, d, \phi, v]^\top$ consists of the longitudinal and lateral position of the center of mass of the vehicle expressed in road-aligned coordinates, (s, d), of the yaw angle ϕ of the vehicle with respect to the reference path, and of the linear velocity v. The input is a two-dimensional vector $\boldsymbol{u} = [a, \delta]^\top$ including the linear acceleration a and the front steering angle δ. The EV is limited in the speed (2.16b), to prevent it from going backwards or violating the maximum allowed speed, in the lateral position (2.16a), to prevent the vehicle from leaving the road, and on the maximum values for acceleration and steering angle (2.16c), as well as on their maximum rate of change (2.16d).

The closed-loop dynamical models presented in Section 2.3.4 are used to describe possible behaviors resulting from the unknown intention. Such models are in a suitable form to be included in the IMM algorithm, which is recalled in Section 2.3.1. The DOs dynamics is modeled as a discrete-time point mass with state $\boldsymbol{\xi}^{\mathrm{DO}} = [x^{\mathrm{DO}}, v_x^{\mathrm{DO}}, y^{\mathrm{DO}}, v_y^{\mathrm{DO}}]^\top$, composed of longitudinal position x^{DO} and velocity v_x^{DO}, and lateral position y^{DO} and velocity v_y^{DO}, expressed in the inertial world frame. The system input $\boldsymbol{u}^{\mathrm{DO}}$, consisting of the longitudinal and lateral accelerations, is modeled as the sum of a feedback term and a perturbation term $\boldsymbol{w}_k^{\mathrm{DO}}$. The feedback term is designed to correct possible deviations from the desired trajectory $\boldsymbol{\xi}_{\mathrm{ref}}^{\mathrm{DO}}$. Then, the perturbation term is modeled as Gaussian noise and represents the uncertainty of the future DO motion. The EV has access to the current position of DOs.

Considering multiple possible future movements of DOs is beneficial to enhance safety. However, it is important to prevent unnecessary conservatism that arises when considering several candidate scenarios, yielding the following objective.

Objective 4.2. *The aim is to compute the trajectory of the EV taking into account multiple possible future intentions of DOs and balancing between safety and efficiency, i.e., the EV trajectory must consider multiple candidate future DO trajectories, limiting excessive conservatism due to very unlikely candidate trajectories.*

We use these simplified models with the ultimate goal of representing different intended behaviors of DOs. Although more precise prediction models representing the motion of traffic participants in more detail could also be adopted without compromising the validity of the presented approach, a precise description of the motion of traffic participants is beyond the scope of this work. For simplicity of notation, in the following a single DO is considered, although the same holds for each of the n_{O} DOs, independently of the number.

4.3.2. Overview of the Proposed IMM+SMPC Approach

Here we give an overview of our novel approach to plan the trajectory of an automated vehicle in presence of twofold uncertainty due to the DOs in the environment. The approach relies on the IMM and on SMPC, recalled in Section 2.3.1 and in Section 2.3.7, respectively.

For each DO an IMM is run to dynamically estimate the probability that the evolution of the trajectory will be the result of each one of the n_{I} possible considered intentions. In fact, prediction models from Section 2.3.4 yield a nominal future DO trajectory that results from an intention. Therefore, the IMM returns the probability of each candidate intention. The set of possible future intentions (and weighing matrices) is assumed to be specified in advance. On one hand, this is a limitation of the method, which will be discussed in Section 4.3.5. On the other hand, this allows to derive closed-loop models for DOs completely offline, storing n_{I} tuples $(\boldsymbol{F}^{(j)}, \boldsymbol{G}^{(j)}, \boldsymbol{H}, \boldsymbol{u}^{(j)})$ independently of the number of DOs that will be encountered. Then, any time a new DO enters the range of the sensors, a new IMM filter is created, using the stored closed-loop models.

We adopt an SMPC approach with probabilistic safety constraints of the form

$$\mathrm{P}[\boldsymbol{\xi}_k \in \mathcal{S}_j] \geq \beta_j \ \forall j = 1, \ldots, n_{\mathrm{I}}, \ \forall k = 1, \ldots, N, \tag{4.15}$$

in which $\mathcal{S}_j$ is a set representation of the collision avoidance conditions, one for each of the n_{I} candidate future trajectories of the DO. In an SMPC approach, instead of guaranteeing constraint satisfaction for every possible outcome of the uncertainty, constraints are required to hold only up to a pre-defined level of probability, $0 \leq \beta_j \leq 1$.

Parameter β_j, sometimes referred to as *safety level*, is a design parameter of the SMPC method that tunes the trade-off between safety and efficiency, and is typically chosen heuristically. Using the same value of parameter β_j in safety constraints (4.15) for all DO trajectories independently of their probability makes the motion planned for the EV conservative. Thus, in this work β_j explicitly depends on the probability of each possible high-level intention estimated by the IMM, see Figure 4.7. The prioritization of safety constraints depending on the probability of each high-level intention allows to reduce unnecessary conservativeness with respect to those trajectories that are currently considered unlikely, maintaining focus on the (possibly multiple) dominant ones. For example, during an initial transient phase in which enough data for accurate inference on the high-level intention are yet to be collected, all the candidate high-level intentions are considered and prioritized; then, as soon as the measurements show that some high-level intentions are safely ruled out, the related safety conditions are gradually relaxed, focusing the attention on the most plausible future motions. Hence, very conservative motions of the controlled system planned taking all the possible high-level intentions equally into account are avoided. Nevertheless, the probability for each trajectory is recursively updated every time new measurements are collected. Thus, if the DO changes intention and starts moving unexpectedly, the previously excluded motions gain higher priority, taking full advantage of the multi-modal prediction.

4.3.3. Trajectory Planning considering Multiple Intentions of Dynamic Obstacles

Here we outline our method by discussing the details of the constraint generation mechanism and formulating the optimal control problem. Further, we discuss improvements to the constraint generation mechanism to consider uncertainty in the estimated probabilities.

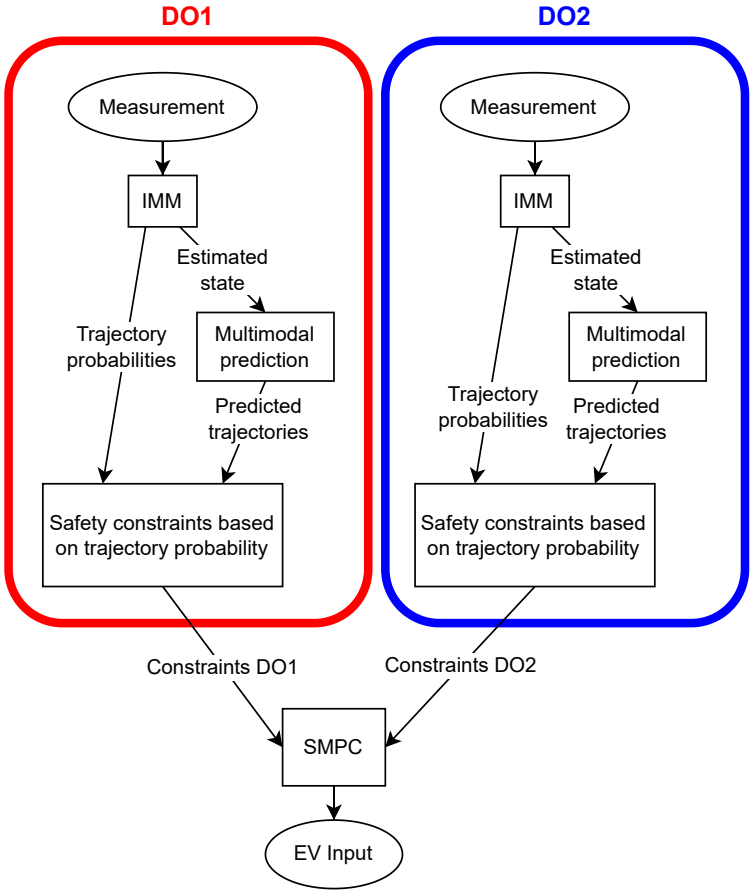


Figure 4.7.: Block diagram of the Interactive Multiple-Model+Stochastic Model Predictive Control (IMM+SMPC) combination. For each DO, measurement data are used to update the probability of each candidate trajectory using the IMM. Then, multi-modal predictions are obtained and eventually safety constraints are generated for all candidate trajectories with different priority in the SMPC Optimal Control Problem. © 2023 IEEE.

Design of Safety Constraints for Multi-modal Uncertainty

Here we discuss the generation of safety constraints preventing collisions of the EV with DOs accounting for multiple intentions. First, the nominal prediction and the associated covariance matrix are obtained through the procedure outlined in Section 2.3.7, for each DO and for each considered intention separately. For every DO, the last combined estimate $\hat{\boldsymbol{z}}$ obtained by the IMM algorithm is used to initialize n_I predictors, each iterating dynamics (2.5) for N steps. This yields $\tilde{\boldsymbol{z}}_k^{(j)}$, that is, the predicted state of the DO k-steps ahead, if trajectory j is followed, for every considered trajectory $j = 1, \ldots, n_\mathrm{I}$, for every future step $k = 1, \ldots, N$. Together with the state, the covariance matrix is also propagated along every candidate trajectory of the DO, yielding the uncertainty about each state prediction.

Remark 4.1. *All the predictors of the DO are initialized with the (same) combined estimate $\hat{\boldsymbol{z}}$ and combined covariance matrix $\hat{\boldsymbol{P}}$, because this is the most accurate information available. Initializing each predictor with the estimate $\hat{\boldsymbol{z}}^{(j)}$ and covariance matrix $\hat{\boldsymbol{P}}^{(j)}$ resulting from each filter j would deteriorate the performance.*

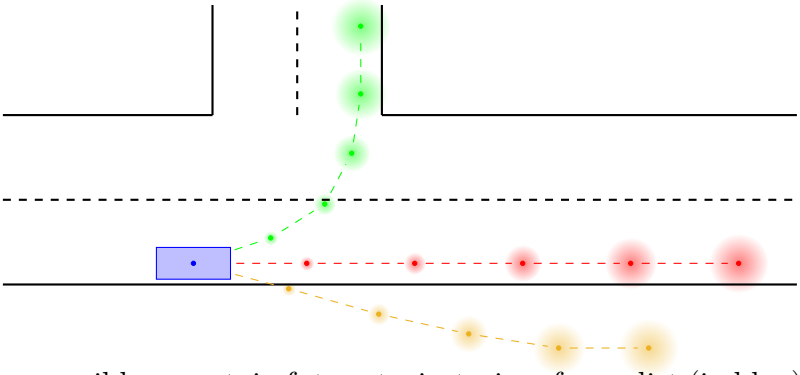


Figure 4.8.: Three possible uncertain future trajectories of a cyclist (in blue) in the proximity of an intersection, for a prediction horizon of $N = 5$. The uncertainty about each nominal prediction increases over time. © 2023 IEEE.

As a result, for every time step $k = 1, \ldots, N$ in the prediction horizon, $n_{\mathrm{I}} \times n_{\mathrm{O}}$ predicted states with associated covariance matrices are available, representing the n_{I} possible future positions for each of the n_{O} DOs surrounding the EV, see Figure 4.8. Next, this information is used to design non-conservative safety constraints.

Then, based on the nominal predictions and propagation of the uncertainty, chance safety constraints for the trajectory planning for automated urban driving are designed and reformulated into deterministic expressions that are included in the numerical optimal control problem discussed in Section 4.3.3. For every predicted step $k = 1, \ldots, N$, n_{I} safety constraints are generated, one for every candidate trajectory of the DO. The aim is to consider multiple possible future trajectories of the DO resulting from different intentions, avoiding excessive conservatism. Thus, constraints must be designed to:

- increase conservativeness for increasing confidence $\hat{\mu}^{(j)}$ in trajectory j for the DO (first level of uncertainty) and for increasing uncertainty about the nominal predicted trajectory $\tilde{\boldsymbol{z}}_k^{(j)}$ (second level of uncertainty);
- vanish if trajectory j is ruled out for the DO, i.e., $\hat{\mu}^{(j)} \to 0$.

We use ellipsoidal regions around the predicted position of the DO as safety conditions, as outlined in Section 2.3.7. Each region obtained is designed to containts the (real) future position of the DO with probability β_j [17]. The regions obtained in (2.39) increase for larger uncertainty about the prediction of the DO future states, and for a higher required probability of constraint satisfaction β_j; furthermore, for $\beta_j \to 0$ these regions collapse to points, that is, the constraints are neglected in practice.

In general, the required probability of constraint satisfaction β_j for a given chance constraint is a design parameter in the SMPC algorithm, which is set to balance between safety and efficiency. The larger β_j, the safer and the less efficient the motion planning. Therefore, a reasonable choice of β_j should allow for a small probability of constraint violation that does not compromise safety, yet reducing the conservatism typical of robust control approaches at the same time. In our method, the goal is to require stricter safety guarantees for those constraints related to the most likely future trajectories of the DO. Hence, β_j is selected explicitly depending on the estimated probability of each future trajectory $\hat{\mu}^{(j)}$ through the function $\beta_j = g(\hat{\mu}^{(j)})$.

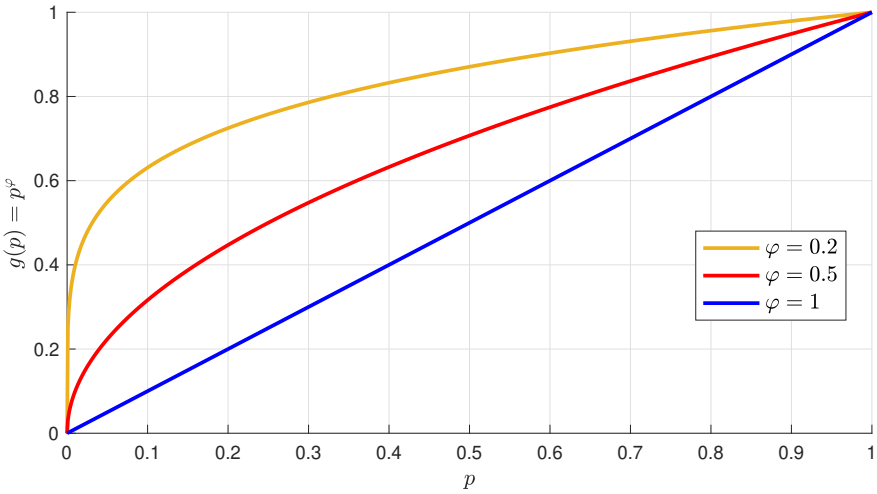


Figure 4.9.: Function $g(p) = p^\varphi$ used to stretch the rate of change of β_j with respect to $\hat{\mu}^{(j)}$ unequally across the interval $p \in (0, 1)$. © 2023 IEEE.

In principle, the identity $\beta_j = \hat{\mu}^{(j)}$ satisfies the premises on the constraints and thus is a good candidate for function $g(p)$, but other options are possible. For example, choosing $g(p) = p^\varphi$, with $0 < \varphi < 1$, the effect is that the required level of safety for relatively large probabilities $\hat{\mu}^{(j)}$ is increased, still allowing for a sudden loss of relevance of the constraints for very unlikely predicted trajectories, see Figure 4.9. In general, the function $g(p)$ is a design parameter which must satisfy

$$g(0) = 0, \tag{4.16a}$$
$$g(1) = 1, \tag{4.16b}$$
$$g(p_1) < g(p_2) \Leftrightarrow p_1 < p_2. \tag{4.16c}$$

This introduces flexibility in the method, as the rate of change of β_j with respect to $\hat{\mu}^{(j)}$ can be shirked or stretched unequally across the interval $p \in (0, 1)$. However, then values β_j do not represent true probabilities anymore.

Remark 4.2. *If the confidence in a trajectory approaches* 1*, the safety areas grow up to the entire support of the disturbance. Therefore, for unbounded uncertainties, like the Gaussian case, thresholding is needed to prevent the constraints from becoming excessively large.*

To conclude the discussion on the safety constraints, observe that the forbidden areas defined in (2.39) only refer to the position of the center of the DO. However, for collision avoidance, the physical dimensions of the EV and of the DO must be included. For this reason, each safety area is expanded to account for the deterministic length l_o and width w_o of the EV and of the DO (independently of the prediction model used), obtaining an ellipse with semi-major axis a_k^j and semi-minor axis b_k^j defined as

$$a_k^j = (\sigma_{x,k,j} + l_\mathrm{o})\sqrt{\zeta(\beta_j)}, \tag{4.17a}$$
$$b_k^j = (\sigma_{y,k,j} + w_\mathrm{o})\sqrt{\zeta(\beta_j)}. \tag{4.17b}$$

Remark 4.3. *In [30], the ellipse sizes are chosen as* $\breve{a}_k^j = \sigma_{y,k,j}\sqrt{\zeta(\beta_j)} + l_o$. *However, this option would not satisfy the requirement of "vanishing" constraints for* $\hat{\mu}^{(j)} \to 0$, *i.e., it would not rule out any constraint also for unlikely predictions, thus resulting in a conservative motion. Furthermore,* l_o *and* w_o *can be artificially increased with respect to the actual dimensions of the EV and of the DO, including additional margins, making the framework arbitrarily safe also using* (4.17).

In summary, safety requirements deliver n_I constraints as

$$q_k^j = 1 - \frac{(s_k - \tilde{x}_k^j)^2}{(a_k^j)^2} - \frac{(d_k - \tilde{y}_k^j)^2}{(b_k^j)^2} \leq 0, \tag{4.18}$$

with a_k^j and b_k^j from (4.17) with $\beta_j = g(\hat{\mu}^{(j)})$, fixed over the whole prediction horizon. As the EV model is referred to the road-aligned components, constraints are generated considering the projection in the Frenet frame of the predicted positions of the DO. If n_O DOs surround the EV, for each step k in the prediction horizon $n_\mathrm{O} \times n_\mathrm{I}$ constraints are generated.

Remark 4.4. *Constraints* (4.18) *are quadratic non-convex expressions. Equivalent representations linearized about the current working point would violate the requirement of neglecting the constraints if the trajectory is very unlikely (*$\hat{\mu}^{(j)} \to 0$*), and thus are not adopted.*

Optimal Control Problem

Considering the feasible state set $\mathcal{X}$ and input $\mathcal{U}$ for the EV dynamics (2.16), and state weighing matrices $\boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0$ and $\boldsymbol{P} = \boldsymbol{P}^\top \geq 0$, and input weighing matrices $\boldsymbol{R} = \boldsymbol{R}^\top > 0$ and $\boldsymbol{S} = \boldsymbol{S}^\top > 0$, at every sampling time, a sequence of control inputs $\boldsymbol{U} = \left[\boldsymbol{u}_0^\top, \dots, \boldsymbol{u}_{N-1}^\top\right]^\top$ for the EV is computed, where N is the prediction horizon, by solving

$$\min_{\boldsymbol{U}} \|\Delta\boldsymbol{\xi}_N\|_{\boldsymbol{P}}^2 + \sum_{k=0}^{N-1} \|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2 + \|\Delta\boldsymbol{u}_k\|_{\boldsymbol{S}}^2 \tag{4.19a}$$

$$\begin{aligned}
\text{s.t. } \boldsymbol{\xi}_{k+1} &= \boldsymbol{f}^\mathrm{d}(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k, \boldsymbol{u}_k), && \forall k = 0, \dots, N-1 && \text{(4.19b)}\\
\boldsymbol{\xi}_k &\in \mathcal{X}, && \forall k = 1, \dots, N && \text{(4.19c)}\\
\boldsymbol{u}_k &\in \mathcal{U}, && \forall k = 0, \dots, N-1 && \text{(4.19d)}\\
q_k^{ij} &\leq 0, \quad \forall i = 1, \dots, n_\mathrm{O}, && \forall j = 1, \dots, n_\mathrm{I}, \\
& && \forall k = 1, \dots, N, && \text{(4.19e)}
\end{aligned}$$

where the sampling time is assumed to b $t = 0$ for simplicity. (4.19b) is a compact representation of the linearized and discretized dynamics (2.15), and (4.19e) refers to safety constraints (4.18). The cost function (4.19a) consists of terms penalizing large inputs and large rates of change in the input, where $\boldsymbol{u}_{k-1}$ is the last applied input, which are included to boost comfort in the motion planning. (4.19a) penalizes also large deviations $\Delta\boldsymbol{\xi} = \boldsymbol{\xi} - \boldsymbol{\xi}^*$ of the EV state from the desired reference state $\boldsymbol{\xi}^*$, consisting of a reference speed and zero lateral displacement and yaw angle with respect to the road.

Regarding the computational complexity of the proposed method, the main bottleneck is the numerical solution of the optimal control problem (4.19) which, because of quadratic

safety constraints (4.19e), is not a quadratic program with linear constraints. However, observe that most of the quantities in (4.19) are computed before the numerical optimization. The IMM algorithm and the multi-modal prediction of the future trajectories of DOs and the error propagation are relatively fast operations that are performed online when the new measurements are collected, before the optimization starts, since the tuples $(\boldsymbol{F}^{(j)}, \boldsymbol{G}^{(j)}, \boldsymbol{H}, \boldsymbol{u}^{(j)})$ describing each considered trajectory are available offline. Furthermore, since such procedure for constraints generation is run for each DO independently, parallelization is possible. Moreover, the matrices of the linear EV model in (2.15) are also quickly evaluated online before the optimization starts, since their structure is known analytically and only depends on the current state of the EV and on the current curvature of the reference path.

Account for Uncertain Probability Estimates in Safety Constraints

Chance constraints described in the previous section might suffer from rapid variability due to the dependency on the probability assigned to each trajectory, which can vary repeatedly and abruptly during a transient phase in which the information collected through the measurements is not enough yet. Constraints changing significantly and frequently between consecutive iterations of the SMPC algorithm are not ideal, because they result in unreliable and therefore sub-optimal planning. The main cause for this is that still untrustworthy information is used to draw conclusions. To mitigate the effect, two solutions are proposed.

A first approach to limit the variability in the constraints consists in substituting $\hat{\mu}^{(j)}$ with the average value over the last $\bar{t}$ instants, i.e., $\beta_j = g\left(\frac{1}{\bar{t}} \sum_{h=-\bar{t}+1}^{0} \hat{\mu}_h^{(j)}\right)$. By doing so, too rapid and frequent fluctuations are attenuated, making the planning more "suspicious", so that it takes longer for the EV to trust the prediction and dare to react consequently when a single high-level behavior emerges as dominant. Heuristics can be used to choose a reasonable value for $\bar{t}$ to rule the trade-off.

Alternatively, a second idea to prevent the EV from taking contradicting decisions over consecutive steps due to the rapidly changing prediction environment consists in adding a further constraint to the first predicted step. The additional constraint is also elliptical, centered in the weighted average of the positions of the first predicted states of the DO $\sum_{j=1}^{n_\mathrm{I}} \hat{\mu}^{(j)} \tilde{\boldsymbol{z}}_1^j$, with a size depending on the uncertainty on the trajectory selection. This constraint is designed so that:

- it remains steadily large during the transient phase, as far as the uncertainty in the trajectory selection is large;
- it gradually moves toward the most likely prediction (being centered in the weighted average);
- it eventually vanishes when one trajectory becomes dominant, allowing the EV to dare overtaking.

Observe that with this advance, when focus is on a trajectory which is currently considered as the most likely, the other options are simply temporarily given lower priority. As required by the premises of the method presented in this work, no option is permanently ruled out. As soon as uncertainty about the trajectory selection increases again, the additional constraint for the first prediction step starts indeed inducing more caution consequently.

Figure 4.10.: Environment used in the simulation in CARLA. © 2023 IEEE.

Remark 4.5. *The additional constraint centered in the average predicted state of the DO might also be enforced to further prediction steps. However, in principle a constraint in the first predicted step only is enough to induce caution, as in an MPC fashion only the first input of the sequence is applied.*

Section 4.4 presents another approach to improve the generation of safety constraints based on BFT instead of classic probabilities.

4.3.4. Simulations of Trajectory Planning of Autonomous Vehicles considering Multiple Intentions of Dynamic Obstacles

The behavior of the EV controlled by the proposed algorithm is simulated in CARLA, to discuss the properties and benefits of the novel combination of the IMM and SMPC. We compare the performance with two similar approaches presented in previous works: one in which the probability of each candidate trajectories of the DOs is estimated but only the most likely is accounted for in the motion planning of the EV [17], and another in which multiple candidate trajectories are considered and regarded as equally likely [93]. Urban environments with significant uncertainty concerning the future trajectories of dynamic obstacles are considered. Although a simplified and high-level model of the EV (2.12) is used to plan the trajectory, the EV position is updated applying the determined input to a more detailed dynamical model. For a realistic physical representation of the involved agents, we rely on the CARLA library. CARLA is an open-source simulator for autonomous driving research, and we implemented our algorithm in Python using publicly available libraries. The optimal control problem (4.19) is solved using the *optimize.minimize* solver from *scipy*. Numerical values and further details are given in Appendix B.1. All simulations are shown at https://youtu.be/ZzJ_h71ccOk.

Interaction with Absentminded Cyclist: Simulation Setup

As first simulation scenario, we consider a straight road, with a sidewalk to the right, and an intersection to the left, as depicted in Figure 4.8, part of the simulation environment "Town01" in CARLA, see Figure 4.10. The EV must proceed straight on the road, considering the uncertain future motion of a cyclist, initially on the sidewalk. The three considered candidate intentions of the cyclist are: A) continue on the bike lane on the sidewalk, B)

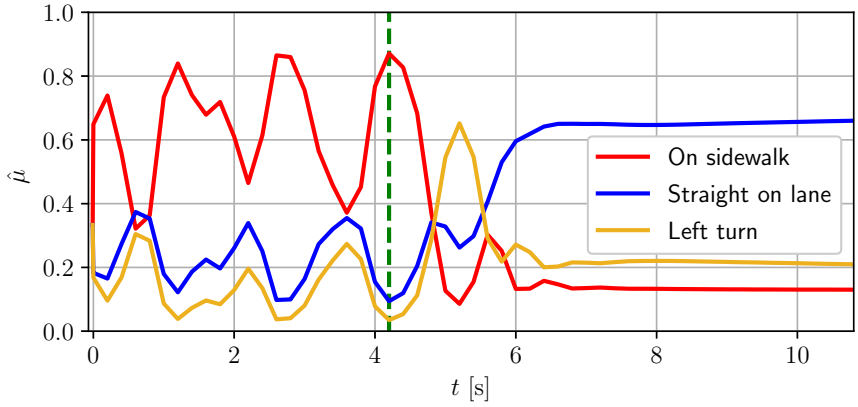


Figure 4.11.: Dynamical estimate of the probabilities of trajectory A (bike lane on the sidewalk, red), B (straight on the right lane, blue), and C (left turn at the intersection, yellow) during the simulation in which the cyclist eventually invades the lane and the IMM+SMPC method is used. The green bar shows when the cyclist finally starts moving toward the lane, at 4.2 seconds. © 2023 IEEE.

move to the road and proceed straight on the right of the lane, C) invade the road to eventually make the left turn. The challenge for the EV is to accommodate sudden and hazardous moves of the cyclist (a lane invasion), without inducing an excessively conservative behavior.

In the simulation, we consider a possible mismatch between the models used to predict the future trajectory of the DOs and their actual dynamical properties. Precisely, the cyclist motion is regulated by two PID controllers, one used to regulate the speed to the desired target value, one in charge for the lateral position. Therefore, all candidate future trajectories considered by the EV are imprecise. Furthermore, during the initial phase, the target lateral position of the cyclist is repeatedly changed, simulating a possibly absentminded behavior challenging to deal with, since at first it is unclear whether it is actually starting a maneuver to invade the EV lane or just proceeding erratically on the sidewalk. The method is tested for both cases, i.e., if the cyclist eventually invades the lane and if it stays on the sidewalk, showing the capability of the novel combination of IMM and SMPC in dealing with mode uncertainty avoiding excessive conservatism.

Efficient Planning Considering Multiple Intentions

In the first simulation, the cyclist starts demonstrating an unclear behavior and eventually leaves the sidewalk and invades the lane. The estimated probability for the three intentions, dynamically updated by the IMM based on the position measurements of the cyclist, are depicted in Figure 4.11. After an initialization with a-priori equal probability, the trajectory on the sidewalk (A) is shortly recognized as dominant, although with great fluctuations in the trajectory probability. Oscillations in the estimated probabilities are due to the fact that the actual behavior of the cyclist does not match any of the candidates trajectories considered by the EV; consequently, the confidence in each of them necessarily changes repeatedly as new measurements are collected. Thus, the EV regards a possible cut-in of the cyclist as sufficiently likely and maintains a safety distance. In fact, since the size of the safety ellipse around each candidate predicted trajectory of the cyclist is directly obtained from the

estimated probability for that candidate trajectory, the safety ellipses of multiple trajectories (straight on the sidewalk and lane invasion) are considerably large. Consequently, the set of feasible positions for the EV is so restricted that the EV is forced to proceed slowly, to be able to safely remain behind the cyclist in case a lane invasion takes place. Eventually, when the cyclist actually moves to the lane (after approximately 4.2 seconds), the EV is sufficiently far and the safety distance is maintained, revealing the enhanced safety resulting from considering multiple candidate intentions of the cyclist. The steady-states value of estimated probabilities of the trajectories depend on the a-priori switching probabilities.

In the second simulation, at first the cyclist shows a similar absentminded behavior (for the first 4.2 seconds), but eventually remains on the sidewalk. Once again, as long as the intention of the cyclist is unclear, the EV is forced to a cautious behavior, so as to be ready for a possible lane incursion by the cyclist, which is regarded to have a non-negligible chance. Thus, the EV is prevented from taking risky decisions while the collected information is not reliable enough to draw conclusions. Nonetheless, at around 4.2 seconds there is a critical change: the motion of the cyclist becomes more regular and the IMM consequently starts steadily estimating that the cyclist will remain on the sidewalk, see Figure 4.12. As soon as it appears clear that the cyclist will proceed straight and will not invade the lane, the other candidate trajectories (B and C) gradually exit the optimization framework of the EV, which hence dares accelerating. Therefore, our novel algorithm allows to consider multiple candidate trajectories for enhanced safety as far as they are sufficiently plausible, without resulting in unnecessary caution if one candidate trajectory is clearly dominant.

Observe that, as previously mentioned, in our simulations all models used to run the IMM and to predict the nominal trajectories of the cyclist are only approximations of real behaviors, i.e., none of the designed models perfectly matches the dynamics of the cyclist used in the CARLA simulator, which is actually controlled by two PIDs. Nevertheless, the goal of the models is to identify trends corresponding to different high-level intentions, and each trajectory obtained is considered to be a nominal realization only. Therefore, additional safety margins are added, and model discrepancies are included in the second level of uncertainty, i.e., uncertainty about the nominal predicted trajectory.

Comparison to State-of-the-Art Approaches

We now propose comparison simulations with two methods inspired by previous works, that do not consider multiple possible future trajectories for the DOs or do not prioritize them. In simulation 3, the EV only takes the (currently) most likely intention of the cyclist into account, with an approach similar to [17]. The same set of candidate intentions of the cyclist is considered and for each the probability is assessed. However, safety constraints are generated only with respect to the currently most likely candidate trajectory of the cyclist. In this simulation, for the first phase the EV, despite the large uncertainty in the trajectory selection, trusts the cyclist to remain on the sidewalk, because this is the dominant mode, although with highly fluctuating confidence. However, at around 4.2 seconds the cyclist starts moving toward the lane and candidate trajectory B gains focus, since the IMM detects another dominant intention. Still, now the EV is too close to the cyclist and thus reacts with an emergency braking and steering maneuver. An excess of confidence in the single most likely trajectory is limiting and multiple candidate trajectories should be accounted for.

Nevertheless, merely considering all candidate trajectories proves inefficient. In simulation 4, all candidate trajectories of the cyclist are taken into account with equally large safety

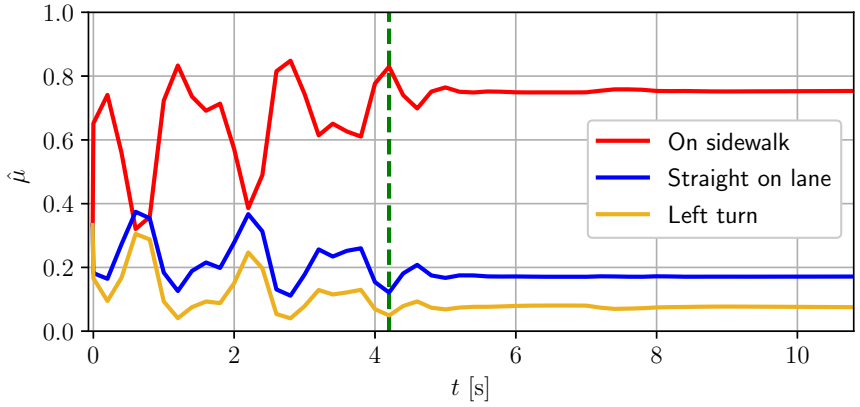


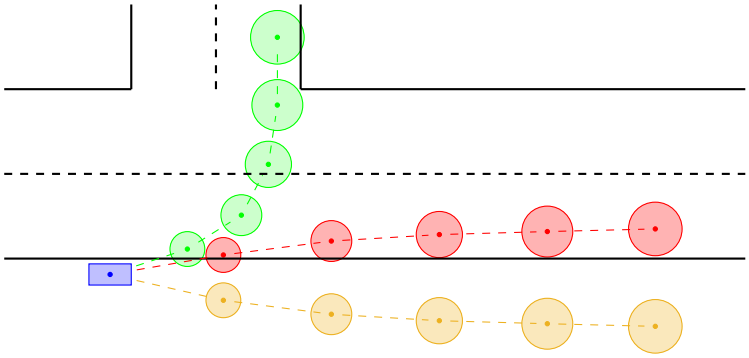
Figure 4.12.: Dynamical estimate of the probabilities of trajectory A (bike lane on the sidewalk), B (straight on the right lane), and C (left turn at the intersection) during the simulation in which the cyclist eventually remains on the sidewalk and the IMM+SMPC method is used. The green bar shows when the cyclist stops behaving absentmindedly, at 4.2 seconds. © 2023 IEEE.

areas, see Figure 4.13a. This approach is inspired by [93]. Initially, the EV behaves similarly to when controlled with the IMM+SMPC algorithm, that is, it maintains a distance and avoids overtaking at first. Yet, even when the cyclist starts moving regularly, clearly showing the intention of staying on the sidewalk, in this case the EV still does not dare accelerating and overtaking. In Figure 4.14, the speed of the EV is depicted. After a short acceleration at the beginning, the EV shortly starts slowing down and gradually reaches the speed of the cyclist. As a matter of fact, if all models are taken into account irrespective of their estimated probability, the future trajectories consisting of a movement toward the lane are never ruled out and thus the EV stays in the distance, slowing down. Equally weighing all candidate future trajectories results in an overly conservative motion planning for the EV, preventing the method from application in real traffic scenarios. With our method, trajectories of the DOs currently considered unlikely result in constraints that can be neglected in practice, without overly restricting the motion of the EV if unnecessary, see Figure 4.13b.

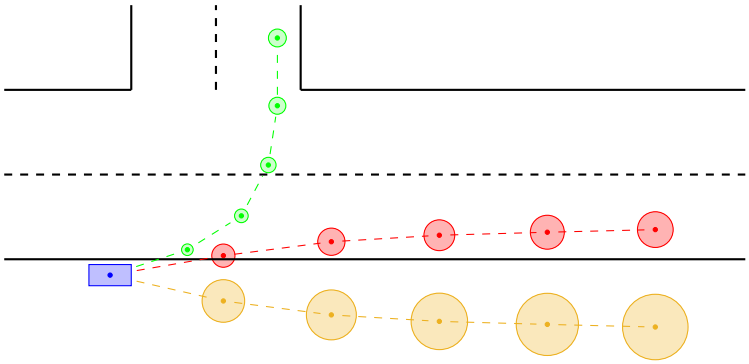
Number of Candidate Future Trajectories

Finally, we discuss the role played by the number of candidate trajectories of the cyclist considered in the IMM through additional simulations. Given that each candidate trajectory is meant to represent a specific intention of the cyclist, including more trajectories is intuitively beneficial. However, there are a few shortcomings that must be taken into account. We test the IMM+SMPC method considering 9 candidate trajectories for the cyclist. For each of the three candidate trajectories initially considered (straight on the sidewalk, lane invasion, and left turn), two more are now included, resulting from different rates of convergence to the same target state, see Appendix B.1.

In simulation 5, the cyclist eventually invades the lane, whereas in simulation 6 the cyclist eventually remains on the sidewalk. Considering multiple candidate trajectories to represent relatively similar motions is not beneficial because it results in unnecessary indecision about the trajectory of the cyclist. Figure 4.15 shows the estimated probability for each candidate



(a) Equal priority independently of the estimated probability. © 2023 IEEE.



(b) Prioritization depending on the estimated probability. © 2023 IEEE.

Figure 4.13.: Forbidden areas around nominal predictions of the cyclist for $N = 5$, 4.13a giving equal priority to each candidate trajectory and 4.13b prioritizing.

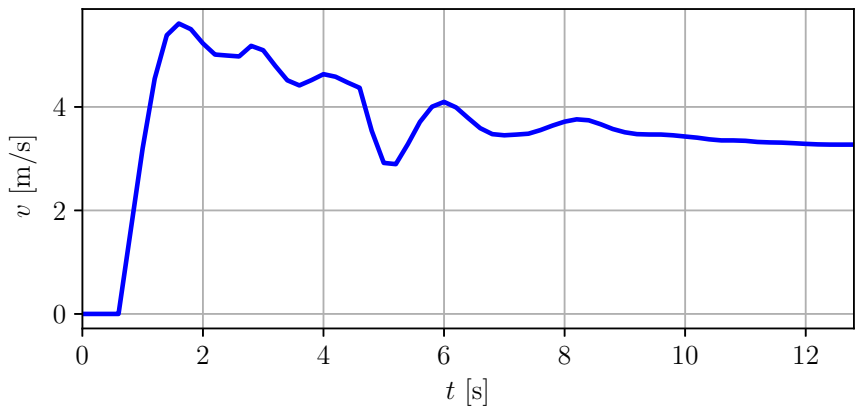


Figure 4.14.: Speed of the EV during the comparison simulation in which the cyclist remains on the sidewalk and collision avoidance constraints are enforced for all three candidate trajectories irrespective of the estimated probability. © 2023 IEEE.

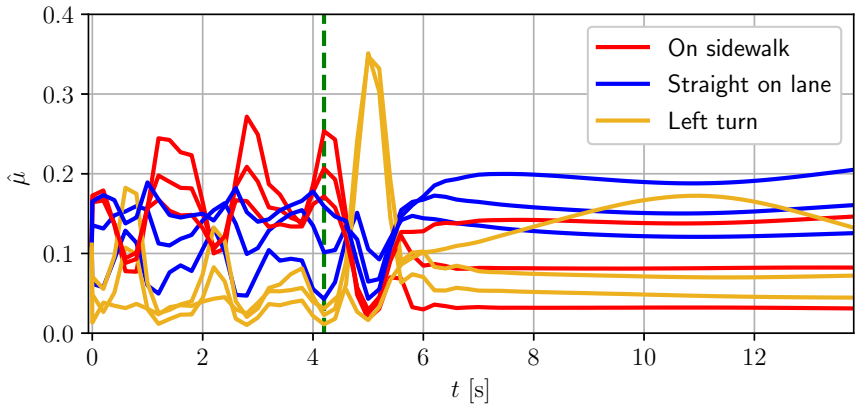


Figure 4.15.: Dynamical estimate of the probabilities of the 9 trajectories of the cyclist consisting of different convergence rates to the three target states: bike lane on the sidewalk, straight on the right lane, and left turn at the intersection. The green bar shows when the cyclist finally starts moving toward to the lane, at 4.2 seconds. © 2023 IEEE.

future trajectory of the cyclist during the simulation in which it eventually invades the lane. Although the behavior of the cyclist is the same as in the previous simulations, due to the numerous candidate trajectories considered, in this case there is persistent indecision about the future trajectory of the cyclist, with the most likely trajectory only estimated at around 0.2 even after the initial phase with the absentminded behavior. As a result, the safety regions around the predicted future positions of the cyclist are rather small, see Section 4.3.3, and the optimization framework becomes very complex. For example, depending on the predicted future positions of the cyclist, it might happen that at a certain iteration the EV can find a feasible path in a given direction and accelerates, whereas at the next sampling time the framework is so different that the EV needs to suddenly slow down and turn. These sudden changes result in a trembling behavior for the EV, visible for example at around 4 seconds of simulation 5 and at around 6 seconds of simulation 6.

Similarly, we test the method in another setup with several candidate trajectories for the cyclist. In these simulations, further candidate trajectories with respect to the original three are designed considering different target speeds, also resulting in 9 candidate trajectories overall. Simulation 7 shows the lane invasion, whereas in simulation 8 the scenario in which the cyclist eventually remains on the sidewalk is depicted. In both simulations a similar trembling behavior can be observed. Furthermore, in this case the safety areas are so small and displaced sufficiently distant from one another that the EV always manages to find a feasible path within the areas to be avoided. Thus, the EV never even slows down. Figure 4.16 shows the speed profile of the EV in case the cyclist eventually remains on the sidewalk, comparing the performance depending on the considered candidate trajectories of the cyclist. If the 9 candidate trajectories are based on different convergence rates to the same target states, the EV starts accelerating but needs to slow down shortly after 3 seconds, in order to keep a safety distance from the cyclist even in case of a possible lane invasion, until the incursion is completely ruled out (red curve in the figure). Conversely, if the 9 trajectories are based on different target speeds of the cyclist, the EV basically does not even slow down (blue curve). In the latter case, although 6 of the 9 considered trajectories

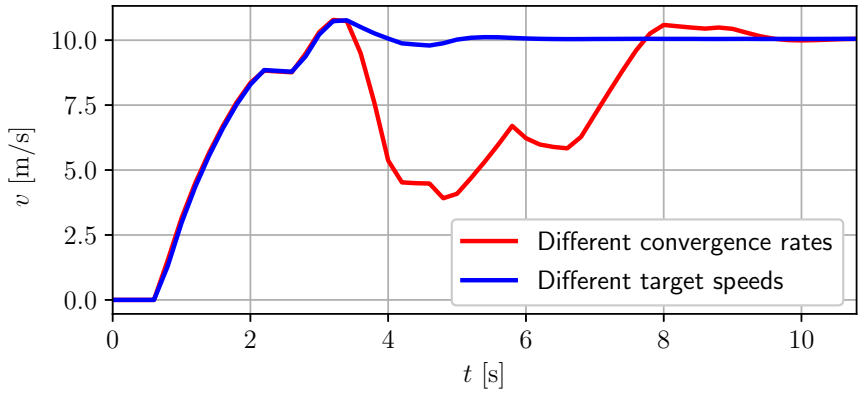


Figure 4.16.: Speed of the EV during the simulation in which the cyclist remains on the sidewalk, considering 9 candidate trajectories of the cyclist consisting of different target speeds or different convergence rates. © 2023 IEEE.

of the cyclist are partially located on the lane, their safety areas are so small that the EV basically does not consider them, because none of trajectories is regarded likely enough to generate a meaningful safety area.

To allow the EV to benefit from the IMM+SMPC method even in presence of several trajectories very similar to one another, the safety area should be made artificially very large also for small probabilities. However, a saturation mechanism should then be added, avoiding unreasonably large safety areas if one model perfectly matches the data and its estimated probability increases. Moreover, this would result in relatively large safety margins with respect to also unlikely trajectories, which is not efficient for the EV motion planning purposes, as discussed in the previous comparison with alternative approach [93].

Interaction with Absentminded Cyclist: Quantitative Comparison

We compare our novel IMM+SMPC algorithm with the two comparison methods from [17], [93] considering two metrics. On one hand, we evaluate the average computation time for each iteration of the IMM+SMPC method for the discussed simulations; specifically, in each iteration, we include the update step of the IMM, the multi-modal trajectory prediction algorithm and the generation of safety collision avoidance constraints, and the solution of the optimal control problem. The average computation time is a primary concern for real-time applicability of the proposed algorithm, as an update must be performed at each sampling time, iteratively re-planning the optimal trajectory of the EV. Secondly, to quantitatively compare the performances in terms of deviations from the desired cruise speed and lateral position of the EV, and in terms of usage of the input, we evaluate the average running-cost of the optimal control problem (4.19). For simulation steps $t = 1, \dots, N_{\text{sim}}$, and for weighing matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{S}$ chosen as in the SMPC cost function used in the simulation, we define:

$$J_{\text{sim}} = \sum_{t=1}^{N_{\text{sim}}} \frac{\left(\|\Delta\boldsymbol{\xi}(t)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(t)\|_{\boldsymbol{R}}^2 + \|\Delta\boldsymbol{u}(t)\|_{\boldsymbol{S}}^2 \right)}{N_{\text{sim}}}. \tag{4.20}$$

The cost (4.20) is a natural choice as a metric to rank the performances of the considered algorithms, given that are all obtained as solution of an optimization problem aiming at

Table 4.1.: Numerical comparison of simulations

Scenario	Number of DO models	Comp. time	Cost J_{sim}
cyclist invades lane	3	0.138 s	328.9
	1 (most likely only, [17])	0.138 s	252.7
	9 (different rates)	0.168 s	309.8
	9 (different target speeds)	0.155 s	317.3
cyclist on sidewalk	3	0.120 s	212.6
	3 (non-prioritized, [93])	0.111 s	348.2
	9 (different rates)	0.144 s	228.5
	9 (different target speeds)	0.123 s	194.2

minimizing such a cost. Performances are summarized in Table 4.1. Observe that, although the EV assumes the cyclist to behave stochastically, the actual cyclist motion in the simulations here is deterministic. Therefore, the reformulation of chance safety constraints in the SMPC optimal control problem (4.19) yields always the same constraint, thus running each scenario once is sufficient.

For all considered methods, the average computation time is significantly shorter than the sampling time $T = 0.2$ s used. Focusing on the scenario in which the cyclist eventually invades the lane, we observe that the method from [17], which only considers the currently most likely prediction of the cyclist, yields a computation time similar to the IMM+SMPC method with the initial 3 candidate trajectories, but a noticeably smaller cost. The explanation for the lower cost lies in the fact that by considering only the most likely future motion of the cyclist, the EV basically ignores the cyclist for most of the simulation and only takes it into account when the relative distance is too small to prevent the collision (which is not considered in the computation of cost J_{sim}). Meanwhile, increasing the number of candidate future trajectories of the cyclist slightly reduces the cost (around 5% of reduction), at the price of considerably increasing the average computation time (between 10% and 20%).

In the second simulation scenario, in which the cyclist in the end remains on the sidewalk, the average computation time is generally shorter than in the first scenario. As a matter of fact, in this case at last the cyclist is predicted to occupy areas outside of the road boundaries, therefore the collision avoidance safety constraints are always satisfied in practice and the optimal control problem is solved quickly. Compared to our proposed IMM+SMPC method with the initial 3 candidate trajectories, the comparison method similar to [93], in which future candidate trajectories are regarded as equally likely, results in a slightly shorter average computation time, but in a 60% increase in the cost. Indeed, without prioritization of the candidate trajectories, the EV never excludes the possible lane invasion, even when the cyclist clearly shows a different intention. Consequently, the EV continues to proceed unnecessarily slow, without daring to speed up, placing excessive focus on an unlikely outcome. Furthermore, we observe the possible shortcoming in the usage of several candidate trajectories. If the 9 candidate trajectories based on different convergence rates

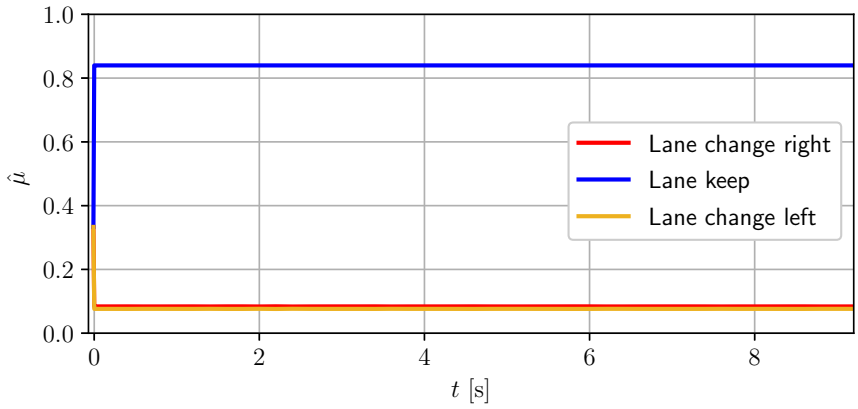
are considered, both the average computation time and the cost J_{sim} marginally increase. However, this is noticeably not the case if we consider the 9 candidate trajectories for the cyclist based on different target speeds. The reason for this lies in the fact that in this case the EV can always find a viable path and all multiple predicted trajectories of the cyclist are essentially disregarded, as previously discussed, compromising the efficacy of the method.

Overtaking Maneuver in Uncertain Traffic: Simulation Setup

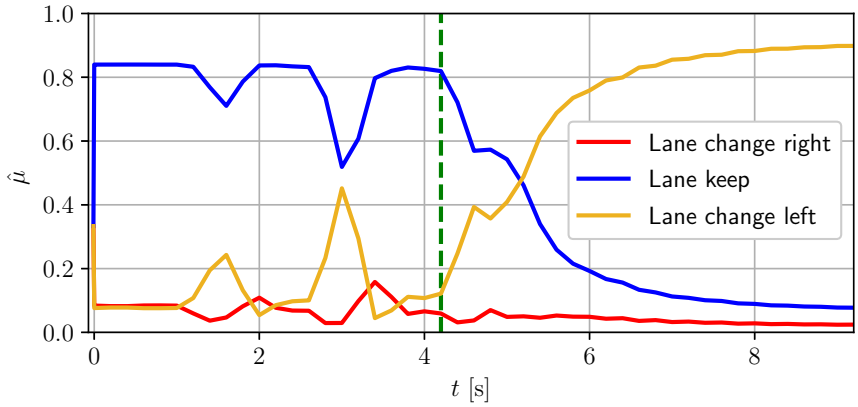
Furthermore, we test our algorithm in a different setting, specifically in a traffic scenario including the interaction with two TV. We consider a three-lane road, part of the simulation environment "Town05" in CARLA, in which the two TVs are both initially on the right-most lane, both located in front of the EV, which approaches from the center lane. The EV must eventually overtake both TVs, who are driving slower, avoiding overly conservative maneuvers when unnecessary. TV1 proceeds on the right most lane regularly, whereas TV2 exhibits a trembling behavior. Such irregular behavior of TV2 could be just an erratic driving on the right-most lane due to distraction of the driver, but could also be understood as a possible initial movement to the left lane to eventually overtake the slower TV1. For both TVs we consider three candidate future trajectories, consisting of executing a lane change to the left, keeping the current lane, and lane change to the right. Also in this case, the mismatch between the models of the DOs assumed by the IMM and the physical actuation is considered, controlling the TVs with PD controllers. The irregular behavior of TV2 is realized along the lines of that of the cyclist in the previous simulations. Observe that in this scenario the lateral motion plays a major role in the trajectory planning of the EV.

Safe Motion Accounting for Multiple Intentions of Traffic Participants

We first consider the case in which TV2, after some hesitation causing an ambiguous movement, changes lane to overtake TV1. The estimated probabilities for the two TVs in this scenario are shown in Figure 4.17. The candidate maneuver *lane keep* for TV1 is quickly recognized as dominant and the corresponding probability is steadily the most likely. Conversely, the trembling behavior of TV2 is reflected on larger fluctuation in the estimated probabilities. Although *lane keep* is recognized as the most likely maneuver, the possibility of *lane change left* is also non-negligible and eventually the latter is recognized as the dominant mode. Videos 9 and 10 represent this scenario both if the novel IMM+SMPC algorithm is used to control the EV and if the comparison method considering only the most likely maneuver of DOs [17] is adopted. Thanks to the IMM+SMPC combination, the EV can pre-account for a possible lane change maneuver of TV2 before TV2 finally initiates a lane change maneuver. Therefore, when the lane change finally occurs, the EV can still guarantee satisfaction of the safety conditions by moving to the left-most maneuver, without the necessity to slow down. Conversely, if the comparison algorithm from [17] is used, the EV trust TV2 to remain on the right-most lane and does not ponder a possible lane-change maneuver until too late. The difference is evident by comparing the planned trajectory of the EV in the two cases. Figure 4.18 represents the traffic configuration and the planned trajectory of the EV at simulation time $t = 4.40$ s, that is, when the lane change has not been clearly initiated yet and TV2 still lies within the right-most lane. Although still relatively unlikely, the IMM+SMPC allows to weigh in a possible lane change maneuver and therefore the planned trajectory is shifted to the left at the end of the prediction horizon



(a) TV1. © 2023 IEEE.



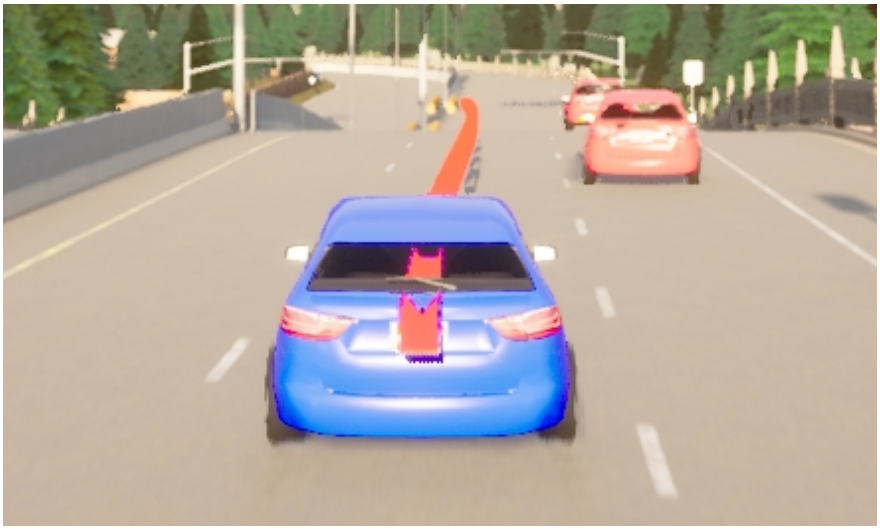
(b) TV2. © 2023 IEEE.

Figure 4.17.: Dynamical estimate of the probabilities of *lane change right*, *lane keep*, and *lane change left* for both TVs in the simulation in which TV2 eventually overtakes TV1. The green bar shows when the ambiguous behavior of TV2 ends and a left lane change is clearly initiated, at 4.2 seconds.

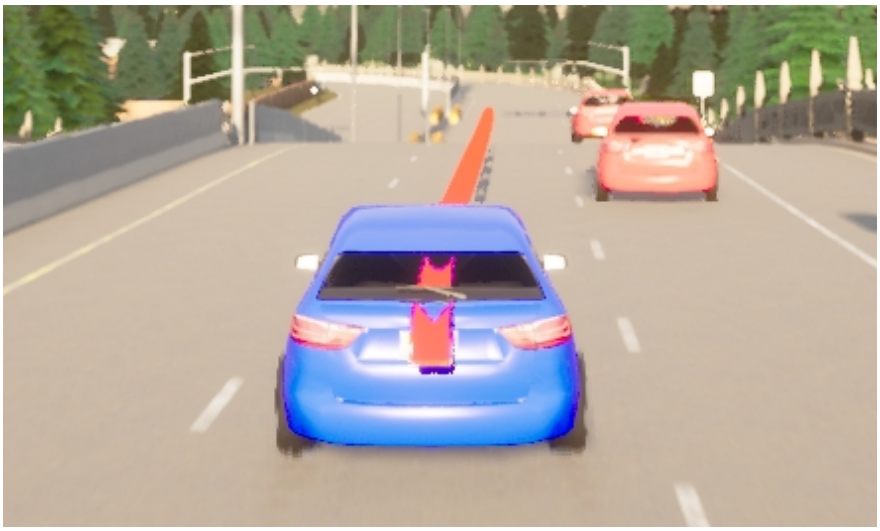
(see Figure 4.18a), i.e., when the EV is predicted to get closer to TV2, in order to observe the safety distance in case TV2 should actually initiate to move to the left. Thus, when this actually happens, the EV can smoothly continue to shift to the left and eventually overtakes TV2 from the left lane, without decelerating. Vice versa, the comparison method from [17] only accounts for the most likely future trajectory of TV2 (lane keeping), and completely neglects a possible lane change; thus, the planned trajectory of the EV continues straight also at the end of the prediction horizon (Figure 4.18b). When TV2 moves to the center lane, the EV is too close to just safely move to the left-most lane, thus steers emergently.

Efficient Motion Accounting for Multiple Intentions of Traffic Participants

Then, we simulated the same traffic scenario, but at the end of the ambiguous phase TV2 eventually keeps the right-most lane. The estimated probabilities for TV2 in this case are represented in Figure 4.19; the behavior of TV1 is unchanged, therefore the estimated probabilities are as in Figure 4.17a. We test both the IMM+SMPC combination and the



(a) Novel IMM+SMPC algorithm. At the end of the prediction horizon, a possible lane change of TV2 is accounted for, reflecting in a visible left turn at the end of the planned trajectory of the EV. © 2023 IEEE.



(b) Comparison method [17]. Only the currently most likely intention (lane keeping) is considered for TV2, thus the planned EV trajectory remains at the center of the lane also at the end of the prediction horizon. © 2023 IEEE.

Figure 4.18.: Planned trajectory of the EV at time $t = 4.40$ s, before TV2 changes lane.

comparison algorithm from [93], which takes all candidate trajectories into account equally; the simulations are represented in videos 11 and 12, respectively. Using the IMM+SMPC combination, the planned trajectory while TV2 behaves ambiguously is consistent with the former case. At first, a possible lane change of TV2 is accounted for and thus the planned trajectory of the EV shows a temporary shift to the left for those instants in the horizon in which the EV is predicted to get close to TV2. Figure 4.20 shows the planned trajectory of the EV at time $t = 4.80$ s, shortly after TV2 ends the swinging behavior. The lane change of TV2 is not completely ruled out yet, so the planned trajectory of the EV includes a shift to the left later in the prediction horizon. However, as soon as TV2 exhibits a more regular behavior consistent with a lane keeping intention, the estimated probability of the maneuvers

become more steady and the lane change is gradually ruled out, so that the shift to the left of the EV is not necessary anymore. As a result, the EV eventually overtakes remaining on the center lane, avoiding overly conservative behaviors. In contrast, the comparison method from [93] never rules out the lane change of TV2, since all candidate trajectories are regarded as equally plausible irrespective of the estimated probabilities. Hence, the EV motion is unnecessarily cautious and the EV moves to the left lane to overtake both vehicles, potentially slowing down the traffic flow on the left-most lane.

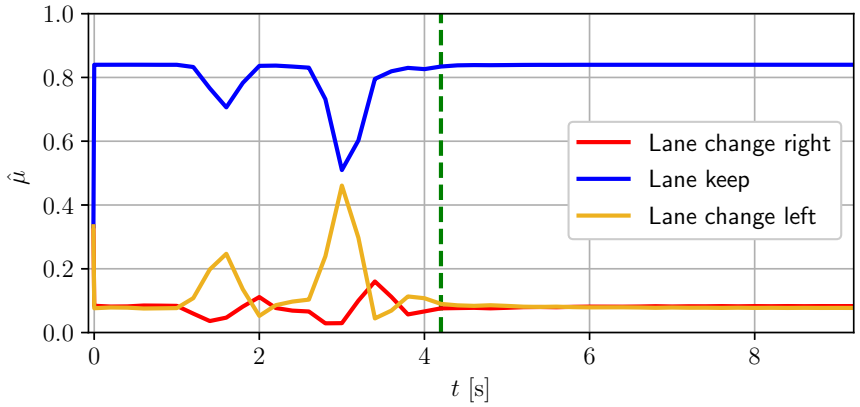


Figure 4.19.: Dynamical estimate of the probabilities of *lane change right*, *lane keep*, and *lane change left* for TV2 in the simulation in which TV2 remains on the right-most lane. The green bar shows the moment in which the ambiguous behavior of TV2 ends, at 4.2 seconds. © 2023 IEEE.

Overtaking Maneuver in Uncertain Traffic: Quantitative Comparison

Table 4.2 proposes a numerical evaluation of the performances and comparison of the algorithms through the same metrics previously introduced, i.e., the average computation time for each iteration of the IMM+SMPC algorithm and metrics (4.20). Considering the scenario in which TV2 eventually changes the lane, the performances of the comparison method from [17] are affected by the fact that the method notably fails in yielding a safe trajectory and thus the IMM+SMPC algorithm yields a considerably lower cost and average computation time. In the other scenario, in which TV2 ultimately remains on the right-most lane, the IMM+SMPC also yields a significantly lower average computation time, due to the fact that eventually the EV disregards the possible lane change of TV2 and the overly conservative lane change maneuver of the EV is avoided. Furthermore, compared to the simulations with the cyclist, we observe that increasing the number of DOs surrounding the EV does not severely affect the computation time, confirming that running a new IMM algorithm for each DO is relatively inexpensive and that the computational burden is driven by the solution of the optimal control problem (4.19). Conversely, comparing the cost J_{sim} for different scenarios is not informative, due to the different setup of the simulations.

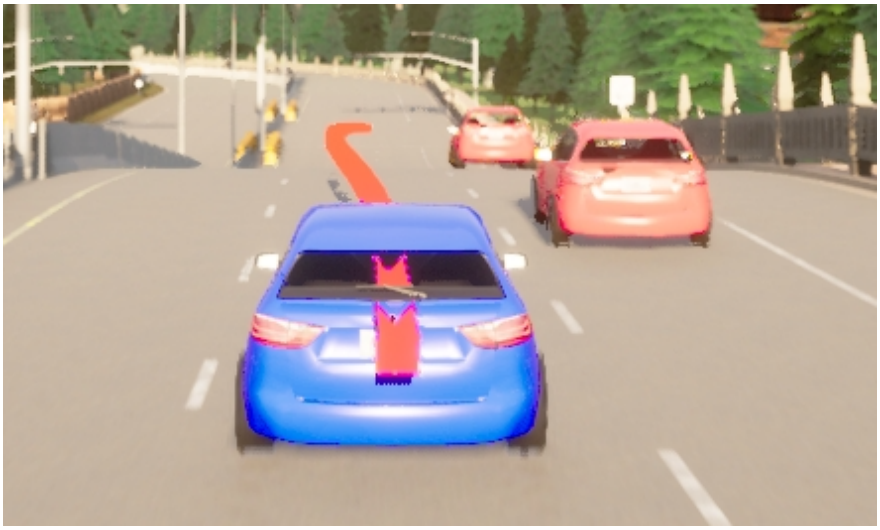


Figure 4.20.: Planned trajectory of the EV using the novel IMM+SMPC algorithm at time $t = 4.80$ s in the scenario in which TV2 eventually does not execute the lane change. © 2023 IEEE.

Table 4.2.: Numerical comparison of simulations

Scenario	Number of DO models	Comp. time	Cost J_{sim}
TV2 changes lane	3	0.135 s	84.8
	1 (most likely only, [17])	0.174 s	137.7
TV2 keeps lane	3	0.121 s	76.8
	3 (non-prioritized, [93])	0.172 s	86.5

4.3.5. Discussion

In this work, the set of candidate possible trajectories resulting from high-level intentions is assumed known. Yet, the choice of such possible trajectories is an open problem. Recorded data can be used to synthesize accurate trajectories describing typical vehicles, pedestrians and cyclists behaviors using Deep Learning [167] or Inverse Optimal Control [46].

Moreover, choosing the number of possible different intentions to be considered is also a challenge, in which conflicting considerations arise. Including numerous trajectories is tempting, as this allows to potentially approximate more accurately an increasing number of motions. For example, in [17] several models are used to take into account different "rates" of convergence in a lane change maneuver. However, in the method proposed in this work this choice would be problematic, leading to persistent indecision in the choice of the trajectory, and therefore to a continuously changing prediction environment, particularly if none of the trajectories considered is "precise enough". Therefore, in practice, it is preferable to consider trajectories sufficiently "differentiated", i.e., representing significantly different high-level intentions. Hence, the uncertainty about the trajectory selection is limited to those cases in which the behavior is really highly unpredictable, and not caused by oscillations between trajectories resulting in a very similar motion, bearing in mind that those candidate

trajectories are anyway only approximations of the real behavior. The approximated models can be dynamically improved by fitting the new data, with an approach consisting of multi-modal variable structures as presented in Section 3.3.

In Section 4.3.3 it was explained how, for every prediction step, $n_{\mathrm{O}} \times n_{\mathrm{I}}$ safety constraints are generated, needed to take into account all the n_{I} possible intentions that each of the n_{O} DOs might be pursuing. However, if candidate trajectories are chosen consistently with the aforementioned considerations, the collected measurements allow the IMM to shortly rule out most of the trajectories, leaving just a few reasonable options for each DO. The unlikely trajectories result in very small elliptical constraints, which have little physical meaning and do not influence the motion planning in practice. However, they can considerably slow down the numerical solution of the optimal control problem. Therefore, it is recommended to set a threshold for the parameters β_j below which no constraint is generated. This does not reduce the capabilities of the method in taking multiple possible trajectories into account, as the unlikely trajectories are still considered in the IMM and the related constraints would be included again in the optimization as soon as the probability should increase. Therefore, this threshold would just speed up the computation by explicitly ignoring trajectories that are currently not considered likely. Hence, several different sufficiently "differentiated" trajectories can be considered, allowing to consider a wide range of intentions.

Furthermore, although a new IMM filter must be implemented and run online for every DO surrounding the EV, the resulting workload remains relatively limited compared to that of the online solution of the constrained optimization problem, which dominates the overall computational demand of the proposed method, as usual for motion planning algorithms.

Finally, we emphasize that the concept of prioritization of chance constraints depending on the probability of a given outcome can be employed in other applications, possibly using multi-modal estimation algorithms other than IMM, whose implementation and computational workload are, however, undemanding. The formulation of chance constraints is not limited to the procedure presented in this work, and different options are possible.

The prioritization of candidate future DO trajectories heavily relies on the estimated probabilities of such trajectories. If the BFT information fusion framework presented in Section 3.2 is used to process the information on DOs, the epistemic uncertainty can also be exploited in the decision of the safety margins of the safety constraints. This approach is presented in the following section.

4.4. Non-Conservative SMPC-Based Planning with Multi-Modal Uncertainty considering Mode Estimation Uncertainty

While in the previous section addresses the problem of prioritizing focus for DO trajectories depending on their estimated probability, in this section we extend the approach by addressing uncertainty in the estimated probability. We assume that the estimation is obtained from the framework presented in Section 3.2, and on top of that we discuss how the SMPC planning can leverage the quantitative measure of the uncertainty in the estimated probabilities. This section is based on the work presented in [22].

4.4.1. Problem Formulation

We consider the trajectory planning problem of an autonomous vehicle in presence of multi-modal uncertainty caused by multiple possible future trajectories of DOs. It is assumed that several sources collect information about the motion of DOs and that information is combined using the BFT approach, that is revised in Section 2.3.5. As frame of discernment $\Theta = \{\theta_1, \ldots, \theta_{n_\mathrm{I}}\}$, that is, as set possible outcomes, we consider a set of n_I mutually exclusive candidate (nominal) DO trajectories. Information about each DO is given in the form of an opinion $\boldsymbol{\omega} = [b_1, \ldots, b_{n_\mathrm{I}}, \mu]^\top$ (2.28), where $b_1, \ldots, b_{n_\mathrm{I}}$ represent the belief masses assigned to candidate trajectories, and μ represents a quantitative measure of the uncertainty of the estimation. Therefore, not only it is estimated how likely each candidate future DO trajectory is, but also a quantitative measure of the subjective confidence in the reliability of the estimation is provided and exploited. Opinions might be produced, e.g, by the framework for to combine information from different sources presented in Section 3.2.

The EV trajectory is planned in SMPC fashion, which is outlined in details in Section 2.3.7. Safety conditions, such as maintaining a minimum distance from DOs, are formulated as chance constraints that must hold at least with probability β, where β is called risk parameter. The probability distribution of the uncertainty is used to derive regions that contain the future position of the DO at least with probability equal to the risk parameter $0 \geq \beta \geq 1$, and constraints are designed to prevent the future EV trajectory from entering such regions. When multiple candidate trajectories for each DO are considered, one chance constraint is designed for each candidate future trajectory of each DO.

For a prediction horizon N, the EV trajectory is planned solving the following optimal control problem with respect to the sequence of future inputs $\boldsymbol{U}_N = [\boldsymbol{u}_0^\top, \ldots, \boldsymbol{u}_{N-1}^\top]^\top$

$$\min_{\boldsymbol{U}_N} J(\boldsymbol{\xi}_0, \boldsymbol{U}_N) \tag{4.21a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k), \qquad \forall k = 0, \ldots, N-1 \tag{4.21b}$$

$$\boldsymbol{\xi}_k \in \mathcal{X}_k, \qquad \forall k = 1, \ldots, N \tag{4.21c}$$

$$\Pr(\boldsymbol{\xi}_k \in \mathcal{S}(\boldsymbol{\xi}_{k,i}^{\mathrm{DO}})) \geq \beta_i, \qquad \forall k = 1, \ldots, N, \forall i = 1, \ldots, n_\mathrm{I}, \tag{4.21d}$$

where for simplicity it is assumed that the current sampling time is $t = 0$. $\boldsymbol{\xi}$ is the state of the EV, $\boldsymbol{f}(\cdot, \cdot)$ represents the discrete-time dynamics of the EV, and $\mathcal{X}_k$ is the constraint set for the EV state at prediction step k, encompassing physical limitations of the EV and traffic rules. Conditions (4.21d) are the safety constraints, requiring the EV to stay in the

set of collision-free states $\mathcal{S}(\boldsymbol{\xi}_{k,i}^{\mathrm{DO}})$ with respect to the i-th candidate trajectory of the DO $\boldsymbol{\xi}_{k,i}^{\mathrm{DO}}$. If multiple DOs are in the proximity of the EV, the approach is repeated for each DO.

In general, the collision avoidance risk parameter β_i can be chosen differently for each candidate trajectory $\boldsymbol{\xi}_{k,i}^{\mathrm{DO}}$ and the choice impacts considerably the planned trajectory of the EV [168]. Both enforcing collision-avoidance only for the most likely trajectory [17] and for all candidate trajectories with equal required risk parameter [93] is not a good trade-off between accounting for several candidate future motions of DOs, without being overly limited by unlikely future trajectories. Even setting the risk parameter β_i for the i-th candidate DO trajectory equal to the estimated probability of that candidate trajectory is not effective if estimated probabilities are not reliable. A systematic approach to choose the risk parameters and generate safety constraints considering the uncertainty in the estimated probability must be obtained, resulting in the following objective.

Objective 4.3. *The goal is to choose the risk parameter β_i for each candidate trajectory depending on the opinion $\boldsymbol{\omega}$ provided by the BFT framework, in order to balance the trade-off between safety and efficiency even in presence of large uncertainty about the probability of candidate future DO trajectories.*

4.4.2. Trajectory Planning considering Uncertainty about the Intention of Dynamic Obstacles

We now present our methods to integrate the BFT opinion for the DO candidate future trajectories in the SMPC scheme. First, we propose a new belief-to-probability transformation, designed not to underestimate the probability of unlikely events if the estimation is highly uncertain. Such probabilities are suitable as risk parameter for SMPC constraints. Then, we discuss a different approach, in which the BFT information is used to tighten constraints, depending on the reliability of the estimation.

Inverse Plausibility Transformation

In this section, we present the *inverse plausibility transformation*, a novel transformation of opinions (2.28) provided by the perception module into probabilities. The goal is to obtain probabilities of candidate DO trajectories suitable as risk parameter for SMPC chance constraints, with an approach similar to that presented in Section 4.3, even when the estimation is highly uncertain. The probabilities obtained are in fact subjective probabilities [169], as in other approaches to convert belief assignments into probabilities [99].

We rely on the concept of plausibility (2.29) with a view at not underestimating the probability of seemingly unlikely trajectories when the estimate is not reliable, i.e., when the uncertainty μ is large. The plausibility of a subset S of the frame of discernment $S \subseteq \Theta$ represents an upper bound to the probability of events contained in the set S itself, and is discussed in more details in Section 2.3.5. The idea behind our novel transformation is to increase the probability of trajectories with respect to their belief mass b_{θ_i}, which represents the amount of evidence that uniquely determines θ_i, if a significant amount of evidence does not exclude θ_i. This is equivalent to say that the plausibility $\mathrm{Pl}(\{\theta_i)\}$ is large. Therefore, we compute the probability of DO candidate trajectories as

$$p_{\theta_i} = b_{\theta_i} + \sum_{\substack{S \in 2^{\Theta} \\ \{\theta_i\} \subset S}} b_S \delta(\theta_i, S), \tag{4.22}$$

where S is any subset of the power set which has the singleton $\{\theta_i\}$ as a proper subset. S represents belief mass that is not uniquely associated to θ_i, but that does not contradict θ_i itself. $\delta(\theta_i, S)$ is a redistribution factor determining the amount of belief mass b_S of set S to be added to b_{θ_i} and is obtained as

$$\delta(\theta_i, S) = \frac{(\mathrm{Pl}(\{\theta_i\}))^{-1}}{\sum_{\theta_j \in S} (\mathrm{Pl}(\{\theta_j\}))^{-1}}. \tag{4.23}$$

The plausibility $\mathrm{Pl}(\{\theta_i\})$ represents an upper bound to the probability of θ_i, computed summing the belief mass of all subsets that contain θ_i, i.e., the belief mass of all evidence not uniquely supporting θ_i, but that does not exclude θ_i. Thus, by redistributing the belief mass of b_S among candidate trajectories $\theta_i \in S$ using the inverse plausibility, we give more belief mass to candidate trajectories θ_i that might incorrectly seem less likely.

For example, in opinion $\boldsymbol{\omega} = [b_{\theta_1}, b_{\theta_2}, \mu] = [0.4, 0.1, 0.5]$, the relative ratio between the beliefs is 4:1. However, depending on how the unspecified belief mass μ is allocated, $\boldsymbol{\omega}$ could correspond to probabilities $p_{\theta_1} = p_{\theta_2} = 0.5$, that is, an equal probability between the two trajectories. Therefore, it is important not to rely too much on the individual beliefs when the uncertainty is large. In this basic example, the probabilities obtained from the inverse plausibility transformation are $p_{\theta_1} = 0.6, p_{\theta_2} = 0.4$: θ_1 is recognized as dominant, but the probability of θ_2 is increased more than proportionally with respect to the beliefs $b_{\theta_1}, b_{\theta_2}$, because the uncertainty $\mu = 0.5$ is large. By contrast, deducing the probability of candidate trajectories from the relative ratio of beliefs and normalizing to one, i.e.,

$$p_i = \frac{b_{\theta_i}}{\sum_{j=1}^{n_\mathrm{I}} b_{\theta_j}}, \ \forall i = 1, \ldots, n_\mathrm{I}, \tag{4.24}$$

is not advisable, since the relative ratio between the probability of the trajectories could be significantly different depending on how the still unspecified belief mass μ is allocated. In the considered example, the latter approach yields probabilities $p_1 = 0.8, p_2 = 0.2$ and the probability of θ_2 to occur is underestimated, potentially with dangerous consequences.

The belief masses of singletons b_{θ_i} are not redistributed in (4.22). Hence, the redistribution of belief mass plays a major role only if the estimation is not reliable, that is, if the uncertainty μ is large. If the estimation is reliable and most of the belief mass is allocated to singletons, the probabilities obtained from (4.22) reflect the ratio between beliefs.

The inverse plausibility transformation (4.22) preserves the bounds coherent with the basic BFT definitions given in Section 2.3.5.

Theorem 4.1. *The inverse plausibility transformation* (4.22) *satisfies the upper-lower-boundary consistency [99], i.e.,*

$$b_{\theta_i} \leq p_{\theta_i} \leq \mathrm{Pl}(\{\theta_i\}), \ \forall \theta_i \in \Theta. \tag{4.25}$$

Proof. Since $\mathrm{Pl}(\{\theta_i\}) \geq 0 \ \forall \theta_i \in \Theta$, then $\delta(\theta_i, S) \geq 0 \ \forall \theta_i, \forall S$. Therefore, since $b_S \geq 0$ $\forall S \in 2^\Theta$, it holds that $p_{\theta_i} = b_{\theta_i} + \sum_{\substack{S \in 2^\Theta \\ \{\theta_i\} \subset S}} b_S \delta(\theta_i, S) \geq b_{\theta_i}$. From $\delta(\theta_i, S) \leq 1$, it is obtained that $p_{\theta_i} \leq b_{\theta_i} + \sum_{\substack{S \in 2^\Theta \\ \{\theta_i\} \subset S}} b_S = \mathrm{Pl}(\{\theta_i\})$. □

The inverse plausibility transformation (4.22) is designed to consider the indefiniteness of the estimation in presence of large uncertainty. The decision making relying on probabilities

so obtained will tend not to be overconfident, and rather to account for all outcomes that are not ruled out by the evidence collected. This approach is opposite to the transformation proposed in [100], in which probabilities are generated with a view at boosting for confidence in the decision making. In autonomous driving, such approach would lead to risky behaviors of the EV. By contrast, the probabilities obtained from (4.22) are designed to be robust against unlikely but not excluded events if the uncertainty is large, so that the EV can still promptly react to such DO behaviors.

Probabilities obtained from the inverse plausibility transformation (4.22) are well suited as risk parameters in the SMPC collision-avoidance constraints, as in Section 4.3, even in presence of unclear motions of DOs. If the estimated probabilities changed repeatedly and significantly because of large uncertainty in the estimation, the SMPC collision-avoidance constraints would also substantially differ between consecutive iterations. Thus, the planning would be frequently updated in possibly contradictory ways, severely reducing the benefit of considering several candidate DO trajectories. By contrast, in these cases the BFT estimation will deliver a large uncertainty and beliefs that do not vary frequently, as discussed in Section 3.2. Consequently, the probabilities obtained from (4.22) and, ultimately, the safety constraints, will also not vary too suddenly. Rather, they are tightened when the information gathered does not allow to confidently recognize the intended behavior of the DO, and thus multiple future trajectories must be considered in the EV planning.

Constraint Tightening

Here, we propose a different approach to leverage BFT in SMPC, consisting of constraint tightening. Rather than obtaining probabilities from BFT opinion as with the inverse plausibility transformation, the belief of singletons are directly used as risk parameter in the SMPC constraints (4.21d) $\beta_i = b_{\theta_i}$. Then, constraints are tightened using the additional information provided by BFT.

Consider, as an example, a deterministic SMPC reformulation resulting in a quadratic distance-based collision-avoidance safety constraint for each i-th candidate DO trajectory

$$\left(\boldsymbol{\xi}_k - \boldsymbol{\xi}_{k,i}^{\mathrm{DO}}\right)^\top \boldsymbol{\Lambda} \left(\boldsymbol{\xi}_k - \boldsymbol{\xi}_{k,i}^{\mathrm{DO}}\right) \geq 1, \tag{4.26}$$

representing an elliptical region around the predicted position $\boldsymbol{\xi}_{k,i}^{\mathrm{DO}}$ of the DO at prediction step k, that the EV state $\boldsymbol{\xi}_k$ must not enter. $\boldsymbol{\Lambda} > 0$ is a weighting matrix determining the size of the ellipse and it depends on the physical dimension of the EV and of the DO, on the uncertainty about the DO state prediction for the i-th candidate DO trajectory, represented by covariance $\boldsymbol{\Sigma}_{k,i}^{\mathrm{DO}}$, and on the risk parameter β_i [30].

We tighten constraint (4.26) considering the reliability of the estimates. Matrix coefficient $\boldsymbol{\Lambda}$, governing the degree of conservatism of the constraint, is scaled as $\boldsymbol{\Lambda}' = \gamma^\lambda \boldsymbol{\Lambda}$, where

$$\lambda = \mathrm{sgn}(\mathrm{Pl}(\{\theta_i\}) - \alpha) \left(\frac{\mu}{\mathrm{Pl}(\{\theta_i\})}\right)^{\mathrm{sgn}(\mathrm{Pl}(\{\theta_i\})-\alpha)}, \tag{4.27}$$

where $\mathrm{sgn}(\cdot)$ is the sign function. $0 < \gamma < 1$ is a tuning parameter that defines the minimum tightening, and $0 < \alpha < 1$ is a threshold on the probability level used to discern which trajectories must be neglected. When the plausibility of a trajectory $\mathrm{Pl}(\{\theta_i\})$ exceeds the

threshold α, the weighing matrix $\mathbf{\Lambda}$ is scaled depending on the ratio between the overall uncertainty μ and the plausibility of the trajectory $\mathrm{Pl}(\{\theta_i\})$

$$\gamma^{\lambda} = \gamma^{\frac{\mu}{\mathrm{Pl}(\{\theta_i\})}}, \tag{4.28}$$

where, since $\mu \leq \mathrm{Pl}(\{\theta_i\})$, it holds $0 \leq \lambda \leq 1$ and, since $0 < \gamma < 1$, we have $0 < \gamma^{\lambda} \leq \gamma \leq 1$, i.e., the weighting matrix $\mathbf{\Lambda}$ is always reduced in norm and constraint (4.26) is strictly tightened. The larger the completely unspecified information, μ, with respect to the evidence non-contradicting θ_i, $\mathrm{Pl}(\{\theta_i\})$, the stronger the tightening, and the scaling factor is upper bounded by γ. There is no tightening for $\mu = 0$, as the information is completely certain.

Conversely, if the plausibility of the i-th trajectory $\mathrm{Pl}(\{\theta_i\})$ is smaller than the threshold α, i.e., the trajectory is to be neglected, matrix $\mathbf{\Lambda}$ is scaled by factor

$$\gamma^{\lambda} = \gamma^{-\left(\frac{\mu}{\mathrm{Pl}(\{\theta_i\})}\right)^{-1}} = \gamma^{-\frac{\mathrm{Pl}(\{\theta_i\})}{\mu}}, \tag{4.29}$$

where, since $\mu \leq \mathrm{Pl}(\{\theta_i\})$, it holds $\lambda \leq -1$ and, since $0 < \gamma < 1$, we have $1 \leq \gamma^{-1} \leq \gamma^{\lambda}$, i.e., the weighting matrix $\mathbf{\Lambda}$ is strictly enlarged in norm and constraint (4.26) strictly relaxed, where the minimum relaxing factor is γ^{-1}. The constraint is relaxed more for smaller uncertainty μ, since in this case the estimate is reliable and outcomes with small plausibility $\mathrm{Pl}(\{\theta_i\}) < \alpha$ can be safely ruled out. For $\mu \to 0$, $\mathbf{\Lambda}'$ grows indefinitely (in norm) and the constraint degenerates and is always satisfied, that is, the i-th trajectory is completely ignored in the planning of the EV. Thus, the feasible set of the constraint is reduced and the motion of the EV becomes more conservative only if the information is highly uncertain. Similar considerations can be adopted to tighten other forms of constraints depending on their geometrical interpretation.

4.4.3. Simulations of Trajectory Planning of Autonomous Vehicles considering Uncertainty about the Intention of Dynamic Obstacles

We compare our approach with existing methods in two numerical simulations in Matlab. We consider a highway scenario and an urban intersection, in which the EV must interact with DOs, whose future behavior is unclear. Not considering the reliability of the estimate provided by BFT results in dangerous situations or inefficient EV behavior.

The EV state $\boldsymbol{\xi} = [x, v_x, y, v_y]^{\top}$ consists of the longitudinal and lateral positions and velocities and the input $\boldsymbol{u} = [a_x, a_y]^{\top}$ of the longitudinal and lateral accelerations. A double integrator system with sampling time $T = 0.2$ s is used for the dynamics of the EV and the SMPC optimal control problem (4.21) is solved using the NMPC toolbox [166]. A precise description of the EV dynamics is beyond the scope of the analysis presented here. However, more sophisticated models can be straightforwardly included in the framework. Collision avoidance constraints are in the form (4.26), that is, consist of ellipsoidal regions around the predicted positions of DOs, that the EV must not enter, as in Section 4.3, relying on nominal future DO trajectories that are assumed given. The actual behavior of DOs and the belief assignments are corrupted by significant noise to simulate challenging scenarios.

Cost function (4.21a) of the SMPC optimal control problem is

$$J = \|\Delta\boldsymbol{\xi}_N\|_{\boldsymbol{P}}^2 + \sum_{k=0}^{N-1} \|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_k\|_{\boldsymbol{R}}^2 + \|\boldsymbol{u}_k - \boldsymbol{u}_{k-1}\|_{\boldsymbol{S}}^2, \tag{4.30}$$

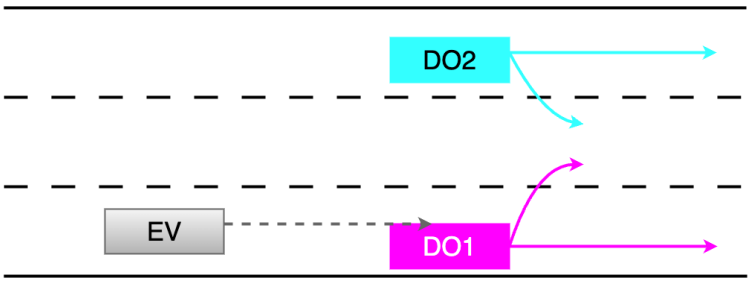


Figure 4.21.: Candidate trajectories for the highway scenario. © 2024 IEEE.

where $N = 8$ is the prediction horizon, $\boldsymbol{Q} = \boldsymbol{P} = \text{diag}(0, 1, 1, 1)$, $\boldsymbol{R} = \text{diag}(0.1, 0.1)$, $\boldsymbol{S} = \text{diag}(0.1, 10)$, and $\boldsymbol{u}_{-1}$ is set equal to the last applied input. The first entry of $\boldsymbol{Q}$ and $\boldsymbol{P}$ is set to zero because the EV is assigned only a longitudinal reference velocity, rather than reference positions. $\Delta\boldsymbol{\xi} = \boldsymbol{\xi} - \boldsymbol{\xi}_{\text{ref}}$, with $\boldsymbol{\xi}_{\text{ref}} = [0, v_{\text{ref}}, y_{\text{lane}}, 0]^\top$, where v_{ref} is the reference longitudinal velocity and y_{lane} is the center of the current lane. Cost function (4.30) penalizes deviations from the center of the lane and from the target speed, and penalizes large accelerations and rapid changes in the accelerations. In both scenarios the EV is subject to lateral position constraints, longitudinal velocity constraints, and lateral velocity constraints in the highway scenario.

We compare the performance of our approaches with implementations of [17], in which only the most likely future DO trajectory is taken into account in the planning, [93], in which all candidate future trajectories are considered with the same risk parameter, and our approach from Section 4.3, in which the risk parameter for each trajectory is the estimated probability of the trajectory itself. The probabilities used by non-BFT-based methods from [17], [93], and Section 4.3, are obtained from the belief assignments re-scaling the belief masses of singletons preserving the relative ratios as in (4.24), i.e., neglecting the uncertainty μ. For [17] and [93], the risk parameter is $\beta = 0.85$. As comparison metrics, we use the cumulative sum of the SMPC stage cost over the full simulation

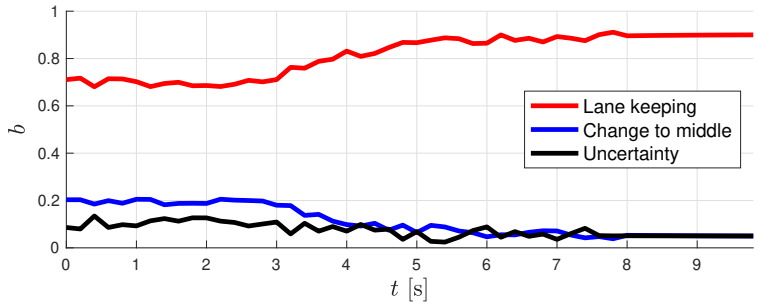
$$J_{\text{sim}} = \sum_{t=1}^{N_{\text{sim}}} \|\Delta\boldsymbol{\xi}_t\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}_t\|_{\boldsymbol{R}}^2 + \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|_{\boldsymbol{S}}^2, \tag{4.31}$$

where all numerical values are as in the SMPC cost (4.30).

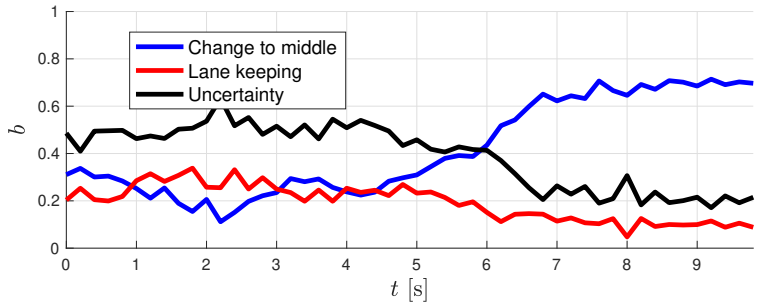
Highway Scenario

In the first scenario, the EV is initially located on the right-most lane of a 3-lane highway and is approaching a slower vehicle, DO1, on the same lane, whereas another vehicle, DO2, is initially located on the left-most lane. For both vehicles we consider two candidate trajectories, namely continuing on the same lane and changing to the middle lane, as in Figure 4.21. The belief estimation for the two vehicles is represented in Figure 4.22. At first, the intention of DO2 is not clear, thus the uncertainty is large. Shortly after $t = 6$ s, it is clear that DO2 has changed to the middle lane and will remain there, thus the belief assignment is gradually updated giving higher confidence to that maneuver, whereas DO1 shows a more regular behavior and it is assumed to stay on the right lane.

Figure 4.23 shows the trajectory of the EV in the traffic scene for the considered methods, in which different approaches to the generation of safety constraints are implemented. The



(a) DO1. © 2024 IEEE.



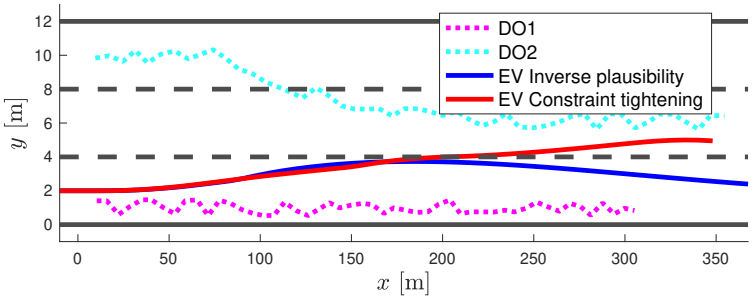
(b) DO2. © 2024 IEEE.

Figure 4.22.: Belief assignment for the candidate DO trajectories.

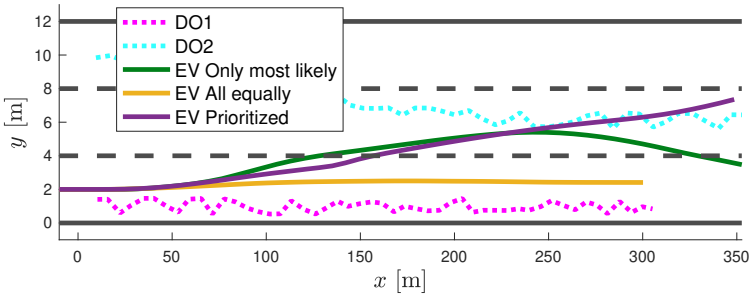
inverse plausibility method yields probabilities designed not to underestimate the probability of trajectories if the uncertainty is high: at first, the EV accounts for the fact that DO2 might move to the center lane and proceeds cautiously while moving to the left. When DO2 actually moves to the center lane, the EV moves back to the right lane, after overtaking DO1. Similarly, the tightened constraints induce a cautious behavior of the EV, so that a sufficient safety distance can be maintained when DO2 moves to the center lane. Yet, in this simulation, the EV does not move back to the right lane at the end, rather remains in an intermediate position between the right and the center lanes, which, although safe, can be undesirable in practical situations.

The other approaches do not account for the uncertainty about the estimation and this is reflected in the behavior of the EV. Constraints generated based only on the most likely future trajectory of the DOs [17] result in an overconfident behavior of the EV, assuming that both DOs will remain in the current lane. As a result, when DO2 moves to the center lane, the EV is too close and the safety distance is violated. Eventually the EV reacts by quickly moving back to the right lane. The safety constraints generated for each candidate trajectory of DOs as in [93] result in a conservative behavior of the EV, which never rules out any candidate future DO trajectory, even when the behavior becomes clear and the probability of some candidate trajectories becomes negligible. Therefore, the EV never overtakes DO1, as it is considered possible at all times that this vehicle will suddenly move to the center lane. Finally, constraints generated for candidate future DO trajectory depending on the trajectory probability as in Section 4.3 produce inconsistent behaviors when the estimated probabilities vary frequently. At the end of the simulation, the EV is forced to move to the left to avoid collisions.

The value of the stage cost (4.31) reflects the qualitative comparison between the approaches. The constraints generated with the inverse plausibility method and with the



(a) Novel methods. © 2024 IEEE.



(b) Comparison methods. © 2024 IEEE.

Figure 4.23.: DO (dashed) and EV (solid) trajectories resulting from different collision-avoidance constraints.

constraint tightening yield cost $J_{\text{sim}} = 617$ and $J_{\text{sim}} = 1722$, respectively. The method from [93] yields a significantly higher cost $J_{\text{sim}} = 3883$, since the velocity of the EV is considerably lower. The cost of the methods from [17] and of the method from Section 4.3 is not informative, as they result in dangerous situations.

Urban Intersection

In the second scenario we consider three DOs, whose movements are less structured and therefore more uncertain. We analyze more in details how accounting for the reliability of the BFT estimation is beneficial also in terms of performance.

The EV is located to the left of an urban intersection and must proceed straight, safely interacting with two cyclists, DO1 and DO2, and a pedestrian, DO3, as in Figure 4.24. The pedestrian is initially located to the right of the EV and could cross either the horizontal or the vertical road, potentially crossing the EV intended path. The two cyclists reach the intersection on the vertical road and could continue straight, turn right, or continue straight on the bike lane on the other side of the road.

The behavior of all DOs is initially unclear, resulting in large uncertainty, Figure 4.26. At first, the EV slows down, considering that the pedestrian and the cyclist from above might cross the road, as eventually happens. Then, it proceeds straight and moves to the left to overtake the remaining cyclist, which clearly shows the intention of proceeding straight after the right turn.

The difference between the considered approaches is visible in the resulting longitudinal velocity of the EV, Figure 4.27, and the resulting cumulative cost J_{sim}, Figure 4.25. The constraints generated considering the reliability of the estimation from BFT allow the EV to move safely in the uncertain environment without excessively decelerating when the

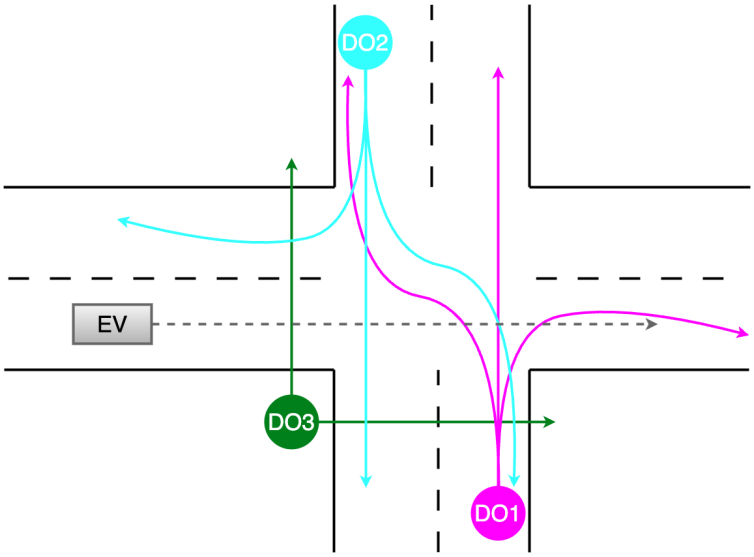


Figure 4.24.: Candidate trajectories for the urban intersection scenario. DO1 and DO2 could turn right, continue straight or take the bike lane on the other side, DO3 could cross vertically or horizontally. © 2024 IEEE.

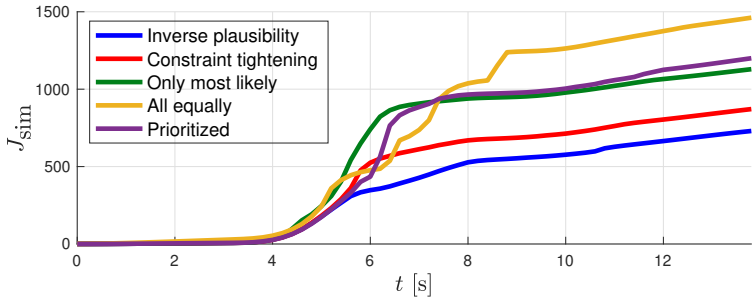
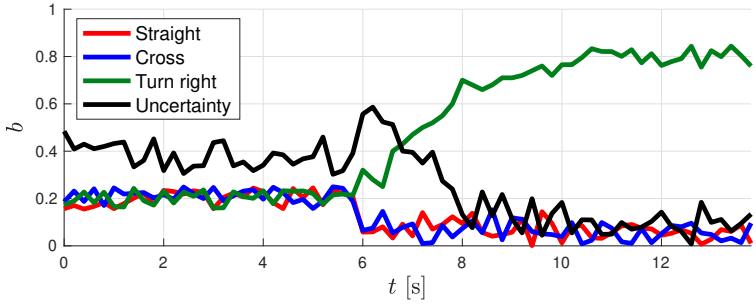
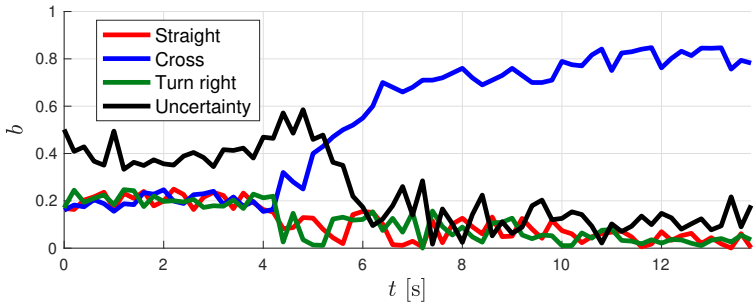


Figure 4.25.: Cumulative cost over time. © 2024 IEEE.

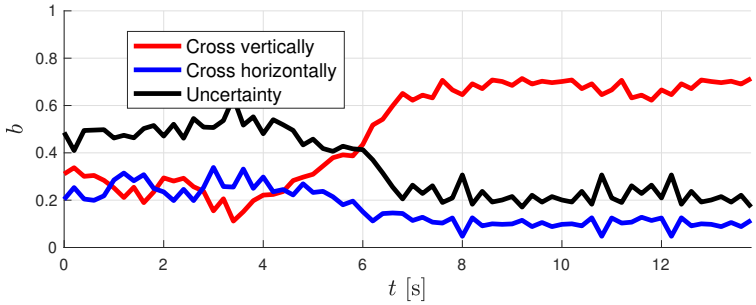
pedestrian is reached (first minimum in the velocity). By contrast, although no dangerous situation is encountered in this scenario, all comparison methods yield a higher cost, because the reliability of the estimation is not considered. The methods from [17] and from Section 4.3 result in higher fluctuations in the EV velocity, because the probabilities used are not reliable during the first part of the simulation, therefore the predictions used in the EV planning change repeatedly, resulting in incoherent behaviors. The method from [17], in particular, produces an overconfident behavior of the EV, which must suddenly come to a full stop when the assumption proves wrong and the pedestrian does not behave according to the most likely trajectory. The method from [93] is especially inefficient, because none of the candidate trajectories of DOs is ever ruled out even when it is safe to do so, and the EV eventually does not pass the cyclist DO1.



(a) DO1 (cyclist). © 2024 IEEE.

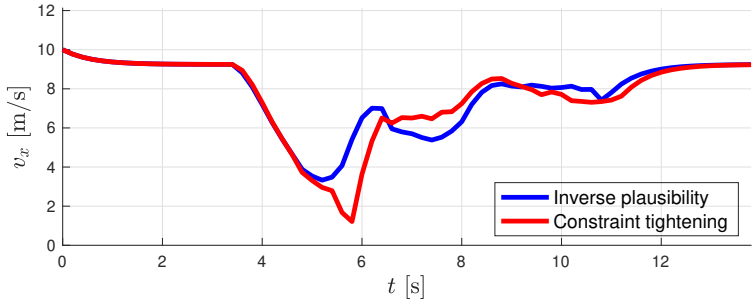


(b) DO2 (cyclist). © 2024 IEEE.

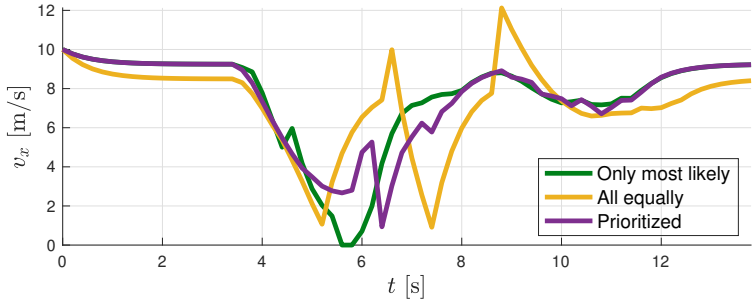


(c) DO3 (pedestrian). © 2024 IEEE.

Figure 4.26.: Belief assignment for the candidate DO trajectories.



(a) Novel methods. © 2024 IEEE.



(b) Comparison methods. © 2024 IEEE.

Figure 4.27.: EV longitudinal speed resulting from the different methods.

4.5. Safe SMPC Planning reacting to Unmodeled Uncertainty

In this last section, we propose a framework to handle the non-zero probability of constraint violation in SMPC, while still yielding an efficient EV motion. Furthermore, we address the unmodeled uncertainty, that is, we propose a strategy to react to scenarios in which the uncertainty violates the assumed bounds, proposing a novel combination of SMPC and CVPM. This section is based on the work presented in [23].

4.5.1. Problem Formulation

We consider again the linearized and discretized kinematic bicycle model in road-aligned coordinates from Section 2.3.3 as EV model. The state of the EV $\boldsymbol{\xi} = [s, d, \phi, v]^\top$ consists of the longitudinal and lateral positions of the center of mass of the EV in the road-aligned reference frame, s and d, respectively, of the yaw angle ϕ, and of the linear velocity v. The input of the EV $\boldsymbol{u} = [a, \delta]^\top$ consists of the acceleration a and of the front steering angle δ. Further, we consider constraints (2.16) on the vehicle's lateral position, velocity, acceleration and steering, as discussed in Section 2.3.3. Such model couples the longitudinal and lateral dynamics of the vehicle and allows efficient computation. Different, more accurate models can also be used without compromising the validity of the presented approach. However, the considerations on the computational complexity given in the following hold true only for linear models. Similarly, uncertainty about the measurements of the EV state might be considered and accounted for in the derivation of the safety collision-avoidance constraints without significant changes in the algorithm.

For the assumed dynamics of the DOs, we adopt a double integrator model with decoupled dynamics for the longitudinal and lateral components. We consider the models presented in Section 2.3.4, but with additional constraints referring to the worst-case movement of DOs. For ease of notation, in the following we refer to one DO only. However, all considerations hold similarly for multiple DOs close enough to the EV. The state of each DO is $\boldsymbol{\xi}^{\mathrm{DO}} = \left[s^{\mathrm{DO}}, v_s^{\mathrm{DO}}, d^{\mathrm{DO}}, v_d^{\mathrm{DO}}\right]^\top$, consisting of longitudinal and lateral positions and velocities, and the input $\boldsymbol{u}^{\mathrm{DO}} = \left[a_s^{\mathrm{DO}}, a_d^{\mathrm{DO}}\right]^\top$ comprises the longitudinal and lateral acceleration, a_s^{DO} and a_d^{DO}, respectively. We model the action of DO drivers as an LQR-based controller tracking a constant reference state that expresses the intention of the driver, for example maintaining a given cruise speed or a lateral target position. Also for DOs more sophisticated prediction models can be used without compromising the validity of the approach.

Here, we further consider the following input bounds for the DO, representing the worst-case behavior

$$\boldsymbol{u}_{\min}^{\mathrm{DO}} \leq \boldsymbol{u}^{\mathrm{DO}} \leq \boldsymbol{u}_{\max}^{\mathrm{DO}}. \tag{4.32}$$

Therefore, the uncertainty about the input for the nominal trajectory is modeled as *truncated* Gaussian, $\boldsymbol{w}^{\mathrm{DO}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}^{\boldsymbol{w},\mathrm{DO}}, \mathcal{W}^{\mathrm{DO}}\right)$. Moreover, we assume that measurements of the DO states available to the EV are corrupted by the measurement noise $\boldsymbol{v}^{\mathrm{DO}}$, which is a zero-mean truncated Gaussian distribution $\boldsymbol{v}^{\mathrm{DO}} \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}^{v,\mathrm{DO}}, \mathcal{V}^{\mathrm{DO}}\right)$. In the approach presented here, stochastic and robust control approaches are combined, therefore the assumption of bounded support for the distributions is later relaxed and unbounded support is assumed for the stochastic control approaches.

Definition 3. *An unanticipated behavior of DOs is any trajectory that is not in the reachable space of model* (2.25) *subject to constraints* (4.32)*, that is, trajectories that cannot be obtained from the dynamics for any input satisfying* (4.32)*.*

Unanticipated behaviors represent both future trajectories that are not feasible given the assumed physical limitations of the DOs and future trajectories that violate traffic rules.

Remark 4.6. *In practice, the actual DO control policy and intended behavior and are not known to the EV and therefore the assumptions could be violated. Thus, the bounds on the DO input* (4.32) *and the Gaussian probability distribution for the stochastic component of the DO input are to be intended as the assumption of the EV on the DO model, on what is considered an anticipated behavior.*

Under the following assumptions on the motion of DOs, it is possible to take advantage of the optimistic SMPC planning, while still verifying that this will not lead to collisions. However, it is important to react to possible violations of the assumptions on the motions of DOs, obtaining the following objective.

Objective 4.4. *The goal is to provide an efficient trajectory for the EV using SMPC, avoiding collisions with DOs when possible, i.e. when they behave as expected, and otherwise to respond systematically to unanticipated behavior of DOs by minimizing the probability of collision while keeping the computational complexity manageable.*

In the following, the SMPC and the CVPM schemes presented in Section 2.3.7 are used. For the SMPC scheme, the admissible region for the EV consists of all positions outside of the safety ellipse of each DO. However, this admissible region is non-convex. Therefore, here we approximate the constraints with linear constraints, preventing the EV from entering the safety areas. The full procedure is outlined in [30, Section VI-A]. If a high-level maneuver planner module as in Section 4.2 is used, the linear constraints should be designed to be consistent with the selected maneuver, for example, so that the EV avoids the ellipse from the appropriate side. However, here we do not assume the presence of a high-level maneuver planner and rather generate safety constraints according to the procedure in Appendix A.1. DOs are considered one at a time and the relative traffic configuration is evaluated. Then, a linear constraint is generated, preventing the EV from entering the forbidden area around the DO and implicitly suggesting the appropriate maneuver to be taken. Eventually, considering all individually-generated linear safety constraints, the set of admissible positions for the EV is convex. Thus, the deterministic optimal control problem replacing (2.37) is a QP problem and is solved efficiently at run time.

The SMPC and the CVPM schemes differ only in the formulation of safety constraints, while they consider the same cost function. For simplicity, we refer to $\boldsymbol{x}$ as the current state of the environment, considering the current states of the EV and of the DO

$$\boldsymbol{x} = \left[\boldsymbol{\xi}^\top, \boldsymbol{\xi}^{\mathrm{DO}\top}\right]^\top. \tag{4.33}$$

Therefore, with the notation SMPC($\boldsymbol{x}$) and CVPM($\boldsymbol{x}$) we refer to problems SMPC and CVPM, respectively, evaluated for the same current traffic configurations, i.e., only differing in the formulation of the safety constraints.

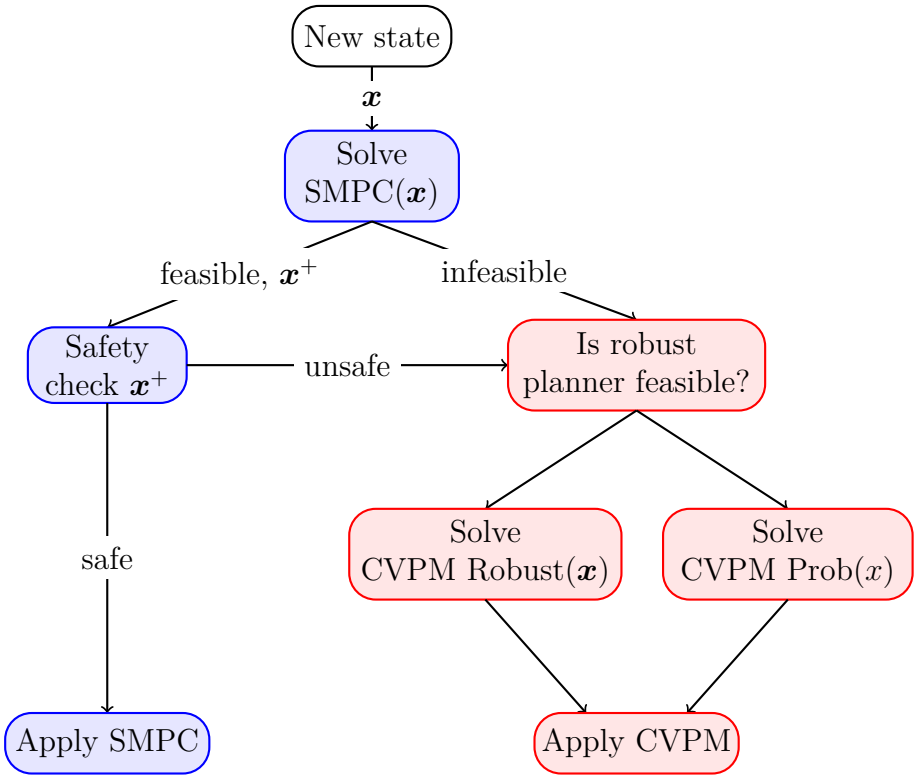


Figure 4.28.: Diagram of the logic of the SMPC+CVPM scheme. © 2024 IEEE.

4.5.2. Novel SMPC+CVPM Scheme

The SMPC+CVPM scheme is now outlined and its properties discussed. The scheme is designed to leverage the optimistic and efficient handling of uncertainty of SMPC. However, this comes at the price of a non-zero risk of collision. Thus, we use CVPM to evaluate the safety of the SMPC trajectory before applying it to the system. Furthermore, CVPM is used to react to situations in which the real behavior of DOs violates the assumptions of the EV, and therefore there might not exist trajectory that are safe for any realization of the uncertainty. Then, CVPM returns a trajectory minimizing the probability of a collision.

The logic to decide whether SMPC or CVPM must be applied is illustrated in Figure 4.28. At each time step, at first, the SMPC($\boldsymbol{x}$) optimal control problem is solved, that is, an optimistic trajectory based on the current traffic configuration $\boldsymbol{x}$ is computed. If a solution to SMPC($\boldsymbol{x}$) exists, the predicted next step is considered, that is, the traffic configuration at the next sampling time, $\boldsymbol{x}^+$ resulting from the first input of the SMPC($\boldsymbol{x}$) solution. From $\boldsymbol{x}^+$, a safety check is performed considering the prediction horizon from prediction step $k = 1$ to $k = N + 1$. The goal of the safety check is to verify whether after applying the solution of SMPC($\boldsymbol{x}$), at the next time step it is still possible to determine a safe trajectory accounting for the worst case movement of the DOs, following a robust approach. Precisely, the existence of a feasible solution to optimal control problem CVPM Robust($\boldsymbol{x}^+$) (2.40) is evaluated. However, a solution to CVPM Robust($\boldsymbol{x}^+$) must not be determined at present, in fact optimal control problem (2.40) is not solved; rather, only the existence of a solution is discussed by performing a set computation and therefore the procedure is computationally undemanding. If the safety check succeeds, then SMPC will lead to a safe state from which a feasible robust solution can be obtained at the next sampling time.

Definition 4. *The solution to the SMPC($\boldsymbol{x}$) optimal control problem is safe if the safety check for $\boldsymbol{x}^+$ succeeds, that is, if it is certified that applying the SMPC solution leads to a state $\boldsymbol{x}^+$ from which a collision free trajectory can be planned, if DOs behave consistently with the assumptions in Section 4.5.1.*

Remark 4.7. *The safety check in particular verifies the safety of the initial condition of the following optimal control problem. Therefore, if the first predicted state of the SMPC trajectory does not satisfy collision avoidance constraints in a robust sense, the safety check fails as well. Rigorously speaking, the safety check succeeds if and only if $\mathcal{U}_{robust} \neq \emptyset$ for $\boldsymbol{x}^+$, with $\mathcal{U}_{robust}$ from* (2.43)*, but also including initial condition in the state sequence $\boldsymbol{\Xi}_N$.*

In such case, the first input of the solution to SMPC($\boldsymbol{x}$) is applied, as the optimistic handling of the uncertainty does not lead to states from which collisions cannot be avoided.

Remark 4.8. *The predicted traffic configuration $\boldsymbol{x}^+$ used for the safety check is not the prediction used in the SMPC scheme, based on the approximated EV model* (2.15)*. Rather, a nonlinear and continuous-time model is used to compute a more accurate prediction of the future state of the EV if the SMPC input is applied. Since the safety check is not part of the SMPC optimal control problem, the predicted traffic configuration $\boldsymbol{x}^+$ is computed only once at each iteration of the whole SMPC+CVPM scheme, after that the SMPC optimal control problem has been solved. Thus, a continuous-time nonlinear model does not severely affect the computation time of the scheme. Nevertheless, the prediction models of DOs must be consisted with the assumptions in Section 4.5.1.*

Since the SMPC($\boldsymbol{x}$) is not recursively feasible, sometimes it does not yield a solution. Furthermore, even when a solution exists, this might not be safe, that is, applying the first input prevents the existence of a solution to the robust planner for the first predicted step. In either of these situations, the optimal control problem CVPM($\boldsymbol{x}$) is solved, as shown in the right branch in Figure 4.28. First, we check which case of the CVPM algorithm must be used. If possible, the robust case is solved, producing a trajectory that is collision-free for any realization of the uncertainty within the assumptions from Section 4.5.1. If this is not possible, the probabilistic case is used to obtain a trajectory which minimizes the probability of collision. As we discuss in Section 4.5.4, the cost function minimized in the probabilistic case might differ from the probability of collision, which is considered here for simplicity. For example, the probabilistic case of CVPM could consider a cost function that accounts for both the probability of a collision and the anticipated harm caused by the impact. In either case, the aim is to plan the EV trajectory by explicitly reacting to the dangerous situation caused by the unanticipated movement of DOs.

Remark 4.9. *If SMPC($\boldsymbol{x}$) does not yield a solution, it is unlikely that the robust case of CVPM($\boldsymbol{x}$) is applicable, since the latter is a robust planner with tighter constraints. Still, because of minor differences in the formulation of the safety constraints, this cannot be completely ruled out for rare cases. Nevertheless, in practice, the robust case of CVPM($\boldsymbol{x}$) is mainly used to plan a safe trajectory when the solution to SMPC($\boldsymbol{x}$) exists but is not safe, i.e., if it fails the safety check.*

The presented SMPC+CVPM scheme allows to benefit from the efficient and optimistic planning of SMPC when it is safe to do so, that is, when a safe trajectory can be obtained for the predicted traffic configuration $\boldsymbol{x}^+$. Yet, the CVPM robust case is used to compute

a safe trajectory if the SMPC solution is not safe. Therefore, if DOs move as expected according to the assumed models in Section 4.5.1, the SMPC+CVPM scheme yields always a collision-free trajectory.

Moreover, compared to the previous SMPC+FTP scheme [30], which is recalled in Appendix B.2 for convenience, the SMPC+CVPM scheme addresses situations where the EV must deal with unforeseen behaviors of DOs, whose motion violates the hypotheses on which the robust motion planner is built. This might happen, for example, due to unanticipated behaviors of traffic participants as per Definition 3. If the actual motion of the DOs violates the worst-case hypotheses on the disturbance distributions considered in the robust approach, the previously established safety guarantees do not longer hold. In such cases, the EV re-plans a new safe trajectory if possible; if not possible, the EV aims at minimizing the probability of a collision due to the possibly unlawful behavior of other traffic participants. Observe that the if DOs behave as expected, the EV returns a collision-free trajectory, thus possible collisions are due to the DO unanticipated behaviors, not caused by the EV.

Remark 4.10. *The robust problem of the robust case of CVPM can be modified to include a terminal constraint that guarantees recursive feasibility, as in previous work [30]. Such terminal constraint can lead to considerable conservatism in the solution and the guarantee of recursive feasibility does not hold if the actual DO motion violates the worst-case assumptions. Therefore, we do not include it and rather rely on the probabilistic case of CVPM to address the situations of loss of feasibility.*

The logical structure of the SMPC+CVPM scheme is computationally efficient and can be parallelized. In fact, the scheme from Figure 4.28 is logically equivalent to the representation depicted in Figure 4.29, in which the blocks have been rearranged based on their dependencies, rather than on the control logic. Two independent branches that are solved in parallel (blue and red branch, respectively) are recognizable. This reflects the fact that the CVPM branch (in red in Figure 4.28) is independent of the solution of the SMPC branch, therefore can be computed in parallel and only applied if necessary. Moreover, in Figure 4.29 rectangular boxes are used to represent the numerical computation of a solution to an optimal control problem, and rounded boxes indicate a set computation procedure. In particular, the safety check for $\boldsymbol{x}^+$, which necessarily follows the SMPC optimal control problem, only consists of a set computation, since no safety backup must be computed or stored. In fact, only the existence of a solution must be evaluated. As a result, the solution of two cascaded optimization problems is never required to obtain the solution. This is reflected in Figure 4.29: all alternative logical paths leading from the measurement of the state to the application of the control consist at most of one set computation and of the solution of one optimal control problem (possibly in reverse order).

4.5.3. Simulations of Trajectory Planning for Autonomous Vehicles with Unanticipated Behaviors of Dynamic Obstacles

We validate the SMPC+CVPM scheme through numerical simulations from the CommonRoad database [170]. CommonRoad provides a rich collection of scenarios meant to be used as benchmarks to compare different motion planning algorithms in challenging traffic situations, giving the traffic environment as well as the trajectory of DOs for each simulation

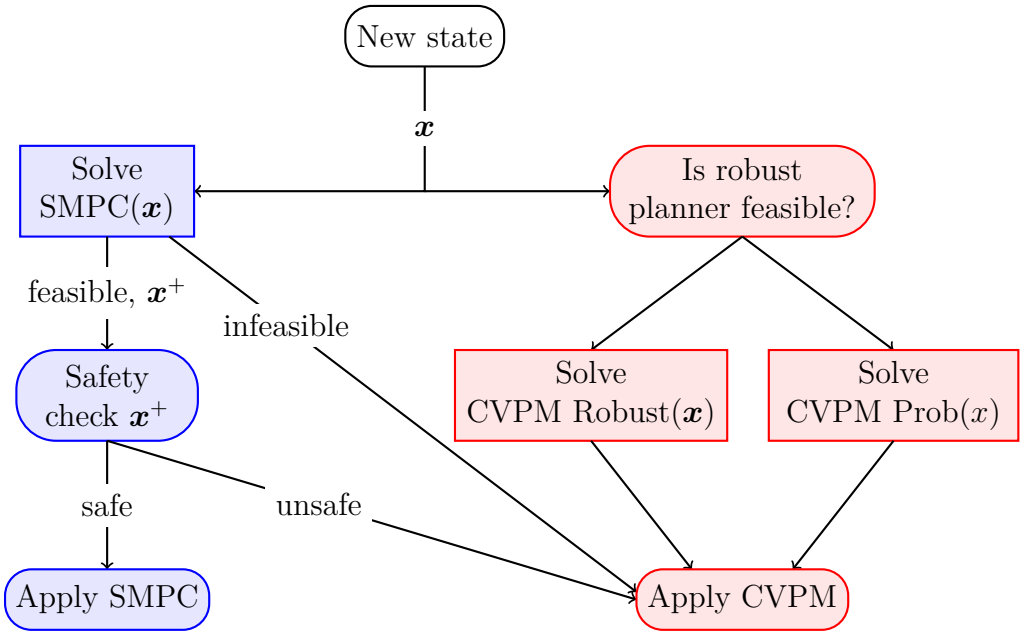


Figure 4.29.: Computational diagram of the procedure of the SMPC+CVPM scheme. Rectangular boxes represent the numerical computation of a solution to an optimal control problem, rounded boxes a set computation. © 2024 IEEE.

instant. We compare the performance of our novel method to that of methods considering a pre-stored fail-safe trajectory as emergency backup plan [30], [171]. Specifically, we tested the SMPC+FTP scheme from [30], recalled in Appendix B.2, for the same traffic scenarios.

Although the controllers use a linear prediction model (2.15), the next states of the EV in the simulations are computed numerically integrating the nonlinear continuous-time kinematic bicycle model. Moreover, the covariance matrix of the probability distribution of the predicted states of the DOs is approximated by a diagonal matrix, i.e., ignoring dependencies between time steps. Both the set computations and the solution to all optimal control problems are computed using the software *CVXPY* [172]. We used the same sampling time of the trajectories given by commonroad, that is, the EV trajectory is replanned every $T = 0.1$ s. Simulations were run on a computer with an AMD Ryzen 5 3500U eight-core processor.

In the following, we present the simulations of two challenging scenarios, that were selected because they consider trajectories of the DOs that trigger a reaction of the EV suitable to analyze the properties of our algorithm. In each simulation, we compare the performance of the control schemes by means of two metrics:

- the computation time, which is of primary concern for the real-time applicability of the algorithms, since the new input must be computed within short sampling times to iteratively re-plan the optimal trajectory of the EV;
- the optimality of the closed-loop trajectory resulting from each control scheme, assessed by computing the average stage cost:

$$J_{\text{sim}} = \frac{1}{N_{\text{sim}}} \sum_{t=1}^{N_{\text{sim}}} \left(\|\Delta\boldsymbol{\xi}(t)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(t)\|_{\boldsymbol{R}}^2 \right), \tag{4.34}$$

  where N_{sim} is the number of steps in the considered scenario.

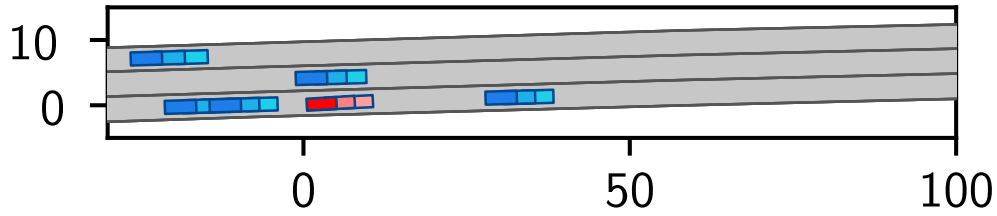


Figure 4.30.: Traffic configuration at time $t = 0\,\mathrm{s}$ of CommonRoad scenario *DEU_A99-1_2_T-1* used in the first simulation. The EV is represented in red. The driving direction is from left to right. Faded colors are used to show the positions of vehicles in the following two steps. © 2024 IEEE.

The weighing matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, as well as the reference target state $\boldsymbol{\xi}^*$ in $\Delta\boldsymbol{\xi} = \boldsymbol{\xi} - \boldsymbol{\xi}^*$, are chosen as in the SMPC optimal control problem (2.2). Thus, metrics J_{sim} is used to evaluate the optimality of the algorithms in closed loop with respect to the efficiency goals of the SMPC. If the SMPC goals are changed by choice of different numerical values (i.e. $\boldsymbol{R}$ and $\boldsymbol{Q}$), the metrics J_{sim} is also adapted consequently.

Overtaking Maneuver

In the first simulation, CommonRoad tag *DEU_A99-1_2_T-1*, the EV is located on the right-most lane of the highway, surrounded by five other vehicles. The initial condition is represented in Figure 4.30. Within the scenario, the vehicle located in the middle lane, just to the left of the EV, performs an overtaking maneuver.

For the first part of the simulation, both the SMPC+FTP [30] and the SMPC+CVPM control scheme apply the SMPC control input, since the SMPC optimal control problem is feasible and the safety check is successful, that is, the SMPC solution is safe. Therefore, both control schemes take advantage of the optimistic planning of SMPC. Observe that, since the control schemes are designed using the same numerical values for the prediction models of the DOs and for the physical limitations of the EV, in these first simulations the same control input is yielded by both schemes. At simulation time $t = 2.2\,\mathrm{s}$, see Figure 4.31, the vehicle performing the overtaking maneuver passes the EV and the optimal input obtained by SMPC fails in the safety check, that is, applying the SMPC solution might lead to a collision. Then, the previous SMPC+FTP scheme [30] applies the backup stored during the previous iteration, as outlined in Appendix B.2. Conversely, in the SMPC+CVPM scheme the CVPM module intercedes. In this case, the set computation shows that there exists a solution guaranteeing zero probability of constraint violation (robust case). In fact, the behavior of the overtaking vehicle did not violate the assumptions on the model. Therefore, CVPM is used only to prevent the risk of collision that the SMPC would lead to, and a solution with zero probability of collision is obtained. The computation of such solution is completely independent from the solution of the SMPC branch, therefore it is computed in parallel and does not compromise the real-time requirements of the scheme.

With respect to the average stage cost (4.34), in this scenario in which no unanticipated situation happens the SMPC+CVPM scheme yields an average stage cost $J_{\mathrm{sim}} = 0.283$, which is marginally lower than the cost $J_{\mathrm{sim}} = 0.302$ of the SMPC+FTP scheme [30]. The

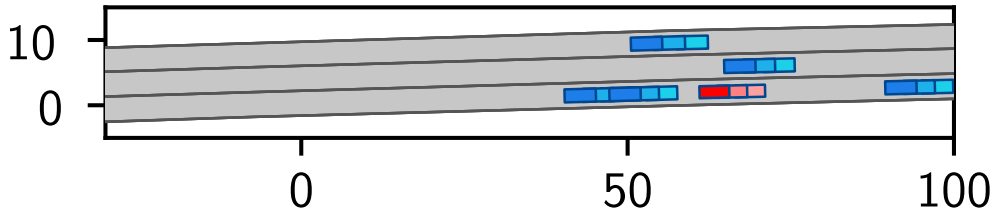


Figure 4.31.: Traffic configuration at time $t = 2.2\,\mathrm{s}$ of CommonRoad scenario *DEU_A99-1_2_T-1* used in the first simulation, when the input obtained by the SMPC branch is not safe and therefore is not applied. Faded colors are used to show the positions of vehicles in the following two steps. © 2024 IEEE.

Table 4.3.: Average computation times over 100 simulations.
Scenario: Overtaking Maneuver.

Scheme	Module	Average computation time [ms]
SMPC + CVPM	SMPC($\boldsymbol{x}$)	3.401 ± 0.232
	Safety check($\boldsymbol{x}^+$)	0.525 ± 0.068
	Total SMPC branch	$\mathbf{3.873 \pm 0.257}$
	Check case CVPM($\boldsymbol{x}$)	0.554 ± 0.068
	Total CVPM branch	$\mathbf{0.554 \pm 0.068}$
	Total	$\mathbf{3.873 \pm 0.257}$
SMPC + FTP [30]	SMPC($\boldsymbol{x}$)	3.158 ± 0.187
	FTP($\boldsymbol{x}^+$)	10.144 ± 0.735
	Total	$\mathbf{12.287 \pm 0.829}$

average computation times are analyzed in Table 4.3, where the results of average and maximum computation times for each module used in the simulation are reported in the format mean $\pm$ std. The numbers reported in Table 4.3 are referred to 100 simulations of the scenario and different colors are used to distinguish modules belonging to the SMPC branch from those of the CVPM branch. The average computation time for the SMPC branch for the SMPC+FTP scheme [30] is 12.287 [ms], whereas in our SMPC+CVPM scheme is 3.873 [ms], with a reduction of 70%. The increased efficiency lies in the fact that the SMPC+FTP scheme [30], similar to [171], requires the solution of two cascaded optimal control problems (SMPC($\boldsymbol{x}$) and FTP($\boldsymbol{x}^+$)), whereas in our SMPC+CVPM scheme the SMPC optimal control problem is followed by a quick set computation (Safety check($\boldsymbol{x}^+$)). Observe that our scheme, although requiring a significantly shorter computation time, still only applies the SMPC solution after checking that it is safe.

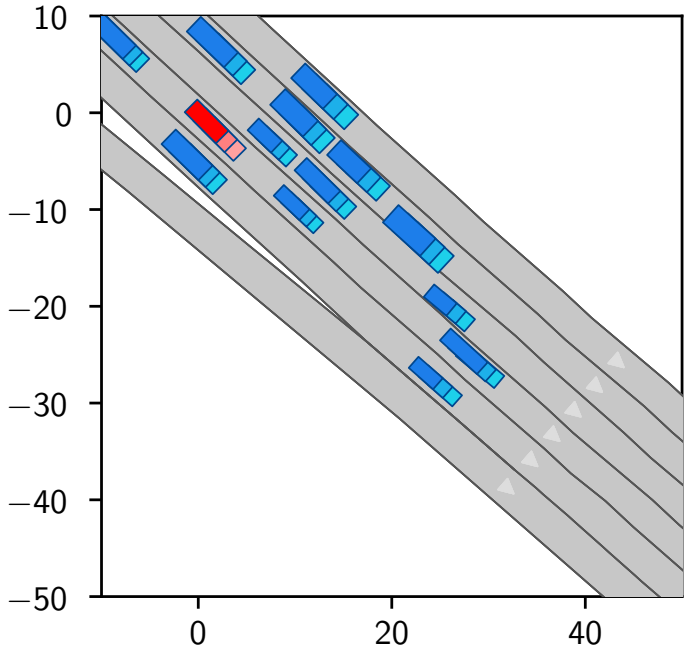


Figure 4.32.: Traffic configuration at time $t = 0\,\mathrm{s}$ of CommonRoad scenario *USA_US101-13_2_T-1* used in the second simulation. The EV is represented in red. The driving direction is from top-left to bottom-right. Faded colors are used to show the positions of vehicles in the following two steps. © 2024 IEEE.

Critical Emergency Braking on a Multi-Lane Highway

In the second simulation, CommonRoad *USA_US101-13_2_T-1*, the EV is located on the second lane from the right of the highway, surrounded by thirteen other cars, motorbikes or trucks. The initial condition is represented in Figure 4.32. Within the scenario, the traffic comes to a critical stop, forcing the EV to perform an emergency break in order to minimize the probability of collision with surrounding vehicles.

Also in this scenario, as long as the SMPC optimal control problem returns a solution which is safe, during the initial part of the braking maneuver the two schemes yield the same control inputs at each time step, during the initial part of the braking maneuver. Therefore, at first both algorithms benefit from the efficient planning of SMPC. However, the SMPC+FTP scheme [30] also computes and stores a safety backup.

From simulation time $t = 2.0\,\mathrm{s}$, represented in Figure 4.33, although the SMPC optimal control problem continues to provide a solution, the safety check fails, that is, the SMPC solution might result in collisions. In this situation, the SMPC+FTP scheme [30] relies on the safety backup stored during the previous iterations. This consists of a rapid deceleration while remaining at the center of the lane. Figure 4.34 shows the position and heading of the EV at the end of the simulation, at time $t = 2.6\,\mathrm{s}$.

In turn, if the SMPC solution is not safe, the SMPC+CVPM scheme applies the solution provided by the CVPM branch, which is run in parallel to the SMPC branch. In this case, no solution with zero probability of constraint violation exists, because the breaking maneuver of the DOs is very sudden and the actual trajectory violates the assumptions

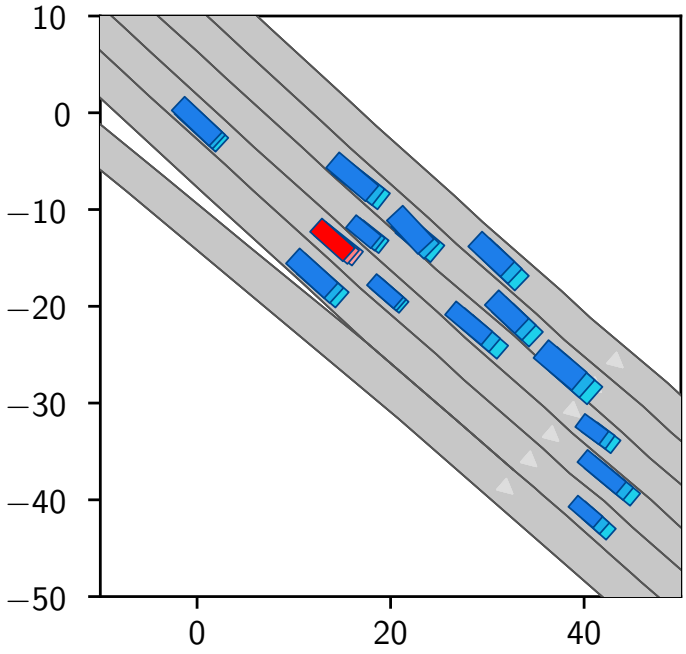


Figure 4.33.: Traffic configuration at time $t = 2.0\,\mathrm{s}$ of CommonRoad scenario *USA_US101-13_2_T-1* used in the second simulation, when the input obtained by the SMPC branch is not safe and therefore is not applied. Faded colors are used to show the positions of vehicles in the following two steps. © 2024 IEEE.

on the DO behavior from Section 4.5.1. This reveals that the previously-stored backup used in the SMPC+FTP scheme [30] is in fact not feasible for this iteration and does not provide any safety guarantee because the traffic participants have not behaved as anticipated. Therefore, the probabilistic case is used, that is, the SMPC+CVPM scheme determines a new sequence of input explicitly aiming at minimizing the overall probability of collision. In this scenario, the trajectory with minimum probability of collision consists of a braking and steering maneuver to the left, similar to the decision of several other human drivers in the recorded scenario. The position and heading of the EV at the end of the simulation is shown in Figure 4.35. Conversely, in the backup stored by the SMPC+FTP scheme the EV does not turn to the left to minimize the probability of collisions with DOs, see Figure 4.34.

In this scenario the SMPC+FTP scheme [30] results in an average stage cost (4.34) of $J_{\mathrm{sim}} = 102.9$, whereas the SMPC+CVPM scheme yields a marginally lower cost, $J_{\mathrm{sim}} = 100.7$. The much higher average stage cost of both algorithms in comparison to the first scenario is due to the fact that here the EV is constantly proceeding at a much slower speed than the reference. Moreover, the benefit in terms of average computation time is clear. Table 4.4 summarizes the results obtained running the scenario 100 times, analogously to Table 4.3. In this scenario, the average computation time for the SMPC branch for the SMPC+FTP scheme [30] is 23.115 [ms], whereas in our scheme is 10.181 [ms], with a reduction of 55%. Considering that the computation time scales significantly with the complexity of the scenario, for example with a larger number of surrounding vehicles, this reduction in the computation time can prove important for the practical application of the scheme with short sampling times.

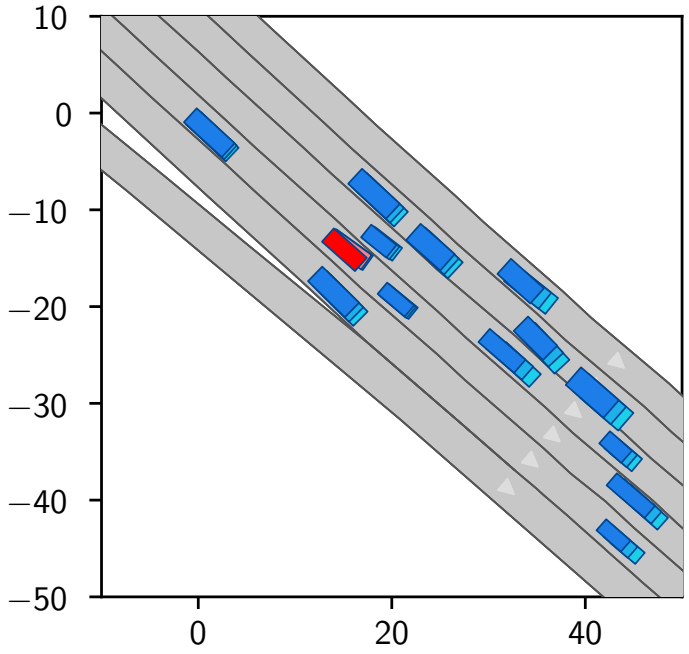


Figure 4.34.: Traffic configuration at the end of the simulation at time $t = 2.4\,\mathrm{s}$ of CommonRoad scenario *USA_US101-13_2_T-1* used in the second simulation, if the SMPC+FTP scheme [30] is applied. Faded colors are used to show the positions of vehicles in the following two steps. © 2024 IEEE.

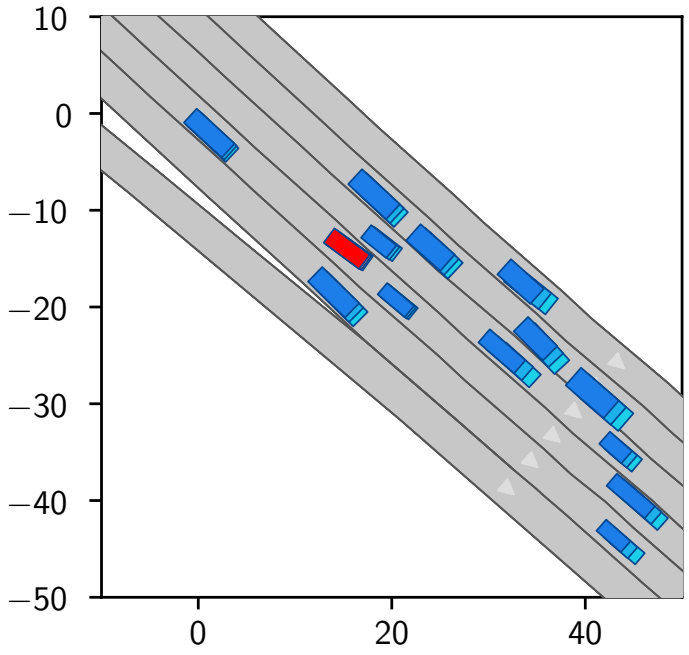


Figure 4.35.: Traffic configuration at the end of the simulation at time $t = 2.4\,\mathrm{s}$ of CommonRoad scenario *USA_US101-13_2_T-1* used in the second simulation, if the SMPC+CVPM scheme is applied. Faded colors are used to show the positions of vehicles in the following two steps. © 2024 IEEE.

Table 4.4.: Average computation times over 100 simulations.
Scenario: Critical Emergency Braking.

Scheme	Module	Average computation time [ms]
SMPC + CVPM	SMPC($\boldsymbol{x}$)	9.434 ± 0.387
	Safety check($\boldsymbol{x}^+$)	0.746 ± 0.024
	Total SMPC branch	$\mathbf{10.181 \pm 0.393}$
	Check case CVPM($\boldsymbol{x}$)	0.753 ± 0.027
	CVPM Prob($\boldsymbol{x}$)	1.316 ± 0.048
	Total CVPM branch	$\mathbf{2.068 \pm 0.066}$
	Total	$\mathbf{10.181 \pm 0.393}$
SMPC + FTP [30]	SMPC($\boldsymbol{x}$)	10.326 ± 0.386
	FTP($\boldsymbol{x}^+$)	12.789 ± 0.390
	Total	$\mathbf{23.115 \pm 0.429}$

4.5.4. Discussion

One major advantage of the SMPC+CVPM scheme is that unanticipated situations, for which a feasible and safe solution cannot be determined, are explicitly addressed with a view at minimizing the probability of collision. Although intuitively reasonable, minimizing the probability of collision should not be the only objective of the EV. For example, between a first trajectory yielding a 40% probability of collision with anticipated 10% severity of the impact and a second trajectory in which the probability of collision is 30% and the expected severity of the impact is 20%, it is not obvious that the latter trajectory, bearing the lowest probability of collision, is preferable. In general, the anticipated harm caused by a collision is to be taken into account and sometimes choosing the trajectory yielding a higher probability of collision proves beneficial, if this in turns significantly reduces the severity caused by the collision. To balance among different types of conflicting objectives, that is, minimizing the probability of collision and the anticipated severity of the impact, respectively, an ad-hoc cost function must be designed. However, the balance between these objectives is worth a special discussion and is not a purely technical matter, rather ethical issues related to the well-known Trolley Problem arise, as discussed in [173]. Nonetheless, for a given cost function designed appropriately to account for both the probability of collision and the anticipated harm of the impact, the SMPC+CVPMalgorithm can be used.

In the presented scheme SMPC+CVPM, three optimal control problems are considered: SMPC($\boldsymbol{x}$), CVPM Robust($\boldsymbol{x}^+$) (although only to assess the existence of a feasible solution), and CVPM($\boldsymbol{x}$). In our formulation in Section 2.3.7 the same prediction horizon length has been used for all of them. However, the scheme can be trivially adapted to account for three different prediction horizon lengths. Moreover, the probability minimized in the CVPM scheme might consider only a subset of the full prediction horizon. If the probability minimization constraint considers only the first few prediction steps, focus is devoted to the

possible collisions in the more immediate prediction steps over further ones. This seems intuitively reasonable, since the first prediction steps are also those with smaller uncertainty in general. Nevertheless, focusing exclusively on the first steps for the minimization of the probability yields the typical disadvantages of greedy approaches. As a result, the chances of collision might worsen for steps later in time because of decisions at earlier steps in which a shorter horizon was considered. Heuristics and application-specific considerations must be adopted to rule the trade-off and select an appropriate horizon length for the probability minimization constraint.

Compared to safe planners relying on a pre-stored fail-safe trajectory [30], [171], the SMPC+CVPM scheme offers several advantages. Concerning the computation complexity, the SMPC+FTP scheme [30] intrinsically requires the cascaded solution of two optimal control problems, since to guarantee safety a backup for the next iteration must be computed and stored. Similar considerations hold for [171]. In practice, for short sampling times, the solution of two cascaded optimization problems is challenging. Alternatively, the scheme from [31] computes jointly the nominal and the backup trajectory, significantly increasing the size of the optimal control problem and, therefore, the computation time. Conversely, our scheme does not require the computation of a backup and therefore requires a significantly smaller computation time, as discussed with the simulations in Section 4.5.3. At the same time, the solution of two branches in parallel is only possible if computational resources allow. However, the SMPC+FTP algorithm likewise requires the solution of two branches independently (the SMPC and the Fail-safe Trajectory Planning (FTP) branch, respectively). If the computational resources are limited, in both algorithms the two branches can be solved sequentially.

Although the backup fail-safe trajectory is not computed, in our scheme the SMPC solution is applied only if it does not lead to a collision. In fact, it is not necessary to pre-compute the backup for following iterations, rather it suffices to verify that from the first predicted state of the SMPC trajectory a safe (robust) solution can be obtained. Then, if necessary, such safe solution will be rapidly computed at the next iteration by the robust case of CVPM, in parallel to the SMPC branch. Therefore, the increased efficiency of our scheme compared to [30], [171] does not compromise safety.

Furthermore, in presence of unanticipated behaviors of DOs that prevents the existence of a robust safe trajectory, the SMPC+FTP scheme [30] applies the backup solution derived at the previous iteration. However, the safety backup is a valid and safe trajectory only under the assumption that the DOs behave as expected at the previous iteration. In doing so, the scheme does not compute a new trajectory to react to the present traffic configuration. Relying on a solution obtained several sampling times before is an hazard and could lead to danger in practice if DOs exhibit unanticipated behaviors. At best, this leads to suboptimal behaviors in terms of minimization of the probability of collisions. Conversely, our SMPC+CVPM scheme uses the probabilistic case of CVPM to plan the trajectory minimizing the probability of constraint violation.

Finally, although the stored backup input sequence from the SMPC+FTP [30] is theoretically not necessary and not optimal for unforeseen situations, still, in practice it might be desirable to compute the backup anyway and to use it to warm-start the optimal control problem of CVPM Robust at the following sampling time. Since the decision on whether or not to apply the SMPC solution depends on the safety check alone, the backup could be computed in another parallel branch not subject to tight computation times.

4.6. Conclusion

The methods proposed in this chapter discussed motion planning challenges in presence of high and multi-modal uncertainty, with a view at balancing safety and efficiency, considering the application of trajectory planning for autonomous driving. First, we proposed a hierarchical framework that allows to take advantage from long-term reasoning by handling short-term and long-term decisions separately, each based on its characteristic time scale.

Second, we proposed a novel combination of IMM and SMPC which is well suited to handle multi-modal uncertainty. Multiple nominal candidate trajectories of DOs, each representing a possible intention, are considered and their probabilities estimated online. Then, the EV trajectory is planned iteratively with in a SMPC fashion, in which the priority given to each trajectory explicitly depends on the probability of such trajectory. Thus, multiple possible intentions of DOs are considered, limiting the conservatism that would result from giving equal importance to all trajectories. The combination of IMM and SMPC is suitable for other contexts in which different possible modes of the uncertainty can be distinguished.

Then, the planning framework for multi-modal uncertainty is extended by considering uncertainty in the estimation of the mode probabilities. We proposed a novel framework for trajectory planning of autonomous vehicles, in which the design of collision-avoidance safety constraints leverages the measure of uncertainty provided by the BFT estimate. When the estimation of the different trends is not reliable, the focus is on not underestimating seemingly unlikely realizations, inducing caution. However, excessive conservatism is prevented if the estimation is reliable. The approaches to combine BFT and SMPC here introduced are applicable to other domains characterized by multi-modal uncertainty, where a trade-off between safety and efficiency is necessary and in which sever consequences can arise if conclusions are drawn from incomplete data.

Finally, we proposed a novel control framework for safe trajectory planning which combines SMPC and CVPM techniques. The efficient SMPC planning is exploited, but it is only applied when safe. Otherwise, a robust planner is used. Furthermore, we addressed the situation in which the uncertainty violates the assumed limits, thus invalidating the safety guarantees provided by robust approaches. In such case, the control input is chosen by minimizing the probability of constraint violation. The presented SMPC+CVPM scheme is suitable for application beyond autonomous driving, especially in frameworks characterized by large uncertainty because of the interaction with humans, e.g., human-robot-collaboration, for which SMPC yields an efficient solution, and CVPM is used to minimize the risk, that is, to give maximum safety given the uncertainty.

Addressing Unmodeled Uncertainty through Active Exploration

5.

In Chapter 3 and 4 we relied on model-based approaches to handle multi-modal uncertainty, allowing both mode recognition and or non-conservative planning. Here, by contrast, we focus on the uncertainty components that are difficult to model and employ machine learning techniques, namely GP regression, to address them. Rather than collecting a large quantity of data during normal operation of the system and use them for the training, we propose an active exploration mechanism. The system is encouraged to visit those regions of the state space where the prediction performance is most uncertain, so that a rich dataset is collected. We consider the context of autonomous racing, tackling both uncertainty coming from modeling errors in the vehicle dynamics and the unknown driving policy of the opponent. The content of this chapter was published in [24].

5.1. Introduction

Planning the control actions in an efficient way in presence of uncertainty is challenging, as the nominal model does not fully describe the behavior of the system. Even though multiple models can be employed to account for several nominal behaviors of a system, in several practical cases, such as in autonomous driving, unmodeled components not captured by the nominal models persist. However, in this application, components for which no deterministic model is available can be learned from data that would otherwise not be exploited.

Among the applications of autonomous driving, autonomous racing has recently gained increased attention [174], also for real-world tests like Roborace and the Indy Autonomous Challenge. Two scenarios are considered: the time trial and head-to-head racing against an opponent. In the former, a single race car drives around a constrained track trying to minimize the lap time. In head-to-head racing with an opponent, the EV must compete with another agent and perform overtaking maneuvers.

5.1.1. Challenges

In the time trial of the autonomous racing application, the main control challenges arise from pushing the vehicle to the handling limits, a task that expert humans can do well, but is challenging for control algorithms. In particular, physics-only models typically used in urban or highway environments are not well suited to represent the vehicle dynamics close to the handling limits. Aerodynamics forces, nonlinear deformations, nonlinear tire dynamics, and weight transfer in the vehicle caused by accelerating, braking, or steering are

effects typically neglected when deriving a simplified nominal vehicle dynamics model [108] for computationally efficient online control. Yet, they must not be ignored if the goal is to push the vehicle to its handling limits and minimize the lap time. Within model-based controllers, MPC relies on a prediction model of the vehicle, and the input is determined by iteratively solving an optimal control problem over a finite horizon. Thus, the large uncertainty introduced by the modeling errors when the vehicle is driven near handling limits must be accounted for, for example adding a learning component to the physics-based model [70], [115], [175] and it has been shown that model-free reinforcement learning can outperform human performance [176]. However, a general challenge with learning-based methods is to obtain data sufficiently representative, while still avoiding dangerous situations.

In the scenario with an opponent, the uncertainty in the vehicle dynamics near the handling limits is compounded by another major challenge: the interaction with the other agent, which is a well-researched problem for autonomous urban and highway driving [177], [178]. Both enforcing collision avoidance and planning a successful overtaking maneuver require the EV to handle the uncertainty about the unknown future position of the opponent. Initial approaches considered *passive* prediction models, that is, predicting the future trajectory of other agents given historical data and the current traffic configuration. Such approaches allow for a simplified planning framework, in which the future trajectories of other agents are assumed to be independent of the current decision of the EV. However, in highly interactive scenarios, such as automated racing, where the other agent is a competing opponent, the reaction of other agents to the EV decision must be considered. Knowledge of the opponent's reaction to own decisions is crucial to allow safe and efficient overtaking maneuvers. To account for the reaction to own decisions, the opponent can be represented as a rational agent in a game-theoretic framework [117], which is, however, computationally demanding. Alternatively, the policy of the opponent to the current and past configurations and the EV own decision can be learned from data [122], [124]. Then, it is fundamental to retrieve a training dataset sufficiently representative to allow for reliable learning of the policy.

5.1.2. Contribution

We deal with the problem of learning a prediction model that addresses both sources of uncertainty in trajectory planning for autonomous racing, i.e., modeling errors in the dynamics and the representation of the unknown policy of the opponent accounting for the reaction to the EV's own decisions, that is, obtaining *interaction-aware* models, as discussed in the literature review in Section 2.2. We use an iterative GP regression algorithm, following the approaches from [113], [124]. GP regression is a non-parametric machine learning framework that provides uncertainty measures over its prediction based on previously collected measurements. We propose a novel active exploration mechanism designed to retrieve representative data and greatly enhance the learning performance. Previous works in autonomous racing did not consider the active exploration of the feature space to improve the learning performance and relied on data collected while maximizing the EV performance. In turn, our novel approach exploits the measure of the model uncertainty provided by GPs, in contrast to other learning tools, such as artificial neural networks, to deliver an enriched dataset. Our novel exploration mechanism delivers a reference trajectory of the EV is designed to encourage the exploration of the regions of the feature space with a high posterior covariance of the prediction error. In doing so, the dataset is rapidly replaced with properly selected

data points that refine the learning performance.

Enriching the dataset through the active exploration mechanism yields a significant improvement in the learning performance and, eventually, in the EV performance. We show that the GP exploration algorithm can be applied successfully both when the GP model is used for error compensation in the time trial and for the opponent modeling in head-to-head racing.

5.2. Active Exporation of Unmodeled Uncertainty in Gaussian Process Regression

5.2.1. Problem Formulation

We discuss how representative data can be systematically collected in two autonomous racing scenarios, in which a learning component is used in the MPC optimal control problem.

We consider a racing vehicle, whose dynamics is modeled using a linearized and discretized dynamic bicycle model [109] referred to the road-aligned Frenet coordinates. The state of the vehicle is $\boldsymbol{\xi} = [v_s, v_d, \dot{\phi}, \phi, d, s]^\top$, where v_s and v_d are the vehicle's longitudinal and lateral velocity, respectively, in the vehicle's body frame, $\dot{\phi}$ is the yaw angular velocity, ϕ and d are the yaw angle and lateral displacement of the center of gravity of the vehicle with respect to the reference path, and s represents the traveled distance along the reference path. The relative yaw angle ϕ and lateral distance d in Frenet coordinates are defined with respect to the closest point of the reference path. The input $\boldsymbol{u} = [\delta, a]^\top$ consists of the front steering angle and of the longitudinal acceleration resulting from the powertrain, which is applied to the rear wheel. Although simplified, the bicycle model represents a good trade-off between keeping the number of parameters low, which is crucial for real-time computations, and having sufficiently accurate dynamics, that reflect the main characteristics of the motion.

When the vehicle is driven near the handling limits, the dynamic bicycle model (2.17), revised in Section 2.3.3, is insufficient to reliably describe the vehicle motion, in particular, neglecting the influence of nonlinear deformations, aerodynamics forces, and weight transfer in the vehicle caused by accelerating, braking, or steering. We integrate the nominal dynamics used in the MPC optimal control problem with a learning component [113], using GP regression, which is recalled in Section 2.3.6. Here, the predicted value is $\boldsymbol{y}_k^{\mathrm{MPC}} = \boldsymbol{\xi}_{k+1} - \boldsymbol{\xi}_{k+1}^{\mathrm{pred}}$, that is, the difference between the next state, $\boldsymbol{\xi}_{k+1}$, and the next state predicted by the nominal vehicle model from the current state (2.18), $\boldsymbol{\xi}_{k+1}^{\mathrm{pred}} = \boldsymbol{\xi}_k + \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k)T$, where T is the sampling time. Specifically, if the GP compensation term added to the nominal system dynamics is defined as

$$\boldsymbol{y}^{\mathrm{MPC}} = \boldsymbol{g}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}}) \sim \mathcal{N}\left(\boldsymbol{\mu}^{\mathrm{MPC}}(\boldsymbol{z}^{\mathrm{MPC}}), \boldsymbol{\Sigma}^{\mathrm{MPC}}(\boldsymbol{z}^{\mathrm{MPC}})\right), \tag{5.1}$$

then the system dynamics can be modeled as

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{A}_k\boldsymbol{\xi}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{d}_k + \boldsymbol{\mu}^{\mathrm{MPC}}(\boldsymbol{z}^{\mathrm{MPC}}), \tag{5.2}$$

where μ^{MPC} models the error of the linearized dynamic bicycle model (2.18) with respect to the real dynamics at the GP input feature $\boldsymbol{z}_{\mathrm{MPC}}$. The mean and standard deviation of the distribution, $\boldsymbol{\mu}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}})$, and $\boldsymbol{\Sigma}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}})$, are obtained from (2.36) to predict the output $\boldsymbol{y}_i$ from the input feature vector $\boldsymbol{z}_i$. Once the GP model is trained, $\boldsymbol{y}_k^{\mathrm{MPC}}$ can be added to $\boldsymbol{\xi}_{k+1}^{\mathrm{pred}}$ to compensate for modeling errors.

Following [113], we use GP to compensate the states having the greatest impact on the prediction error, v_d and $\dot{\phi}$, i.e., $\boldsymbol{\mu}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}}) = [0, \mu_{v_d}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}}), \mu_{\dot{\phi}}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}}), 0, 0, 0]^\top$. In this application, the GP input feature vector is $\boldsymbol{z}_{\mathrm{MPC}} = \boldsymbol{G}[\boldsymbol{\xi}^\top, \boldsymbol{u}^\top]^\top = [v_d, \dot{\phi}, \delta]^\top$. Furthermore, defining the $\boldsymbol{z}^{\mathrm{MPC}}$ with respect to a nominal predicted trajectory rather than on the actual predicted state $\boldsymbol{\xi}_k$ and predicted input $\boldsymbol{u}_k$ allows real-time computation of the MPC [110], [113]. Precisely, $\boldsymbol{z}^{\mathrm{MPC}}$ is computed from the nominal state $\tilde{\boldsymbol{\xi}}_k$ and $\tilde{\boldsymbol{u}}_k$ of the linearized

dynamics from the previous MPC iteration. We record the values of $(\boldsymbol{z}_k^{\text{MPC}}, \boldsymbol{y}_k^{\text{MPC}})$ during real-time control to construct the training dataset $\mathcal{D}$. Measurements collected while tracking the optimal path are not necessarily diverse enough to train the GP models since the states encountered will be concentrated in a small subset around the reference path being tracked. It limits the learning performance and, consequently, the improvement yielded by the GP compensation both in the planning and MPC tracking.

In racing scenarios with an opponent, a model of the opponent is necessary, to predict the opponent future trajectory and plan overtaking maneuver, while avoiding collisions. The combined effect of the opponent's dynamics and its policy is modeled as a GP, namely, we consider

$$y_k^{\text{O}} = \boldsymbol{\xi}_{k+1}^{\text{O}} = \boldsymbol{\xi}_k^{\text{O}} + \boldsymbol{f}(\boldsymbol{\xi}_k^{\text{O}}, \boldsymbol{\pi}_k(\boldsymbol{\xi}_k^{\text{O}}, \boldsymbol{\xi}_k))T, \tag{5.3}$$

where $\boldsymbol{f}$ represents the real dynamics of the opponent and $\boldsymbol{\pi}$ represents the one-step opponent policy, which depends on the current opponent state $\boldsymbol{\xi}_k^{\text{O}}$ and on the current EV state $\boldsymbol{\xi}_k$, in order to incorporate the opponent reaction to the EV decisions in the prediction steps [124]. We model the one-step closed-loop dynamics of the opponent in (5.3) with the GP

$$\boldsymbol{g}^{\text{O}}(\boldsymbol{z}_{\text{O}}) \sim \mathcal{N}\left(\boldsymbol{\mu}^{\text{O}}(\boldsymbol{z}_{\text{O}}), \boldsymbol{\Sigma}^{\text{O}}(\boldsymbol{z}_{\text{O}})\right), \tag{5.4}$$

where z_{O} is the GP input feature vector defined as:

$$\boldsymbol{z}_{\text{O}} = [s^{\text{O}} - s, d^{\text{O}} - d, \phi, v_s, \phi^{\text{O}}, v_s^{\text{O}}, \bar{\boldsymbol{\kappa}}]^\top. \tag{5.5}$$

$\boldsymbol{z}_{\text{O}}$ contains the longitudinal and lateral distance between the EV and the opponent, the yaw angle and longitudinal velocity of both vehicles, and the vector $\hat{\boldsymbol{\kappa}}$ which contains the track curvature at a few look-ahead points, with a view to considering that humans typically choose their actions accounting for the future evolution of the track [124]. The GP input features (5.5) consist only of the relative configuration of the EV and of the opponents and of their position in the curvilinear Frenet coordinates rather than in the absolute coordinates, with a view to boosting the generalization capability of the GP prediction. Furthermore, the prediction of the opponent's trajectory is obtained by averaging over several samples from the GP model as in [124, Algorithm 1].

Also the GP predicting the opponent's future trajectory, as it held true for the GP compensating for the model errors in the dynamics, is only accurate if the training data are sufficiently representative, yielding the following objective.

Objective 5.1. *The goal is to control the EV in order to collect representative data of the dynamics or of the opponent's behavior during the race. The approach should be applicable to both the time trial and head-to-head racing.*

Previous work [110] found that adopting a more complicated model, e.g., a four-wheeled model, only marginally improved the racing performance compared to the bicycle model and thus introduced unnecessary computational costs to the optimal control problem. Also, introducing a more sophisticated nominal dynamics model, especially one with a higher dimensional state space, results in a larger parameter space for the GP compensation model, since a larger list of features would be necessary to span the feature space. It would then lead to increasing computation time and memory usage to run the active exploration algorithm, which will be apparent after we introduce the algorithm in the following.

5.2.2. Novel Active Exploration Scheme for Autonomous Racing

Here, we introduce our active exploration framework that can iteratively train GP models in autonomous racing scenarios. The exploration mechanism is augmented into the objective function of an MPC, and it strategically explores locations of the state space where the GP model has the highest uncertainty.

We first detail the active exploration framework for the time trial racing challenge, where a GP model is used to compensate for EV modeling error in an offline optimal trajectory planner and online trajectory-tracking MPC. Then, we consider head-to-head racing with an opponent, where the GP model is used to predict the behavior of the opponent and the EV employs an online MPC for trajectory planning and control. We discuss the necessary changes we made to adapt the proposed active exploration framework to this scenario.

Minimum Lap Time Application

In the minimum lap time task, our iterative exploration-based controller is used to compensate for modeling errors in the dynamics when the vehicle approaches handling limits. To minimize the lap time, first, a time-optimal trajectory for the EV is planned, then, the vehicle is driven around the track by the MPC. Relying on the nominal model of the vehicle dynamics does not suffice to minimize the lap time. Therefore, we use a GP compensation model to improve the prediction of the EV state. It is important to account for the influence of unmodeled effects of the dynamics also on the optimal path, therefore we use the GP compensation both in the planning and in the MPC tracking phase, adopting the Double GP compensation scheme that was presented in [113]. At the end of the trial, the measurements collected are used to retrain both GP models, and the trials are repeated iteratively.

Our method leverages the uncertainty in the GP prediction as a heuristic to guide active data collection in regions of high uncertainty, with the goal of improving the prediction accuracy of the GP. In the first trials, active exploration takes place; namely, the EV control is determined as a trade-off between the MPC performance objective and the exploration objective. In doing so, the enriched data can be used iteratively to re-train the GP; thus, the GP will be more accurate, particularly in regions of high uncertainty in previous iterations. Our goal is to improve the overall performance of the MPC after exploration is complete, and the MPC may fully exploit the more accurate GP model. In the following, we discuss the components and main aspects of our proposal, which is implemented in the online tracking phase for the time trial. The derivation of the time-optimal reference is detailed in [24].

1) MPC The optimal control problem of tracking MPC at time step $t = 0$ is

$$\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \sum_{k=0}^{N-1} \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^{\text{ref}}\|_{\boldsymbol{Q}} + r_\delta \Delta\delta_k^2 \tag{5.6a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{A}_k\boldsymbol{\xi}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{d}_k + \boldsymbol{\mu}^{\text{MPC}}(\boldsymbol{z}^{\text{MPC}}(\tilde{\boldsymbol{\xi}}_k, \tilde{\boldsymbol{u}}_k)), \quad \forall k = 0, \dots, N-1 \tag{5.6b}$$

$$w_{\text{r},k} + \gamma_k \leq d_k \leq w_{\text{l},k} - \gamma_k, \qquad \forall k = 1, \dots, N \tag{5.6c}$$

$$\boldsymbol{u}_{\min,k} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max,k}, \qquad \forall k = 0, \dots, N-1, \tag{5.6d}$$

where N is the MPC prediction horizon. The cost function (5.6a) is designed to penalize rapid changes in the steering angle according to weight $r_\delta > 0$, with $\Delta\delta_k = \delta_k - \delta_{k-1}$ and δ_{-1}

is set equal to the last applied steering angle δ_{t-1} at time $t-1$. $\boldsymbol{Q} \geq 0$ is the weight to penalize deviations of the state $\boldsymbol{\xi}_k$ from the reference $\boldsymbol{\xi}_k^{\text{ref}}$, which plays an important role in encouraging the exploration of the feature space depending on the value of α. Constraint (5.6b) relies on a linearized version of the bicycle model dynamics, computed with respect to a nominal trajectory $\tilde{\boldsymbol{\xi}}, \tilde{\boldsymbol{u}}$ [110]. Since the GP model is not embedded into the optimization [110], minimization problem (5.6) is a quadratic problem that can be solved in real-time.

Prior work [113] has shown that iteratively collecting data with the MPC and retraining the GP model (5.1) can improve the performance of the GP prediction and MPC. We further propose a mechanism that will employ the knowledge of the fact that $\boldsymbol{y}^{\text{MPC}}$ is predicted by a GP to purposefully explore regions of the state space where the prediction of $\boldsymbol{y}^{\text{MPC}}$ has larger uncertainty. Inspired by [179], we use large posterior covariance of the GP to indicate regions of the state space that need further exploration. Inspired by [137], we realize active exploration by appropriately changing the state reference $\boldsymbol{\xi}^{\text{ref}}$ in the optimal control problem (5.6). By doing so, the MPC cost function in (5.6a) is a function of the predicted states and control actions *and* the posterior covariance of the GP model. The cost function will trade-off between the MPC's original performance objective and the exploration objective as described in the next paragraph.

2) Active Exploration The goal of the active exploration mechanism is to solve (5.6) such that it encourages the exploration of the feature space and collects new measurements that enrich the dataset $\mathcal{D}$. For this purpose, we change the reference $\boldsymbol{\xi}_k^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}}$, obtained from the optimal planner, to visit states where the uncertainty in the prediction is large.

We consider the feature vector $\boldsymbol{z}$ used for GP prediction and determine its target value $\boldsymbol{z}^{\text{ref}}$ that the MPC should explore to improve the prediction accuracy of the GP, while considering the performance objectives at the same time. The target GP input feature vector $\boldsymbol{z}^{\text{ref}}$ is chosen from a list of n_{G} candidate feature vectors, $\{\boldsymbol{z}^{(i)}\}_i^{n_{\text{G}}}$. The list is designed to span the feature space: First, the candidate values of each feature component are selected to cover its maximum range estimated from collected data; Then, candidate feature vectors are created combinatorially from the candidate values of all the feature components. It is worth noting that by first determining the exploration objective $\boldsymbol{z}^{\text{ref}}$ and following it in terms of tracking a reference trajectory in the MPC cost function, the computational complexity required to solve the optimal control problem can be kept low, in contrast to directly incorporating the feature uncertainty as an exploration intrinsic in the cost function (5.6a), which will then embed the GP model into the optimization problem and cause non-convexity [134].

Our algorithm to set a new reference takes as input the reference state and action trajectory computed by the offline planner, $\boldsymbol{\xi}_k^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}}$, and outputs the updated reference $\boldsymbol{\xi}_k^{\text{ref}}, \boldsymbol{u}_k^{\text{ref}}$, selected so that the target GP input feature vector $\boldsymbol{z}^{\text{ref}}$ is visited. The algorithm consists of the following steps:

1. First, $\tilde{\boldsymbol{z}} = \boldsymbol{G}[(\boldsymbol{\xi}_k^{\text{ref}})^\top, (\boldsymbol{u}_k^{\text{ref}})^\top]^\top$ is computed, that is, the feature vector that would be visited following the trajectory of the optimal planner.

2. Then, the candidate feature vectors $\boldsymbol{z}^{(i)}$ are ranked based on two competing criteria: 1) *their proximity to $\tilde{\boldsymbol{z}}$, the features of the optimal reference state.* For each candidate feature vector $\boldsymbol{z}_i, i = 1, \ldots, n_{\text{G}}$, its proximity is quantified by the rank of its distance to $\tilde{\boldsymbol{z}}$ among all the candidates, denoted by D_i, with $D_i = 1$ implying the farthest

from $\tilde{\boldsymbol{z}}$ and $D_i = n_{\mathrm{G}}$ implying the closest; and 2) *their posterior covariance*, which are calculated using the GP model and weighted by the matrix S.

3. Analogous to D_i, we introduce the ordering index V_i with the covariance as the sorting criterion, with $V_i = 1$ implying the lowest posterior covariance and $V_i = n_{\mathrm{G}}$ implying the highest posterior covariance.

4. In order to trade-off between these objectives, we select the target feature vector by maximizing their convex combination with weight $\alpha \in [0, 1]$. Once the target feature vector is selected, we find its corresponding state and action to update the reference state used for the tracking MPC.

The covariance of GP has previously been used as a mechanism for data selection [179], as well as for autonomous racing [71]. By selecting $\boldsymbol{z}^{\mathrm{ref}}$ to balance the distance from the racing objectives and covariance criterion, the MPC balances exploration while remaining near the original MPC reference. Increasing α places more weight on the exploration of the feature space, and when $\alpha = 0$, the MPC defaults to use its standard reference. The target value $\boldsymbol{z}^{\mathrm{ref}}$ should not be too far $\tilde{\boldsymbol{z}}$ to prevent significant deterioration in the performance of the controlled system, and to prevent possibly dangerous behaviors and loss of stability during the exploration. Conversely, $\boldsymbol{z}^{\mathrm{ref}}$ should correspond to values with large posterior covariance to explore uncertain regions of the state space. We opt to weight the posterior covariance $\boldsymbol{\Sigma}^{\boldsymbol{\xi}}(\boldsymbol{z}^{(i)})$ by a positive semi-definite matrix $\boldsymbol{S} \geq 0$. The weighing matrix $\boldsymbol{S}$ is a hyperparameter used to reflect the relative importance of the uncertainty in different GP components. We set $\boldsymbol{S}$ equal to the weights from the MPC cost function (5.6a), with a view to giving priority to features whose posterior covariance is larger for those components that are more relevant for the MPC tracking. It is also worth observing that the posterior covariance for each candidate feature in the list can be computed immediately after the training of the GP and stored prior to MPC run time, significantly reducing the computational demand of the algorithm at run time.

In the first few iterations, we set α to be large, so that the focus is placed on the exploration of the feature space and the collection of data points that enrich the dataset. In later iterations, α is decreased to zero; thus, the focus is entirely on the performance objectives of the MPC, taking advantage of the accurate GP prediction model obtained from training with the dataset from the exploration.

Remark 5.1. *We adopt the algorithm to select the target GP input feature vector [137], but replace the mutual information-based exploration metric used in [137] with the weighted posterior covariance. Computing the mutual information-based exploration metric requires selecting a set of data points to quantify the information gain of a given feature vector candidate. The selection procedure itself introduces additional computational costs. Moreover, computing mutual information can also be time-consuming, since it involves high-dimensional matrix inversion and multiplication. To this end, we use the weighted posterior covariance, which can be directly obtained from the GP model, to simplify the computation. Meanwhile, as we will introduce later, we also use the weighted posterior covariance as a heuristic to diversify the selected training data. We intend to use the same metrics to specify consistent data priority for the active exploration and data selection phases.*

3) Diverse Data Selection During the repeated trials, a large number of data points are collected. Using all such data to train the GP model is impractical and unnecessary, as a smaller dataset of appropriately selected data points suffices to represent the input-output relation. Yet, creating a smaller dataset by randomly sampling from the collected data points, as in [113], does not guarantee that the diverse data points collected during the exploration phase are appropriately exploited. For this reason, at the end of each iteration, we train the GP using a smaller dataset of points obtained with the data selection approach described in [179].

The goal is to select a (small) collection of points $\mathcal{D} = \{(\boldsymbol{z}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$ to represent the feature space and allow for an accurate GP prediction. To add a data point $(\boldsymbol{z}_i, \boldsymbol{y}_i)$ to the dataset or replacing existing ones, our policy leverages a similarity measure between the new data point and the present collection, i.e., the posterior prediction covariance (2.36b) at $\boldsymbol{z}_i$ given all other data points in the dataset $\mathcal{D}$. The policy to update the dataset works as follows:

- If a data point's posterior covariance given the current dataset is larger than the median of the posterior covariance of all data points currently in the dataset, for at least one of its output features, it is added to the dataset;
- If the dataset is full, the new data point replaces the data point in the dataset with the smallest posterior covariance.
- If the data points yielding the lowest posterior covariance for different output dimensions are different, we consider the dimension in which the ratio between the new posterior covariance of the new point and the minimum posterior covariance of points in the dataset is the largest. Moreover, we use the outlier rejection mechanism described in [71, Section V-B].

In contrast to [71], we do not consider a decay factor to encourage the removal of older data points first, with the goal of prioritizing data points that contribute the most to maximizing the data covariance, rather than the most recent points. In fact, older data points that have been collected during the exploration in previous iterations are, in general, more significant than recent points collected during the last iterations, in which the focus is on the maximization of the performance.

Remark 5.2. *Two GP models are used to compensate for the errors in the dynamics, one during the planning phase and one during the MPC tracking phase. The two GP models differ since they compensate for different nominal dynamics, namely the nonlinear dynamics used in the offline planning problem and the linearized dynamics used in the MPC tracking problem, respectively. Consequently, two different datasets are extracted separately from the set of data collected during the trials, with each dataset diversified considering the prediction error and covariance with respect to the nominal dynamics used in the planning and MPC tracking phase respectively.*

4) Constraint Tightening Constraints (5.6c) and (5.6d) ensure that the lateral position of the vehicle and the input stay within the track bounds and the actuation bounds, respectively. Since the prediction of the lateral error in (5.6b) is influenced by the GP compensation $\boldsymbol{g}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}})$, in contrast to [113], we tighten the constraints to address the uncertainty in the prediction. Taking uncertainty into account in the constraints is crucially important

to reduce the risk of dangerous movements of the EV during the exploration phase. The modeling error $\epsilon_{y,k}$ for d at prediction step k is an affine transformation of Gaussian variables, therefore, is also Gaussian distributed. Thus, the support of the uncertainty $\epsilon_{y,k}$ is unbounded, and a robust tightening, guaranteeing constraint satisfaction for all realizations of the uncertainty $\epsilon_{y,k}$, is not possible. Hence, we implement a stochastic tightening requiring

$$\Pr\left(d_k + \epsilon_{y,k} \leq w_{\mathrm{l},k}\right) \geq \beta, \tag{5.7}$$

where $0 \leq \beta \leq 1$ is the risk parameter. Constraint (5.7) yields a deterministic formulation for the tightening parameter γ_k in (5.6c). The covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{\xi}}$ of the predicted state $\boldsymbol{\xi}_k$ at step $k = 1, \ldots, N$ is obtained recursively from the dynamics (5.6b) and from the covariance $\boldsymbol{\Sigma}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}})$ of the GP compensation $\boldsymbol{g}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}})$ as

$$\boldsymbol{\Sigma}_{k+1}^{\boldsymbol{\xi}} = \boldsymbol{A}_k \boldsymbol{\Sigma}_k^{\boldsymbol{\xi}} \boldsymbol{A}_k^{\top} + \boldsymbol{\Sigma}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}}), \tag{5.8}$$

where for all prediction steps k the state $\boldsymbol{\xi}_k$ and the GP compensation $\boldsymbol{g}^{\mathrm{MPC}}(\boldsymbol{z}_{\mathrm{MPC}})$ are uncorrelated because the compensation is computed from a nominal trajectory, $\tilde{\boldsymbol{\xi}}_k, \tilde{\boldsymbol{u}}_k$. From $\boldsymbol{\Sigma}_k^{\boldsymbol{\xi}}$, the covariance $\sigma_{d,k}^2$ of the prediction error $\epsilon_{y,k}$ at prediction step k is obtained and the tightening parameter γ_k is computed as in [17]

$$\gamma_k = \sqrt{2}\sigma_{d,k}\mathrm{erf}(2\beta - 1). \tag{5.9}$$

Because of symmetry, the same tightening parameter is applied to the lower bound in (5.6c).

Head-to-Head Racing Application

In head-to-head racing, the EV needs to predict the opponent's future trajectory in order to plan overtaking maneuvers. The opponent's future trajectory depends on the opponent's reaction to the EV's own decision. It would be unrealistic to assume that the opponent's policy is known, and this fact represents a source of uncertainty. Following the approach from [124], we model the policy and dynamics of the opponent as a GP model, which is included in the EV controller.

In the following, we present the active exploration mechanism for head-to-head racing, which is an adaptation of the active exploration mechanism used in the time trial case. Here, the aim is to retrieve informative data from the opponent's reaction to several overtaking attempts of the EV. Further, we discuss specific limitations and challenges that pertain to the active exploration in head-to-head racing, due to the fact that the EV does not have full control over the feature space, and we outline how the active exploration takes place in the setting of a single competition with the opponent. In this racing scenario, we consider the *extended* state of the system $\boldsymbol{\xi}^{\mathrm{E}} = [\boldsymbol{\xi}, \boldsymbol{\xi}^{\mathrm{O}}]^{\top}$, which contains both the state of the ego vehicle $\boldsymbol{\xi}$ and of the opponent $\boldsymbol{\xi}^{\mathrm{O}}$. The opponent's state is defined as $\boldsymbol{\xi}^{\mathrm{O}} = [s^{\mathrm{O}}, d^{\mathrm{O}}, \phi^{\mathrm{O}}, v_s^{\mathrm{O}}]^{\top}$. Here s^{O} and d^{O} are the longitudinal and lateral position of the opponent on the track, ϕ^{O} is the yaw angle with respect to the reference of the track, and v_s^{O} is the longitudinal velocity.

1) Active Exploration and MPC The EV trajectory is computed iteratively by an MPC, based on the formulation in [124], where the cost function consists of several performance-based objectives. Unlike the time-trial case, the control objective in head-to-head racing is not to track an offline planned optimal trajectory over the track, which is infeasible to obtain

in a prior, since the dynamic reaction of the opponent at run time must be considered. We aim to adapt the active exploration mechanism we developed for the time trial with minimal modifications to the baseline approach [124] for a close comparison. To this end, we introduce an additional term in the cost function that penalizes the deviation from a target reference state set for exploration, as we did in the time trial. The EV's target reference state $\boldsymbol{\xi}^{\text{ref}}$ is selected to test the opponent's reaction to EV's overtaking attempts, in order to collect informative data for accurate opponent modeling. Since we no longer have access to an offline planned reference trajectory, we determine the target reference states as follows.

First, we determine the initial reference state $\boldsymbol{\xi}^{\text{ref}}$ based on the *nominal* behavior of the two vehicles: we approximate the one-step future relative configuration of the two vehicles, assuming they follow their current linear velocities and yaw angles. Then, such reference state $\boldsymbol{\xi}^{\text{ref}}$ is modified using the algorithm from used in the time trial application. On the one hand, the distance of a candidate feature from the feature $\tilde{\boldsymbol{z}}$ visited following the nominal behavior is penalized, to prevent the EV from moving in a possibly dangerous way; on the other hand, visiting regions of the feature space with high posterior covariance is encouraged, to test the reaction of the opponent to EV behaviors for which the prediction is more uncertain.

Given the target GP input feature vector $\boldsymbol{z}^{\text{ref}}$ from the algorithm for the time trial application, the EV reference state $\boldsymbol{\xi}_k^{\text{ref}}$ is obtained with the following procedure: 1) The yaw angle ϕ and the longitudinal velocity v_s are obtained from the third and fourth entry of $\boldsymbol{z}^{\text{ref}}$, since $\boldsymbol{z}^{\text{ref}}$ is defined as in (5.5). 2) We use the current GP model to predict the opponent's trajectory over the prediction horizon. Notably, the prediction is conditioned on a hypothetical future trajectory of the EV, which is the open-loop solution of the MPC problem at the last iteration. In doing so, we account for the opponent's reaction to the planned EV future movements without coupling the GP model with the optimization problem or the active exploration algorithm, as in [124]. 3) Then, the (time-varying) target longitudinal and lateral positions of the EV are obtained from the predicted trajectory of the opponent, subtracting the first and second entry of vector $\boldsymbol{z}^{\text{ref}}$, respectively.

Eventually, the MPC optimal control problem solved at each iteration is:

$$\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \alpha \sum_{k=1}^{N} \|\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^{\text{ref}}\|_{\boldsymbol{Q}}^2 + (1-\alpha)\left(\sum_{k=0}^{N-1} q_c d_k^2 + \boldsymbol{u}_k^\top \boldsymbol{R} \boldsymbol{u}_k + \Delta \boldsymbol{u}_k^\top \boldsymbol{R}_d \Delta \boldsymbol{u}_k - q_s s_N^2 \right) \tag{5.10a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k) T, \qquad \forall k = 0, \dots, N-1 \tag{5.10b}$$

$$s_{k+1} = s_k + v_{s,k} T, \qquad \forall k = 0, \dots, N-1 \tag{5.10c}$$

$$w_{\text{r},k} \leq d_k \leq w_{\text{l},k}, \qquad \forall k = 1, \dots, N \tag{5.10d}$$

$$\boldsymbol{u}_{\min,k} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max,k}, \qquad \forall k = 0, \dots, N-1 \tag{5.10e}$$

$$\boldsymbol{0} \geq \boldsymbol{h}(\boldsymbol{\xi}_k, \boldsymbol{\xi}_k^{\text{O}}), \qquad \forall k = 1, \dots, N. \tag{5.10f}$$

Other than the penalty for deviations with respect to the reference state $\boldsymbol{\xi}^{\text{ref}}$, weighted by matrix $\boldsymbol{Q} \geq 0$, cost function (5.10a) includes racing objectives from [124]: namely penalties for lateral offset from the center line d, and for large inputs and large rates of change of the input, where $q_c > 0$ and $\boldsymbol{R}, \boldsymbol{R}_d \geq 0$. Moreover, the last term is included to maximize the progress of the EV along the track depending on $q_s > 0$, where s is the longitudinal position along the track, initialized as $s_0 = s(\boldsymbol{\xi}_t)$ based on the current state of the EV and predicted based on the predicted longitudinal velocity $v_{s,k}$ of the EV (5.10c), as in the baseline [124]. Alternatively, s could be handled as an independent optimization variable linked to the state [180]. The cost function is a convex combination of the reference tracking

term and of the original racing term, ruled by parameter α. If $\alpha = 0$, the active exploration mechanism is completely disregarded and the cost function coincides with that from the baseline work [124], allowing a close comparison.

The EV dynamics (5.10b) is modeled as the dynamic bicycle model without compensation, as in this scenario, we focus exclusively on the uncertainty introduced by the unknown policy of the opponent to allow a close comparison with [124]. Constraints (5.10d) and (5.10e) enforce track boundary and input constraints, respectively. Constraint (5.10f) enforces collision avoidance with the opponent, whose predicted state $\boldsymbol{\xi}_k^{\mathrm{O}}$ at step k is the GP prediction. The EV state reference $\boldsymbol{\xi}_k^{\mathrm{ref}}$ also depends on the predicted opponent state $\boldsymbol{\xi}_k^{\mathrm{O}}$ and on the selected GP input feature reference $\boldsymbol{z}^{\mathrm{ref}}$ through the procedure outlined earlier in this subsection. Collision avoidance constraints also take the uncertainty of the prediction into account, as explained in the following.

2) Probabilistic Collision Avoidance Constraints Collision avoidance constraints in (5.10f) consist of ellipsoidal regions around the predicted positions of the opponent that the EV must not enter. At first, the minimum covering ellipse given the physical dimensions of the opponent is considered; then, the ellipse is expanded by considering the uncertainty about the prediction of the opponent in longitudinal and lateral directions. Observe that the posterior covariance provided by the GP is fundamental to expanding the forbidden ellipsoidal regions. Finally, the constraints are implemented as soft constraints [124, Section IV], to allow for small violations of the expanded ellipsoidal regions if this yields a significant advantage in terms of performance, although such violation is disincentivized. More details on the collision-avoidance constraints are reported in [124, Section IV]. It is worth observing that, although the quadratic collision avoidance constraints make the optimal control problem non-convex, the solution is obtained efficiently using the `FORCESPRO` software [181].

3) Challenges and Limitations in Head-to-Head Racing Relying on the target GP input feature selected via the algorithm used in the time trial might not result in sufficiently diverse data. In fact, the opponent's future moves are not controlled by the EV, therefore the EV cannot arbitrarily enforce the future configuration of the two racing vehicles. It is possible that, while the EV is attempting to reach a given traffic configuration, the opponent reacts in a way to counterbalance the movement of the EV, and the configuration of the two reaches an equilibrium. In this case, although the absolute position of the two vehicles changes, the relative position does not change. If such equilibrium is reached and the change in the relative configuration is smaller than a threshold for several consecutive steps, we heuristically modify the algorithm to encourage the exploration of GP input features corresponding to configurations of the two agents that are different from the current configuration by reversing vector $\boldsymbol{D}_i$ in the algorithm for target GP input feature selection, with a view to breaking the stalemate. Furthermore, the target GP input feature is not updated for a few iterations to avoid reaching the same equilibrium.

Also, it is worth noting that, other than the GP input features that depend on both the EV and the opponent, namely the longitudinal distance, $s^{\mathrm{O}} - s$, and the lateral distance, $d^{\mathrm{O}} - d$, there exist several GP input features over which the EV has no control, i.e., the opponent's yaw angle and lateral velocity and the curvature of the look-ahead points. Exploration cannot be encouraged for such features. Therefore, when generating the list of candidate GP input features $\{\boldsymbol{z}_{\mathrm{O}}^{(i)}\}_i^{n_{\mathrm{G}}}$, we consider features that differ only in the components over which the EV

has influence. Otherwise, the algorithm might choose a target GP input feature $\boldsymbol{z}_{\mathrm{O}}^{\mathrm{ref}}$ because of the high posterior covariance given by the curvature value, for example, which the EV cannot impose, and possibly resulting in a relative configuration for which the posterior covariance is already small. Focusing only on the input features over which the EV has influence is beneficial also to limit the number of candidate target GP input features n_{G}.

4) Iterative Framework in Head-to-Head Racing Finally, we outline how our iterative scheme works for the scenario with the opponent. At first, an initial GP model is used during the exploration phase. At this stage, it is not required that the GP model is accurate since it will improve in the training after the exploration phase. Nevertheless, a GP model is needed, as the regions of the feature space that must be explored are chosen using the posterior covariance. Therefore, a coarse model is trained at first, possibly from data collected with other opponents, and therefore not tailored for the current opponent.

Then, the exploration phase takes place, which starts with a high value of α in the EV optimal control problem (5.10). The goal of this phase is a trade-off between winning the race and collecting a variety of informative data points about the opponent's policy in reaction to several attempts of the EV. At the end of the exploration phase, which can last for a few minutes, the GP is retrained on a remote platform, while the EV continues the competition. As soon as the training of the updated GP model has been completed, the updated GP model is transferred to the EV, which now can leverage an accurate prediction of the opponent's behavior and focus on winning the race, that is, setting parameter $\alpha = 0$ in problem (5.10).

5.2.3. Simulations of Autonomous Racing Scenarios using Active Exploration

In this section, we describe the simulations that were conducted to validate our iterative GP regression framework with active exploration mechanisms in both autonomous racing scenarios. In the time trial, we compare performance with previous work [113], where a Double iterative GP regression framework is used without active exploration. We show that our approach yields an improvement in the minimum lap time as a result of the more accurate learning performance that the enriched dataset from the exploration allows. Then, in the challenge with the opponent, we compare our approach with the approach from [124], in which a GP predictor of the future trajectory of the opponent is trained on a large dataset of shorter runs. Notably, our approach results in an improvement in the average EV performance as a consequence of the improved prediction of the opponent for further prediction steps, although our approach relies on a significantly smaller dataset of measurements collected during a single phase of exploration.

Time Trial

We used the same simulation setup as in previous work [113] for an accurate and fair comparison. The closed-loop simulations were carried out in the highly realistic racing simulation platform Gran Turismo Sport from Sony Interactive Entertainment Inc [182], using as EV the Audi TT Cup running on the Tokyo Expressway Central Outer Loop Track. The desktop computer wired connected to the Play Station 5 is an Alienware-R13, with CPU Intel i9-12900 and GPU Nvidia 3090. Our code is developed in Python. The QP MPC optimal

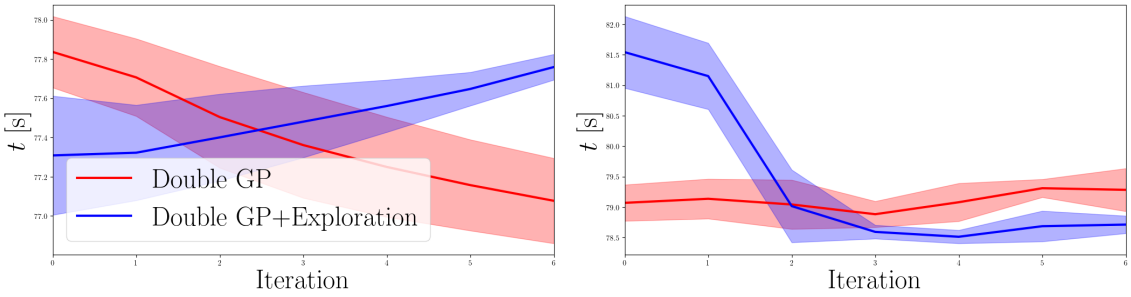
Table 5.1.: Parameters of Audi TT Cup in GTS

Parameter	Value
Total mass m	1161.25 kg
Length from CoG to front wheel l_f	1.0234 m
Length from CoG to rear wheel l_r	1.4826 m
Width of chassis	1.983 m
Height of CoG h_c	0.5136 m
Friction ratio μ	1.5
Wind drag coefficient C_{xw}	0.1412 kg/m
Moment of inertia I_{zz}	2106.9543 Nm

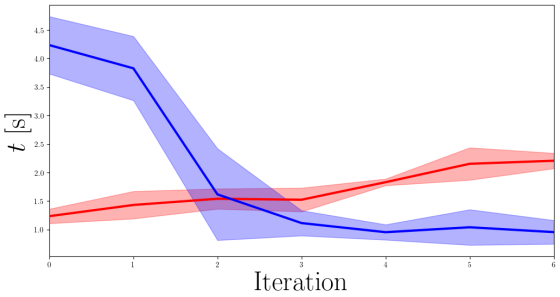
control problem (5.6) is solved using `qpsolvers` [183]. The MPC frequency is 20 Hz and the prediction horizon $N = 20$. The vehicle parameters are reported in Table 5.1. The GP regression is implemented using `GPyTorch` [184], which exploits the GPU and adopts an efficient and general approximation of GPs based on black-box matrix-matrix multiplication.

In our implementation, the maximum size of the dataset is $M = 2000$. The risk parameter in the tightened constraints (5.7) is $\beta = 0.6$. It is important to observe that since the target GP input feature is changed at each iteration of the MPC algorithm, we collect data points also about sudden changes in the features, which are relevant for the GP model in the planning problem. To speed up the target GP input feature selection online, we evaluate and store the covariance of each point in the list of candidate features $\{\boldsymbol{z}^{(i)}\}_i^{n_\mathrm{G}}$ before the start of each trial, after retraining the GP. Inspired by the discussion on the value of α provided in the previous work [137], during the first two iterations, we encourage the exploration setting $\alpha_0 = \frac{6}{7} = 0.857$ and $\alpha_1 = \frac{5}{7} = 0.714$. These values were chosen to linearly decrease α to zero in 7 steps, to completely stop the exploration mechanism in 7 rounds. However, the results suggested that after just a few rounds of exploration, the dataset of selected diverse measurements would not change significantly because the collected measurements are sufficiently diverse, making further exploration not helpful. Therefore, from the third iteration, α is set to zero, instead of gradually decreasing it to zero, that is, from iteration number 2, the focus is exclusively on minimizing the lap time.

In the first run, the EV uses the nominal MPC, without the GP compensation, to track a curvature-optimal path [110], and the measurements are used to train the two GPs. In the following iterations, the GPs are exploited and the time-optimal path is re-planned and tracked. The planning problem is warm-started with the planned trajectory from the previous iteration to discourage large deviations from the previous trajectory since this could result in planning infeasible trajectories. Furthermore, we have heuristically observed that the optimal planned trajectory does not improve significantly after the first iteration of the optimization, therefore we only run one iteration. The iterative framework has been repeated for 7 iterations. Because of small uncertainties in the timing of the communication network between the computer implementing our algorithm and the simulation environment in the Play Station, slight variations are observed when the same simulation is repeated. Thus, we have repeated each simulation three times, considering the mean of the measured times and the standard deviation between the three trials.



(a) Time of the planned path. © 2024 IEEE. (b) Time measured during closed-loop test. © 2024 IEEE.



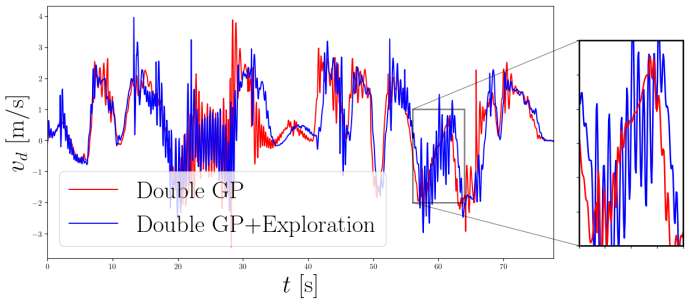
(c) Gap between planned and measured. © 2024 IEEE.

Figure 5.1.: Lap times obtained in each iteration in the time trial simulations. Each simulation has been repeated for three trials: solid lines indicate the mean over the three trials, and the shaded areas represent the standard deviation across the three trials. At each iteration, Double GP+Exploration uses $\alpha_0 = 6/7$, $\alpha_1 = 5/7$, and $\alpha_{2+} = 0$ to transition from high exploration to no exploration.

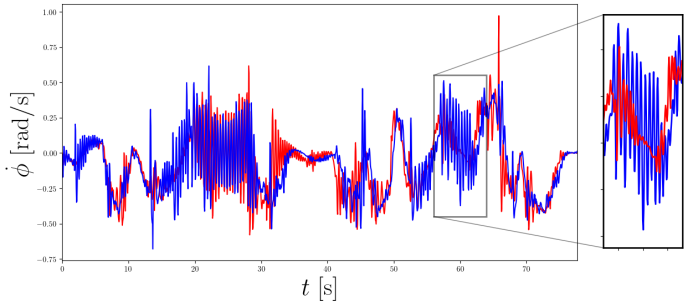
First, we discuss the advantages of our method evaluating the lap time of the optimal planned trajectory and the recorded lap time during the experiments, since the compensates the nominal dynamics both at run time and in the planning phase. Figure 5.1 shows the lap times obtained with the comparison with the Double GP method, the previous approach [113], and our proposed Double GP+Exploration which includes our active exploration method in Algorithm 1. For convenience, iteration 0 is the first trial in which the compensation is used, that is, neglecting the curvature-optimal run. As shown in Figure 5.1a, in our approach, the time of the optimal planned path increases over the iterations, although the dispersion within repeated simulations decreases. This is understood as a consequence of the improvement in learning performance. In fact, the goal of the path planner is to derive the optimal path that is feasible for the actual dynamics of the EV, therefore it is reasonable that a more accurate dynamics results in an optimal planned path with higher lap time. Considering the actual lap time measured at each iteration, reported in Figure 5.1b, we first observe a significant increase in the lap time yielded by our algorithm, which is due to the fact that in the first two iterations, the exploration takes place, and therefore the performance objectives are partially compromised to collect diverse measurements. Yet, from iteration 2, the EV focuses on minimizing the lap time and both the minimum time and the deviation between repeated simulations decrease compared to the baseline [113]. Finally, the difference between the lap time of the planned path and the actual lap time of

the run, Figure 5.1c, shows that the diverse dataset yielded by active exploration reduces the gap between the time of the planned path and the actually achieved minimum time.

Figure 5.2 presents the data collected in the first iteration. Both from the signal of the longitudinal velocity v_d, Figure 5.2a, and the signal of the derivative of the yaw angle $\dot{\phi}$, Figure 5.2b, we observe that during the run the EV dynamics is tested by repeatedly deviating from the trajectory of the planned path.



(a) Data collected for v_d. © 2024 IEEE.



(b) Data collected for $\dot{\phi}$. © 2024 IEEE.

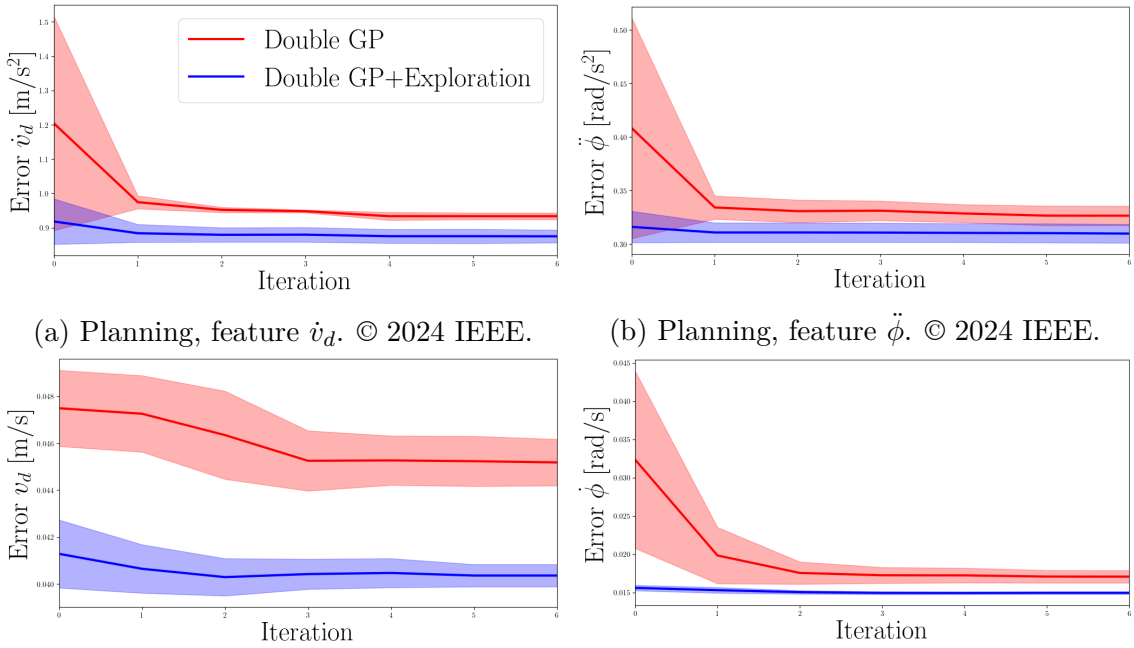
Figure 5.2.: State analysis of the time trial simulations in Gran Turismo Sport. Data was collected in the first iteration, in which the EV dynamics is tested by repeatedly deviating from the trajectory of the planned path ($\alpha_0 = 6/7$).

Finally, we have evaluated the learning performance over the iterations, both for the used in the planning and for the used in the tracking phase. The evaluation, shown in Figure 5.3, investigates the prediction error over a dataset of diverse measurements, collected during a run in which the EV repeatedly deviated from the center line of the track. With respect to the previous work [113], the diverse dataset collected during the active exploration allows a reduction of the prediction error for both the used in the planning and the used in the tracking phase, confirming that the improved performance in the measured lap time is a consequence of increased prediction accuracy.

Head-to-head Racing

For the competition against the opponent, we use the simulation setup from [124], which implements a racing environment for miniature racing cars[1]. The optimal control problem (5.10) is solved using the `FORCESPRO` software [181]. The control frequency is 10 Hz, and

[1]https://github.com/MPC-Berkeley/gp-opponent-prediction-models.



(a) Planning, feature $\dot{v}_d$. © 2024 IEEE.

(b) Planning, feature $\ddot{\phi}$. © 2024 IEEE.

(c) MPC tracking, feature v_d. © 2024 IEEE.

(d) MPC tracking, feature $\dot{\phi}$. © 2024 IEEE.

Figure 5.3.: Prediction error between compensated model and data of the dynamics, for each iteration in time-trial task. Each simulation has been repeated for three trials: solid lines indicate the mean, and the shaded areas represent the standard deviation across the three trials.

the prediction horizon is $N = 10$. All simulations are run on a laptop with an AMD Ryzen 5 3500U eight-core processor.

Vector $\boldsymbol{\kappa}$ in the GP input feature (5.5) consists of the curvature at three look-ahead points, to facilitate the comparison with [124]. In all simulations, the opponent is implemented as an MPC-controlled agent with a blocking policy [122]. Other than performance-based objectives, the cost function of the opponent penalizes deviations from the current lateral position on the track of the EV, so that the opponent "mirrors" the EV lateral behavior and blocks overtaking attempts. To encourage overtaking attempts of the EV, the parameter q_s ruling the progress maximization reward of the EV in (5.10a) is set higher than for the TV. Further details are given in [124].

We compare two methods for modeling the opponent with the GP. In the baseline [124], the GP model for the opponent is trained using closed-loop trajectories from an offline dataset of 500 runs in which the EV starts behind the opponent on randomly generated tracks. In contrast, our proposed Data Selection + Exploration method uses the iterative and exploration-based approach presented in Section 5.2.2. For the initial GP model, we use a *smaller* initial offline dataset of 20 runs, generated with the same mechanism as the baseline method. This allows us to test how well the GP model can be improved during the exploration. The exploration lasts 10 minutes of simulation time and is run on the closed track provided by [124]. During the exploration, we set $\alpha = 0.9$, so that the EV focuses primarily on testing the opponent's reaction to several EV movements. We simulate only one exploration and retrain, thus the GP model is retrained only at the end of the exploration. Then, the parameter we set $\alpha = 0$, i.e., the EV focuses exclusively on winning the race.

Table 5.2.: Head-to-head racing results over 100 tracks

GP type	Hit border	Average overtaking time mean±std [s]	Prediction error mean±std [m] 1-step-ahead	9-step-ahead
Baseline [124]	2	12.772±4.353	0.003±0.003	0.119±0.119
Data Selection + Exploration	5	12.442±5.041	0.004±0.004	0.055±0.055
Data Selection[1]	/	/	0.005±0.005	0.131±0.127

[1] The Data Selection GP predictor is only used for the offline analysis of the prediction accuracy. Thus, we only report the average prediction errors computed in the offline analysis with respect to the closed-loop opponent trajectories.

In order to test each method, we randomly generate a set of 100 scenarios. A track is randomly generated from straight, curved, and chicane stretches, and the length and curvature of each stretch are randomly selected. For each track, the initial longitudinal position and velocity of the cars are randomly generated, but the EV is always behind, to test the overtaking ability. Each simulation is interrupted 1.5 seconds after overtaking occurs, or when the EV reaches the track end.

We summarize the results in Table 5.2. In 100 simulations, no major collision is observed—the EV never leaves the track or crashes during the trials. Nevertheless, the EV hits the track border in 2 simulations when the GP predictor from [124] is used, and 5 times when using our GP trained with the exploration data. These minor collisions could be prevented by adding safety margins to the constraints, at the price of compromising the performance. For the average overtaking time, we consider the simulations on the 93 tracks in which the EV stays strictly inside the track boundaries with both predictors. On average, the EV overtakes the opponent 0.33 seconds earlier when using the GP trained on the exploration data compared to the baseline approach. It should be observed that this improvement in the EV performance is achieved with a significantly smaller training dataset, that is, roughly 600 data points collected during the exploration, as opposed to the dataset of sample runs used to train the baseline GP [124], consisting of roughly 5000 data points.

Finally, we analyze the GP prediction accuracy to test how well the data selection and exploration mechanisms reduce model error. We compare the methods to a third GP, the purely Data Selection method, which trains a GP on a small dataset of the most diverse measurements within the dataset used for the baseline GP, therefore without exploration. The dataset is selected using the procedure outlined in Section 5.2.2. We evaluate the impact of employing only the most diverse data points collected within several runs, without employing the active exploration mechanism in the data collection. We assess model accuracy by comparing the prediction error of the lateral position of the opponent, which is of primary concern for overtaking maneuvers. We perform the analysis offline using the 200 closed-loop trajectories of the opponent collected from the simulations on the 100 tracks in which the EV first uses the baseline GP and then our GP with Data Selection and Exploration. Since the trajectory of the opponent depends on the EV's own behavior, we repeat the prediction offline using data from all 200 trajectories for all three GP predictors, for a fair comparison.

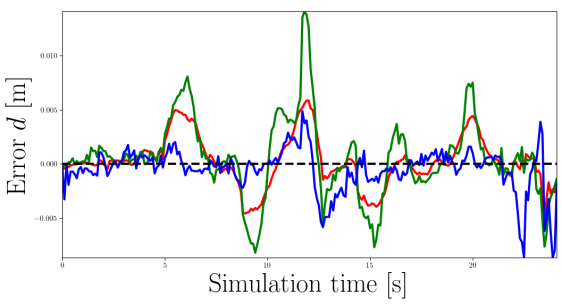
The analysis of the accuracy is shown in Figure 5.4, using the data from one of the 200 closed-loop opponent trajectories. Furthermore, on the right of Table 5.2 we report the average results over all 200 closed-loop trajectories. Each GP method has a comparable accuracy in the 1-step-ahead prediction of the lateral position of the opponent, shown in Figure 5.4a. However, there are occasional spikes in prediction error, especially with the GP only using Data Selection to improve dataset diversity. This likely indicates that exploration is necessary to improve the diversity of the dataset and, thus, the accuracy of the GP.

While 1-step prediction accuracy is important, accurately predicting the opponent's behavior over *long* time horizons is also very important, given the fact that the EV's behavior is predicted using an N-step prediction horizon in the MPC. Thus, we compare each method's 9-step prediction accuracy in Figure 5.4b. Our proposed GP with data selection and exploration significantly outperforms the other methods, resulting in a smaller 9-step prediction error compared to the two other predictors. In fact, comparing the average k-step prediction error in Figure 5.4c as a function of the number of steps in the horizon k, we see that the exploration mechanism decreases modeling error compared to the baseline GP or Data Selection alone. As expected, each method's accuracy deteriorates for further prediction steps; however, our exploration-based GP results in the slowest increase in the mean and standard deviation of prediction error as the prediction horizon increases. Since the baseline GP and GP with Data Selection achieve similar modeling performance, this indicates that the exploration mechanism can make a notable improvement in the training dataset by purposefully opting to collect data in regions with greater modeling uncertainty. Thus, with the greater long-step prediction accuracy of the opponent's model, our GP with Data Selection and Exploration improves the performance of the EV, since the strategic decisions especially rely on the prediction for further steps.

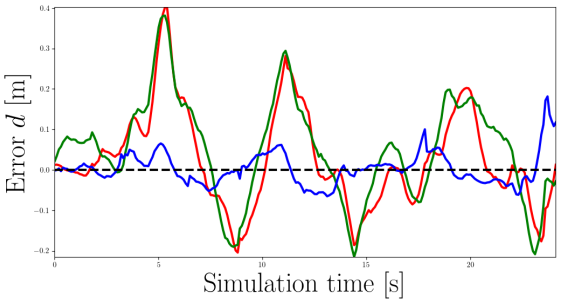
5.2.4. Discussion

An important aspect regarding active exploration, especially in head-to-head racing, is the criterion to terminate the exploration phase. The decision should be grounded on the improvement in GP's prediction accuracy. However, assessing the prediction accuracy requires retraining the GP with the updated dataset, which is time-consuming and, although it could be possible in principle between different runs for the time trial, it is not feasible in real time for head-to-head racing. We instead assess the diversity of collected measurements. During operation, the dataset is incrementally extended with new measurements following the procedure outlined in Section 5.2.2. In practice, the dataset reaches a steady state after a few minutes of exploration, then most new measurements are discarded as they no longer enhance data diversity. Thus, the data update frequency serves as a good heuristic to determine when to conclude the exploration phase. The update rate is also used to guide the systematic scheduling of the exploration weight, α, during exploration: higher rates suggest that the dataset does not well represent the feature space, thus exploration should be encouraged; instead, lower rates suggest the dataset is sufficiently diverse, thus exploration should be discouraged, and α should gradually decay to zero to focus on winning the race.

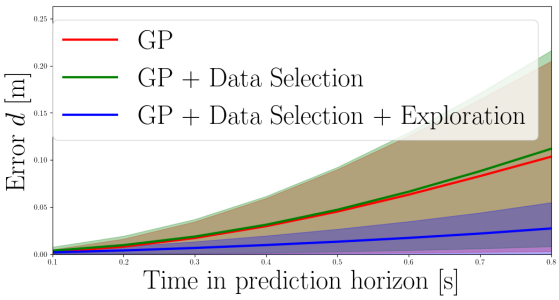
Potentially, multiple rounds of exploration with more fine-grained data update strategy and α scheduling can be investigated to further optimize the exploration procedure. However, it should be noted that the exploration and following training phase must not take too long with respect to the duration of the competition. Otherwise, too little time remains to focus



(a) 1-step-ahead prediction. © 2024 IEEE.



(b) 9-step-ahead prediction error. © 2024 IEEE.



(c) Average prediction error k-seconds-ahead. Solid lines indicate the mean, the shaded areas the standard deviation. © 2024 IEEE.

Figure 5.4.: Prediction error of the opponent lateral position d with respect to the same closed-loop trajectory from one of the closed-loop trajectories opponent.

on taking advantage of the improved model to win the race. Our simulation results show that a single phase of exploration with the simple termination strategy described above already yields a significant improvement, as we discuss in the simulations of head-to-head racing.

Furthermore, the framework proposed in this work implicitly assumes that the behavior of the opponent and its reaction to the EV is stationary—which is a common assumption in literature [124]—and thus can be accurately learned with data collected in finite exploration time. If the opponent is similarly featured with exploration capabilities of the EV behavior, this might not be the case, since the opponent policy will keep changing. One possibility to deal with a time-varying opponent policy is to run several phases of exploration, then exploitation, and re-exploration when the prediction error grows again. However, in the presence of an opponent with symmetric exploration capabilities, this strategy might result in a deadlock. A thorough discussion and a practical solution to this problem are beyond the scope of this work, which we leave as potential future extensions.

5.3. Conclusion

In this chapter, the problem of learning the residual uncertainty, generated by unmodeled effects, was addressed considering the application of autonomous racing. Two types of uncertainty were considered: the uncertainty in the system, caused by imprecise modeling of the vehicle dynamics, and uncertainty in the surrounding environment, caused by the unknown reaction of the opponent to the controlled vehicle's own behavior. With a view at boosting for accuracy in the compensation of unmodeled phenomena, we presented an iterative GP regression scheme for autonomous racing implementing an active exploration mechanism. During the first iterations, the EV trajectory is planned trying to collect measurements for the states with high posterior covariance. Among the collected measurements, a smaller dataset is obtained, selecting the most diverse data points, and is used to retrain the GP model. Then, in further iterations of the algorithm, the focus is exclusively on improving the performance of the EV, leveraging the improved prediction accuracy. We showed that the GP exploration method can be applied both when the GP model is used for error compensation and for opponent modeling. We tested the framework to compensate for both types of mentioned uncertainty. In both scenarios, we obtained significant improvements in the prediction accuracy and, consequently, in the EV performance.

6. Conclusion

Recent technological advances have greatly expanded the scope of control algorithms, resulting in more frequent interaction with humans. Especially in safety-critical applications, it is important to find a trade-off between ensuring safety without overly limiting the potential of automation. Interaction with humans is particularly challenging because the intent is unknown and may result in significantly different future behaviors. To reduce the overall level of uncertainty, it is beneficial to identify and treat different trends separately whenever possible. However, such multi-modal uncertainty poses several challenges both in the estimation and prediction of the unknown phenomenon and in the subsequent action planning for the controlled agent.

This thesis focuses on improvements in the handling of multi-modal uncertainty, considering both the description of the uncertainty and estimation of the modes, as well as the control design problem. Several examples of automated driving were considered, as this is one of the most impactful and rapidly growing safety-critical applications.

6.1. Summary of Contributions

After discussing the related work in the field of MPC-based schemes to handle uncertainty and especially multi-modal uncertainty and introducing an efficient scheme for constraint generation schemes in Chapter 2, the main contributions of the thesis are outlined in the following three chapters. In the following, we reconsider them in light of the challenges articulated in Section 1.1.

Chapter 3 - Effective Description of Multi-Modal Uncertainty. We proposed two novel algorithms which provide an effective description of multi-modal uncertainty, addressing challenge 1.

The first algorithm allows to combine individual assessments on the uncertainty mode derived from multiple sensors. Our novel approach leverages the BFT framework to explicitly quantify the reliability of the estimation. Rather than averaging the estimations provided by several sources, in which difference in the assessments would be compensated, we analyze possible conflicts between individual estimations and increase the quantification of the uncertain in the estimation. This approach overcome issues that arise when highly uncertain environments are considered, causing estimation on the mode to rapidly and repetitively change, which makes the estimation of little use for decision making purposes. Although applied to the problem of intention recognition of DOs, the approach is well suited for other information fusion scenarios, in which the reliability of the information is of primary importance, such fault detection for predictive maintenance, considering the multiple sensors, and

diagnosis of health conditions, with specific attention to diagnostic uncertainty [4] due to aspecific symptoms and noisy measurements.

In the second proposed approach a combination of the IMM algorithm and MLE is used to obtain models representing different candidate uncertainty modes and tuning the parameters to optimally fit the recent data. Furthermore, mode probabilities are jointly estimated. We discussed two approaches to overcome overfitting problems that can arise when too short data sequences are used. In practice, this is an important aspect, as using only the most recent data allows to promptly identify mode changes, and to maintain the computational burden limited. The method is applicable to other fields in which distinct uncertainty modes can be recognized and must be iteratively estimated, for example in design of instant models for demand prediction in energy systems and fault modeling for predictive maintenance.

Chapter 4 - Safe and Non-Conservative SMPC-Based Planning with Multi-modal Uncertainty. We proposed several novel algorithms well suited to plan the future trajectory of the controlled system balancing between safety and efficiency in highly uncertain environments, addressing challenge 2. We proposed a framework to include long-term reasoning in the planning of the controlled system exploiting a hierarchical SMPC structure. Explicitly distinguishing between short-term and long-term decisions of the controlled agent is beneficial in presence of high uncertainty.

Then, we proposed two algorithm to balance between safety and efficiency in planning the behavior of the controlled agent in presence of multi-modal uncertainty. Constraints are designed in a SMPC fashion for several candidate modes, allowing a high probability of constraint violation for those modes that are currently considered unlikely. In doing so, a balanced control is achieved, in which multiple modes of the uncertainty are considered, while the exaggerated conservatism caused by accounting for (currently) unrealistic scenarios is avoided. Then, we advanced the approach with a new algorithm, in which the uncertainty about the estimated mode probabilities is addressed. The BFT description of the mode probabilities allows to design constraint tightening mechanisms which prevent risky decision making if the information is not reliable, without introducing conservatism if the information can be trusted.

Finally, we proposed an approach in which the benefits of the optimistic SMPC planning are compounded with safety guarantees. The optimistic solution is only used if it is previously checked that it will not lead the system to unsafe states. However, given that such safety certifications rely on the worst-case assumptions on the uncertainty, we addressed the case in which the uncertainty realization violates the assumed limits. In such cases, our algorithm computes the control input with a view at minimizing the probability of constraint violation, systematically reacting to the unanticipated occurrence. Although the algorithms have been developed with the autonomous driving application in mind, the approaches can be applied to different settings, such as human-robot collaboration [1], optimization of balance in energy systems [2], and dynamic allocation of the financial risk [3]. In particular, the systematic criterion to react to unanticipated occurrences of the uncertainty can be set to risk minimization, with a view to considering the severity of the violation, rather than just the probability of the occurrence.

Chapter 5 - Addressing Unmodeled Uncertainty through Active Exploration. We addressed challenge 3, that is, we discussed the data collection problem when machine

learning approaches are used to compensate for inaccuracies of model-based settings. We considered a specific application setting, autonomous racing, and proposed an approach to systematically collect representative data by guiding the system to states where the prediction is more uncertain. We developed frameworks to implement such an approach in multiple challenging racing scenarios, considering both uncertainty in the model of the controlled system and environmental uncertainty. With our active exploration approach the machine learning component is accurate even with a small amount of data. This is especially relevant for applications in which collecting a large amount of data is expensive or might lead to unacceptable risk.

6.2. Future Research Directions

This thesis developed novel research contributions that greatly enhance the applicability of control strategies in presence of high and multi-modal uncertainty. Nonetheless, several aspects require further investigation and some challenges remained unaddressed. We discuss some of these interesting future research directions in the following.

Interaction Models. This thesis investigated in depth the problem of determining a safe and not excessively conservative trajectory for the EV in frameworks with multi-modal uncertainty. It was considered that the uncertainty is characterized by distinct modes, that may have their own varying parameters that must be repeatedly estimated. However, it is assumed that the uncertainty is independent of the behavior of the control action, which is an important limitation. In the autonomous driving application, this is equivalent to assume that DOs do not react to the decision of the EV. However, conservatism can be further reduced by considering the interactive behavior. A human driving on the highway often decides to start a lane changing maneuver even though another vehicle is approaching on the target lane, because it is assumed that the other driver will slow down and permit the lane change, if the distance is large enough. In congested urban scenarios, such approaches are especially important to optimize traffic flow and can be implemented safely, as the low speed allows for almost immediate stopping in risky situations.

However, optimizing the behavior of the EV while considering that DOs will adapt their motion to react to EV's decisions is challenging for two main reasons. On the one hand, the problem of modeling the reaction of humans to the decisions of other agents has been scarcely addressed in the literature [40], with the majority of studies focusing on pure machine learning techniques, such as GPs [124]. This approach is also adopted in Chapter 5 to model the behavior of the opponent. However, machine learning makes it difficult to interpret and analyze the results and giving guarantees is extremely challenging. Therefore, new models should be designed, considering this feature.

On the other hand, a major problem caused by the interaction is that the strict separation between predicting the behavior of other agents and planning the control of the ego agent is no longer possible. In fact, in most approaches from literature, including those presented in this thesis, the future trajectory of DOs is evaluated numerically upstream of the optimal control problem that determines the trajectory of the EV. In doing so, the dimension of the optimization problem is reduced, since the movement of the behavior of traffic participants is not optimized for, and the numerical complexity of game-theoretical approaches is avoided, so that the trajectory of the controlled agents is obtained rapidly at run time. Nevertheless,

including the reaction of other agents in the optimization of the own agent's trajectory would allow to achieve further efficiency, similar to human reasoning. Hence, it is necessary to design novel interaction-aware models that are simple enough not to significantly affect the computational complexity of motion planning for the controlled system.

Unobservable States in Active Exploration. In Chapter 5, the machine learning compensation for the vehicle dynamics is introduced to capture the dynamic effects not modeled by the nominal dynamics used, in this case the dynamic bicycle model. However, the feature space of the machine learning compensation is still defined from the state space of the nominal dynamics. Therefore, certain unmodeled phenomena are still not captured by the compensation due to the missing states. In the example considered, the roll and pitch angles of the vehicle are not considered as features for the GP compensation, although it is well known that they have an influence on the dynamics. On the one hand, it is beneficial to have a small number of features, since it makes the computation less demanding and the requires a smaller amount of data for training. On the other hand, the absence of such features limits the potential of the machine learning compensation.

One potential solution to further enhance the model's accuracy is to account for the missing states adding the history of the observation to the GP input features, so that the model can learn to implicitly infer the unobservable states. However, this approach requires to change the active exploration mechanism, since now the target GP input features to be designed cannot be chosen independently for different time steps.

A. Details on Preparatory Work

This appendix chapter provides additional details for Chapter 2, in which preparatory work and existing approaches are outlined.

A.1. Efficient Constraint Generation Scheme

Here, we present a scheme to systematically generate constraints for motion planning algorithms in environments with several obstacles. The approach was presented in [25].

Motion planning of automated vehicles must guarantee safety, i.e., avoiding collision between the AV and other traffic participants. DOs are particularly challenging because their uncertain future motion must be considered in the planning phase. MPC is often used for trajectory planning because it allows to iteratively update the trajectory online once new information has been collected, rapidly adapting to the changing environment, and because constraints, such as collision avoidance, can be included in the control scheme. The optimal control problem must be solvable efficiently, thus safety constraints design should consider the resulting computational complexity of the numerical optimization.

Most existing approaches propose formulation of safety constraints that result in non-convex optimization, for example exploiting potential fields [185]. However, these approaches are not applicable if several surrounding traffic participants are considered, as they render the set of feasible positions for the ego vehicle non-convex. Another common approach is the derivation of quadratic safety regions (for example ellipses) around each predicted future position of the DO [186], making the optimization problem non-convex and thus significantly increasing the computation time. Therefore, they might not be appropriate for applications with short sampling times or limited computation resources.

To keep the computational effort limited, other approaches employ convex optimization. In [43] the linearization of elliptical constraints is proposed. Although this approach makes the set of feasible positions convex, a linearization of quadratic constraints lacks strategic planning to implicitly suggest an appropriate maneuver, based on the current situation. Other approaches derive convexified zones in which the ego vehicle can drive to avoid collisions with traffic participants [187], but proposed a procedure that is unlikely to scale well enough in presence of several vehicles. The method in [188] explores the best choice of behavior of the controlled vehicle in presence of several DOs in front, making the optimization problem highly complex, by solving several simpler optimization problems in parallel.

In order to keep the computational burden limited, we propose a novel systematic procedure to generate non-conservative linear safety constraints, in which the overall traffic configuration results from the combination of several basic scenarios, where a safety constraint is generated by considering one surrounding DO at a time. In doing so, a relatively

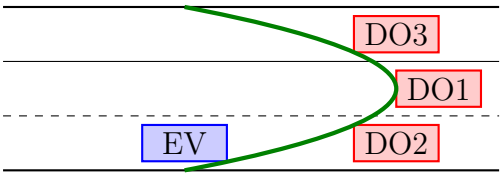


Figure A.1.: Safety constraint considering all DOs at once. © 2022 AVEC.

limited number of possible basic scenarios must be considered and for each an ad-hoc linear constraint is generated, with a view at limiting conservatism by implicitly suggesting the appropriate maneuver to be executed. Eventually, the set of admissible positions for the ego vehicle is convex, resulting from the intersection of linear constraints, and thus the optimization can be performed efficiently. Further, the procedure is scalable by design and additional traffic configurations can be easily implemented by extending the set of basic scenarios.

Designing constraints based on full traffic scenarios, i.e., taking all DOs into account at once (see Figure A.1), is not advisable, since numerous possibilities must be considered separately, depending on the number of DOs and on their relative position. Thus, we generate linear constraints for base cases, i.e., considering the relative configuration of the EV with one DO at a time. Therefore, the number of base cases to be considered is limited. However, new base cases can be easily included, adapting the procedure to cover new scenarios at one's convenience, and complex traffic configurations are systematically handled as the combination of several base cases. Furthermore, although suitable for MPC, the procedure is independent of the control algorithm applied, and other techniques, e.g., Reinforcement Learning, can profit from such a constraint formulation.

Linear Safety Constraints

Here, we outline our proposal for linear safety constraints generation, enforcing collision avoidance of the EV with DOs. We assume that the motion planning algorithm uses the road-aligned coordinates, rather than the Cartesian coordinates. Instead of information about the absolute position of the traffic participants, their location within the so-called Frenet frame [189], centered on the road, is used.

Our method addresses the design of constraints given a prediction of the future trajectory of the DOs and their safety area. The approach is independent of the prediction algorithm used, but for each DO the following is needed:

- a sequence of points $(s_k^{\mathrm{DO}}, d_k^{\mathrm{DO}})$ representing the predicted future positions of the DO for every step k in the considered prediction horizon, where s_k is the position along the road and d_k is the lateral displacement;
- a safety shape S representing the area around the DO that the EV must not enter to avoid collisions.

For motion prediction, deterministic models [40], [42] or learning-based models [6], [46], [167] can be used. The predicted trajectory might as well be the result of communication between vehicles. Furthermore, the prediction can also use Cartesian coordinates if the trajectory is projected onto the Frenet frame.

The safety shape must primarily consider the physical dimension of the DO. However, additional margins can be included, accounting for the uncertainty about the prediction

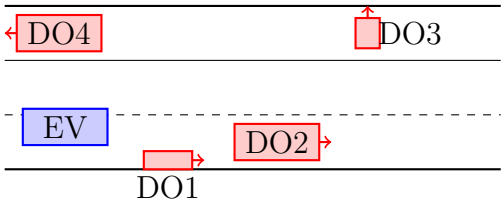


Figure A.2.: Traffic scenario considered in the example.

of the future positions. Furthermore, given that the uncertainty about the prediction in general increases over time, the safety shape might also not be constant, but rather increase for further steps in the prediction horizon.

The aim of the algorithm is to deliver a linear constraint preventing the EV from entering the region S_k, i.e., the safety shape of the DO at time k, centered in the position $(s_k^{\mathrm{DO}}, d_k^{\mathrm{DO}})$, for any of the steps in the prediction horizon. However, the use of linear constraints comes at the price of limiting the possible future moves of the EV. Thus, the design of the constraints must be adapted to the current configuration of the DOs, so that the constraints do not restrict excessively the EV motion, but rather implicitly suggest the appropriate maneuver.

Consequently, the resulting constraints are a (possibly nonlinear) function of the current and predicted position of the DOs, but linear with respect to the future position of the EV. Precisely, the linear safety constraints can be expressed in the form

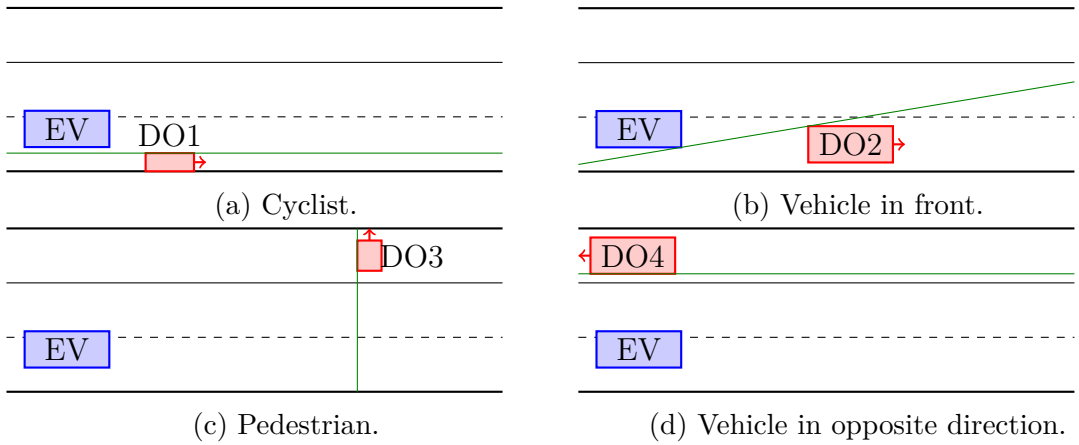
$$\boldsymbol{q}_{s,k} s_k + \boldsymbol{q}_{d,k} d_k \boldsymbol{q}_{t,k} \leq 0, \;\; k = 0, \ldots, N, \tag{A.1}$$

where N is the prediction horizon, (s_k, d_k) is the position of the EV in the Frenet frame at prediction step k, and vector coefficients $\boldsymbol{q}_{s,k}, \boldsymbol{q}_{d,k}$, and $\boldsymbol{q}_{t,k}$ are functions of the current and predicted traffic configuration. The latter can be obtained numerically before the optimization starts, allowing a quick evaluation of the constraint for every new guess of the control sequence during the numerical solution of the optimization problem. The task is then to systematically generate the coefficients $\boldsymbol{q}_{s,k}, \boldsymbol{q}_{d,k}$, and $\boldsymbol{q}_{t,k}$ appropriately, to cover every possible traffic configuration.

The most common traffic scenarios of an urban environment are analyzed and rules to generate a non-conservative linear safety constraint for one DO at a time are derived. The set of admissible positions for the EV that results from all the constraints generated considering each DO individually is convex, since all constraints are linear, thus the computational effort required by the optimization is small and allows for an efficient solution. Complex traffic scenarios result from the combination of base cases. Hence, from a relatively small number of basic configurations the method scales to a possibly high number of DOs.

Composing Scenarios from Basic Configurations

Here we present an example in which basic configurations are used to compose the safety constraints for a traffic scenario scenario including two vehicles, a cyclist, and a jaywalking pedestrian, represented in Figure A.2. Individual basic configurations with one DO at a time are analyzed individually, in parallel. The constraints for each basic example are shown in Figure A.3 and the cases discussed in the following.



(a) Cyclist. (b) Vehicle in front. (c) Pedestrian. (d) Vehicle in opposite direction.

Figure A.3.: Individual constraints for each DO determined through basic configurations.

Cyclist. The cyclist is to the right of the EV, proceeding in the same direction. Given that cyclists are significantly narrower than cars, the EV can overtake without requiring a full lane change. In this case, just a horizontal constraint is generated, requiring the EV to stay sufficiently to the left of the cyclist. Observe that this the constraint is generated for every prediction step separately, based on the predicted position of the preceding vehicle and on its (possibly time-varying) safety area.

Vehicle in front. If to the left of the preceding vehicle there is an available free lane, which the EV can drive, then an inclined constraint is generated. In this situation, the design of the constraint follows a twofold goal. On the one hand, the EV must not enter the safety area about the predicted future positions of the preceding vehicle. On the other hand, the EV must be implicitly pushed to the left lane, initiating an overtaking maneuver. Thus, the inclined constraint requires the predicted positions of the EV to stay to the left of the line connecting the rear left corner of the safety areas of the preceding vehicle and the front right corner of the EV shape at the current position. As a result, the EV is encouraged to move to the left of the lane to maintain a higher speed compared to the preceding vehicle.

Pedestrian. Jaywalking pedestrians pose a serious risk, as they can hazardously cross the intended path of the EV, possibly also outside of signalized intersections and crosswalks. Therefore, when the future positions of the pedestrian are predicted to occupy a neighboring area of the intended path of the EV, a vertical constraint is generated. Thus, the EV is prevented from entering the area that the pedestrian will cross, based on the predicted trajectory and on the pedestrian safety area.

Vehicle in opposite direction. If in the proximity of the EV there is a vehicle driving in the other direction, a horizontal constraint is generated. Such a constraint forces the future positions of the EV to be to the right of the other vehicle, so that the safety area around it is not violated. This constraint is purely meant to enforce collision avoidance and not the traffic rules, e.g., to remain within the lanes for the EV direction. In fact, if the vehicle in the opposite direction is proceeding irregularly and its safety area partially invades the left-most lane in the EV direction, this constraint requires the EV to maintain safety margins.

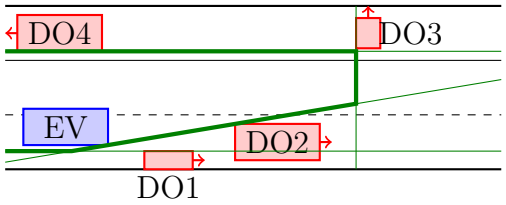


Figure A.4.: Resulting constraints for the whole traffic scenario.

Figure A.4 shows the allowed area for the EV, composed by the intersection of individual constraints. Note that the set of allowed positions for the EV is a convex set, as it results from the combination of linear constraints. We emphasize once more that our proposed method is independent of the algorithms for prediction of the future positions of the DOs and for derivation of the safety shapes, and of the motion planning method.

The safety area around each DO is considered as an input parameter for the constraints generation procedure. It is worth mentioning that in particular for pedestrians the safety area might be significantly larger than the actual physical dimension, accounting for the great variability about the future motion. Furthermore, it is not required that the shape must be a rectangle. In case of different shapes, the linear constraint is generated with respect to the point of the shape being the closest to the initial position of the EV.

Considering directly full traffic scenarios would require storing a large number of different possibilities and furthermore systematically determining convex drivable areas for the EV would prove challenging. By contrast, analyzing complicated traffic scenarios as the composition of individually generated constraints allows to cover a large variety of complex traffic scenarios with a relatively small number of basic individual configurations. This method is more conservative than nonlinear non-convex constraints. However, conservatism is reduced with a strategic design that aims at excluding maneuvers that would not be convenient anyway. Moreover, the coefficients representing constraints in form (A.1) only depend on the predicted future position of the DOs, on their safety areas, and on the *initial* position of the EV. Thus, the coefficients are numerically evaluated outside of the motion planning algorithm, since they do not depend on the predicted position of the EV.

A.2. Explicit Formulation of the Approximated EV Model

Matrices $\boldsymbol{A}_\mathrm{d}$ and $\boldsymbol{B}_\mathrm{d}$ of model (2.15), used for the EV prediction in the OCP, resulting from the linearization of (2.12) about state $\boldsymbol{\xi}^* = [s, d, \phi, v]^\top$ and zero input $\boldsymbol{u}^* = [0, 0]^\top$, and discretization with sampling time T, are given as

$$\boldsymbol{A}_\mathrm{d} = \begin{bmatrix} 1 & \frac{q_6 \cos\phi}{1-k(s)d} & a_{13} & a_{14} \\ 0 & \frac{q_7}{2} & a_{23} & a_{24} \\ 0 & -\frac{k(s) q_6 \cos\phi}{2(1-k(s)d)} & \frac{q_8}{2} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B}_\mathrm{d} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ T & 0 \end{bmatrix}, \tag{A.2a}$$

with $\alpha' = \frac{l_r}{l_f+l_r}$, $\lambda = \frac{vk(s)}{1-k(s)d}$, $\omega = \frac{1-k(s)d}{k(s)l_r} + \sin\phi$, and

$$q_0 = \sqrt{5 - \cos^2(\phi)}, \tag{A.3a}$$
$$q_1 = \sin\phi + q_0, \tag{A.3b}$$
$$q_2 = \sin\phi - q_0, \tag{A.3c}$$
$$e_1 = \exp(\tfrac{\lambda q_1 T}{2}), \tag{A.3d}$$
$$e_2 = \exp(\tfrac{\lambda q_2 T}{2}), \tag{A.3e}$$
$$q_3 = e_1 - e_2, \tag{A.3f}$$
$$q_4 = q_1 e_2 - q_2 e_1, \tag{A.3g}$$
$$q_5 = q_1 e_1 - q_2 e_2, \tag{A.3h}$$
$$q_6 = \tfrac{q_3}{q_0}, \tag{A.3i}$$
$$q_7 = \tfrac{q_4}{q_0}, \tag{A.3j}$$
$$q_8 = \tfrac{q_5}{q_0}, \tag{A.3k}$$
$$q_9 = \tfrac{2}{\lambda q_0}[\tfrac{e_1-1}{q_1} - \tfrac{q_2-1}{q_2}], \tag{A.3l}$$
$$q_{10} = \tfrac{2}{\lambda q_0}[\tfrac{q_1}{q_2}(e_2-1) - \tfrac{q_2}{q_1}(e_1-1)], \tag{A.3m}$$
$$q_{11} = \tfrac{2q_6}{\lambda}, \tag{A.3n}$$

and

$$a_{13} = (1 - \tfrac{q_8}{2})\tfrac{1}{k(s)}, \tag{A.4a}$$
$$a_{23} = \tfrac{(1-k(s)d)q_6\cos\phi}{k(s)}, \tag{A.4b}$$
$$a_{14} = ((1 - \tfrac{q_8}{2q_0})\sin\phi + q_6)\tfrac{1}{vk(s)\cos\phi}, \tag{A.4c}$$
$$a_{24} = (q_6 \sin\phi\cos^2\phi + \tfrac{q_7}{2} - 1)\tfrac{1}{\lambda\cos^2\phi}, \tag{A.4d}$$
$$a_{34} = ((\tfrac{q_8}{2} - 1)\sin\phi - q_6)\tfrac{1}{v\cos\phi}, \tag{A.4e}$$
$$b_{11} = ((T - \tfrac{q_{11}}{2})\sin\phi + q_9\tfrac{1}{vk(s)\cos\phi}, \tag{A.4f}$$
$$b_{21} = (q_9 \sin\phi\cos^2\phi + \tfrac{q_{10}}{2} - T)\tfrac{1}{\lambda\cos^2\phi}, \tag{A.4g}$$
$$b_{31} = ((\tfrac{q_{11}}{2} - T)\sin\phi - q_9)\tfrac{1}{v\cos\phi}, \tag{A.4h}$$
$$b_{12} = (\cos^2\phi q_9 + (T - \tfrac{q_{11}}{2})\omega - T\sin\phi)\tfrac{v\alpha'}{1-k(s)d}, \tag{A.4i}$$
$$b_{22} = (\tfrac{q_{10}}{2} + q_9\omega)v\alpha'\cos\phi, \tag{A.4j}$$
$$b_{32} = (\tfrac{q_{11}\omega}{2} - q_9\cos^2\phi)\lambda\alpha'. \tag{A.4k}$$

All variables in (A.3) and (A.4) are extended by continuity at possible singularities.

Details on Safe and Non-Conservative SMPC-Based Planning with Multi-modal Uncertainty

B.

This appendix chapter provides additional information for Chapter 4, which addressed safe and non-conservative motion planning in presence of multi-modal uncertainty.

B.1. Details about the CARLA Simulations

All simulations were run on a desktop computer with an AMD Ryzen 7 1700X eight-core processor. We implemented the CARLA client using the CARLA Python library. At each sampling time, the CARLA client receives updated information on the involved agents and runs the code yielding the control input to move the agents. The numerical value of the current position of the DOs used to run the IMM is directly obtained through the CARLA client, whereas no information about the current speed or acceleration of DOs is used. The EV and the TVs are spanned using CARLA's BMW Grandtourer blueprint, the crossbike blueprint is used for the cyclist.

Units are omitted, as all quantities are given in SI units. The server updates the world in CARLA with a frequency of 30. The EV control is updated every $T = 0.2$ and the prediction horizon length is $N = 10$. The parameters of the nonlinear bicycle model are $l_\mathrm{f} = l_\mathrm{r} = 1.9$, and the weighing matrices in cost function (4.19a) of the EV are $\boldsymbol{Q} = \boldsymbol{P} = \mathrm{diag}(0, 1, 1, 1)$, $\boldsymbol{R} = \mathrm{diag}(0.1, 0.1)$, and $\boldsymbol{S} = \mathrm{diag}(0.1, 10)$. Among the two solutions proposed in Section 4.3.3 to mitigate the possible sudden and persistent variability of the optimization framework in transient phases, in the simulations an additional safety constraint for the first step is used. Parameters β_j are chosen equal to the estimated probabilities $\hat{\mu}^{(j)}$, i.e., $g(p) = p$ in (4.16).

The closed-loop models used in the IMM and to predict the future trajectories of the cyclists are obtained through the procedure outlined in Section 2.3.4, where all models share the same input weighing matrix $\boldsymbol{R} = \mathrm{diag}(0.2, 0.2)\ \forall j$, process noise covariance matrix $\boldsymbol{\Sigma^w} = \mathrm{diag}(0.1, 0.5, 0.1, 0.5)$, and measurement noise covariance matrix $\boldsymbol{\Sigma^\nu} = \mathrm{diag}(0.05, 0.05)$. Model-specific state weighing matrices and target states are:

- $\boldsymbol{Q}^\mathrm{A} = \mathrm{diag}(10, 1, 0, 1)$, $\boldsymbol{z}^{*\mathrm{A}} = \left[x_\mathrm{A}, 0, 0, v^\mathrm{DO}\right]^\top$
- $\boldsymbol{Q}^\mathrm{B} = \mathrm{diag}(10, 1, 0, 1)$, $\boldsymbol{z}^{*\mathrm{B}} = \left[x_\mathrm{B}, 0, 0, v^\mathrm{DO}\right]^\top$
- $\boldsymbol{Q}^\mathrm{C} = \mathrm{diag}(0, 10, 0.01, 10)$, $\boldsymbol{z}^{*\mathrm{C}} = \left[0, v^\mathrm{DO}, y_\mathrm{C}, 0\right]^\top$,

where x_{A}, x_{B}, and y_{C} are the position of the center of the sidewalk, of the right lane, and of the lane after the left turn, respectively, and v^{DO} is the current speed of the cyclist. The a-priori switching probability matrix is

$$\boldsymbol{\Pi} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}. \tag{B.1}$$

For the simulations with multiple models consisting of different rates of convergence to the original target states $\boldsymbol{z}^{*\mathrm{A}}, \boldsymbol{z}^{*\mathrm{B}}$, and $\boldsymbol{z}^{*\mathrm{C}}$, the additional trajectories are obtained considering the following additional state weighing matrices:

- $\boldsymbol{Q}^{\mathrm{D}} = \boldsymbol{Q}^{\mathrm{E}} = \mathrm{diag}(1, 1, 0, 10)$
- $\boldsymbol{Q}^{\mathrm{F}} = \mathrm{diag}(0, 10, 0.0001, 1)$
- $\boldsymbol{Q}^{\mathrm{G}} = \boldsymbol{Q}^{\mathrm{H}} = \mathrm{diag}(0.1, 1, 0, 5)$
- $\boldsymbol{Q}^{\mathrm{I}} = \mathrm{diag}(0, 1, 0.1, 1)$

whereas the additional trajectories resulting from different target speeds are obtained using the original state weighing matrices $\boldsymbol{Q}^{\mathrm{A}}, \boldsymbol{Q}^{\mathrm{B}}$, and $\boldsymbol{Q}^{\mathrm{C}}$, and the additional target states:

- $\boldsymbol{z}^{*\mathrm{D}} = \left[x_{\mathrm{A}}, 0, 0, v^{\mathrm{DO}} - 1.38\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{E}} = \left[x_{\mathrm{B}}, 0, 0, v^{\mathrm{DO}} - 1.38\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{F}} = \left[0, v^{\mathrm{DO}} - 1.38, y_{\mathrm{C}}, 0\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{G}} = \left[x_{\mathrm{A}}, 0, 0, v^{\mathrm{DO}} + 1.38\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{H}} = \left[x_{\mathrm{B}}, 0, 0, v^{\mathrm{DO}} + 1.38\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{I}} = \left[0, v^{\mathrm{DO}} + 1.38, y_{\mathrm{C}}, 0\right]^{\top}$.

The LQR approximations for the additional simulations are obtained from $\boldsymbol{Q}^{j} = \mathrm{diag}(0, 1, 10, 1)\ \forall j$ and

- $\boldsymbol{z}^{*\mathrm{A}} = \left[0, v^{\mathrm{TV}} - \Delta v, \bar{y}^{\mathrm{TV}} - 3.5, 0\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{B}} = \left[0, v^{\mathrm{TV}}, \bar{y}^{\mathrm{TV}}, 0\right]^{\top}$
- $\boldsymbol{z}^{*\mathrm{C}} = \left[0, v^{\mathrm{TV}} + \Delta v, \bar{y}^{\mathrm{TV}} + 3.5, 0\right]^{\top}$,

with $\Delta v = 1.39$ and $\bar{y}^{\mathrm{TV}}$ is the center of the current lane.

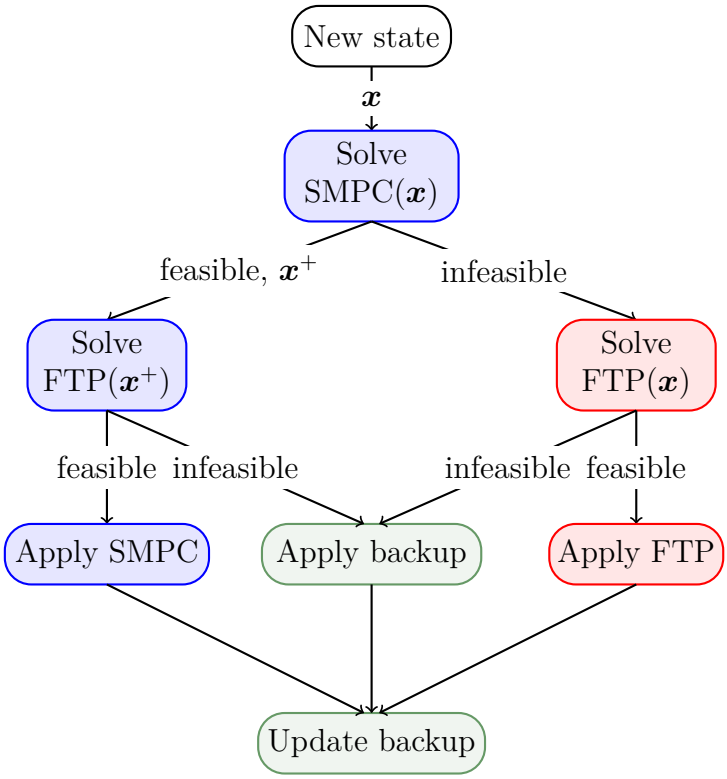


Figure B.1.: Diagram of the previous SMPC+FTP scheme from [30].

B.2. SMPC+FTP Scheme

Here we revise the control scheme introduced in [30]. The procedure is summarized in Figure B.1. Given the current traffic configuration $\boldsymbol{x}$, the SMPC($\boldsymbol{x}$) optimal control problem is solved, formulating safety constraints as probabilistic chance constraints. If the SMPC($\boldsymbol{x}$) optimal control problem admits a solution, the first step of the prediction is considered, that is, the next traffic situation $\boldsymbol{x}^+$ if the solution of SMPC($\boldsymbol{x}$) were applied and the FTP($\boldsymbol{x}^+$) optimal control problem is solved, that is, a robust planner. If FTP($\boldsymbol{x}^+$) admits a solution, then the first step of the SMPC($\boldsymbol{x}$) solution is safe. Then, the first input of the SMPC($\boldsymbol{x}$) solution is applied to the EV and the solution of the FTP($\boldsymbol{x}^+$) problem is stored as a backup for the following iteration. Conversely, if the FTP($\boldsymbol{x}^+$) optimal control problem does not yield a solution, then the SMPC($\boldsymbol{x}$) input is not safe and it is discarded. In this case, the first element of the backup obtained during previous iterations is applied and the tail of the backup is stored as new backup for the following iterations.

If the SMPC optimal control problem from the current traffic configuration SMPC($\boldsymbol{x}$) does not yield a solution, then the FTP optimal control problem starting from the current traffic configuration FTP($\boldsymbol{x}$) is solved. If a solution to FTP($\boldsymbol{x}$) exists, then the first optimal input is applied and the tail of the solution is stored as new backup. Finally, if also FTP($\boldsymbol{x}$) is infeasible, then the first element of the backup obtained from the previous iterations is applied and the tail of the backup is stored as new backup for the following iterations.

In the simulations in Section 4.5.3, the SMPC+FTP scheme [30] is implemented using CVPM robust case for the FTP optimal control problem to facilitate the comparison with the novel SMPC+CVPM scheme.

Notation

Acronyms and Abbreviations

AGV	Automated Guided Vehicle
BFT	Belief Function Theory
CAVs	Connected and Automated Vehicles
CVPM	Constraint Violation Probability Minimization
DO	Dynamic Obstacle
EV	Ego Vehicle
FTP	Fail-safe Trajectory Planning
FDD	Fault Detection and Diagnosis
GP	Gaussian Process
IMM	Interacting Multiple-Model
IMM+SMPC	Interactive Multiple-Model+Stochastic Model Predictive Control
IOC	Inverse Optimal Control
IMPC	Inverse Model Predictive Control
MLE	Maximum Likelihood Estimation
MPE	Model Parameter Estimation
MPC	Model Predictive Control
MM	Multiple Model
PE	Parameter Estimation
POMDP	Partially Observable Markov Decision Process
QP	Quadratic Program
RegSPME	Regularized Simultaneous Parameter and Model Estimation
RevSPME	Reverse Simultaneous Parameter and Model Estimation
RMPC	Robust Model Predictive Control

SPME	Simultaneous Parameter and Model Estimation
SMPC	Stochastic Model Predictive Control
TV	Target Vehicle
WBF	Weighted Belief Fusion

Sets, Conventions, Functions and Operators

$\emptyset$	empty set
$\mathbb{N}$	set of natural numbers
$\mathbb{R}$	set of real numbers
a, A	scalar quantity (lowercase or uppercase)
$\boldsymbol{a}$	vector (bold lowercase)
$\boldsymbol{A}$	matrix (bold uppercase)
$\boldsymbol{A}^\top$	transpose of a matrix
$\boldsymbol{A}^{-1}$	inverse of a matrix
Δ	difference operator
$\|\boldsymbol{a}\|_{\boldsymbol{A}}^2$	weighted norm of a vector, i.e., $\boldsymbol{a}^\top \boldsymbol{A}\boldsymbol{a}$
$\delta(\cdot)$	dirac delta function
$\mathbf{Cov}(\cdot, \cdot)$	covariance of two random vectors
$\Pr(A)$	Probability of event A
$\mathbf{Pl}(A)$	plausibility of event A
$m(\cdot)$	belief mass assignment

Variables

Systems, Control Theory and Stochastic Signals

$\boldsymbol{A}$	state matrix
$\boldsymbol{B}$	input matrix
β	risk parameter in SMPC
$\boldsymbol{\gamma}$	output

J	cost function
$\boldsymbol{K}$	input feedback gain
k	prediction step
$\boldsymbol{\mu}$	mean of a Gaussian random vector
N	prediction horizon
$\boldsymbol{P}$	terminal state weighting matrix
$\boldsymbol{Q}$	state weighting matrix
$\boldsymbol{R}$	input weighting matrix
$\boldsymbol{S}$	input difference weighting matrix
$\boldsymbol{\Sigma}$	covariance matrix of a Gaussian random vector
$\mathcal{S}$	probabilistic state constraint set
t	current time step
T	sampling time
$\boldsymbol{u}$	control input
$\boldsymbol{U}$	control input sequence
$\mathcal{U}$	input constraint set
$\boldsymbol{v}$	output disturbance
$\mathbb{V}$	support of the output disturbance
$\boldsymbol{w}$	state disturbance or uncertainty
$\mathbb{W}$	support of the state disturbance
$\boldsymbol{\xi}$	state
$\boldsymbol{\Xi}$	state sequence
$\mathcal{X}$	state constraint set

Interacting Multiple-Model Algorithm and Maximum Likelihood

$\boldsymbol{\epsilon}$	prediction error
$\boldsymbol{F}$	state matrix
$\boldsymbol{G}$	input matrix
$\boldsymbol{H}$	output matrix
$\boldsymbol{L}$	Kalman gain
$m^{(j)}$	j-th model
$\hat{\boldsymbol{\mu}}$	vector of estimated probabilities
$n_{\mathbf{I}}$	number of intentions
$n_{\mathbf{W}}$	length of the data sequence
$\boldsymbol{\theta}$	parameter vector
$\boldsymbol{\Theta}$	parameter set
$\hat{\boldsymbol{P}}$	state covariance matrix
$\boldsymbol{\Pi}$	matrix of a-priori switching probabilities
$\hat{\boldsymbol{z}}$	state estimate

Vehicles

a	linear acceleration
d	lateral position in Frenet coordinates
δ	steering angle
κ	local curvature
$l_{\mathbf{f}}$	distance of CoM from front axle
$l_{\mathbf{r}}$	distance of CoM from rear axle
$n_{\mathbf{O}}$	number of obstacles
ϕ	yaw angle
s	longitudinal position in Frenet coordinates
v	linear velocity
x	longitudinal position
y	lateral position

Belief Function Theory

b	belief mass
μ	uncertainty parameter
$n_{\mathbf{S}}$	number of sources
θ	hypothesis
$\boldsymbol{\Theta}$	frame of discernment
$\boldsymbol{w}$	opinion

Gaussian Process Regression and Exploration

α	active exploration tuning parameter
$\mathcal{D}$	dataset of measurements
k^a	kernel
M	size of the dataset
$n_{\mathbf{G}}$	number of candidate GP input feature vectors
$\boldsymbol{y}$	GP output feature vector
$\boldsymbol{z}$	GP input feature vector

List of Figures

List of Tables

Bibliography

[1] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal Probabilistic Model-Based Planning for Human-Robot Interaction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3399–3406.

[2] P. Wang, C. Wang, Y. Hu, L. Varga, and W. Wang, "Power Generation Expansion Optimization Model Considering Multi-Scenario Electricity Demand Constraints: A Case Study of Zhejiang Province, China," *Energies*, vol. 11, no. 6, p. 1498, 2018.

[3] N. H. Chan and H.-Y. Wong, *Simulation Techniques in Financial Risk Management*. John Wiley & Sons, 2006.

[4] C. Cox, T. Hatfield, and Z. Fritz, "How and why do doctors communicate diagnostic uncertainty: An experimental vignette study," *Health Expectations*, vol. 27, no. 1, e13957, 2024.

[5] J. J. Montero Jimenez, S. Schwartz, R. Vingerhoeds, B. Grabot, and M. Salaün, "Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics," *Journal of Manufacturing Systems*, vol. 56, pp. 539–557, 2020.

[6] Q. Tran and J. Firl, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *2014 IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 918–923.

[7] H. Woo, Y. Ji, H. Kono, *et al.*, "Lane-Change Detection Based on Vehicle-Trajectory Prediction," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1109–1116, 2017.

[8] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple Object Tracking Using K-Shortest Paths Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[9] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[10] L. Zhao and C. Thorpe, "Stereo- and neural network-based pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148–154, 2000.

[11] X. Song, K. Chen, X. Li, *et al.*, "Pedestrian Trajectory Prediction Based on Deep Convolutional LSTM Network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3285–3302, 2021.

[12] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, Special Issue on Modeling People: Vision-based Understanding of a Person's Shape, Appearance, Movement and Behaviour, vol. 104, no. 2, pp. 90–126, 2006.

[13] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[14] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[15] R. Soloperto, J. Köhler, F. Allgöwer, and M. Müller, "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust Model Predictive Control," in *2019 European Control Conference (ECC)*, 2019, pp. 811–817.

[16] A. Mesbah, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[17] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *International Symposium on Advanced Vehicle Control*, 2014.

[18] T. Benciolini, X. Zhang, D. Wollherr, and M. Leibold, "Information Fusion for Online Estimation of the Behavior of Traffic Participants using Belief Function Theory," *Frontiers in Future Transportation, section Connected Mobility and Automation*, 2023.

[19] A. Daniels, T. Benciolini, D. Wollherr, and M. Leibold, "Adaptive Multi-Model Fault Diagnosis of Dynamic Systems for Motion Tracking," *IEEE Access (accepted)*, 2024.

[20] T. Benciolini, T. Brüdigam, and M. Leibold, "Multistage Stochastic Model Predictive Control for Urban Automated Driving," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2021, pp. 417–423.

[21] T. Benciolini, D. Wollherr, and M. Leibold, "Non-Conservative Trajectory Planning for Automated Vehicles by Estimating Intentions of Dynamic Obstacles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2463–2481, 2023.

[22] T. Benciolini, Y. Yan, D. Wollherr, and M. Leibold, "Combining Belief Function Theory and Stochastic Model Predictive Control for Multi-Modal Uncertainty in Autonomous Driving," in *2024 American Control Conference (ACC)*, 2024, pp. 5042–5048.

[23] T. Benciolini, M. Fink, N. Güzelkaya, D. Wollherr, and M. Leibold, "Safe and Non-Conservative Trajectory Planning for Autonomous Driving Handling Unanticipated Behaviors of Traffic Participants," in *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2024.

[24] T. Benciolini, C. Tang, M. Leibold, C. Weaver, M. Tomizuka, and W. Zhan, "Active Exploration in Iterative Gaussian Process Regression for Uncertainty Modeling in Autonomous Racing," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2024.

[25] T. Benciolini, T. Brüdigam, D. Wollherr, and M. Leibold, "Linear Safety Constraints for Urban Automated Driving: A Case Differentiation," in *International Symposium on Advanced Vehicle Control (AVEC)*, 2022.

[26] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*, A. Garulli and A. Tesi, Eds., Springer, 1999, pp. 207–226.

[27] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, 2009.

[28] S. Dixit, U. Montanaro, M. Dianati, *et al.*, "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.

[29] 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.

[30] 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*, vol. 8, no. 1, pp. 22–36, 2023.

[31] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency Model Predictive Control for Automated Vehicles," in *2019 American Control Conference (ACC)*, 2019, pp. 717–722.

[32] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999.

[33] M. Ono and B. C. 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*, 2008, pp. 3427–3432.

[34] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng, "Explicit use of probabilistic distributions in linear predictive control," in *UKACC International Conference on Control 2010*, 2010, pp. 1–6.

[35] M. Cannon, B. Kouvaritakis, S. V. Raković, and Q. Cheng, "Stochastic Tubes in Model Predictive Control With Probabilistic Constraints," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 194–200, 2011.

[36] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[37] G. C. Calafiore and L. Fagiano, "Robust Model Predictive Control via Scenario Optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, 2013.

[38] 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.

[39] P. Ghorai, A. Eskandarian, Y.-K. Kim, and G. Mehr, "State Estimation and Motion Prediction of Vehicles and Vulnerable Road Users for Cooperative Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 983–17 002, 2022.

[40] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, 2014.

[41] E. Käfer, C. Hermes, C. Wöhler, H. Ritter, and F. Kummert, "Recognition of situation classes at road intersections," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 3960–3965.

[42] S. Atev, G. Miller, and N. Papanikolopoulos, "Clustering of Vehicle Trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 647–657, 2010.

[43] T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr, "Combining Stochastic and Scenario Model Predictive Control to Handle Target Vehicle Uncertainty in an Autonomous Driving Highway Scenario," in *21st International IEEE Conference on Intelligent Transportation Systems (ITSC 2018)*, 2018, pp. 1317–1324.

[44] T. Brüdigam, J. Zhan, D. Wollherr, and M. Leibold, "Collision Avoidance with Stochastic Model Predictive Control for Systems with a Twofold Uncertainty Structure," in *24th International IEEE Conference on Intelligent Transportation Systems (ITSC 2021)*, 2021, pp. 432–438.

[45] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1671–1678.

[46] L. Guo and Y. Jia, "Inverse Model Predictive Control (IMPC) Based Modeling and Prediction of Human-Driven Vehicles in Mixed Traffic," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 501–512, 2021.

[47] G. Tanzmeister, J. Thomas, D. Wollherr, and M. Buss, "Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6090–6095.

[48] G. Tanzmeister and D. Wollherr, "Evidential Grid-Based Tracking and Mapping," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1454–1467, 2017.

[49] Z. Zhou, O. S. Oguz, M. Leibold, and M. Buss, "A General Framework to Increase Safety of Learning Algorithms for Dynamical Systems Based on Region of Attraction Estimation," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1472–1490, 2020.

[50] Z. Zhou, O. S. Oguz, M. Leibold, and M. Buss, "Learning a Low-Dimensional Representation of a Safe Region for Safe Reinforcement Learning on Dynamical Systems," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[51] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor fusion using Dempster-Shafer theory [for context-aware HCI]," in *IEEE Instrumentation and Measurement Technology Conference*, vol. 1, 2002, 7–12 vol.1.

[52] A. Martin, "Conflict Management in Information Fusion with Belief Functions," in *Information Quality in Information Fusion and Decision Making*, ser. Information Fusion and Data Science, É. Bossé and G. L. Rogova, Eds., Springer International Publishing, 2019, pp. 79–97.

[53] A. Jøsang, "Categories of Belief Fusion," *Journal of Advances in Information Fusion*, 2019.

[54] X. Dai and Z. Gao, "From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[55] J. Cen, Z. Yang, X. Liu, J. Xiong, and H. Chen, "A Review of Data-Driven Machinery Fault Diagnosis Using Machine Learning Algorithms," *Journal of Vibration Engineering & Technologies*, vol. 10, no. 7, pp. 2481–2507, 2022.

[56] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[57] T. Ekanayake, D. Dewasurendra, S. Abeyratne, L. Ma, and P. Yarlagadda, "Model-based fault diagnosis and prognosis of dynamic systems: A review," *Procedia Manufacturing*, Digital Manufacturing Transforming Industry Towards Sustainable Growth, vol. 30, pp. 435–442, 2019.

[58] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: Basics and beyond," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3639–3664, 2021.

[59] X. R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic Control*, vol. 41, no. 4, pp. 478–493, 1996.

[60] X. R. Li, X. Zwi, and Y. Zwang, "Multiple-model estimation with variable structure. III. Model-group switching algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 1, pp. 225–241, 1999.

[61] X. R. Li and Y. Zhang, "Multiple-model estimation with variable structure. V. Likely-model set algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 448–466, 2000.

[62] X. R. Li, V. Jilkov, and J. Ru, "Multiple-model estimation with variable structure. VI: Expected-mode augmentation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 853–867, 2005.

[63] J. Ru and X. Rong Li, "Variable-Structure Multiple-Model Approach to Fault Detection, Identification, and Estimation," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 5, pp. 1029–1038, 2008.

[64] Y. Hu, X. Xue, Z. Jin, and C. Sun, "Actuator Fault Diagnosis for Multi-sensor Systems with Asynchronous IMM and MLE," *Journal of Applied Science and Engineering*, vol. 20, no. 4, pp. 533–540, 2017.

[65] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian Prediction by Planning Using Deep Neural Networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5903–5908.

[66] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "A data-driven approach for pedestrian intention estimation," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2607–2612.

[67] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[68] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 797–802.

[69] D. Phillips, T. Wheeler, and M. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1665–1670.

[70] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning Model Predictive Control," in *2017 American Control Conference (ACC)*, 2017, pp. 5115–5120.

[71] 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.

[72] Z. Xu, C. Tang, and M. Tomizuka, "Zero-shot Deep Reinforcement Learning Driving Policy Transfer for Autonomous Vehicles based on Robust Control," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2865–2871.

[73] P. Cai, Y. Sun, Y. Chen, and M. Liu, "Vision-Based Trajectory Planning via Imitation Learning for Autonomous Vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2736–2742.

[74] 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, 2017.

[75] N. Wan, C. Zhang, and A. Vahidi, "Probabilistic Anticipation and Control in Autonomous Car Following," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 30–38, 2019.

[76] 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.

[77] A. Wischnewski, M. Euler, S. Gümüs, and B. Lohmann, "Tube model predictive control for an autonomous race car," *Vehicle System Dynamics*, vol. 60, no. 9, pp. 3151–3173, 2022.

[78] A. Wischnewski, T. Herrmann, F. Werner, and B. Lohmann, "A Tube-MPC Approach to Autonomous Multi-Vehicle Racing on High-Speed Ovals," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 368–378, 2023.

[79] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario Model Predictive Control for Lane Change Assistance and Autonomous Driving on Highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 23–35, 2017.

[80] K. Ren, H. Ahn, and M. Kamgarpour, "Chance-Constrained Trajectory Planning With Multimodal Environmental Uncertainty," *IEEE Control Systems Letters*, vol. 7, pp. 13–18, 2023.

[81] A. Musa, M. Pipicelli, M. Spano, *et al.*, "A Review of Model Predictive Controls Applied to Advanced Driver-Assistance Systems," *Energies*, vol. 14, 2021.

[82] 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)*, 2016, pp. 2763–2768.

[83] T. Brüdigam, J. Teutsch, D. Wollherr, and M. Leibold, "Combined Robust and Stochastic Model Predictive Control for Models of Different Granularity," *IFAC-PapersOnLine*, 21st IFAC World Congress, vol. 53, no. 2, pp. 7123–7129, 2020.

[84] R. C. Shekhar and C. Manzie, "Optimal move blocking strategies for model predictive control," *Automatica*, vol. 61, pp. 27–34, 2015.

[85] S. M. El-Feky, A. M. Zaki, A. M. Bahaa-Eldin, and M. H. El-Shafey, "Improving model predictive controller by using non-uniform sampled model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1862–1868.

[86] M. A. Brdys, M. Grochowski, T. Gminski, K. Konarczak, and M. Drewa, "Hierarchical predictive control of integrated wastewater treatment systems," *Control Engineering Practice*, Special Section on Large Scale Systems, vol. 16, no. 6, pp. 751–767, 2008.

[87] N. D. Van, M. Sualeh, D. Kim, and G.-W. Kim, "A Hierarchical Control System for Autonomous Driving towards Urban Challenges," *Applied Sciences*, vol. 10, no. 10, p. 3543, 2020.

[88] J. Ziegler, P. Bender, M. Schreiber, *et al.*, "Making Bertha Drive—An Autonomous Journey on a Historic Route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[89] Q. Wang, T. Weiskircher, and B. Ayalew, "Hierarchical Hybrid Predictive Control of an Autonomous Road Vehicle," in *ASME 2015 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers Digital Collection, 2016.

[90] U. Rosolia, D. C. Guastella, G. Muscato, and F. Borrelli, "Model Predictive Control in Partially Observable Multi-Modal Discrete Environments," *IEEE Control Systems Letters*, vol. 7, pp. 2161–2166, 2023.

[91] J. Zhou, B. Olofsson, and E. Frisk, "Interaction-Aware Motion Planning for Autonomous Vehicles With Multi-Modal Obstacle Uncertainty Predictions," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1305–1319, 2024.

[92] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, "Joint Multi-Policy Behavior Estimation and Receding-Horizon Trajectory Planning for Automated Urban Driving," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2388–2394.

[93] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic MPC with Multi-modal Predictions for Traffic Intersections," in *25th International IEEE Conference on Intelligent Transportation Systems (ITSC 2022)*, 2022, pp. 635–640.

[94] S. H. Nair, H. Lee, E. Joa, Y. Wang, H. E. Tseng, and F. Borrelli, *Predictive Control for Autonomous Driving with Uncertain, Multi-modal Predictions*, 2023.

[95] L. Heuer, L. Palmieri, A. Rudenko, A. Mannucci, M. Magnusson, and K. O. Arras, "Proactive Model Predictive Control with Multi-Modal Human Motion Prediction in Cluttered Dynamic Environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 229–236.

[96] V. Popescu and S. Nedevschi, "Cut-in maneuver recognition and behavior generation using Bayesian networks and fuzzy logic," in *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*, 2012, pp. 123–130.

[97] R. Adla, Y. A. Bazzi, and N. Al-Holou, "Bayesian network based collision avoidance systems," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*, 2015, pp. 605–610.

[98] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[99] M. Daniel, "On transformations of belief functions to probabilities," *International Journal of Intelligent Systems*, vol. 21, no. 3, pp. 261–282, 2006.

[100] Z. Deng and J. Wang, "A novel decision probability transformation method based on belief interval," *Knowledge-Based Systems*, vol. 208, p. 106 427, 2020.

[101] Ö. Ş. Taş and C. Stiller, "Limited Visibility and Uncertainty Aware Motion Planning for Automated Driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1171–1178.

[102] L. Zhihui, W. Wei, H. Zhen, and X. Jinpeng, "Minimize Collision Risk Global Planner Based Voronoi Graph for Mobile Robot," in *2022 4th International Conference on Robotics and Computer Vision (ICRCV)*, 2022, pp. 360–363.

[103] L. Lützow, Y. Meng, A. C. Armijos, and C. Fan, "Density Planner: Minimizing Collision Risk in Motion Planning with Dynamic Obstacles using Density-based Reachability," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7886–7893.

[104] D. Li, J. Zhang, B. Xiao, B. Tang, and Z. Hu, "Vehicle crash mitigation strategy in unavoidable collision scenarios: Focusing on motion planning by considering a generalized crash severity model," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 44, no. 12, p. 581, 2022.

[105] 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.

[106] M. Fink, T. Brüdigam, D. Wollherr, and M. Leibold, "Constraint Violation Probability Minimization for Norm-Constrained Linear Model Predictive Control," in *2022 European Control Conference (ECC)*, 2022, pp. 839–846.

[107] M. Fink, T. Brüdigam, D. Wollherr, and M. Leibold, "Minimal Constraint Violation Probability in Model Predictive Control for Linear Systems," *IEEE Transactions on Automatic Control*, pp. 1–8, 2024.

[108] R. Rajamani, *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[109] 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)*, 2015, pp. 1094–1099.

[110] C. Hao, C. Tang, E. Bergkvist, *et al.*, *Outracing Human Racers with Model-based Autonomous Racing*, 2022.

[111] H. Pacejka, *Tire and Vehicle Dynamics*. Elsevier, 2005.

[112] A. Jain, M. O'Kelly, P. Chaudhari, and M. Morari, "BayesRace: Learning to race autonomously using prior experience," in *Proceedings of the 2020 Conference on Robot Learning*, PMLR, 2021, pp. 1918–1929.

[113] S. Su, C. Hao, C. Weaver, C. Tang, W. Zhan, and M. Tomizuka, *Double-Iterative Gaussian Process Regression for Modeling Error Compensation in Autonomous Racing*, 2023.

[114] M. Malmström, I. Skog, D. Axehill, and F. Gustafsson, "Uncertainty quantification in neural network classifiers—A local linear approach," *Automatica*, vol. 163, p. 111 563, 2024.

[115] 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)*, 2018, pp. 1341–1348.

[116] 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)*, 2019, pp. 1–6.

[117] Z. Wang, R. Spica, and M. Schwager, "Game Theoretic Motion Planning for Multi-robot Racing," in *Distributed Autonomous Robotic Systems*, N. Correll, M. Schwager, and M. Otte, Eds., ser. Springer Proceedings in Advanced Robotics, Springer International Publishing, 2019, pp. 225–238.

[118] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, "A Game-Theoretic Approach to Replanning-Aware Interactive Scene Prediction and Planning," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3981–3992, 2016.

[119] A. Dreves and M. Gerdts, "A generalized Nash equilibrium approach for optimal control problems of autonomous cars," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 326–342, 2018.

[120] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, "Multi-Hypothesis Interactions in Game-Theoretic Motion Planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8016–8023.

[121] B. Evens, M. Schuurmans, and P. Patrinos, "Learning MPC for Interaction-Aware Autonomous Driving: A Game-Theoretic Approach," in *2022 European Control Conference (ECC)*, 2022, pp. 34–39.

[122] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian Process-based Stochastic Model Predictive Control for Overtaking in Autonomous Racing," in *International Conference on Robotics and Automation (ICRA), Workshop on Opportunities and Challenges with Autonomous Racing*, 2021.

[123] J. L. V. Espinoza, A. Liniger, W. Schwarting, D. Rus, and L. V. Gool, "Deep Interactive Motion Prediction and Planning: Playing Games with Motion Prediction Models," in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, PMLR, 2022, pp. 1006–1019.

[124] E. L. Zhu, F. L. Busch, J. Johnson, and F. Borrelli, "A Gaussian Process Model for Opponent Prediction in Autonomous Racing," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 8186–8191.

[125] M. Li and I. Sethi, "Confidence-based active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[126] R. Rickenbach, J. Köhler, A. Scampicchio, M. N. Zeilinger, and A. Carron, "Active Learning-based Model Predictive Coverage Control," *IEEE Transactions on Automatic Control*, pp. 1–16, 2024.

[127] D. Silver, J. A. Bagnell, and A. Stentz, "Active learning from demonstration for robust autonomous navigation," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 200–207.

[128] I. Abraham and T. D. Murphey, "Active Learning of Dynamics for Data-Driven Control Using Koopman Operators," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1071–1083, 2019.

[129] S. Müller, A. von Rohr, and S. Trimpe, "Local policy search with Bayesian optimization," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 20 708–20 720.

[130] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Recent Advances in Bayesian Optimization," *ACM Computing Surveys*, vol. 55, no. 13s, 287:1–287:36, 2023.

[131] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.

[132] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, "Active learning in robotics: A review of control principles," *Mechatronics*, vol. 77, p. 102 576, 2021.

[133] P. Ren, Y. Xiao, X. Chang, *et al.*, "A Survey of Deep Active Learning," *ACM Computing Surveys*, vol. 54, no. 9, 180:1–180:40, 2021.

[134] J. Park and K. H. Law, "Bayesian Ascent: A Data-Driven Optimization Scheme for Real-Time Control With Application to Wind Farm Power Maximization," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1655–1668, 2016.

[135] S. Bin-Karim, A. Bafandeh, A. Baheri, and C. Vermillion, "Spatiotemporal Optimization Through Gaussian Process-Based Model Predictive Control: A Case Study in Airborne Wind Energy," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 798–805, 2019.

[136] A. Siddiqui, J. Borek, and C. Vermillion, "A Fused Gaussian Process Modeling and Model Predictive Control Framework for Real-Time Path Adaptation of an Airborne Wind Energy System," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 1, pp. 475–482, 2023.

[137] S. Yang, N. Wei, S. Jeon, R. Bencatel, and A. Girard, "Real-time optimal path planning and wind estimation using Gaussian process regression for precision airdrop," in *2017 American Control Conference (ACC)*, 2017, pp. 2582–2587.

[138] A. F. Genovese, "The Interacting Multiple Model Algorithm for Accurate State Estimation of Maneuvering Targets," *J. Hopkins Technical Digest*, vol. 22, no. 4, p. 10, 2001.

[139] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME*, 1960.

[140] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[141] K. Yoo, J. Chun, and J. Shin, "Target Tracking and Classification for Missile Using Interacting Multiple Model," in *IEEE RADAR*, 2018, pp. 1–6.

[142] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "SpAGNN: Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting from Sensor Data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9491–9497.

[143] A. Kampker, M. Sefati, A. S. Abdul Rachman, K. Kreisköther, and P. Campoy, "Towards Multi-Object Detection and Tracking in Urban Scenario under Uncertainties:" in *VEHITS*, 2018, pp. 156–167.

[144] A. S. Willsky and H. L. Jones, "A generalized likelihood ratio approach to state estimation in linear systems subjects to abrupt changes," in *1974 IEEE Conference on Decision and Control Including the 13th Symposium on Adaptive Processes*, 1974, pp. 846–853.

[145] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.

[146] H. Kwakernaak, "Maximum likelihood parameter estimation for linear systems with singular observations," *IEEE Transactions on Automatic Control*, vol. 24, no. 3, pp. 496–498, 1979.

[147] M. Werling and L. Groll, "Low-level controllers realizing high-level decisions in an autonomous vehicle," in *2008 IEEE Intelligent Vehicles Symposium (IV)*, 2008, pp. 1113–1118.

[148] R. P. Srivastava, "An introduction to evidential reasoning for decision making under uncertainty: Bayesian and belief function perspectives," *International Journal of Accounting Information Systems*, Special Issue on Methodologies in AIS Research, vol. 12, no. 2, pp. 126–135, 2011.

[149] A. Jøsang, "Belief Mosaics of Subjective Opinions," in *International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8.

[150] N. Milisavljevic and I. Bloch, "Sensor fusion in anti-personnel mine detection using a two-level belief function model," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 33, no. 2, pp. 269–283, 2003.

[151] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

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

[153] D. Krajzewicz, G. Hertkorn, C. Feld, and P. Wagner, *SUMO (Simulation of Urban MObility); An Open-Source Traffic Simulation*. 2002, p. 187.

[154] T. Reineking, "Belief Functions: Theory and Algorithms," Ph.D. dissertation, Universität Bremen, 2014.

[155] T. Denoeux, "The cautious rule of combination for belief functions and some extensions," in *2006 9th International Conference on Information Fusion*, 2006, pp. 1–8.

[156] F. Pukelsheim, "The Three Sigma Rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[157] Y. Zhao, B. Lehman, R. Ball, J. Mosesian, and J.-F. de Palma, "Outlier detection rules for fault detection in solar photovoltaic arrays," in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2013, pp. 2913–2920.

[158] J. Chen, B. Yuan, and M. Tomizuka, "Model-free Deep Reinforcement Learning for Urban Autonomous Driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2765–2771.

[159] J. Hawke, R. Shen, C. Gurau, *et al.*, "Urban Driving with Conditional Imitation Learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 251–257.

[160] J. Chen, B. Yuan, and M. Tomizuka, "Deep Imitation Learning for Autonomous Driving in Generic Urban Scenarios with Enhanced Safety," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2884–2890.

[161] J. Levinson, J. Askeland, J. Becker, *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.

[162] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using Robust Model Predictive Control," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 233–238.

[163] N. A. Nguyen, D. Moser, P. Schrangl, L. del Re, and S. Jones, "Autonomous overtaking using stochastic model predictive control," in *2017 11th Asian Control Conference (ASCC)*, 2017, pp. 1005–1010.

[164] 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, 2018.

[165] 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)*, 2020, pp. 1–7.

[166] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer, 2011.

[167] L. Paparusso, S. Melzi, and F. Braghin, "Real-Time Forecasting of Driver-Vehicle Dynamics on 3D Roads: A Deep-Learning Framework Leveraging Bayesian Optimisation," *arXiv preprint:2103.03825*, 2021.

[168] N. Dang, T. Brüdigam, Z. Zhang, F. Liu, M. Leibold, and M. Buss, "Distributed Stochastic Model Predictive Control for a Microscopic Interactive Traffic Model," *Electronics*, vol. 12, no. 6, p. 1270, 2023.

[169] D. Kahneman and A. Tversky, "Subjective probability: A judgment of representativeness," *Cognitive Psychology*, vol. 3, no. 3, pp. 430–454, 1972.

[170] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.

[171] C. Pek and M. Althoff, "Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1447–1454.

[172] S. Diamond and S. Boyd, "CVXPY: A Python-Embedded Modeling Language for Convex Optimization," *Journal of Machine Learning Research*, 2016.

[173] M. Geisslinger, F. Poszler, J. Betz, C. Lütge, and M. Lienkamp, "Autonomous Driving Ethics: From Trolley Problem to Ethics of Risk," *Philosophy & Technology*, vol. 34, no. 4, pp. 1033–1055, 2021.

[174] J. Betz, H. Zheng, A. Liniger, *et al.*, "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.

[175] 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.

[176] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürr, "Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.

[177] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "PiP: Planning-Informed Trajectory Prediction for Autonomous Driving," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., ser. Lecture Notes in Computer Science, Springer International Publishing, 2020, pp. 598–614.

[178] J. Liu, W. Zeng, R. Urtasun, and E. Yumer, "Deep Structured Reactive Planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4897–4904.

[179] D. Nguyen-Tuong and J. Peters, "Incremental online sparsification for model learning in real-time robot control," *Neurocomputing*, Adaptive Incremental Learning in Neural Networks, vol. 74, no. 11, pp. 1859–1867, 2011.

[180] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[181] A. Domahidi and J. Jerez, *FORCES Professional*, Embotech AG, 2014.

[182] S. I. E. Inc., *Gran Turismo Sport*.

[183] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *2013 European Control Conference (ECC)*, 2013, pp. 3071–3076.

[184] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration," in *Advances in Neural Information Processing Systems*, 2018.

[185] J. ji, A. Khajepour, W. Melek, and Y. Huang, "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 1–1, 2016.

[186] 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.

[187] M. Ammour, R. Orjuela, and M. Basset, "Collision avoidance for autonomous vehicle using MPC and time varying Sigmoid safety constraints," *IFAC-PapersOnLine*, 6th IFAC Conference on Engine Powertrain Control, Simulation and Modeling E-COSM 2021, vol. 54, no. 10, pp. 39–44, 2021.

[188] V. K. Adajania, A. Sharma, A. Gupta, H. Masnavi, K. M. Krishna, and A. K. Singh, "Multi-Modal Model Predictive Control Through Batch Non-Holonomic Trajectory Optimization: Application to Highway Driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4220–4227, 2022.

[189] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét Frame," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993.