TECHNISCHE UNIVERSITÄT MÜNCHEN
TUM School of Computation, Information and Technology

Distributed Stochastic Model Predictive Control for a Microscopic Interactive Traffic Model

Ni Dang

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Eckehard Steinbach

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Martin Buss
2. Prof. Dr.-Ing. Heike Vallery

Die Dissertation wurde am 20.02.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 13.09.2024 angenommen.

Preamble

This dissertation marks the culmination of a profound academic journey, one that has been made possible by the invaluable contributions, support, and encouragement of countless individuals. As I reflect upon the challenges and triumphs of my doctoral studies, I am deeply humbled and profoundly grateful to those who have played pivotal roles in shaping this path.

First, my deep appreciation goes to the head of the chair and also my esteemed advisor Univ. Prof. Dr.-Ing./Univ. Tokio Martin Buss, whose expertise, mentorship, and unwavering belief in my potential have been my guiding lights. Furthermore, I extend my deepest appreciation to my esteemed supervisor PD Dr.-Ing. habil. Marion Leibold. Your intellectual guidance, patience, and dedication to my research have been instrumental in shaping my scholarly pursuits. Your insights and belief in my potential have served as constant sources of inspiration. Also, I would like to thank Prof. Sofie Haesaert and Prof. Zhiyong Sun for your warm welcome to Eindhoven Univerity of Technology.

I am indebted to my fellow colleagues and peers of both Chair of Automatic Control Engineering (LSR) and Chair of Information-oriented Control (ITR), especially to our LSR colleagues Volker Gabler, Khoi Hoang Dinh, Fangzhou Liu, Zengjie Zhang, Yingwei Du, Tim Brüdigam, Tong Liu, Zhehua Zhou, Cong Li, Yongchao Wang, Sebastian Kerz, Salman Bari, Yuhong Chen, Tommaso Benciolini, Michael Fink, Annalena Daniels and Johannes Teutsch, and ITR colleagues Yongxu He, Qingchen Liu, Lanlin Yu, Junjie Jiao, Siyi Wang, Xiaobing Dai and Zewen Yang. Thank you for the camaraderie, stimulating discussions, and shared moments of triumph and tribulation. Your dedication to teaching and research has provided an environment conducive to intellectual exploration. Your insights and diverse perspectives have broadened my horizons and enriched my academic journey.

Thanks to the technical and administrative staffs: Wolfgang Jaschik, Larissa Schimid, and Brigitta Renner. Your invaluable assistance, whether it was in the laboratory, the library, or the administrative office, has facilitated the smooth progression of my research and academic pursuits. I also want to express my gratitude to Dr. Stephen Starck from the Language Center at the Technical University of Munich for his valuable advice on academic writing.

Many students also contribute to this work, and I express particular gratitude to Tao Shi for his valuable contributions to the research related to driving style.

Lastly, I reserve my most heartfelt thanks for my parents. Your endless love, sacrifices, and unwavering encouragement have been the foundation upon which I have built my academic aspirations. Your belief in me has been a driving force, propelling me forward with the determination to excel and make a meaningful contribution to my field. My special appreciation goes to Johannes Ringwald for his constant support during the thesis writing phase and valuable assistance in translating the Abstract.

In presenting this thesis, I offer my gratitude to each of you, for it is through your collective support that I have been able to reach this point. As I step into the next chapter of my journey, I carry with me the lessons, memories, and relationships forged during these transformative years.

With heartfelt appreciation,
Ni Dang

Munich, February 2024

Abstract

Autonomous driving has the potential to bring significant societal, economic, and environmental benefits, such as improving safety and productivity. Consequently, it has been drawing increasing attention in recent decades. Several vital modules, including perception, localization, planning, and control, collaborate cohesively to enable autonomous vehicles to safely move and interact with their surroundings. Perception involves the vehicle's ability to interpret and understand its surroundings using sensors like cameras, Lidar, etc. Localization ensures the vehicle knows its precise position within its environment. Planning involves high-level decision-making, where the autonomous vehicle determines the optimal maneuvers and motions to respond to complex traffic scenarios. Control is responsible for executing the decisions made by the high-level planner. Autonomous vehicles operate in inherently uncertain environments. The uncertainty can arise from various factors including vehicle interactions, unpredictable movements of other road users, sensor limitations, road conditions, map inconsistencies, localization accuracy, and unforeseen emergencies. In this thesis, we handle uncertainties arising from dynamic interactions with other vehicles and the unpredictable nature of these interactions from both planning and control perspectives, with a particular focus on control.

Model Predictive Control (MPC) can address uncertainties, as well as enable real-time decision-making, handle complex constraints, integrate with trajectory planning, and respond to changing environments, and consequently is a crucial technology for autonomous driving. Stochastic Model Predictive Control (SMPC), an advanced variation of MPC, has attracted increasing attention for autonomous driving in recent years since it can address uncertainties without being overly conservative. This is achieved by allowing a certain probability of constraint violations, where the probability is determined by a factor named risk parameter. Due to these characteristics, we have selected MPC and SMPC as the vehicle controller for this work. Specifically, we propose a planning method that is efficiently suitable for vehicles controlled by general MPC and explore further how to better use SMPC to control vehicles.

The planning module determines the best maneuvers for the current traffic situation and conveys them to the low-level MPC to execute. Generally, for a vehicle driving in an uncertain environment, e.g., a highway, there are several potentially feasible maneuvers. How to select the best maneuver for the given traffic situation from all potential ones is a significant challenge. We introduce a maneuver planning approach consisting of an existing maneuver generation algorithm and a novel selection method and integrate it with low-level MPC-based trajectory control. The maneuver generation algorithm delivers all potential maneuvers in both lateral and longitudinal directions. Our maneuver selection method chooses the best maneuvers from all generated ones for both directions, respectively, based on multiple factors including driving goals, safety, efficiency, and passanger comfort.

An individual SMPC-controlled vehicle can successfully move in an uncertain environment with its driving goals, traffic rules, and collision avoidance requirements being realized. Nevertheless, the potential to extend the success of an individual SMPC-controlled vehicle to an interactive system involving multiple SMPC-controlled vehicles remains uncertain, since the interactive behaviors of multiple SMPC-controlled vehicles have not been examined to date. This motivates us to investigate how multiple SMPC-controlled vehicles in an interactive system will interact with each other. Further we investigate how to impact these interactions to enhance the performances of both individual vehicles and the overall system. To investigate these, we first model the interactive multi-vehicle system using a Distributed SMPC (DSMPC) framework, where each vehicle is

controlled by SMPC and interacts with its surrounding vehicles by observing their current states and predicting their future behaviors in order to avoid potential collisions. In our investigation, we specifically examine the influence of SMPC risk parameters on both non-interactive and interactive vehicle systems in context of different highway scenarios and provide insights into the appropriate configuration of risk parameters for vehicles operating within interactive systems. Furthermore, precisely predicting the behaviors of the neighboring vehicles is crucial for improving the safety of an SMPC-controlled vehicle because the predicted behaviors are incorporated into the collision avoidance constraints of the SMPC formulation. However, the prediction remains challenging due to numerous unknown factors including their driving goals and particularly their interactive behaviors. Nevertheless, vehicles tend to drive habitually exhibiting a consistent driving style, e.g., how they achieve their driving goals and how they interact with other road users. This motivates our work about predicting the behaviors of surrounding vehicles based on their driving styles. We identify the driving styles including interaction characteristics of SMPC-controlled vehicles, subsequently predict the behaviors of these vehicles according to their identified driving styles, and eventually incorporate the predictions into the collision avoidance constraints of SMPC.

Zusammenfassung

Autonomes Fahren hat das Potenzial erhebliche gesellschaftliche, wirtschaftliche und ökologische Vorteile, wie verbesserte Sicherheit und Produktivität, mit sich zu bringen. Mehrere vitale Bereiche wie Perzeption, Lokalisierung, Planung und Steuerung arbeiten kohäsiv zusammen, um autonomen Fahrzeugen eine sichere Bewegung und Interaktion mit der Umgebung zu ermöglichen. Zur Perzeption bzw. Wahrnehmung gehört die Fähigkeit des Fahrzeugs die eigene Umgebung mit Hilfe von Sensoren wie Kameras, Lidar etc. zu interpretieren und zu verstehen. Lokallisierungssysteme stellen sicher, dass das Fahrzeug seine genaue Position in der eigenen Umgebung kennt. Planung umfasst höhere Entscheidungsprozesse, bei denen die für das autonome Fahrzeug optimalen Manöver und Bewegungen bestimmt werden, um auf komplexe Verkehrsszenarien zu reagieren. Die Steuerung ist für die Ausführung der vom übergeordneten Planer getroffenen Entscheidungen verantwortlich. Autonome Fahrzeuge agieren in von Natur aus unsicheren Umgebungen. Diese Unsicherheit kann durch verschiedene Faktoren wie unvorhersehbare Bewegungen anderer Verkehrsteilnehmer, Sensoreinschränkungen, Straßenbedingungen, Karteninkonsistenzen, Lokalisierungsgenauigkeit oder unvorhergesehene Notfälle verursacht werden. Gegenstand dieser Arbeit sind, sowohl aus Planungs- als auch aus Regelungssicht, mit besonderem Schwerpunkt auf der Regelung, Unsicherheiten, die sich aus dynamischen Interaktionen mit anderen Fahrzeugen und der unvorhersehbaren Natur dieser interaktionnen ergeben.

Model Predictive Control (MPC) kann Unsicherheiten berücksichtigen, Entscheidungen in Echtzeit ermöglichen, mit komplexen Einschränkungen umgehen, in die Trajektorienplanung integriert werden sowie auf sich ändernde Umgebungen reagieren und stellt deshalb eine entscheidende Technologie für autonomes Fahren dar. Stochastic Model Predictive Control (SMPC), eine weiterentwickelte Variante von MPC, hat in den letzten Jahren im Bereich autonomes Fahren immer mehr Aufmerksam auf sich gezogen, da SMPC Unsicherheiten berücksichtigen kann ohne signifikant konservativ zu sein. Dies wird erreicht indem eine gewisse Wahrscheinlichkeit von Beschränkungsverletzungen zugelassen wird, wobei diese Wahrscheinlichkeit durch einen Faktor namens Risikoparameter bestimmt wird. Aufgrund der beschriebenen Eigenschaften haben wir MPC bzw. SMPC als Fahrzeugregler für diese Arbeit ausgewählt. Konkret schlagen wir eine Planungsmethode vor, die für Fahrzeuge geeignet ist, welche über MPC gesteuert werden und untersuchen wie SMPC besser zur Regelung von Fahrzeugen genutzt werden kann.

Das Planungsmodul ermittelt die besten Manöver für die aktuelle Verkehrssituation und übermittelt sie zur Ausführung an den untergeordneten MPC. Im Allgemeinen gibt es für ein Fahrzeug, das in einer unsicheren Umgebung, z. B. einer Autobahn, fährt, mehrere potentiell umsetzbare Manöver. Aus all diesen möglichen Kandidaten, das, für die aktuelle Situation, optimale Manöver auszuwählen, stellt eine signifikante Herausforderung dar. Wir stellen einen Manöverplanungsansatz vor, welcher einen existierenden Manövergenerierungsalgorithmus sowie eine neuartige Auswahlmethode umfasst und integrieren diesen Ansatz mit einer MPC-basierten untergeordneten Trajektorienregelung. Der Manövergenerierungsalgorithmus liefert alle potenziellen Manöver sowohl in Quer- als auch in Längsrichtung. Unsere Manöverauswahlmethode wählt, basierend auf mehreren Faktoren wie Fahrziele, Sicherheit, Effizienz und Fahrgastkomfort, aus den generierten Manövern jeweils die besten Kandidaten für beide Richtungen aus.

Ein einzelnes SMPC-gesteuertes Fahrzeug kann sich erfolgreich in einer unsicheren Umgebung bewegen, wenn seine Fahrziele, die vorhandenen Verkehrsregeln und Anforderungen zur Kollisionsvermeidung umgesetzt werden. Dennoch bleibt das Potenzial, den Erfolg eines einzelnen SMPC-gesteuerten Fahrzeugs auf ein interaktives System mit mehreren SMPC-gesteuerten

Fahrzeugen auszudehnen, unsicher, da das interaktive Verhalten mehrerer SMPC-gesteuerter Fahrzeuge bisher nicht untersucht wurde. Dies motiviert unsere Studien, wie mehrere SMPC-geregelte Fahrzeuge in einem interaktiven System miteinander interagieren. Weiter untersuchen wir wie wir diese Interaktionen beeinflussen können, um die Leistung sowohl einzelner Fahrzeuge als auch des Gesamtsystems zu verbessern. Hierzu modellieren wir zunächst das interaktive Mehrfahrzeugsystem mithilfe eines Distributed SMPC (DSMPC)-Frameworks, bei dem jedes Fahrzeug über SMPC geregelet wird sowie mit den umliegenden Fahrzeugen interagiert, indem es deren aktuellen Zustand beobachtet und das zukünftige Verhalten vorhersagt um mögliche Kollisionen zu vermeiden. In unserer Studie untersuchen wir insbesondere den Einfluss von SMPC-Risikoparametern sowohl in nicht-interaktiven als auch in interaktiven Systemen im Kontext verschiedenenr Autobahnszenarien und geben Einblicke in die geeignete Konfiguration von Risikoparametern für Fahrzeuge, die in interaktiven Systemen betrieben werden. Darüber hinaus ist die genaue Vorhersage des Verhaltens benachbarter Fahrzeuge von entscheidender Bedeutung für die Verbesserung der Sicherheit eines SMPC-geregelten Fahrzeugs, da das vorhergesagte Verhalten in die Kollisionsvermeidungsbeschränkungen der SMPC-Formulierung einbezogen wird. Die Vorhersage bleibt jedoch aufgrund zahlreicher unbekannter Faktoren, wie Fahrziele oder insbesondere das interaktive Verhalten, schwierig. Dennoch neigen Fahrzeuge zu einem gewohnheitsmäßigen bzw. einheitlichen Fahrstil. Beispielsweise wie diese Ihre Fahrtziele erreichen oder wie diese mit anderen Verkehrsteilnehmern interagieren. Dies motiviert unsere Arbeit, das Verhalten umliegender Fahrzeuge anhand ihres Fahrstils vorherzusagen. Wir identifizieren die Fahrstile einschließlich der Interaktionseigenschaften von SMPC-geregeleten Fahrzeugen, prognostizieren anschließend das Verhalten dieser Fahrzeuge entsprechend ihren identifizierten Fahrstilen und integrieren die Vorhersagen abschließend in die Kollisionsvermeidungsbeschränkungen von SMPC.

Contents

1. Introduction

Autonomous driving has been drawing increasing attention in recent decades due to its potential to bring significant societal, economic, and environmental benefits, such as improving safety, reducing the number of traffic accidents, increasing mobility for people with disabilities, and improving productivity. Therefore, vital modules including perception, localization, planning, and control, work together in real-time to approach these benefits by enabling autonomous vehicles to navigate and drive safely without human intervention. The perception is responsible for gathering information about the environment around the vehicle using various sensors, e.g., LiDAR, Radar, and Cameras, and processing the data to identify and understand relevant objects, obstacles, road markings, and traffic signs. The localization determines the precise position of the vehicle within its environment, which is crucial for accurate navigation. The planning module makes high-level decisions about the vehicle's motions and responds to complex traffic scenarios. Control systems implement the specific actions needed to execute the higher-level decisions made by the planning module, ensuring the vehicle moves safely and smoothly. In this thesis, we focus on planning and control.

Autonomous vehicles move through environments inherently fraught with uncertainties that contribute to the complexity of driving. Addressing uncertainties is crucial for ensuring the safety and reliability of autonomous driving systems [1]. Mitigating uncertainties requires advanced sensor technologies, robust perception algorithms, real-time planning processes, advanced control strategies, and ongoing improvements in system reliability. Many researchers have been working on these in the past decades [2]–[7]. The uncertainties stem from a multitude of sources, including unpredictable movements of other road users, sensor limitations, map inconsistencies, localization inaccuracy, and unforeseen emergencies. In this thesis, we tackle uncertainties arising from dynamic interactions with other vehicles and the unpredictable nature of these interactions.

The planning encompasses strategic high-level decision-making processes that govern the vehicle's movements and responses in uncertain environments. In the realm of motion planning, various technical terms are present in the literature, such as maneuver planning, task planning, motion planning, trajectory planning, and path planning [8]–[15]. Although these terms lack universal definitions, their meanings are typically clear within their specific contexts. In this thesis, we focus on planning maneuvers that involve high-level decisions such as selecting desired lanes and speeds based on factors including driving goals, safety, efficiency, and passenger comfort. The planning module needs to coordinate seamlessly with the subsequent controller module, conveying information and signals that the controller can comprehend and execute effectively.

The controller of the autonomous vehicles must be able to manage the uncertainties. Nominal control approaches that lack the capacity to handle system uncertainties might cause hazardous performance [16]. In contrast, while robust control methods, e.g., Robust Model Predictive Control (RMPC), can account for uncertainties, they tend to be overly conservative because they also factor in worst-case scenarios [17]. Stochastic Model Predictive Control (SMPC) has been used to control

autonomous vehicles due to its capability to account for uncertainties while avoiding excessively cautious behaviors [16], [18]–[20]. SMPC is an advanced variation of Model Predictive Control (MPC). MPC iteratively solves an optimal-control problem with multiple constraints being satisfied on a finite prediction horizon. In contrast, SMPC permits the violation of constraints with a specified low probability through the application of probabilistic chance constraints. Autonomous vehicles controlled by SMPC treat collision-avoidance constraints with adjustable risk parameters as probabilistic constraints [16], [19], resulting in non-conservative behaviors [21], [22].

In previous studies of applying SMPC in autonomous driving [16], [18]–[20], simulations that confirmed the suitability of SMPC-based controllers for autonomous vehicles assumed that the surrounding vehicles used a much simpler controller and, in particular, did not consider predictions of the surrounding vehicles and, thus, also did not react to SMPC-controlled vehicles, meaning mutual interactions were not considered. However, in real traffic, the surrounding vehicles are also controlled vehicles, having their individual driving goals and reacting to their nearby vehicles. Therefore, an SMPC-controlled vehicle needs to predict the behaviors of the surrounding vehicles and interact with these vehicles based on its predictions; even further, the potential interactions between vehicles should also be considered in the predictions because the interactions will influence the future behaviors of the surrounding vehicles. These motivate us to study how an SMPC-controlled vehicle interacts with its surrounding vehicles and how to improve their interactions by more precisely predicting the behaviors of the surrounding vehicles with the interactions taken into consideration.

Evaluating the performance of a novel controller requires simulating it in different scenarios employing either microscopic traffic models or macroscopic traffic models. Microscopic traffic models focus on studying the traffic phenomena of individual vehicles and analyzing how they interact with each other [23]. In microscopic traffic models, the dynamics of each traffic participant are individually modeled [24]; this allows us to know each vehicle's detailed information, such as the location, velocity, inertial heading, acceleration, and steering angle [24]. Macroscopic traffic models research the overall characteristics, e.g., the intensity, density, and mean speed, of the traffic flow, in which the details of individual interactions between vehicles are ignored [23], [25]. The selection of the model type depends on the demanded level of the analytical details. In this thesis, we study the performance of individual MPC/SMPC-controlled vehicles and the interactions between them, and, consequently, select a microscopic traffic model.

Driving in diverse environments like highways, urban areas, suburbs, rural regions, and mountainous terrain involves specific challenges and characteristics, each requiring unique skills and considerations. We select highway scenarios as the driving environment because the uncomplicated driving structure and limited driving behaviors render highways the most feasible context for the initial deployment of microscopic interactive traffic systems [15].

In this thesis, our primary emphasis lies in 1) designing a high-level maneuver planner that can effectively issue instructions to MPC-controlled vehicles and 2) enhancing vehicle controller SMPC with interactions taken into consideration for controlling individual vehicles based on a microscopic traffic model in the highway environment.

1.1. Challenges

The brief introduction provided above motivates us to address the ensuing challenges related to the design of the vehicle planner and controller, emphasizing considerations for interactions.

Challenge 1: *How can a planner select the best maneuvers for the lower-level MPC to execute?*
Maneuvers represent high-level strategies that delineate the actions the vehicle will undertake in the short term or long term. In the context of a vehicle driving on a multi-lane highway, there exist numerous potential maneuvers, e.g., lane changing and decelerating, in both longitudinal and lateral directions. The maneuver planner must take into consideration multiple factors including driving goals, safety, efficiency, and passenger comfort. Selecting the optimal maneuvers from a range of options poses a significant challenge.

Challenge 2: *How will multiple SMPC-controlled vehicles interact with each other?*
An individual SMPC-controlled vehicle can successfully move while realizing its driving goals, traffic rules, and collision avoidance requirements. However, for multiple SMPC-controlled vehicles in the driving environment, their interactive behaviors have not been studied yet. This makes extending the success of an individual SMPC vehicle to an interactive system involving multiple SMPC-controlled vehicles uncertain. Studying the interactions between SMPC-controlled vehicles is the foundation of this extension. The study includes finding the answers to the following questions. How will their behaviors change while interacting with each other? What factors, e.g., risk parameters that determine how aggressively the SMPC-controlled vehicles are driving, can influence the interactions? How will the interactions influence the performances of the individual vehicles as well as the overall multi-vehicle system?

Challenge 3: *How to precisely predict the behaviors of surrounding vehicles?*
Accurately predicting the future behaviors of nearby vehicles is critical in enhancing the safety of SMPC-controlled vehicles [26]. However, predicting the future behaviors of surrounding vehicles is challenging due to numerous unknown factors, such as their driving goals, and particularly the impact of vehicles' interactive behaviors. On the other hand, a vehicle tends to drive habitually exhibiting a consistent driving style, including how they achieve their driving goals and how they interact with other road users. This opens up the possibility of predicting the behaviors of surrounding vehicles based on their driving styles.

The subsequent section outlines how these three challenges are tackled within this thesis. Given that a specific challenge may be tackled in more than one chapter, we delineate the contributions of each chapter in the following.

1.2. Contributions and Outline

This thesis provides solutions to the above-mentioned challenges motivated by the design of the vehicle planner and controller within the context of microscopic interactive traffic. We illustrate the outline of the thesis in Figure 1.1. Before presenting our solutions, we first introduce the preliminaries that will be used in addressing these challenges in **Chapter 2**. To resolve **Challenge 1**, we proposed a maneuver planning method that selects the best maneuver from various maneuvers and efficiently conveys it to a low-level MPC-based trajectory generation algorithm in **Chapter 3**. We handle **Challenge 2** in **Chapter 4** by first modeling the microscopic interactive control system of multiple-controlled vehicles using a Distributed SMPC (DSMPC) framework and then investigating the interactive behaviors of vehicles in this system. **Challenge 3** is systematically addressed through the collaborative efforts of both **Chapter 5** and **Chapter 6**. Specifically, we identify the

driving styles including interaction characteristics of SMPC-controlled vehicles using an Maximum Entropy Inverse Reinforcement Learning (ME-IRL) method in **Chapter 5**; then, the idea of the driving style identification is further used in **Chapter 6** to predict future behaviors of surrounding vehicles with the interactions between vehicles taken into consideration. **Chapter 7** concludes the work in this thesis and provides an outlook to future research directions. (Besides, a deep learning-based trajectory prediction method is introduced in Appendix A [27].)

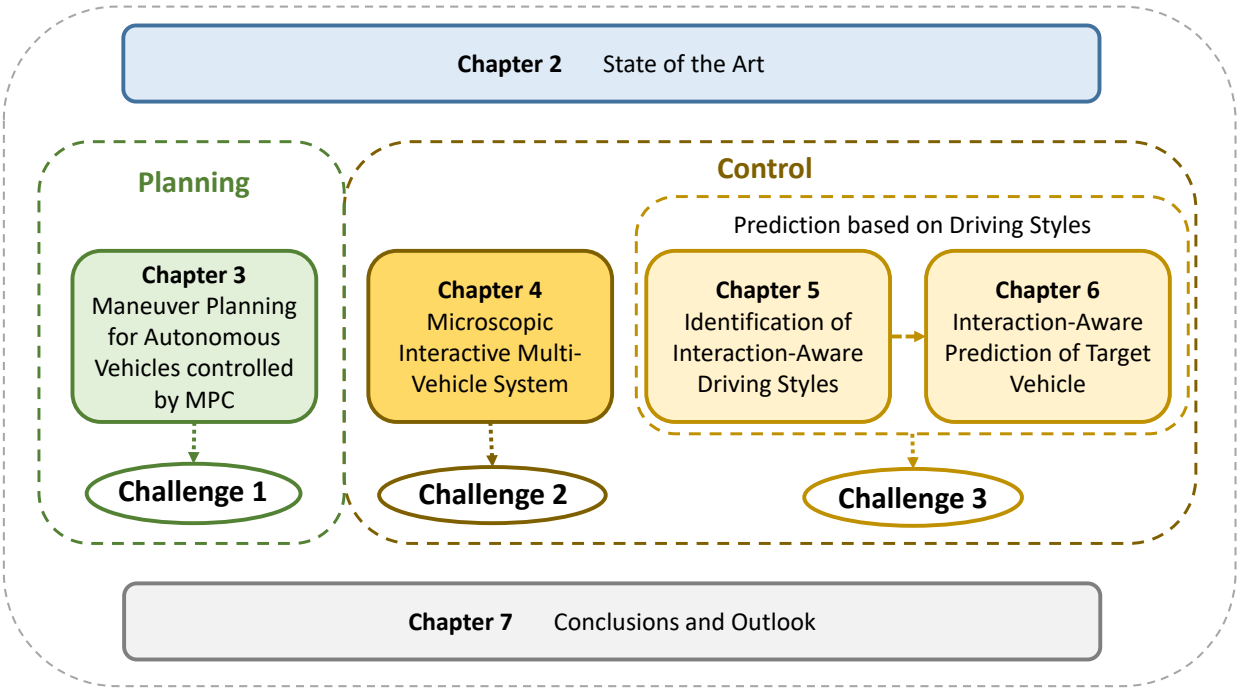


Figure 1.1.: The outline of the thesis.

The contributions of this thesis are to address the challenges in the planning and control of autonomous vehicles within the context of microscopic interactive traffic. We introduce the contributions in Chapters 3, 4, 5 and 6. These contributions are summarized by chapters as follows.

Chapter 3: Maneuver Planning for Autonomous Vehicles controlled by MPC.

We introduce a high-level maneuver planning approach and integrate it with a low-level MPC-based trajectory control in this chapter. Our methodology is based on the maneuver generation approach introduced in [13]. We leverage the maneuver generation method presented in [13] and refine it to enhance practical applicability by eliminating the need to explore an excessive number of trajectories. Specifically, we first generate all possible maneuvers using the maneuver generation method in [13]; then, our novel approach selects the best maneuver from all possible maneuvers generated using the method in [13]; subsequently, the selected maneuver is input to the lower-level trajectory tracking controller.

(The method and results presented in Chapter 3 have been published in [28].)

Chapter 4: Simulation of Microscopic Interactive Multi-SMPC-Vehicle System.

In this chapter, we explore the interactions among vehicles controlled by SMPC in a microscopic interactive multi-vehicle system designed for a highway environment. To do this, we model the microscopic interactive multi-vehicle system using a DSMPC framework [29]–[31], where each

vehicle interacts with its surrounding vehicles by observing their current states and predicting their future behaviors in order to avoid potential collisions. The risk parameter of an SMPC-controlled vehicle plays an important role in our interaction studies because it determines the aggressiveness/conservativeness level of the vehicle when reacting to nearby vehicles. Specifically, we investigate the effects of SMPC risk parameters on non-interactive and interactive vehicle-control systems on highways and provide insights into how to set risk parameters for vehicles in interactive systems.

(The method and results presented in Chapter 4 have been published in [32].)

Chapter 5: Identification of Interaction-Aware Driving Styles.

The driving style of an autonomous vehicle pertains to its habitual driving behaviors including how it interacts with other vehicles. The driving style of a vehicle tends to be consistent over time, which makes predicting future behaviors of the vehicle according to its driving style reasonable. Identifying the driving styles of vehicles serves as the preparatory phase for subsequent behavior prediction based on driving styles. This chapter solves the driving style identification problem based on a Maximum Entropy Inverse Reinforcement Learning (ME-IRL) method introduced in [33], where the driving style is represented as a cost function of a series of weighted features that capture the characteristics of the demonstration trajectory. The ME-IRL method endeavors to learn the optimal weights of the features. We propose novel features to capture the interaction characteristics of the vehicles; since some novel features are active only when the vehicle is close to the nearby vehicle, we design a triggering condition to activate these features.

(The method and results presented in Chapter 5 have been published in [34].

Chapter 6: Interaction-Aware Prediction of Vehicle Trajectories.

In this chapter, we partially adopt the idea of identifying the driving styles of the vehicles in Chapter 5 and introduce how to predict the behaviors of surrounding vehicles using the knowledge of their driving styles. The prediction is used within the microscopic interactive multi-vehicle system modeled by the DSMPC framework. The novelties in identifying the driving styles include a novel idea for demonstrations that allows for the inclusion of more details, enabling a more comprehensive understanding of the behaviors of the vehicles. To predict the behaviors of surrounding vehicles, we generate trajectories with similar driving styles to the surrounding vehicles, treat these trajectories as the predictions of the surrounding vehicles, and incorporate these interaction-aware predictions into SMPC-controlled vehicles in the microscopic interactive multi-vehicle system.

(The work presented in Chapter 6 will be published.)

More state-of-the-art information will be provided in the chapters.

1.3. List of Publications

The contributions presented in this thesis are based on the following work.

- Chapter 3 is based on [28]:
  N. Dang, T. Brüdigam, M. Leibold, and M. Buss, "Combining event-based maneuver selection and MPC based trajectory generation in autonomous driving," *Electronics*, vol. 11, no. 10, p. 1518, 2022.

- Chapter 4 is based on [32]:
  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.

- Chapter 5 is based on [34]:
  N. Dang, T. Shi, Z. Zhang, W. Jin, M. Leibold, and M. Buss, "Identifying reaction-aware driving styles of stochastic model predictive controlled vehicles by inverse reinforcement learning," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2023, pp. 2887–2892.

- Chapter 6 is based on our recent work that will be submitted to a journal.

- Appendix A is based on [27]:
  N. Dang, Z. Zhang, J. Liu, M. Leibold, and M. Buss, "Incorporating target vehicle trajectories predicted by deep learning into model predictive controlled vehicles," *arXiv preprint arXiv:2310.02843, 2023.*

2. Preliminaries

This chapter introduced the preliminaries that will serve as the foundation for our approaches in this thesis. The preliminary elements consist of vehicle modeling, vehicle control, Maximum Entropy Inverse Reinforcement Learning (ME-IRL), trajectory representation, and the evaluation metrics we will use to evaluate our control approaches.

2.1. Vehicle Modeling

Vehicle modeling mathematically represents the motions and behaviors of vehicles in response to external inputs. Various vehicle models, with different modeling depths, have been proposed in the literature, including, e.g. Fiala tire model, dynamic bicycle model, point-mass model and kinematic bicycle model[16], [19], [35]–[37]. The Fiala tire model and dynamic bicycle model are physically more accurate regarding representing vehicle dynamics as well as computationally expensive due to requiring numerous details, e.g., tire friction and stiffness, compared to the point-mass model and the kinematic bicycle model. On the other hand, the point-mass model[19] and kinematic bicycle model [38] are relatively coarse models, ignoring many details, and thus contribute to avoiding excessive computational load [38]. In this thesis, vehicle models are used in the design of model-based vehicle control, e.g., model predictive control, for controlled vehicles to predict their own behaviors and those of the surrounding vehicles. We choose the point-mass model[19] and kinematic bicycle model [38], but any other model that includes the position and speed of the vehicle is also suitable. The point-mass model and the kinematic bicycle model are introduced as follows.

2.1.1. Point-Mass Model

This point-mass model is described by the following equation

$$\xi_{k+1} = A\xi_k + Bu_k \tag{2.1}$$

where $\xi_k = \left[x_k, y_k, v_{x,k}, v_{y,k}\right]^\top$ is the state vector. x_k and y_k represent the longitudinal and lateral positions of the vehicle at time k, respectively. $v_{x,k}$ and $v_{y,k}$ are the longitudinal and lateral velocities at time k, respectively. $u_k = \left[a_{x,k}, a_{y,k}\right]^\top$ is the input vector, containing the longitudinal acceleration $a_{x,k}$ and the lateral acceleration $a_{y,k}$ at time k. Moreover, system matrices A and B are shown below,

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \tag{2.2}$$

where T is the sampling interval.

2.1.2. Kinematic Bicycle Model

The original kinematic bicycle model [38] is nonlinear and continuous in time. In order to use the kinematic bicycle model for vehicles controlled by Model Predictive Control (MPC), this original model must be transformed into a linear discrete-time version. We introduce the nonlinear and continuous-time kinematic bicycle model as well as the linearized and discretized version as follows.

Nonlinear and Continuous-Time Model

The kinematic bicycle model is given by the following nonlinear continuous-time equations [38],

$$\dot{x} = v\cos(\psi+\beta) \tag{2.3a}$$

$$\dot{y} = v\sin(\psi+\beta) \tag{2.3b}$$

$$\dot{\psi} = \frac{v}{l_r}\sin\beta \tag{2.3c}$$

$$\dot{v} = a \tag{2.3d}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f+l_r}\tan\delta\right) \tag{2.3e}$$

where x and y represent the longitudinal position and lateral position of the center of mass of the vehicle, respectively. The inertial heading is given by ψ, and the velocity of the vehicle is denoted by v. The distances from the center of mass of the vehicle to the front and rear axles are l_f and l_r, respectively. The angle of the vehicle with respect to the longitudinal axis of the road is β. The acceleration of the center of mass of the vehicle is represented by a. The front steering angle is δ. The state and control input vectors are $\xi = (x, y, \psi, v)^\top$ and $u = (a, \delta)^\top$, respectively. The nonlinear continuous kinematic bicycle model is summarized as $\dot{\xi} = \mathscr{F}^{\mathrm{c}}(\xi, u)$. Note that the elements in state vector ξ and control input vector u are different from that in the state and control input vectors of the point-mass model (2.1).

Linear Discrete-Time Model

In simulations, a linearized and discretized version of the kinematic bicycle model is used. We adopt the methods in [20], [39] to get the linearized and discretized model, denoted as

$$\xi_{k+1} = \xi_0 + T\mathscr{F}^{\mathrm{c}}(\xi_0, 0) + A(\xi_k - \xi_0) + Bu_k, k = 0, \ldots, N-1, \tag{2.4}$$

where the state and control input at step k are represented by ξ_k and u_k, respectively. The initial state is ξ_0, and the sampling time is T. The linearization is around the initial vehicle state ξ_0 and input $[0,0]^\top$. The linearized, discretized system matrices A and B [39] are given by

$$A = \begin{bmatrix} 1 & 0 & -Tv\sin z_1 & T\cos z_1 - \frac{z_2 \sin z_1}{2z_4} \\ 0 & 1 & Tv\cos z_1 & T\sin z_1 - \frac{z_2 \cos z_1}{2z_4} \\ 0 & 0 & 1 & \frac{T\tan\delta}{z_4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5a}$$

and

$$B = \begin{bmatrix} \frac{T^2\cos z_1}{2} & -\frac{T^2 v z_7 \sin z_1}{2} - \frac{z_8 \sin z_1}{z_9} \\ \frac{T^2\sin z_1}{2} & \frac{T^2 v z_7 \cos z_1}{2} + \frac{z_8 \cos z_1}{z_9} \\ \frac{T^2\tan\delta}{2z_4} & Tz_7 \\ T & 0 \end{bmatrix} \tag{2.5b}$$

with $z_1 = \psi + \arctan\left(\frac{l_r \tan\delta}{l_r+l_f}\right)$, $z_2 = T^2 v\tan\delta$, $z_3 = (l_r\tan\delta)^2$, $z_4 = \left(l_r+l_f\right)\left(\frac{z_3}{(l_r+l_f)^2}+1\right)^{\frac{1}{2}}$, $z_5 = v\left((\tan\delta)^2+1\right)$, $z_6 = \left(l_r+l_f\right)^3\left(\frac{z_3}{\left(l_r+l_f\right)^2}+1\right)^{\frac{3}{2}}$, $z_7 = \frac{z_5}{z_4} - \frac{z_3 z_5}{z_6}$, $z_8 = Tl_r z_5$ and $z_9 = \left(l_r+l_f\right)\left(\frac{z_3}{\left(l_r+l_f\right)^2}+1\right)$.

Note that the system matrices A and B here are different from that in the point-mass model (2.1),(2.2).

2.2. Vehicle Control

In this thesis, we employ Model Predictive Control (MPC) and Stochastic Model Predictive Control (SMPC) as vehicle controllers. MPC/SMPC aims to get the optimal tracking performance in terms of cost and constraints. The vehicle models introduced in Section 2.1 are used to generate predictions of the vehicle behaviors.

2.2.1. Model Predictive Control

Model Predictive Control (MPC) iteratively solves a constrained optimal-control problem on a finite prediction horizon [16], [18]–[20], [28]. That means a cost function is minimized while satisfying multiple constraints [16], [18]–[20], [28], including a system dynamic model used for generating predictions and constraints on states and control inputs. The optimal control problem that MPC solves at any current time t is given as

$$\min_{\boldsymbol{u}} V(\boldsymbol{\xi}, \boldsymbol{u}) \tag{2.6a}$$

$$\text{s. t. } \xi_{k+1} = \mathscr{F}(\xi_k, u_k),\ k = 0,1,\cdots,N-1, \tag{2.6b}$$

$$\xi_k \in \Xi,\ k = 1,2,\cdots,N, \tag{2.6c}$$

$$u_k \in \mathscr{U},\ k = 0,1,\cdots,N-1, \tag{2.6d}$$

where prediction time k counts from current time t on, with prediction horizon N. The control input sequence $\boldsymbol{u} = (u_0, u_1, \cdots, u_{N-1})^\top$ is optimized while minimizing the cost $V(\boldsymbol{\xi}, \boldsymbol{u})$. The sequence $\boldsymbol{\xi} = (\xi_0, \xi_1, \cdots, \xi_N)^\top$ contains all states over the prediction horizon N. $\mathscr{F}$ is the system dynamic model to generate predictions. The states ξ_k ($k = 1,2,\cdots,N$) and control inputs u_k ($k = 0,1,\cdots,N-1$) are constrained by sets Ξ and $\mathscr{U}$, respectively.

When applying MPC to control Autonomous Vehicles (AVs), the MPC formulation (2.6) is realized in a way that fulfills the requirements of autonomous driving. We consider the trajectory tracking and energy consumption in the cost function. A discrete-time vehicle model, e.g., the point-mass model and the kinematic bicycle model, is used as a system dynamic to generate behavior predictions of the controlled vehicle. The traffic rules, physical limitations of the vehicle, and safety are treated in constraints (2.6c) and (2.6d). Among them, the safety constraint is what we are focusing on in this thesis. To simplify the explanation, we separate the safety constraints from the overall state constraints (2.6c). All states that can ensure safety are summarized in set Ξ_{safe} and the safety constraints are therefore shown below

$$\xi_k \in \Xi_{\text{safe}},\ k = 1, 2, \cdots, N. \tag{2.6e}$$

The realization of the constraints, including the safety constraint, will be introduced in Chapters 3, 4 and 6.

Generally, in order to ensure safety, the controlled vehicle must predict the actions or behaviors of the surrounding vehicles to avoid colliding with them. However, accurately predicting the future behaviors of surrounding vehicles is challenging due to numerous unknown factors including their driving goals and the interactive behaviors of other vehicles. The imprecise prediction can bring uncertainties to collision avoidance constraints. Therefore, relying on the prediction can result in safety problems and suboptimal performances of the controller in general. Robust Model predictive control (RMPC) and stochastic model predictive control (SMPC) can take the effect of the prediction uncertainties into account [16]–[20]. RMPC considers the worst-case scenario and derives an optimal solution that guarantees that the constraints hold for any possible prediction uncertainties, which is over-conservative [17]. SMPC is less conservative, because it requires the constraints to be satisfied with a user-defined probability, allowing a small probability of constraint violations [16], [18]–[21]. Therefore, SMPC is selected to control autonomous vehicles in this thesis.

2.2.2. Stochastic Model Predictive Control

Stochastic Model Predictive Control (SMPC) provides a probabilistic framework for MPC with stochastic uncertainty [16], [18]–[21]. SMPC formulation includes chance constraints, which enables a trade-off between control performance and the probability of constraint violations. To obtain the SMPC formulation, a modification to the above-mentioned standard MPC is required. Specifically, the uncertainty-related hard constraints are transformed into chance constraints.

SMPC has been increasingly attractive in autonomous driving because of its capability to tackle the dynamics of vehicles, address constraints arising from the physical limitations of vehicles, and handle collision-avoidance requirements without being excessively conservative [16], [18]–[20], [32]. Generally, the collision-avoidance requirements are realized using probabilistic constraints [16], [19], [20], [32] where constraint violation with a specified small probability is allowed.

In our applications of SMPC on autonomous driving, the safety constraints are probabilistic because of the uncertainties caused by imprecise predictions of the surrounding vehicles. Therefore, we transform the safety constraints (2.6e) along the prediction horizon into the chance constraints below:

$$\mathbf{Pr}\left(\xi_k \in \Xi_{\text{safe}}\right) \geq p,\ p \in [0.5, 1],\ k = 1, \cdots, N. \tag{2.7}$$

Employing $\mathbf{Pr}(*) \geq p$, we ensure that the event $*$ occurs with a probability of not less than p. Incorporating this probabilistic constraint (2.7) into the formulation of MPC, we obtain the following optimal control problem that SMPC solves at any current time t:

$$\min_{\boldsymbol{u}} V(\boldsymbol{\xi}, \boldsymbol{u}) \tag{2.8a}$$

$$\text{s. t. } \xi_{k+1} = \mathscr{F}(\xi_k, u_k),\ k = 0, 1, \cdots, N-1, \tag{2.8b}$$

$$\xi_k \in \Xi,\ k = 1, \cdots, N, \tag{2.8c}$$

$$u_k \in \mathscr{U},\ k = 0, 1, \cdots, N-1, \tag{2.8d}$$

$$\mathbf{Pr}(\xi_k \in \Xi_{\text{safe}}) \geq p,\ p \in [0.5, 1],\ k = 1, 2, \cdots, N. \tag{2.8e}$$

2.3. Maximum Entropy Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) is a machine learning framework employed to infer the underlying reward or cost function that drives the behavior of an expert agent. Maximum Entropy IRL (ME-IRL) is a specific IRL approach. ME-IRL extends the basic IRL framework by incorporating the idea of maximum entropy and provides an alternative perspective that is particularly useful in scenarios involving uncertainty and stochasticity.

ME-IRL method has been employed to learn the driving style of vehicles [33], [40]–[43]. The driving style is quantified by a cost function represented by a linear combination of features that capture the important characteristics of the vehicle trajectories. The ME-IRL method aims to identify the weights of the features that best fit the driving style of the demonstration trajectory and reproduce trajectories that mimic the driving style of the expert demonstration [33], [42], [43]. The features together with their weights describe the driving style and can be used to measure the similarity between trajectories. How to define the features and learn feature weights is introduced as follows.

2.3.1. Features

We adopt the following six features for an individual AV from [33] and [44]. These features include accelerating, approaching or maintaining desired speed, and approaching or remaining in the target lane [33].

a) x-acceleration: $f_{\text{ax}} = \int_t \|\dot{r}^x(\tau)\|^2 \mathrm{d}\tau$.

b) y-acceleration: $f_{\text{ay}} = \int_t \|\dot{r}^y(\tau)\|^2 \mathrm{d}\tau$.

c) Desired velocity: $f_{\text{v}} = \int_t \|v^x_{\text{des}} - \dot{r}^x(\tau)\|^2 \mathrm{d}\tau$, where v^x_{des} is the desired velocity in the longitudinal direction.

d) Desired lane: $f_{\text{l}} = \int_t |l_{\text{des}} - r^y(\tau)| \mathrm{d}\tau$, where l_{des} is the desired lane.

e) Initial lane: $f_{\text{il}} = \int_0^{t_{\text{turn}}} |l_{\text{initial}} - r^y(\tau)| \mathrm{d}\tau$, where l_{initial} is the initial lane of the EV and t_{turn} is the time when the EV remains in l_{initial}.

f) End lane: $f_{\text{el}} = \int_{t_{\text{end}-1}}^{t_{\text{end}}} |l_{\text{target}} - r^y(\tau)| \mathrm{d}\tau$, where l_{target} is the target lane of the EV at the ending time t_{end}.

Therefore, the features can be recognized as functions of a trajectory $\boldsymbol{r}$. The trajectory $\boldsymbol{r}$ is represented using piecewise quintic spline segments (see Section 2.4). This ensures smoothness of trajectory $\boldsymbol{r}$, allowing for second-order derivative in the features. All features are collected in a feature vector $\boldsymbol{f}(\boldsymbol{r})$.

2.3.2. Learning of Feature Weights

We quantify the driving style using a cost function that is a linear combination of features, where each feature is multiplied by a weight. We summarize the weights of the features in a weight vector $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_m]^\top \in \mathbb{R}^m$, where m represents the number of the features. A trajectory can be reproduced by a learned weight vector $\boldsymbol{\theta}$, denoted as $\boldsymbol{r}_{\boldsymbol{\theta}}$. We aim to find the optimal feature weight $\boldsymbol{\theta}^*$ such that the features of the reproduced trajectory $\boldsymbol{r}_{\boldsymbol{\theta}^*}$ are closest to those of the demonstration trajectory $\boldsymbol{r}_{\mathscr{D}}$, i.e. [33],

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \varepsilon_{\boldsymbol{\theta}} \tag{2.9}$$

where $\varepsilon_{\boldsymbol{\theta}} = \left\| \boldsymbol{f}(\boldsymbol{r}^*_{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{r}_{\mathscr{D}}) \right\|_2$ is the learning error, $\boldsymbol{r}^*_{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{r}} L(\boldsymbol{\theta}, \boldsymbol{r})$, where $L(\boldsymbol{\theta}, \boldsymbol{r}) = \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{r})$ is a cost function that represents the driving style. This is a bilevel optimization problem [45] where we do not consider constraints. The optimal weight $\boldsymbol{\theta}$ can be found using an updating law $\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}}$ with a learning rate α, where gradient $\nabla_{\boldsymbol{\theta}}$ is approximated using [46]

$$\nabla_{\boldsymbol{\theta}} \approx \boldsymbol{f}(\boldsymbol{r}^*_{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{r}_{\mathscr{D}}) \tag{2.10}$$

The learning is conducted in an iterative manner, i.e., the weights $\boldsymbol{\theta}$ are used to derive a trajectory $\boldsymbol{r}_{\boldsymbol{\theta}}$ which is then used to produce the gradient $\nabla_{\boldsymbol{\theta}}$ for the update of $\boldsymbol{\theta}$. The iteration of the learning process terminates when the incremental learning error is smaller than a given threshold, i.e., $\left| \varepsilon^i_{\boldsymbol{\theta}} - \varepsilon^{i-1}_{\boldsymbol{\theta}} \right| < \bar{\varepsilon}$, where $\varepsilon^i_{\boldsymbol{\theta}}$ is the learning error at iteration $i \in \mathbb{N}^+$ and $\bar{\varepsilon} \in \mathbb{R}^+$ is a predefined threshold.

We summarize the learning process in Algorithm 1.

2.4. Trajectory Representation

We represent the trajectories in Inverse Reinforcement learning (IRL) related methods using piecewise quintic spline segments to ensure their smoothness. The spline segments are parameterized using control points comprising the positions, velocities, and accelerations in the longitudinal and lateral directions, respectively [47].

Control Points

We employ piecewise quintic splines $\boldsymbol{s}_j(t)$ to represent a trajectory $\boldsymbol{r}$, i.e., $\boldsymbol{r}(t) = \boldsymbol{s}_j(t)$ for $t \in [t_j, t_{j+1}]$, $j \in \{0, \cdots, N_s - 1\}$, where N_s denotes the number of spline segments. Spline $\boldsymbol{s}_j$ is parameterized employing the pair of control inputs $\boldsymbol{c}_j$ and $\boldsymbol{c}_{j+1}$ at interval $[t_j, t_{j+1}]$ [33]. Control input $\boldsymbol{c}_j$ is comprised of positions, velocities and accelerations in the longitudinal and lateral directions at

Algorithm 1 Learning from demonstration trajectory $\boldsymbol{r}_{\mathscr{D}}$

Input: Demonstration trajectory $\boldsymbol{r}_{\mathscr{D}}$ and threshold $\bar{\varepsilon}$ for ending learning process
Output: Optimal feature weights $\boldsymbol{\theta}^*$

Compute the feature $\boldsymbol{f}(\boldsymbol{r}_{\mathscr{D}})$ of trajectory $\boldsymbol{r}_{\mathscr{D}}$
Initialize all elements of $\boldsymbol{\theta}$ to 1, learning rate α to 0.01, learning error $\varepsilon^0_{\boldsymbol{\theta}}$ to 0, the counter of the learning iterations i to 1, and guess of trajectory $\boldsymbol{r}$
while $i \neq 0$ **do**
    Optimize trajectory $\boldsymbol{r}$ by minimizing cost function $L(\boldsymbol{\theta},\boldsymbol{r}) = \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{r})$, obtaining $\boldsymbol{r}^*_{\boldsymbol{\theta}}$
    Calculate the approximated gradient $\nabla_{\boldsymbol{\theta}} \approx \boldsymbol{f}\left(\boldsymbol{r}^*_{\boldsymbol{\theta}}\right) - \boldsymbol{f}(\boldsymbol{r}_{\mathscr{D}})$
    Obtain learning error $\varepsilon^i_{\boldsymbol{\theta}} = \|\nabla_{\boldsymbol{\theta}}\|_2$
    if $\left|\varepsilon^i_{\boldsymbol{\theta}} - \varepsilon^{i-1}_{\boldsymbol{\theta}}\right| < \bar{\varepsilon}$ **then**
        break
    else
        $i = i+1$
        Update weight vector using updating law $\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha\nabla_{\boldsymbol{\theta}}$
    end if
end while
return $\boldsymbol{\theta}^* = \boldsymbol{\theta}$

interval $[t_j, t_{j+1}]$ [33],

$$\boldsymbol{c}_j = \begin{bmatrix} \boldsymbol{c}^x_j \\ \boldsymbol{c}^y_j \end{bmatrix} = \begin{bmatrix} [r^x_j & v^x_j & a^x_j]^\top \\ [r^y_j & v^y_j & a^y_j]^\top \end{bmatrix}, \tag{2.11}$$

where r^x_j, v^x_j and a^x_j are the position, velocity, and acceleration of the vehicle in the longitudinal direction, respectively; r^y_j, v^y_j and a^y_j are the lateral counterparts. All control points $\boldsymbol{c}_j$ ($j \in \{0,\cdots,N_s\}$) at any time t_j constitute the set of control points $\bar{\boldsymbol{c}} = [\boldsymbol{c}_0^\top\ \boldsymbol{c}_1^\top\ \cdots\ \boldsymbol{c}_{N_s}^\top]^\top \in \mathbb{R}^{6(N_s+1)}$. The control point $\boldsymbol{c}_0$ is fixed during the learning process. How we get control points for demonstration trajectories and reproduced trajectories in this thesis will be introduced in Section 5.3.4 and Section 6.2.3.

Quintic Polynomials

To define a 2D quintic polynomial [48] of spline

$$\boldsymbol{s}_j(\tau) = \begin{bmatrix} q^x_5\tau^5 + q^x_4\tau^4 + \cdots + q^x_1\tau + q^x_0 \\ q^y_5\tau^5 + q^y_4\tau^4 + \cdots + q^y_1\tau + q^y_0 \end{bmatrix} \tag{2.12}$$

where $\tau \in [t_j, t_{j+1}]$, twelve coefficients $q^x_5,\cdots,q^x_0,q^y_5,\cdots,q^y_0$ are required. The coefficients $q^x_5,\cdots,q^x_0$ for the longitudinal direction can be obtained using control points $\boldsymbol{c}^x_j = [r^x_0, v^x_0, a^x_0]$ at time t_j and $\boldsymbol{c}^x_{j+1} = [r^x_T, v^x_T, a^x_T]$ at time t_{j+1}, as shown in equations (7) and (8) in [48]. If we replace the longitudinal part of the control points with their lateral parts, we obtain coefficients $q^y_5,\cdots,q^y_0$. For spline $\boldsymbol{s}_{j+1}$, we use control points $[\boldsymbol{c}_{j+1}, \boldsymbol{c}_{j+2}]$ to calculate the coefficients. Since two adjacent spline segments share control points, we have continuous velocity $\boldsymbol{v}(t)$ and acceleration $\boldsymbol{a}(t)$ along the entire trajectory. Given the above, the set of all control points $\bar{\boldsymbol{c}}$ parameterizes the trajectory $\boldsymbol{r}$ consisting of N_s spline segments.

2.5. Evaluation Metrics

This section introduces evaluation metrics employed to assess the control efforts of the controller as well as the precision of the prediction methods used in our controller. For evaluating the controller, we consider control efforts that measure the effort invested in implementing the control inputs including acceleration and steering (AccEff and SteEff). The metrics for evaluating prediction methods, particularly their precision, consist of Root Mean Squared Error (RMSE) and Average Displacement Error (ADE).

2.5.1. Evaluation of Controller

The control efforts used to evaluate the controllers are introduced as follows.

Control Efforts

'Control Effort' quantitatively measures how much effort is required to conduct the control inputs, e.g., acceleration and steering, along the entire trajectory. Control efforts can reflect the driving experience and energy consumption. Low efforts generally indicate a seamless driving experience and reduced energy consumption. The control efforts of acceleration and steering [49] are given as follows.

$$\mathrm{AccEff}(a) = \frac{1}{K}\frac{1}{\Delta a}\sum_{k=1}^{K}|a_k| \tag{2.13}$$

$$\mathrm{SteEff}(\delta) = \frac{1}{K}\frac{1}{\Delta \delta}\sum_{k=1}^{K}|\delta_k| \tag{2.14}$$

where AccEff and SteEff represent the acceleration effort and steering effort, respectively. The acceleration and steering angle of the vehicle at each time step k are denoted using a_k and δ_k, respectively, while K is the number of time steps along the entire trajectory. The parameters Δa and $\Delta\delta$ represent the nominal ranges of the acceleration and the steering angle, which are given in $\Delta a = |a_{\max} - a_{\min}|$ and $\Delta\delta = |\delta_{\max} - \delta_{\min}|$, respectively. Here, $a_{\max}$ is the upper bound of the acceleration and $a_{\min}$ is the lower bound; $\delta_{\max}$ and $\delta_{\min}$ are the upper and lower bounds of the steering angle, respectively. Smaller values of AccEff and SteEff imply lower energy consumption and more comfortable driving experiences.

2.5.2. Evaluation of Prediction

We introduce the metrics for evaluating the precision and reliability of prediction methods below.

Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) [50] is commonly used to measure the accuracy of a prediction model. It measures the difference between the predicted and the actual values, here trajectories, using the following formula:

$$\mathrm{RMSE}(\boldsymbol{r}_{\mathrm{p}}, \boldsymbol{r}_{\mathrm{r}}) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(r_{\mathrm{p}k} - r_{\mathrm{r}k})^2} \tag{2.15}$$

where $\boldsymbol{r}_{\mathrm{p}}$ and $\boldsymbol{r}_{\mathrm{r}}$ are the predicted trajectory and the real trajectory, respectively. K represents the number of time steps along the entire trajectory. The positions on the predicted and real trajectories at time step k are denoted by $r_{\mathrm{p}k}$ and $r_{\mathrm{r}k}$, respectively. RMSE measures the gap/error between the predicted and actual values. The smaller the RMSE, the smaller the gap/error, meaning better prediction accuracy. However, the weakness of RMSE is that it doesn't distinguish between positive and negative gaps/errors, and the calculation of the square root makes the effect of the large gaps/errors on the overall result discounted.

Average Displacement Error (ADE)

Average Displacement Error (ADE) [51] measures how accurately the prediction method predicts the vehicle trajectory over time. Lower values of ADE indicate better prediction performance, as they indicate smaller discrepancies between the predicted and actual trajectories. ADE is defined as follows: $\|a\|$

$$\mathrm{ADE}(\boldsymbol{r}_{\mathrm{p}},\boldsymbol{r}_{\mathrm{r}}) = \frac{1}{K}\sum_{k=1}^{K}\left\|r_{\mathrm{p}k} - r_{\mathrm{r}k}\right\|_2 \tag{2.16}$$

where the notations $\boldsymbol{r}_{\mathrm{p}}$, $\boldsymbol{r}_{\mathrm{r}}$, K, k, $r_{\mathrm{p}k}$, $r_{\mathrm{r}k}$ represent the same as that in RMSE in (2.15). The greater the ADE value, the lower the prediction accuracy.

2.6. Conclusions

In this chapter, we introduced the preliminaries that will be employed in our approaches in this thesis. We first introduced the vehicle models including the point-mass model and kinematic bicycle model. This is followed by the introduction of the vehicle controller, Model Predictive Control (MPC) and Stochastic Model Predictive Control (SMPC), where how to use the vehicle models in the controllers is also described. After that, we presented Maximum Entropy Inverse Reinforcement Learning (ME-IRL), which will be used to learn the driving style of vehicles in this thesis. Moreover, a trajectory representation method that can ensure the smoothness of the trajectories in the ME-IRL-related methods is introduced. In the end, the metrics we will use to evaluate our control approaches are exhibited.

Maneuver Planning for Autonomous Vehicles controlled by MPC

3.

Maneuver planning that involves high-level decisions such as selecting desired lanes and speeds plays a crucial role in enabling safe, efficient, and comfortable autonomous driving. Generally, for a vehicle driving on a multi-lane road, there are several potential maneuvers in both longitudinal and lateral directions. Selecting the best maneuver from the various options represents a significant challenge.

In this chapter, we describe our maneuver planning method for autonomous vehicles controlled by MPC. We propose a maneuver selection algorithm and combine it with a trajectory generation algorithm, which is based on model predictive control (MPC). The maneuver selection method is a higher-level planner, which selects only one maneuver from all possible maneuvers based on the current situation and delivers it to a lower-level MPC-based trajectory tracking controller. The effectiveness of the proposed algorithm is validated by simulating an overtaking scenario on a multi-lane highway. The content of this chapter was published in [28].

3.1. Introduction

Planning appropriate maneuvers and tracking reference trajectories are fundamental tasks for an autonomous vehicle and thus must be incorporated in a control framework for autonomous vehicles. In this chapter, a maneuver (e.g., lane changing and decelerating) results from a higher-level planner. Our approach selects the maneuver that the controlled vehicle will execute at the next time step, and the selected maneuver is input to the lower-level trajectory tracking controller.

Trajectory planning techniques for autonomous vehicles can be broadly classified into four groups [11]: sampling based planning [52], [53], interpolating curve planning [54]–[58], graph search based planning [10], [59]–[61], and numerical optimization [62]–[65]. Sampling-based planning algorithms [52], [53], [66] plan trajectories by randomly sampling the configuration space and finding connections [11]. These methods can provide fast solutions and have therefore been used in self-driving vehicles [66]–[68]. However, the trajectories are not continuous and therefore are uncomfortable for passengers. Interpolating curve planning methods, e.g. using Clothoid curves, polynomial curves, Bézier curves, or Spline curves [54]–[58], deliver smoother paths [11]. These methods are e.g. used in autonomous driving when comfort and safety are major concerns and the driving environment is structured and modelled [11]. However, in general it is difficult to obtain a global model of the environment. Graph search based planning methods, e.g. applying Dijkstra's algorithm [59], [60], the A* algorithm [10], [61], or the D* algorithm [11], [69], determine a path from one point to another. However, these methods are not suitable for real time applications, where dynamic obstacles present one major challenge. Numerical optimizations [62]–[65] obtain the optimal trajectory by solving a constrained optimization problem. These methods can deal with constraints and uncertainties, so the controlled vehicle can take other traffic participants'

behavior into consideration by setting appropriate collision avoidance constraints [11]. Additional advantages for application in autonomous driving are that they can improve comfort by setting constraints on control inputs, such that the global road information is not needed when combined with a receding horizon strategy, as e.g. in MPC, and these methods are applicable in real time and can avoid moving obstacles. However, the computational complexity of MPC is a challenge, because the optimization of the cost functions needs to be carried out at every time step for a potentially long prediction horizon [62].

Trajectory planning methods based on MPC have been proposed e.g. in [62], [70]–[72]. In [62], multiple feasible maneuvers are obtained using the maneuver planning method and these maneuvers are delivered to the lower-level MPC controller for trajectory control. However, delivering multiple maneuvers to the lower-level MPC might cause unnecessary computational burden. This has motivated us to identify a better maneuver selection method, which finally selects only one maneuver from all feasible/possible maneuvers, and then delivers this maneuver to the lower-level MPC controller.

3.1.1. Approach Overview

We propose a high-level maneuver planning method and combine it with a low-level MPC-based trajectory control [28]. We summarize the relation between the maneuver planning and the trajectory control in a framework, shown in Figure 3.1. The framework consists of four main modules: scenario perception, maneuver planning, MPC-based trajectory control, and action. Our main concerns here are maneuver planning and MPC-based trajectory control. The maneuver planning module contains two consecutive sub-modules: maneuver generation and maneuver selection. Similarly, the MPC-based trajectory control consists of two sub-modules: MPC-based reference trajectory generation and MPC-based trajectory tracking.

Our method is based upon the method presented in [13]. We utilize the maneuver generation method from [13] and improve it to make it more applicable in reality by removing the necessity to explore too many trajectories. Then we employ an MPC controller instead of the sliding mode controller for tracking to avoid high switching frequencies and to be able to take into account constraints.

Our maneuver planning and trajectory control method works as follows: at each time step, 1) we generate all possible maneuvers employing the maneuver generation method from [13]; then, 2) select the desired lane and the longitudinal speed maneuver using our maneuver selection method; after that, the trajectory control module 3) generates a reference trajectory based on the selected maneuver, and 4) tracks it. More specifically, a group of candidate maneuvers is initially generated with respect to the environment in which the method in [13] is employed. Next, we exclude maneuvers that contribute less to meeting the criteria for the best maneuver from these candidate maneuvers. The criteria for excluding maneuvers involve the goal lane, the roadside edges, the time to collision (TTC) [13], [73], and the intervehicular time (TIV) [13]. In the trajectory control module, the desired reference trajectory is generated by adjusting the cost function and constraints in the MPC controller to account for the selected maneuver. The reference trajectory is then tracked by MPC which, in addition, ensures that no constraints are violated.

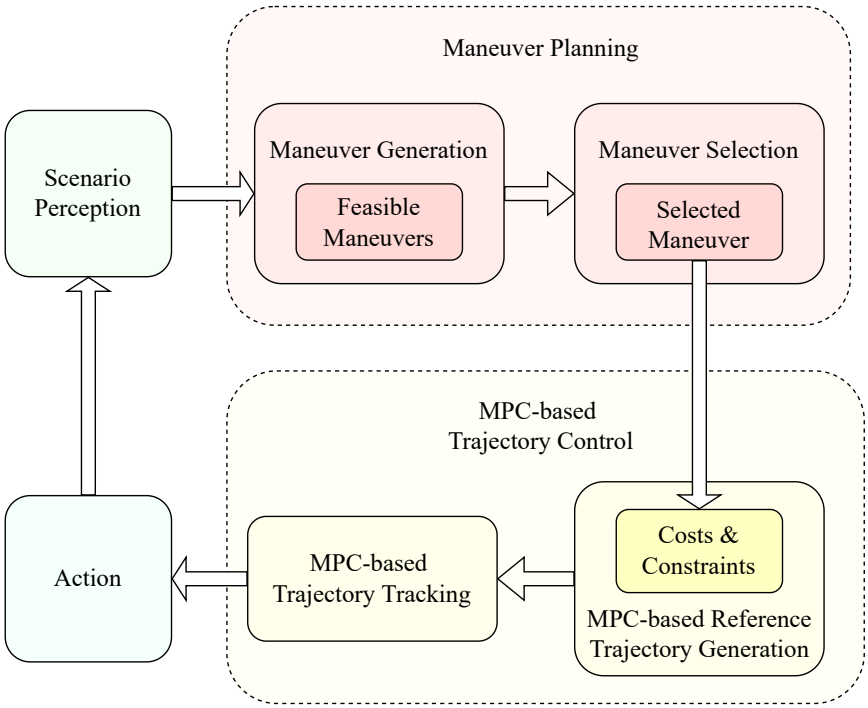


Figure 3.1.: Interconnection of maneuver planning and trajectory tracking

3.1.2. Chapter Overview

The rest of this chapter is organized as follows. Section 3.2 introduces the maneuver planning method, including maneuver generation and maneuver selection. In Section 3.3, we explain how MPC is applied by recalling the optimization problem of the MPC controller introduced in Section 2.2.1, as well as introducing constraints, cost function, and the trajectory generation for maneuvers. Simulation results are presented in Section 3.4, and Section 3.5 summarizes Chapter 3 and provides prospects for future research.

3.2. Maneuver Planning

In autonomous driving, maneuvers are the high-level strategies that describe what the vehicle will do in the short term or long term. For example, lane changing is a short-term goal and overtaking is a long-term goal. In this chapter [28], the maneuver planning method is executed at each time step. This maneuver planning method determines what the controlled vehicle will do in the next time step, and we assume that the controlled vehicle knows the long-term goal. Our method is designed for scenarios consisting of two vehicles, where the controlled vehicle is the ego vehicle (EV), and the other one is the target vehicle (TV). The EV assumes that the TV will maintain its current velocity at each time step. In this chapter, we make the common assumption that the position and velocity of the TV can be observed by the EV [20], [74], [75].

Maneuver planning consists of maneuver generation and maneuver selection (see Figure 3.1). For the maneuver generation, we use an approach presented in [13] and add a novel approach for maneuver selection that first selects only one maneuver from the generated options and then provides the maneuver information to the lower-level controller. In [13] the lower-lever controller is

a second-order sliding-mode controller. Ideal sliding modes require high switching frequencies, which can burden the actuators and cause rapid wear, and, therefore, lead to safety issues. We propose to replace the sliding mode controller by MPC to combine trajectory generation and tracking. This might again improve safety in addition to safe maneuver selection by enforcing that no safety constraints are violated.

The main task of the maneuver selection sub-module is to select the appropriate lane in the lateral direction and the speed strategy in the longitudinal direction (see Section 3.2.3). Additionally, the safety criteria used in selecting longitudinal maneuvers are presented in Section 3.2.2.

3.2.1. Maneuver Generation

In [13], a vehicle driving on a multi-lane road is considered. Before a trajectory is generated, nine feasible maneuvers are generated by a combination of both lateral and longitudinal directions. In more detail, three feasible maneuvers can be chosen in the longitudinal direction: decelerating (DE), maintaining current speed (CS), accelerating (AC); meanwhile, there are three possible maneuvers in the lateral direction: changing to the left lane (LCL), keep moving in the current lane (LK), or changing to the right lane (LCR). Combining the maneuvers in both directions, nine combined maneuvers [13] can be obtained, as shown in Figure 3.2.

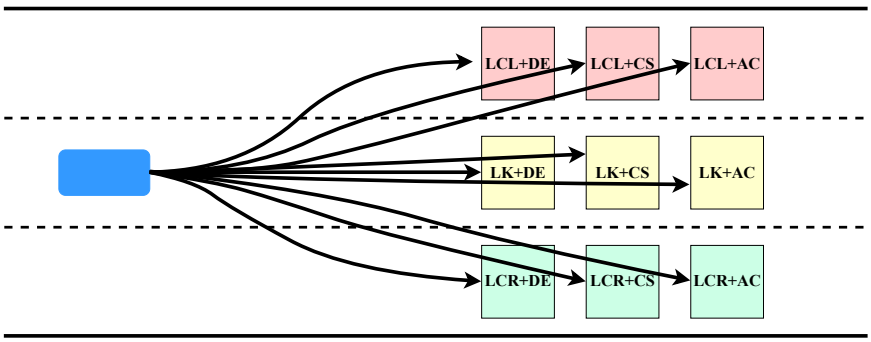


Figure 3.2.: Nine feasible combined maneuvers. Boxes with the same color contain three different maneuvers in the longitudinal direction: decelerating (DE), staying at/maintaining current speed (CS), and accelerating (AC). Boxes with different colors represent distinct maneuvers in the lateral direction. The red, yellow, and green boxes illustrate changing to left lane (LCL), keeping/continue moving in current lane (LK), and changing to right lane (LCR), respectively.

High-level maneuvers are then delivered to the lower-level controller for trajectory tracking. If the maneuver planning method delivers multiple feasible maneuvers to the controller, the controller will have to experience superfluous computational burden. Additionally, it is even worse when the controller is computationally expensive, for instance, MPC. Autonomous vehicles can benefit from the ability of MPC to deal with constraints. However, in order to efficiently use MPC as a low-level controller, we have to reduce the computational burden. Our maneuver selection method contributes to this by selecting only one maneuver during maneuver planning and, therefore, delivering only one maneuver to the lower-level MPC controller, as shown in Section 3.2.3.

3.2.2. Safety Criteria

When selecting one maneuver from all feasible ones, safety is one of our major concerns in both longitudinal and lateral directions. The safety criteria we use in selecting the maneuver in the longitudinal direction are the time to collision (TTC) and the intervehicular time (TIV). The risk of collision is higher for 1) high speed differences and 2) low intervehicle distances [13]. TTC [13], [73] and TIV [13] can be used to quantify this risk.

TTC was first proposed by Hayward in [73] to measure the risk level of the two-vehicle scenarios where the two cars are close to each other and have different velocities. TTC can deal with both lane-keeping scenarios and lane-chaning scenarios [73]. However, there exists a dangerous situation that cannot be detected by TTC: when two vehicles, at the same speed, are close to each other. In this situation, the TTC is not small, even though the two vehicles have a high probability of colliding. In order to also take this situation into consideration, the intervehicular time (TIV) is introduced [13].

TTC

In [73], Hayward gave the definition of TTC: TTC represents the time required for two vehicles to collide if they maintain their current longitudinal velocities. TTC is formulated as follows:

$$\mathrm{TTC} = \frac{|\Delta x_{\mathrm{EV,TV}}|}{|\Delta v_{\mathrm{EV,TV}}|} \tag{3.1}$$

where $\Delta x_{\mathrm{EV,TV}}$ is the relative distance between the EV and the TV in the longitudinal direction with $\Delta x_{\mathrm{EV,TV}} = x_{\mathrm{EV}} - x_{\mathrm{TV}}$, where x_{EV} and x_{TV} represent the longitudinal positions of the EV and TV, respectively. We denote with $\Delta v_{\mathrm{EV,TV}}$ the relative velocity between these two vehicles in the longitudinal direction with $\Delta v_{\mathrm{EV,TV}} = v_{\mathrm{EV}} - v_{\mathrm{TV}}$, where v_{EV} and v_{TV} are the longitudinal velocities of the EV and TV, respectively.

In general, a great TTC stands for a safer situation. In [13], if the TTC is greater than 10s, it is supposed that there is no interaction between two vehicles. If the TTC is smaller than 1.5s, the two vehicles are judged to be interacting intensely with each other, viewed as a risky situation that activates a warning system.

TIV

The TIV [13] can deal with the risky situation that TTC fails to detect: two vehicles with the same (or similar) high speed are close to each other in the same lane. In this situation, the TTC can be great because of the relative velocity between the two vehicles is zero (or very small). However, this is actually a risky situation, because once the leading vehicle decelerates or the following vehicle accelerates, they will most probably collide. The TIV is defined as follows:

$$\mathrm{TIV} = \frac{|\Delta x_{\mathrm{EV,TV}}|}{v^*} \tag{3.2}$$

where $\Delta x_{\mathrm{EV,TV}}$ is the same as that in TTC, and v^* is the velocity of the vehicle which is following the other vehicle in the longitudinal direction. Here, $v^* = v_{\mathrm{EV}}$ if the EV is behind the TV, and $v^* = v_{\mathrm{TV}}$ if the EV is in front of the TV. Great TIV represents safer situations. According to [13], a TIV of 2s is commonly used as a boundary for safety.

Great TTC and TIV indicate a lower risk level [13]. In this chapter [28], a situation with great TTC and TIV at the same time is considered safe. Additionally, TTC and TIV might consider the following safe situation dangerous: two vehicles without the intention of changing lanes are driving in different lanes but close to each other in the longitudinal direction. However, this false danger can be recognized with further knowledge of the lateral positions of the vehicles.

In the maneuver selection, the aim of the selection of the longitudinal maneuver is to choose a maneuver with low risk, so the maneuver which can lead to greater TTC and TIV will be selected.

3.2.3. Maneuver Selection

The maneuver selection method selects only one maneuver from all possible maneuvers obtained from the maneuver generation method in [13]. The main idea of our maneuver selection method is: remove less appropriate maneuvers from all possible ones obtained from the maneuver generation based on selection criteria (details will be shown later); therefore, only one maneuver is kept at the end of the maneuver selection.

Our maneuver selection method has two tasks: i) selecting a lane choosing maneuver in the lateral direction and ii) obtaining a speed generating maneuver in the longitudinal direction. The maneuver selection method is shown in Figure 3.3, where the tasks i) and ii) are colored green and yellow, respectively. These two tasks in the maneuver selection method will be explained in detail below.

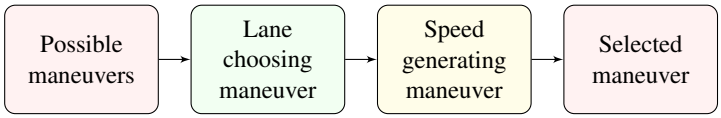


Figure 3.3.: The maneuver selection method.

Lane choosing maneuver in the lateral direction

In the lane choosing maneuver in the lateral direction, we remove the less appropriate maneuvers based on the constraints from the road edges and the current goal lane of the vehicle. The process is shown as follows:

1. *Remove the lateral maneuvers that will cause the road edge constraints to be violated.* For instance, if the vehicle is in the rightmost lane of the road, the maneuver of changing to the right lane (LCR) is inadmissible.

2. *Exclude the lateral maneuvers with which the vehicle is not heading towards* the goal lane. Here, we consider two cases: a) the goal lane is the current lane; b) the goal lane is a different lane. In case a), we simply remove the lane-changing maneuvers, LCL and LCR.

   Case b) is more complicated. If the goal lane is the lane at the left (right) side of the current lane, we first exclude the lateral maneuver with which the vehicle turns to the opposite/wrong direction, LCR (LCL) is therefore removed. Then, we consider whether the lane-changing maneuver is satisfied or not. If the conditions for changing lanes are satisfied, we exclude the LK maneuver; otherwise the lane-changing maneuver LCL (LCR) is removed.

After the lane choosing maneuver, only one maneuver in the lateral direction remains. Thus, before the only longitudinal maneuver has been selected from all three possible ones, there are three combined maneuvers left.

Speed generating maneuver in the longitudinal direction

To select the longitudinal maneuver, the criteria used to remove the less appropriate maneuvers from all possible maneuvers are safety, efficiency, and smoothness. Among them, safety is our prime concern. If, after all less safe maneuvers have been removed, there is more than one speed maneuver left, we take into account the efficiency and smoothness to, finally, retain just one maneuver.

The safety criteria we use are TTC and TIV. Great TTC and TIV correspond to safer situations. Therefore, maneuvers that contribute to increasing TTC and TIV are kept in the selection process. The process of using TTC and TIV as criteria to select the longitudinal maneuver/maneuvers is shown in Table 3.1.

Table 3.1.: Selection of longitudinal maneuver based on the safety criteria TTC and TIV

$\Delta x_{\mathrm{EV,TV}}$	$\Delta v_{\mathrm{EV,TV}}$	Possible maneuver	$\lvert\Delta x_{\mathrm{EV,TV}}\rvert$	$\lvert\Delta v_{\mathrm{EV,TV}}\rvert$	v^*	TTC	TIV	Result
$\Delta x_{\mathrm{EV,TV}} < 0$	$\Delta v_{\mathrm{EV,TV}} < 0$	DE	↑	↑	↓	?	↑	CS
		CS	↑	–	–	↑	↑	
		AC	↑	↓	↑	↑	↓	
	$\Delta v_{\mathrm{EV,TV}} > 0$	DE	↓	↓	↓	?	?	DE
		CS	↓	–	–	↓	↓	
		AC	↓	↑	↑	↓	↓	
	$\Delta v_{\mathrm{EV,TV}} = 0$	DE	↑	↑	↓	/	↑	DE
		CS	–	–	–	/	–	
		AC	↓	↑	↑	/	↓	
$\Delta x_{\mathrm{EV,TV}} > 0$	$\Delta v_{\mathrm{EV,TV}} < 0$	DE	↓	↑	–	↓	↓	AC
		CS	↓	–	–	↓	↓	
		AC	↓	↓	–	?	↓	
	$\Delta v_{\mathrm{EV,TV}} > 0$	DE	↑	↓	–	↑	↑	DE/CS
		CS	↑	–	–	↑	↑	
		AC	↑	↑	–	?	↑	
	$\Delta v_{\mathrm{EV,TV}} = 0$	DE	↓	↑	–	/	↓	AC
		CS	–	–	–	/	–	
		AC	↑	↑	–	/	↑	

In Table 3.1, the first two columns show the current relative position and velocity of the two vehicles. The third column enumerates the possible maneuvers obtained from maneuver generation. The fourth, fifth and sixth columns demonstrate how $|\Delta x_{\mathrm{EV,TV}}|$, $|\Delta v_{\mathrm{EV,TV}}|$ and v^* will change if the specific maneuver has been executed. The seventh and eighth columns show the trends of TTC and TIV, respectively, with the change of the $|\Delta x_{\mathrm{EV,TV}}|$, $|\Delta v_{\mathrm{EV,TV}}|$ and v^*. The last column shows the longitudinal maneuver/maneuvers left after removing maneuvers based on TTC and TIV.

We use labels ('↑', '↓', '−', '?', and '/') to briefly represent the trend of the change in the case of a specific maneuver. '↑' means the value of corresponding item will increase; '↓' means it is decreasing; '−' means that there will be no change in terms of the corresponding value; '?' means the future change is unknown; '/' shows that we do not calculate the TTC when $\Delta v_{\mathrm{EV,TV}} = 0$.

In the following, we will discuss how to choose the longitudinal maneuver in two distinct situations according to the relative positions of the two vehicles in the longitudinal direction: the EV is behind the TV, and the EV is in front of the TV.

1. The EV is behind the TV; $\Delta x_{\mathrm{EV,TV}} < 0$.
   - $\Delta v_{\mathrm{EV,TV}} < 0$. The velocity of the EV is smaller than that of the TV.
     - When the EV maintains current longitudinal speed (CS), the distance between the two vehicles increases. Thus, both TTC and TIV increase as time goes on.
     - The EV's choice of deceleration (DE) will cause the distance between the two vehicles ($|\Delta x_{\mathrm{EV,TV}}|$) to increase, but also cause the relative velocity ($|\Delta v_{\mathrm{EV,TV}}|$) to increase. Thus, the effect of deceleration (DE) on TTC is unknown.
     - Choosing acceleration (AC) will cause the TIV to decrease. Thus, we will not keep the AC.

     Therefore, maintaining current longitudinal speed (CS) is the best choice.
   - $\Delta v_{\mathrm{EV,TV}} > 0$. The velocity of the EV is greater than that of the TV, which is dangerous.
     - Both maintaining current speed (CS) and acceleration (AC) will definitely cause a decrease of the TTC and TIV.
     - However, by executing deceleration (DE), TTC and TIV will probably trend upward.

     Consequently, deceleration (DE) is selected as the longitudinal maneuver performed in this situation.
   - $\Delta v_{\mathrm{EV,TV}} = 0$. The two vehicles have the same longitudinal speed.
     - Under this circumstance, only TIV is used to estimate the risk.
     - Whether the distance between the two vehicles is small or not, decreasing the speed of the EV is a safe maneuver, which contributes to obtaining a greater TIV.

     Thus, deceleration (DE) is selected for the EV in this situation.

2. The EV is in front of the TV; $\Delta x_{\mathrm{EV,TV}} > 0$.
   - $\Delta v_{\mathrm{EV,TV}} < 0$. The velocity of the EV is less than that of the TV.
     - Not only deceleration (DE) but also maintaining current speed (CS) will cause a decreasing trend of the TTC and TIV.
     - Additionally, when choosing acceleration (AC), TIV will experience a rapid drop, while TTC might show an upward trend.

     This is not a safe situation, but increasing the speed of the EV will probably contribute to avoiding possible collisions, so AC is taken as the longitudinal maneuver.
   - $\Delta v_{\mathrm{EV,TV}} > 0$. The velocity of the EV is greater than that of the TV.

  - Both decelerating (DE) and maintaining current speed (CS) are beneficial to increasing TTC and TIV, so they can be taken as candidate maneuvers.
  - Furthermore, DE has a negative effect on the efficiency and smoothness.

  Therefore, we finally keep CS as the longitudinal maneuver in this situation.

- $\Delta v_{\mathrm{EV,TV}} = 0$. The EV and TV drive at same speed.
  - In this situation, TTC is not calculated and only TIV is considered as a safety criterion.
  - Increasing the speed (AC) will help obtain greater TIV.
  - Neither decreasing (DE) nor maintaining current speed (CS) will contribute to increasing TIV.

  Therefore, acceleration (AC) is selected.

All the situations we discuss require that the TV be within the detectable area of the EV. This is also mathematically described as $|\Delta x_{\mathrm{EV,TV}}| \leq \Delta x_{\max}$, where $\Delta x_{\max}$ is the longitudinally maximum detectable distance.

After excluding less appropriate maneuvers in both lateral and longitudinal directions, only one longitudinal maneuver and one lateral maneuver are left. Combining the selected maneuvers in both directions, we eventually obtain a single maneuver.

3.3. Model Predictive Control

The maneuver selected by the maneuver selection method is delivered to the lower-level MPC-based trajectory control module, which contains two sub-modules, reference trajectory generation and trajectory tracking, as shown in Figure 3.1.

MPC is used to generate and track trajectories because it will not only consider a cost function and thus allow for energy efficiency but also take into account collision avoidance constraints and thus further contribute to safety. As a low-level trajectory tracking controller, MPC realizes the maneuver selected from high-level maneuver planning by setting reference states in the cost function and by adjusting constraints.

Reference trajectory generation is the first step in the MPC-based trajectory control module. However, maneuver planning and trajectory planning are coupled by specifying parameters in the cost function and in the constraints of MPC, so we will define a model, a cost function and constraints for MPC before we introduce the trajectory generation.

3.3.1. Optimization Problem of the MPC Controller

At each time step, MPC minimizes an optimal control cost function with respect to constraints on a short prediction horizon to obtain the current control input. A dynamic system model is used to predict states in the prediction horizon for different choices of control input trajectories which finally allows to obtain the optimal inputs.

As we introduced in Section 2.2.1, the constrained optimization problem that MPC solves at each time step is shown in (2.6), which consists of the system model, constraints and cost function,

as shown below.

$$\begin{aligned} & \min_{\boldsymbol{u}} V(\boldsymbol{\xi}, \boldsymbol{u}) \\ \text{s. t. } & \xi_{k+1} = \mathscr{F}(\xi_k, u_k),\ k = 0, 1, \cdots, N-1, \\ & \xi_k \in \Xi,\ k = 1, 2, \cdots, N, \\ & u_k \in \mathscr{U},\ k = 0, 1, \cdots, N-1, \\ & \xi_k \in \Xi_{\text{safe}},\ k = 1, 2, \cdots, N. \end{aligned}$$

In this chapter, we choose the point-mass model demonstrated in Section 2.1.1 as the system model $\xi_{k+1} = \mathscr{F}(\xi_k, u_k)$ to predict the future states of the vehicles controlled by MPC. The point-mass model is

$$\xi_{k+1} = A\xi_k + Bu_k$$

where the state vector and the input vector are $\xi_k = \left[x_k, y_k, v_{x,k}, v_{y,k}\right]^\top$ and $u_k = \left[a_{x,k}, a_{y,k}\right]^\top$, respectively. System matrices A and B are given in (2.2).

We will discuss the constraints and the cost function of MPC for autonomous driving in the following subsections.

3.3.2. Constraints

The constraints are used to enforce restrictions on states and inputs. For our autonomous driving application, we include constraints for traffic rules, collision avoidance, and physical limitations of vehicles.

State Constraints

This part contains the states' constraints for obeying traffic rules and avoiding collisions.

- Constraints for Traffic Rules:

$$x_k \in [0, +\infty] \tag{3.4}$$

$$y_k \in [w_{\text{veh}}/2, m w_{\text{lane}} - w_{\text{veh}}/2] \tag{3.5}$$

$$v_{x,k} \in [v_{x_{\min}}, v_{x_{\max}}] \tag{3.6}$$

$$v_{y,k} \in [v_{y_{\min}}, v_{y_{\max}}] \tag{3.7}$$

  where w_{veh} is the width of vehicles, w_{lane} is the width of each lane, and m is the number of lanes. As shown in (3.4), the vehicle can move forward freely in the longitudinal direction. (3.5) is designed for keeping the vehicle within the road edges. Constraints (3.6) and (3.7) give the upper and lower limits to velocities in longitudinal and lateral directions, respectively.

- Safety constraint:
  Here, we present a realization of the safety constraint $\xi_k \in \Xi_{\text{safe}}$ shown in (2.6e) of Section 2.2.1. In order to avoid collisions, a region around each vehicle is defined that other vehicles are not allowed to enter [16], [19]. The region can be any convex shape larger than the shape

of the vehicle [16]. As in [19], [76]–[78], an elliptic region is chosen for our implementation, as shown in Figure 3.4.

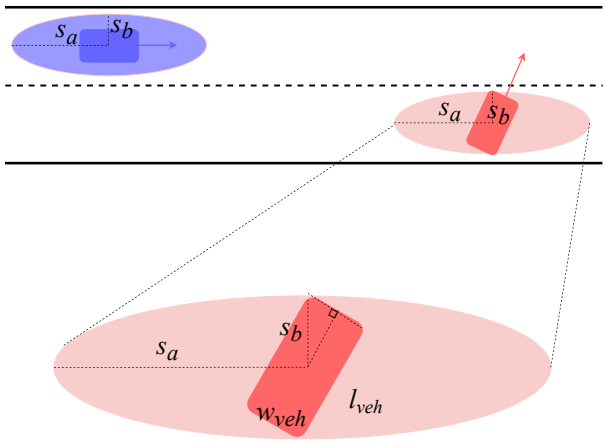


Figure 3.4.: Safe region for a vehicle that is turning left.

This safety constraint is then realized by the following inequality:

$$\frac{\Delta x_{\mathrm{EV,TV}}^2}{s_a^2} + \frac{\Delta y_{\mathrm{EV,TV}}^2}{s_b^2} > 1, \tag{3.8}$$

where s_a and s_b are the semi-major and semi- minor axes of the ellipse, respectively. The longitudinal distance between the EV and one TV is $\Delta x_{\mathrm{EV,TV}}$. The lateral distance between these two vehicles is $\Delta y_{\mathrm{EV,TV}}$. The center of the ellipse-shaped region is the same as the center of the vehicle. Since we use point-mass model that does not provide information about the current orientation of the vehicle, the major axis of the ellipse is chosen aligned with the lanes.

For safety, the ellipse-shaped region has to be large enough such that the vehicle shape is covered by the ellipse for all possible vehicle orientations. In order to find appropriate s_a and s_b, we consider the vehicle turning left or right (see Figure 3.4). If the vehicle is covered by the ellipse here, it will also be covered for all other orientations. Let l_{veh} and w_{veh} denote the length and the width of a vehicle, respectively. Then it holds for s_a and s_b:

$$0 < \sqrt{(\frac{l_{\mathrm{veh}}}{2})^2 + (\frac{w_{\mathrm{veh}}}{2})^2} + \zeta < s_b < s_a \tag{3.9}$$

where $\zeta > 0$.

Input Constraints

Input constraints are restrictions on the accelerations in both longitudinal and lateral directions. These constraints come from the physical limitations of the vehicles.

$$a_{x,k} \in [a_{x_{\min}}, a_{x_{\max}}] \tag{3.10}$$

$$a_{y,k} \in [a_{y_{\min}}, a_{y_{\max}}] \tag{3.11}$$

3.3.3. Cost Function

The cost function for an MPC-controlled vehicle is designed for tracking the reference trajectory while realizing smooth and energy efficient driving.

The cost function $V(\boldsymbol{\xi},\boldsymbol{u})$ at each time step is

$$V(\boldsymbol{\xi},\boldsymbol{u}) = \sum_{k=0}^{N-1} \left[u_k^\top Q u_k + \Delta\xi_k^\top R \Delta\xi_k \right] + \Delta\xi_N^\top Q_N \Delta\xi_N \tag{3.12}$$

where $\Delta\xi_k = \xi_k - \xi_k^{\text{ref}}$ and $\Delta\xi_N = \xi_N - \xi_N^{\text{ref}}$. ξ_k^{ref} and ξ_N^{ref} are the reference states for the vehicle to track. $\Delta\xi$ is the error between the predicted states and the reference states. The weighting matrices Q, R and Q_N are defined by $Q = \text{diag}(q_1, q_2)$, $R = \text{diag}(r_1, r_2, r_3, r_4)$, and $Q_N = \text{diag}(q_{N,1}, q_{N,2}, q_{N,3}, q_{N,4})$, where q_1, q_2, r_1, r_2, r_3, r_4, $q_{N,1}$, $q_{N,2}$, $q_{N,3}$, and $q_{N,4}$ are elements in the matrices, and non-negative real numbers. The terms $\Delta\xi_k^\top R \Delta\xi_k$ and $\Delta\xi_N^\top Q_N \Delta\xi_N$ are used for tracking reference states, where in particular the reference positions (longitudinal and lateral) are relevant. The term $u_k^\top Q u_k$ is designed to punish large control inputs, to ensure that the vehicle drives smoothly and in an energy-efficient way.

The cost function acts as a bridge in our combined maneuver planning and trajectory control method. The maneuver selected from the maneuver planning method determines the reference trajectories in the cost function. The reference trajectories will be tracked by minimizing the cost function with all constraints being satisfied simultaneously. In the following subsection, we will introduce the reference trajectory generation and trajectory tracking.

3.3.4. MPC-Based Reference Trajectory Generation

MPC-Based reference trajectory generation aims to translate the selected semantic maneuver to a reference trajectory. At each time step, the selected maneuver is executed by setting appropriate costs and constraints of the MPC controller, as shown in Figure 3.1.

We set values of the reference states ξ_k^{ref} and ξ_N^{ref} in the cost function (3.12) based on the selected maneuver. In the reference states $\xi_k^{\text{ref}} = \left[x_k^{\text{ref}}, y_k^{\text{ref}}, v_{x,k}^{\text{ref}}, v_{y,k}^{\text{ref}} \right]^\top$, the first two elements x_k^{ref} and y_k^{ref} are reference positions in longitudinal and lateral directions, respectively. Corresponding reference velocities are represented by $v_{x,k}^{\text{ref}}$ and $v_{y,k}^{\text{ref}}$. Among these reference states, only y_k^{ref} and $v_{x,k}^{\text{ref}}$ are directly determined by the selected maneuvers. The maneuver planning has no impact on the reference states x_k^{ref} and $v_{y,k}^{\text{ref}}$. However, this does not mean that the maneuver planning will not affect the longitudinal position x_k and lateral velocity v_k. It will affect them implicitly by the coupling of states and control inputs in the constrained optimal control problem.

The elements in weighting matrices $R = \text{diag}(r_1, r_2, r_3, r_4)$ and $Q_N = \text{diag}(q_{N,1}, q_{N,2}, q_{N,3}, q_{N,4})$ in the cost function (3.12) allow tracking performances to be tuned. The maneuver planning will directly set the lateral reference position and the longitudinal reference velocity of the vehicle. Therefore, we set the related elements $r_2, r_3, q_{N,2}, q_{N,3} > 0$ and small constants are assigned to the elements $r_1, r_4, q_{N,1}$, and $q_{N,4}$.

The selected lane maneuver will determine the value of the lateral reference position y^{ref}, and the speed maneuver will determine the longitudinal reference velocity v_x^{ref}. y^{ref} is set to be the lateral center of the target lane. This applies to both lane keeping (LK) and lane changing (LCL

and LCR). The rules for setting v_x^{ref} vary according to the relative position of the two vehicles in the lateral direction. For the situation where two vehicles are in the same lane: 1) If the selected maneuver is maintaining the current speed (CS), let v_x^{ref} be the same as its current longitudinal velocity; 2) For decelerating (DE), the reference velocity v_x^{ref} is the minimum between 75% of the current longitudinal velocity of the EV and the longitudinal velocity of the vehicle in front of the EV; 3) If the vehicle intends to accelerate (AC), we specify v_x^{ref} to be the maximum of 125% of the current longitudinal velocity of the EV, the longitudinal velocity of the TV, and the longitudinal speed limit. When the two vehicles are in the same lane, v_x^{ref} is equal to the speed limit.

3.4. Simulation Results

To validate the effectiveness of the proposed method, a simulation is conducted in MATLAB for a highway environment. In this section, we will first introduce two scenarios. Then, we explain how the maneuver planning and trajectory control method is applied. Finally, we show simulation results and discuss the effectiveness of our method.

3.4.1. Scenario Description

We consider two scenarios, a vehicle-following scenario and an overtaking scenario. We discuss maneuver and trajectory planning for both scenarios. However, we simulate only the overtaking scenario, because it includes all lateral and longitudinal maneuvers, and car-following will occur when overtaking.

The environment is a three-lane highway. All these three lanes have the same width and there is no curvature. There are two vehicles, an EV (red) and an TV (blue), as shown in Figure 3.5. The three lanes are labelled by L_0, L_1 and L_2, respectively. The rightmost lane L_0 is the slow lane. The leftmost lane L_2 is the fast lane, which is meant just for overtaking. Initially, we assume that the EV is driving in the slow lane L_0, with longitudinal velocity $35\mathrm{m/s}$. The TV is driving in the middle lane L_1 with a lower longitudinal velocity $20\mathrm{m/s}$. The initial longitudinal positions of EV and TV are 10m and 90m, respectively. On a highway, overtaking on the right side is forbidden [9]. Therefore, with this initial settings, the EV has two possible options: either following the TV in L_0, or overtaking left.

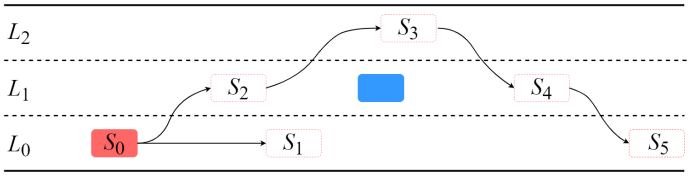


Figure 3.5.: Car-following and overtaking.

In order to better describe the maneuver and trajectory planning, the two possible scenarios are divided into six stages: S_0, S_1, S_2, S_3, S_4 and S_5, as shown in Figure 3.5. S_0 is the initial stage. In stages S_0 and S_1 car-following is realized ($S_0 \to S_1$). The overtaking consists of stages S_0, S_2, S_3, S_4 and S_5 ($S_0 \to S_2 \to S_3 \to S_4 \to S_5$). In these stages, the TV is always in the middle lane L_1, while the EV drives in different lanes. In detail:

- S_0: The EV is driving behind the TV in L_0 with a higher longitudinal velocity.

- S_1: The EV is following the TV in L_0 with a lower longitudinal velocity.
- S_2: The EV is driving in lane L_1, waiting for a chance to change to L_2. This is a transition stage.
- S_3: The EV reaches lane L_2 and drives in L_2 before overtaking the TV from the left, looking for an opportunity to go back to lane L_1. In this stage, also lane keeping occurs.
- S_4: The EV reaches lane L_1, and is preparing for moving to lane L_0.
- S_5: The EV drives in lane L_0.

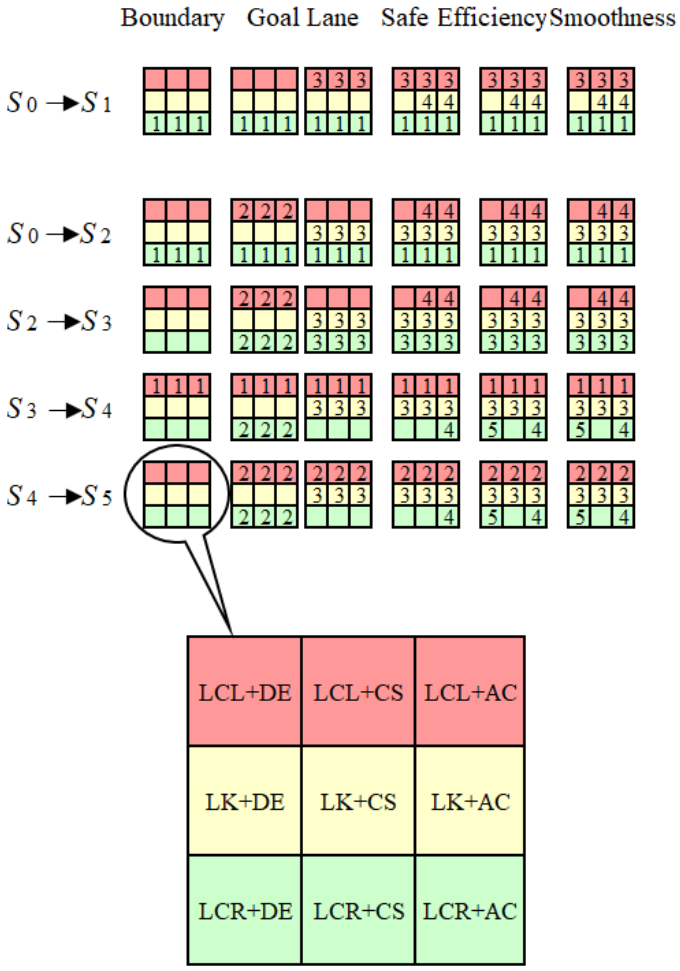


Figure 3.6.: Maneuver Selection: The box displays the 9 possible maneuvers: LCL+DE, LCL+CS, LCL+AC, LK+DE, LK+CS, LK+AC, LCR+DE, LCR+CS and LCR+AC.

3.4.2. Maneuver Planning

The procedure of maneuver selection is shown in Figure 3.6. The box represents the nine possible combinations of maneuvers in the lateral and longitudinal directions. The tiny cubes without numbers on them stand for current candidate maneuvers. The tiny cubes with numbers inside

represent the maneuvers that are excluded from the candidate maneuvers. The numbers, '1', '2', '3', '4', '5' and '6', represent the reasons for removing maneuvers. Among them, '1', '2' and '3' are for removing lateral maneuvers, and '4', '5' and '6' are the reasons for removing longitudinal maneuvers. They stand for:

- 1– The vehicle will reach the edge of the road if it turns left (or right).
- 2– The goal is to change lane but the the safety constraints for changing lane is not satisfied.
- 3– By conducting these maneuvers, the vehicle can not move to the target lane even though the conditions for changing lane are fulfilled.
- 4– These maneuvers contribute less/nothing to increasing TTC and TIV.
- 5– These maneuvers contribute less to improving efficiency.
- 6– Selecting these maneuvers has a negative or no positive impact on smoothness of driving behaviors.

In the following, we explain how the proposed maneuver selection method works in the two possible scenarios, the car-following scenario and the overtaking scenario.

Maneuver Selection in Car-following Scenario: $S_0 \rightarrow S_1$

Following the TV in the current lane is an excessively conservative but feasible choice in terms of collision avoidance. To implement the maneuver selection, one maneuver will be selected from all possible maneuvers in each direction. The lateral maneuver is first selected. *Changing to the right lane* (LCR) is excluded because of the boundary constraints. Then, we remove *changing to the left lane* (LCL) because it is inconsistent with current goal scenario (car-following), so only *lane keeping* (LK) is left. Finally, longitudinally, both TTC and TIV will definitely decrease if keeping the current speed (CS) or accelerating (AC), but decelerating (DE) probably causes an increase in TTC and TIV, so decelerating is selected as longitudinal maneuver. Combining the results of maneuver selection in both directions, we obtain the LK+DE maneuver for this car-following scenario.

Maneuver Selection in Overtaking Scenario: $S_0 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5$

Overtaking, although highly efficient for the EV, will increase the risk of collision with other traffic participants and requires careful trajectory design. The stage sequence $S_0 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_5$ can be divided into 4 steps: i) $S_0 \rightarrow S_2$, ii) $S_2 \rightarrow S_3$, iii) $S_3 \rightarrow S_4$ and iv) $S_4 \rightarrow S_5$. Note that the target lane of each stage change is not the current lane. In contrast to car-following, the EV now has to examine the environment to judge whether the conditions for changing lanes are satisfied. If not, the EV has to stay in its current lane and wait. The only difference to the maneuver selection procedure in car-following is that now the lane change condition is taken into account. Moreover, we do not use TTC and TIV in stage S_3, where the EV and TV are in different lanes, because this configuration is safe in the lateral direction. However, the distance in the longitudinal direction might be small, leading to small TTC and TIV, which is a false risk indicator (see discussion in Section 3.2.2).

3.4.3. MPC-based Trajectory Control

The maneuvers selected from the maneuver planning are delivered to the MPC controller for trajectory control. In order to validate our combined maneuver planning and MPC-based trajectory control method, we simulate the overtaking scenario shown in Figure 3.5.

The parameters of the road and vehicle, the limits of speed and acceleration, the parameters of the safety constraints, the prediction horizon, the sampling time and the weighting matrices are set to:

$$\begin{aligned}
&w_{\text{lane}} = 5.25\text{m}, \quad w_{\text{veh}} = 1.83\text{m}, \quad v_{x_{\min}} = 13.6\text{m/s}, \quad v_{x_{\max}} = 70\text{m/s},\\
&a_{x_{\min}} = -9\text{m/s}^2, \quad a_{x_{\max}} = 6\text{m/s}^2, \quad a_{y_{\min}} = -0.5\text{m/s}^2, \quad a_{y_{\max}} = 0.5\text{m/s}^2,\\
&s_a = 5\text{m}, \quad s_b = 2.625\text{m}, \quad N = 25, \quad T = 0.2,\\
&q_1 = 1, \quad q_2 = 0.1, \quad r_1 = 0, \quad r_2 = 10, \quad r_3 = 100, \quad r_4 = 0,\\
&q_{N,1} = 0, \quad q_{N,2} = 10, \quad q_{N,3} = 100, \quad q_{N,4} = 0.
\end{aligned} \tag{3.13}$$

The initial positions of both vehicles are shown in Figure 3.7. From Section 3.4.1, we know

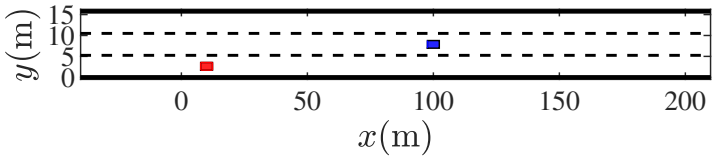


Figure 3.7.: Initial states of the vehicles

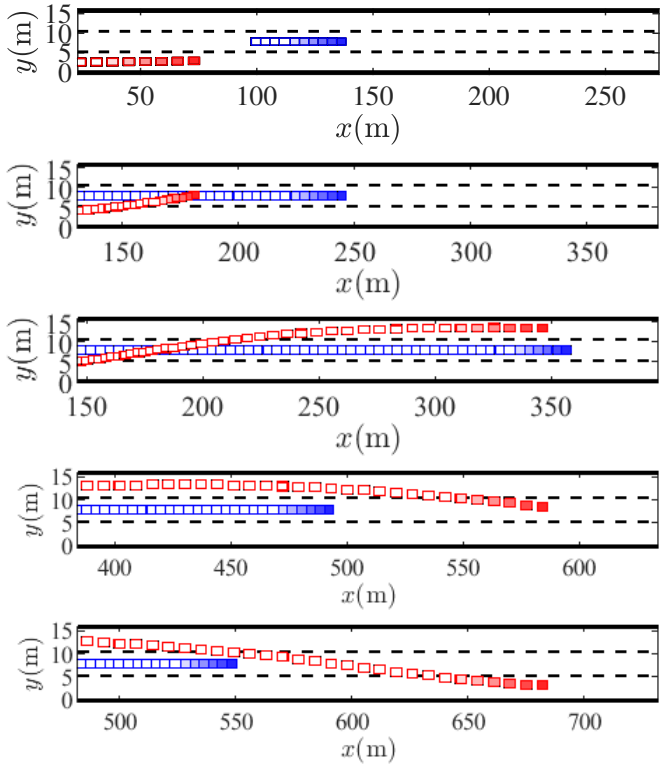


Figure 3.8.: Five stages for the overtaking scenario: S_0, S_2, S_3, S_4, S_5

that there are five stages and four stage changes in the overtaking scenario, as shown in Figure 3.5. The simulation results of our combined maneuver planning and MPC-based trajectory control method are shown in Figure 3.8. In Figure 3.8, the successive five subfigures show the five stages in the overtaking scenario: S_0, S_2, S_3, S_4, S_5. The dark red rectangle is the current position of the EV, and the other red rectangles stand for the past trajectory of the EV. Similarly, the dark blue rectangle is the current position of the TV, and all other blue rectangles indicate the trajectory of the TV.

The maneuvers that the EV selects can also be extracted from Figure 3.8. The lateral maneuvers can be observed directly from the lateral positions of the EV (red). The changes of density of the rectangles indicate the longitudinal maneuvers. When the density of the rectangles in a trajectory changes from sparse to dense, the vehicle executes a deceleration maneuver and vice versa. In addition, no change in the longitudinal distance between rectangles implies that the vehicle drives at a constant longitudinal velocity.

3.5. Conclusions and Future Work

In this chapter (based on [28]), we propose a maneuver selection method which chooses one maneuver from all potential ones by excluding less appropriate maneuvers based on five rules involving lane edges, the goal lane, safety, efficiency and smoothness. Our novel maneuver selection method cooperates with an existing maneuver generation method, constituting our high-level maneuver planning approach. Furthermore, we use an MPC-based trajectory generation method to execute the maneuvers selected from high-level planning by designing a corresponding cost function and constraints for the MPC controller. Besides, we validate the proposed method for an overtaking scenario on a three-lane highway. The whole overtaking scenario can be divided into five stages. The results show that at each stage the ego vehicle can successfully reach the desired lane, conduct the desired speed maneuver, and avoid collisions.

Our current maneuver planning method is executed at each time step, which allows the autonomous vehicle to rapidly adjust its behavior in risky situations. However, the high frequency of updating also causes unnecessary computational burden when the driving environment is mostly safe. Therefore, a trade-off between ensuring fast response when necessary and avoiding unnecessary computation should be found in future work. Moreover, the maneuver planning method is validated in the simulations of a two-vehicle system. However, real traffic often includes multiple vehicles. It will also be worth researching how to extend the usage of our maneuver planning method in multi-vehicle systems. Additionally, our maneuver planning method can cooperate successfully with the MPC-based trajectory generation method that executes our high-level maneuver. Whether our maneuver planning method can smoothly cooperate with other controllers remains unknown. In the future, we would explore the possibility of generalizing our maneuver planning method to be able to cooperate with other lower-level controllers.

Simulation of Microscopic Interactive Multi-SMPC-Vehicle System

4.

Many promising strategies have been proposed on how to use Stochastic Model Predictive Control (SMPC) to control an autonomous vehicle in uncertain environments. The limitation of these approaches is that they focus on scenarios where only one vehicle is controlled by SMPC and is, thus, reacting to the surrounding vehicles; however, the surrounding vehicles do not react to the SMPC-controlled vehicle, which means there is no mutual interaction. However, when multiple autonomous vehicles are driving on the road, each individual vehicle will take the behavior of the other surrounding vehicles into account and adjust its individual decisions accordingly in trajectory planning.

This chapter, therefore, examines in simulations how the interactive control system of multiple SMPC-controlled vehicles behaves. To do this, we model the microscopic interactive multi-vehicle system using a Distributed SMPC (DSMPC) framework. We investigate the effects of the risk parameter of the collision avoidance probabilistic constraint on non-interactive and interactive vehicle systems based on a three-lane highway scenario and provide insights into how to parameterize the controllers in interactive vehicle systems. The content of this chapter was published in [32].

4.1. Introduction

In this chapter [32], for simplicity of description, vehicles that react to other vehicles are called reactive vehicles, and those that do not react to other vehicles are non-reactive vehicles. A system consisting of only one reactive vehicle and multiple non-reactive vehicles is a non-interactive system. In real traffic, all vehicles tend to react to the vehicles in the environment, forming an interactive system. Previous SMPC algorithms for autonomous driving focus on scenarios where the controlled vehicle treats all surrounding vehicles as non-reactive vehicles. However, it is essential to also embed this controlled vehicle system in an environment where all surrounding vehicles are reactive, meaning in an interactive system. Not only is it important to investigate the performance of the overall interactive control system but the findings from this investigation must be included in the design of the SMPC for an individual autonomous vehicle.

In a multi-vehicle highway environment, considering the interactions between reactive vehicles contributes to more precise traffic prediction [79], which is fundamentally required in intelligent transportation systems [25]. Interactions between reactive vehicles have previously been investigated in microscopic traffic simulations [79], [80]; however, to the best of our knowledge, the interactions between SMPC-controlled vehicles, where it is of interest to see the impact and the interplay of different risk parameters that determine the aggressiveness/conservativeness level of vehicles when reacting to other vehicles, have not been investigated. This motivated us to examine the interactions between SMPC-controlled vehicles in a multi-vehicle interactive system for a highway environment in this chapter [32].

4.1.1. Approach Overview

In this chapter [32], we examine the interactions between SMPC-controlled vehicles in a multi-vehicle interactive system for a highway environment. To do this, we model the multi-vehicle interactive system using a Distributed SMPC (DSMPC) framework [29]–[31]. In this framework, each vehicle interacts with its surrounding vehicles by observing their current states and predicting their future behaviors and avoiding potential collisions. Distributed MPC (DMPC) has been applied to solve vehicle platooning problems [81]–[84], where multiple vehicles are typically involved and are controlled to cruise at a constant speed. However, DMPC has not been used for problems where individual vehicles do not have a common driving goal. In this chapter [32], we use a DSMPC framework to model multi-vehicle interactive systems, where individual vehicles have unique driving goals, which are usually different. Here, we assume that all vehicles have the same controller but with different parameterizations. In particular, the risk parameter is chosen differently.

We summarize the contributions of this chapter [32] as follows:

1. Investigating the effects of SMPC risk parameters on non-interactive and interactive vehicle-control systems on highways.

2. Providing guidelines on how to set risk parameters for vehicles in interactive systems.

4.1.2. Chapter Overview

The rest of Chapter 4 is organized as follows. Section 4.2 presents the communication topology used in the distributed framework and formulates the individual SMPC controller and how all individual SMPC controllers are combined together. In Section 4.3, we introduce elements of these SMPC problems and transform the stochastic optimal-control problem into a deterministic problem. Finally, our simulation results are presented in Section 4.4 followed by our conclusions in Section 4.5.

4.2. Model of the Multi-Vehicle System

In our multi-vehicle interactive control system model, each vehicle detects the position in the lateral and longitudinal direction, velocity, and inertial heading angle of all currently neighboring vehicles, which we refer to as 'information' in the following; a graph theoretic time-varying communication topology [83] models this information transmission.

In this section, we introduce the communication topology and the SMPC problem that is solved by an individual vehicle and how all SMPC problems are combined into the distributed SMPC control framework.

Our work regarding interactive systems of SMPC-controlled vehicles is based on the hypotheses below:

Hypothesis 1 *The behaviors of a vehicle are determined not only by its own controller but also by the controllers of other vehicles.*

Hypothesis 2 *The behaviors of one vehicle can influence the performance of the whole system.*

4.2.1. Communication Topology

The communication topology shows which of the surrounding vehicles is considered in the controller of one particular vehicle. We assume that all vehicles are equipped with sensors to detect information about their surrounding vehicles at a specified detectable distance. This distance depends on the detection ability of each vehicle's sensors. For simplicity, we assume that the sensors of all vehicles have the same detection ability, which means the detectable distance is the same.

Here, we introduce a communication topology at one time step as shown in Figure 4.1. The communication topology is updated at each time step to account for changing vehicle positions. It

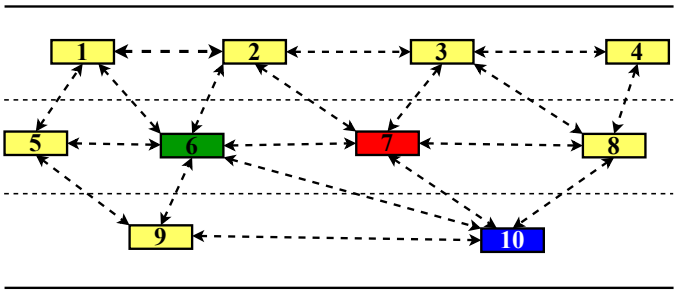


Figure 4.1.: Communication topology at one time step. This topology shows, at one time step, which of the surrounding vehicles is considered in the controller of one particular vehicle. This topology is updated at each time step.

is modeled as an undirected graph $\mathbb{G} = \{\mathbb{V}, \mathbb{E}\}$, where $\mathbb{V} = \{1, 2, ..., N_v\}$ is the set of nodes, which represent vehicles, and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is the set of edges describing the information detection among vehicles. The number of nodes (vehicles) in the graph is given by N_v. The graph $\mathbb{G}$ can be denoted with an adjacency matrix $\mathscr{A} \in \mathbb{R}^{N_v \times N_v}$

$$\mathscr{A} = [a_{ij}] = \begin{cases} a_{ij} = 1, \text{ if } \{i, j\} \in \mathbb{E} \\ a_{ij} = 0, \text{ if } \{i, j\} \notin \mathbb{E} \end{cases}, \tag{4.1}$$

where $\{i, j\} \in \mathbb{E}$ means vehicle i senses the information about vehicle j, which is within the detectable distance of vehicle i. Vehicle j is, therefore, a neighbor of vehicle i. The set consisting of the neighbors of vehicle i is denoted by $\mathbb{N}_i = \{j \mid a_{ij} = 1, j \in \mathbb{V}\}$. We define a dual set $\mathbb{O}_i = \{j \mid a_{ji} = 1, j \in \mathbb{V}\}$, which includes all vehicles that identify i as a neighbor. The union of the two sets $\mathbb{N}_i$ and $\mathbb{O}_i$ is $\mathbb{N}_i \cup \mathbb{O}_i$. All vehicles in $\mathbb{N}_i \cup \mathbb{O}_i$ categorize i as one of their neighbors and are themselves simultaneously neighbors of vehicle i. The sets $\mathbb{N}_i$, $\mathbb{O}_i$, and $\mathbb{N}_i \cup \mathbb{O}_i$ are updated at each sampling time. Assuming that all vehicles have the same detectable distance, the sets $\mathbb{N}_i$, $\mathbb{O}_i$, and $\mathbb{N}_i \cup \mathbb{O}_i$ are equal.

Each controlled vehicle will attempt to avoid collisions with its neighboring vehicles, and probabilistic constraints in the optimal-control problem reflect this requirement. Any vehicle i incorporates the information about its neighbors from $\mathbb{N}_i$ into its collision-avoidance constraints. All vehicles in $\mathbb{O}_i$ take the information of vehicle i into account in their collision-avoidance constraints.

4.2.2. Vehicle Controllers

The multi-vehicle interactive control system consists of a number of individual vehicles that are interactive because their individual controllers consider information about the current states of

surrounding vehicles. We assume that each vehicle is controlled by SMPC; thus, the overall system is modeled using a distributed SMPC framework. The SMPC optimal-control problem that is solved by each vehicle at every sampling time is introduced in this section.

To decide on the current optimal control, each controlled vehicle, denoted as an Ego Vehicle (EV), must consider its predicted behaviors as well as those of vehicles, denoted as Target Vehicles (TVs), within a detectable distance. Simultaneously, an EV might be a TV of other Ego Vehicles (EVs). For each vehicle i ($i \in \mathbb{V}$) and each prediction time step k ($k = 0,...,N-1$), we define the predicted state $\xi^{\mathrm{p}}_{i,k}$ and predicted control input $u^{\mathrm{p}}_{i,k}$ that will later be optimized over a prediction horizon of length N. Further, we introduce assumed states $\xi^{\mathrm{a}}_{i,k}$ and assumed control inputs $u^{\mathrm{a}}_{i,k}$ [82], [83] to describe what other vehicles (vehicles in $\mathbb{O}_i$) assume about the future behaviors of vehicle i. Finally, $\xi^{*}_{i,k}$ and $u^{*}_{i,k}$ define the optimal trajectories that vehicle i determines by solving the SMPC optimal control problem.

The SMPC optimal-control problem for EV i ($i \in \mathbb{V}$) is specified by a cost function V_i and constraints. The cost function V_i is minimized over all admissible control input trajectories $\boldsymbol{u}^{\mathrm{p}}_i = \left(u^{\mathrm{p}}_{i,0}, u^{\mathrm{p}}_{i,1}, \ldots, u^{\mathrm{p}}_{i,N-1}\right)^{\top}$, where admissibility means that the control inputs $\boldsymbol{u}^{\mathrm{p}}_i$ as well as the corresponding state trajectory $\boldsymbol{\xi}^{\mathrm{p}}_i = \left(\xi^{\mathrm{p}}_{i,0}, \xi^{\mathrm{p}}_{i,1}, \ldots, \xi^{\mathrm{p}}_{i,N}\right)^{\top}$, which is found by iterating the system dynamics

$$\xi^{\mathrm{p}}_{i,k+1} = \mathscr{F}^{\mathrm{p}}\left(\xi^{\mathrm{p}}_{i,k}, u^{\mathrm{p}}_{i,k}\right), i \in \mathbb{V}, k = 0, \ldots, N-1, \tag{4.2a}$$

do not violate constraints. The initial predicted state $\xi^{\mathrm{p}}_{i,0}$ is the current state of the EV i. A first version of the optimal-control problem is, thus, given by

$$\min_{\boldsymbol{u}^{\mathrm{p}}_i} V_i\left(\boldsymbol{\xi}^{\mathrm{p}}_i, \boldsymbol{u}^{\mathrm{p}}_i\right) \tag{4.2b}$$

subject to state and input constraints

$$\xi^{\min}_i \leq \xi^{\mathrm{p}}_{i,k} \leq \xi^{\max}_i, k = 0, \ldots, N \tag{4.2c}$$

$$u^{\min}_i \leq u^{\mathrm{p}}_{i,k} \leq u^{\max}_i, k = 0, \ldots, N-1. \tag{4.2d}$$

Remark: We used only box constraints here, though more general constraints would be allowed.

We still have to add collision-avoidance constraints that involve assumptions on the surrounding vehicles' behaviors. Summarizing the assumed models of all TVs in

$$\xi^{\mathrm{a}}_{\mathrm{TV},k+1} = \mathscr{F}^{\mathrm{a}}_{\mathrm{TV}}\left(\xi^{\mathrm{a}}_{\mathrm{TV},k}, u^{\mathrm{a}}_{\mathrm{TV},k}, \omega^{\mathrm{a}}_{\mathrm{TV},k}\right), k = 0, \ldots, N-1, \tag{4.2e}$$

where $\omega^{\mathrm{a}}_{\mathrm{TV},k}$ is the uncertainty in the prediction of TV behaviors, we obtain assumptions for all times in the prediction horizon used to formulate probabilistic collision-avoidance constraints for each TV summed up in

$$\mathbf{Pr}\left(\xi^{\mathrm{p}}_{i,k} \in \Xi^{\mathrm{safe,TV}}_{i,k}\right) \geq p_i,, \; p_i \in [0.5, 1], k = 1, \ldots, N. \tag{4.2f}$$

These constraints are probabilistic constraints in our approach. The requirement $\xi^{\mathrm{p}}_{i,k} \in \Xi^{\mathrm{safe,TV}}_{i,k}$ means that states $\xi^{\mathrm{p}}_{i,k}$ have to be in the safe set $\Xi^{\mathrm{safe,TV}}_{i,k}$ to avoid potential collisions with the TVs at prediction step k. The set $\Xi^{\mathrm{safe,TV}}_{i,k}$ is determined from the predicted states of EV i and the

assumed states of all its TVs $\xi^{\mathrm{a}}_{\mathrm{TV},k}$. Employing $\mathbf{Pr}(*) \geq p_i$, we ensure that the event $*$ occurs with a probability of not less than p_i. The probabilistic constraints (4.2f) are designed to soften the collision-avoidance constraint between the EV i and its TVs.

A small probability of collisions between the EV i and its TVs is acceptable. This softening prevents overly conservative driving behaviors caused by hard constraints in robust MPC. In the following, p_i in constraints (4.2f) is identified as a risk parameter of EV i and is specified in advance. A smaller risk parameter p_i corresponds to more aggressive driving behaviors, which might increase the probability of collisions. Conversely, a larger p_i results in more conservative behaviors, a defensive driving mode.

We refer to the expressions (4.2a)−(4.2f) as 'the SMPC optimal-control problem' in the following. The model in (4.2e) collects the system models of all TVs of EV i. If we assume that EV i takes m TVs labeled $i_1, i_2, \ldots, i_m$ ($i_1, i_2, \ldots, i_m \in \mathbb{N}_i$) into account, then (4.2e) summarizes

$$\begin{cases} \xi^{\mathrm{a}}_{i_1,k+1} = \mathscr{F}^{\mathrm{a}}\left(\xi^{\mathrm{a}}_{i_1,k}, u^{\mathrm{a}}_{i_1,k}, \omega^{\mathrm{a}}_{i_1,k}\right) \\ \xi^{\mathrm{a}}_{i_2,k+1} = \mathscr{F}^{\mathrm{a}}\left(\xi^{\mathrm{a}}_{i_2,k}, u^{\mathrm{a}}_{i_2,k}, \omega^{\mathrm{a}}_{i_2,k}\right) \\ \qquad\qquad \vdots \\ \xi^{\mathrm{a}}_{i_m,k+1} = \mathscr{F}^{\mathrm{a}}\left(\xi^{\mathrm{a}}_{i_m,k}, u^{\mathrm{a}}_{i_m,k}, \omega^{\mathrm{a}}_{i_m,k}\right) \end{cases} ,k = 0,\ldots,N-1. \tag{4.3}$$

The assumed states $\xi^{\mathrm{a}}_{i_1,k}$, $\xi^{\mathrm{a}}_{i_2,k}$, and $\xi^{\mathrm{a}}_{i_m,k}$ correspond to TVs i_1, i_2, and i_m, respectively. Similarly, the assumed control inputs are $u^{\mathrm{a}}_{i_1,k}$, $u^{\mathrm{a}}_{i_2,k}$, and $u^{\mathrm{a}}_{i_m,k}$; the prediction uncertainties are denoted by $\omega^{\mathrm{a}}_{i_1,k}$, $\omega^{\mathrm{a}}_{i_2,k}$, and $\omega^{\mathrm{a}}_{i_m,k}$. The dynamic model of the EV and TVs will be discussed in more detail in Section 4.3.

In the same way, expression (4.2f) contains the collision-avoidance constraints between EV i and all its TVs ($i_1, i_2, \ldots, i_m \in \mathbb{N}_i$):

$$\begin{cases} \mathbf{Pr}\left(\xi^{\mathrm{p}}_{i,k} \in \Xi^{\mathrm{safe},i_1}_{i,k}\right) \geq p_i \\ \mathbf{Pr}\left(\xi^{\mathrm{p}}_{i,k} \in \Xi^{\mathrm{safe},i_2}_{i,k}\right) \geq p_i \\ \qquad\qquad \vdots \\ \mathbf{Pr}\left(\xi^{\mathrm{p}}_{i,k} \in \Xi^{\mathrm{safe},i_m}_{i,k}\right) \geq p_i \end{cases} , p_i \in [0.5,1], k = 1,\ldots,N. \tag{4.4}$$

Here, $\Xi^{\mathrm{safe},i_1}_{i,k}$, $\Xi^{\mathrm{safe},i_2}_{i,k}$ and $\Xi^{\mathrm{safe},i_m}_{i,k}$ are the sets of safe states of EV i for preventing collisions with TVs i_1, i_2, …, i_m at prediction step k, respectively.

4.3. Elements of the SMPC Problem

In this section, we introduce the elements of the SMPC optimal-control problem, including the vehicle models in Section 4.3.1, constraints in Section 4.3.2, and cost function in Section 4.3.3.

Additionally, due to the presence of stochastic disturbances $\omega^{\mathrm{a}}_{\mathrm{TV},k}$ in the TV model (4.2e) and the probabilistic chance constraints (4.2f), the SMPC optimal-control problem cannot be solved directly [16]. To solve this, we transfer the stochastic optimal-control problem into a deterministic one by (1) reformulating the dynamic model of the TV, as shown in Section 4.3.1; and (2) tightening the probabilistic constraints as shown in Section 4.3.2.

4.3.1. Vehicle Models

A predictive controller requires a system model (4.2a). Thus, for our application, we need a system model of each EV, which is used by the EV to decide on optimal controls. In addition, we need a system model of each TV, which EVs use to predict TV trajectories to avoid potential collisions. In Chapter 4, the models of EVs and TVs are based on the linearized, discretized version of kinematic bicycle model [20], [38], [39] introduced in (2.4) of Section 2.1.2.

Model of EVs

We use the model (2.4) for each EV i to generate predictions:

$$\xi^{\mathrm{p}}_{i,k+1} = \xi^{\mathrm{p}}_{i,0} + T\mathscr{F}^{\mathrm{c}}\left(\xi^{\mathrm{p}}_{i,0}, 0\right) + A_i\left(\xi^{\mathrm{p}}_{i,k} - \xi^{\mathrm{p}}_{i,0}\right) + B_i u^{\mathrm{p}}_{i,k}, i \in \mathbb{V}, k = 0, \ldots, N-1, \tag{4.5}$$

where $\xi^{\mathrm{p}}_{i,k} = \left(x^{\mathrm{p}}_{i,k}, y^{\mathrm{p}}_{i,k}, \psi^{\mathrm{p}}_{i,k}, v^{\mathrm{p}}_{i,k}\right)^\top \in \mathbb{R}^{N_{\xi,i}}$ and $u^{\mathrm{p}}_{i,k} = \left(\delta^{\mathrm{p}}_{i,k}, a^{\mathrm{p}}_{i,k}\right)^\top \in \mathbb{R}^{N_{u,i}}$ are the predicted states and control inputs of EV i in prediction step k, respectively.

Model of TVs

For TVs, we choose a slightly adapted version of model (2.4) to include prediction uncertainty. Let vehicle $\breve{i}$ be one TV of any EV i ($\breve{i} \in \mathbb{N}_i$), then $\xi^{\mathrm{a}}_{\breve{i},k}$ is the assumed trajectory of TV $\breve{i}$ at prediction step k, and the TV model is

$$\xi^{\mathrm{a}}_{\breve{i},k+1} = \xi^{\mathrm{a}}_{\breve{i},0} + T\mathscr{F}^{\mathrm{c}}\left(\xi^{\mathrm{a}}_{\breve{i},0}, 0\right) + A_{\breve{i}}\left(\xi^{\mathrm{a}}_{\breve{i},k} - \xi^{\mathrm{a}}_{\breve{i},0}\right) + B_{\breve{i}} u^{\mathrm{a}}_{\breve{i},k} + G_{\breve{i}}\omega^{\mathrm{a}}_{\breve{i},k}, k = 0, \ldots, N-1, \tag{4.6}$$

where $\xi^{\mathrm{a}}_{\breve{i},k} = \left(x^{\mathrm{a}}_{\breve{i},k}, y^{\mathrm{a}}_{\breve{i},k}, \psi^{\mathrm{a}}_{\breve{i},k}, v^{\mathrm{a}}_{\breve{i},k}\right)^\top \in \mathbb{R}^{N_{\xi,\breve{i}}}$ and $u^{\mathrm{a}}_{\breve{i},k} = \left(\delta^{\mathrm{a}}_{\breve{i},k}, a^{\mathrm{a}}_{\breve{i},k}\right)^\top \in \mathbb{R}^{N_{u,\breve{i}}}$ are the assumed states and control inputs of TV $\breve{i}$ at prediction step k, respectively. The system matrices $A_{\breve{i}}$ and $B_{\breve{i}}$ can be found in [39]. The vector $\omega^{\mathrm{a}}_{\breve{i},k} \in \mathbb{R}^{N}_{\omega,\breve{i}}$ is included to account for the uncertainty at any prediction step k, which comes from the imprecision of the prediction.

The uncertainties $\omega^{\mathrm{a}}_{\breve{i},k} \in \mathbb{R}^{N}_{\omega,\breve{i}}$ are assumed to be subject to a Gaussian distribution with zero mean and covariance matrix $\sum_{\omega^{\mathrm{a}}_{\breve{i}}}$, and thus $\omega^{\mathrm{a}}_{\breve{i},k} \sim \mathscr{N}\left(0, \sum_{\omega^{\mathrm{a}}_{\breve{i}}}\right)$.

Reformulation of the TV Model

The SMPC optimal-control problem in expressions (4.2a)–(4.2f) is replaced by an equivalent deterministic problem that is numerically tractable. Here, we prepare this replacement by splitting the TV model into deterministic and stochastic equations (see [85]).

The state of TV $\breve{i}$ at prediction step k is decomposed into two components: the deterministic, nominal component $z^{\mathrm{a}}_{\breve{i},k}$ ($z^{\mathrm{a}}_{\breve{i},k} = \left(\widetilde{x}^{\mathrm{a}}_{\breve{i},k}, \widetilde{y}^{\mathrm{a}}_{\breve{i},k}, \widetilde{\psi}^{\mathrm{a}}_{\breve{i},k}, \widetilde{v}^{\mathrm{a}}_{\breve{i},k}\right)^\top \in \mathbb{R}^{N_{z,\breve{i}}}$) and a zero-mean stochastic error component $e^{\mathrm{a}}_{\breve{i},k}$

$$\xi^{\mathrm{a}}_{\breve{i},k} = z^{\mathrm{a}}_{\breve{i},k} + e^{\mathrm{a}}_{\breve{i},k}. \tag{4.7}$$

The following assumption is made (see [16]):

Assumption 1 *The state feedback is perfect, i.e.,* $\xi^{\mathrm{a}}_{\breve{i},0} = z^{\mathrm{a}}_{\breve{i},0}$, *which suggests* $e^{\mathrm{a}}_{\breve{i},0} = 0$ *with a probability of* 1.

We incorporate a prestabilizing error feedback (see [21]) into the control input

$$u^{\mathrm{a}}_{\check{i},k} = K_{\check{i}} e^{\mathrm{a}}_{\check{i},k} + v^{\mathrm{a}}_{\check{i},k}, \tag{4.8}$$

where $K_{\check{i}}$ is a stabilizing feedback gain that is obtained by applying a linear quadratic control strategy, and $v^{\mathrm{a}}_{\check{i},k} = \left(\widetilde{\delta}^{\mathrm{a}}_{i,k}, \widetilde{a}^{\mathrm{a}}_{i,k}\right)^{\top} \in \mathbb{R}^{N_{v,\check{i}}}$ is the assumed control input used for an EV to predict the behaviors of its TV $\check{i}$. In the following, we set $v^{\mathrm{a}}_{\check{i},k} = 0$ $(k = 0, \ldots, N-1)$, so that the EVs assume that TVs will drive with almost constant speed in the prediction horizon. The equations for the TV model are summarized as

$$z^{\mathrm{a}}_{\check{i},k+1} = z^{\mathrm{a}}_{\check{i},0} + T\mathscr{F}^{\mathrm{c}}\left(z^{\mathrm{a}}_{\check{i},0}, 0\right) + A_{\check{i}}\left(z^{\mathrm{a}}_{\check{i},k} - z^{\mathrm{a}}_{\check{i},0}\right) + B_{\check{i}} v^{\mathrm{a}}_{\check{i},k}, \tag{4.9a}$$

$$e^{\mathrm{a}}_{\check{i},k+1} = \Phi_{\check{i}} e^{\mathrm{a}}_{\check{i},k} + G_{\check{i}} \omega^{\mathrm{a}}_{\check{i},k}, \tag{4.9b}$$

where $\Phi_{\check{i}} = A_{\check{i}} + B_{\check{i}} K_{\check{i}}$ is strictly stable for the system $\left(A_{\check{i}}, B_{\check{i}}\right)$ of TV $\check{i}$. The deterministic equation (4.9a) will generate predictions of TV behavior, while the stochastic equation (4.9b) will be used to evaluate the collision-avoidance constraints.

The distribution of all predicted errors $e^{\mathrm{a}}_{\check{i},k}$ is determined iteratively from the distributions of the initial error $e^{\mathrm{a}}_{\check{i},0}$ and the disturbances $\omega^{\mathrm{a}}_{\check{i},k}$. Let $e^{\mathrm{a}}_{\check{i},k} \sim \mathscr{N}\left(0, \sum_{\check{i},k}\right)$, then $e^{\mathrm{a}}_{\check{i},k+1} \sim \mathscr{N}\left(0, \sum_{\check{i},k+1}\right)$, where $\sum_{\check{i},k+1} = \Phi_{\check{i}} \sum_{\check{i},k} \Phi_{\check{i}}^{\mathsf{T}} + G_{\check{i}} \sum_{\omega^{\mathrm{a}}_{\check{i}}} G_{\check{i}}^{\mathsf{T}}$ (see [16]). From Assumption 1, we find that the covariance of the initial error $e^{\mathrm{a}}_{\check{i},0}$ is 0, and thus $\sum_{\check{i},0} = 0$.

4.3.2. Constraints

In this subsection, we introduce constraints on states and inputs for the SMPC optimal-control problems of EVs. We consider (1) road boundaries, limitations on the inertial heading, speed, and acceleration; and (2) collision avoidance, where collision-avoidance constraints are probabilistic constraints, and all others are hard constraints.

Hard Constraints

For any EV i, we incorporate the following hard constraints into the SMPC problem

$$\xi_i^{\min} \leq \xi_{i,k} \leq \xi_i^{\max} \tag{4.10a}$$

$$u_i^{\min} \leq u_{i,k} \leq u_i^{\max} \tag{4.10b}$$

where $\xi_i^{\min} = \left(0, y^{\mathrm{r,l}} + w_i^{\mathrm{veh}}/2, \psi_i^{\min}, v_i^{\min}\right)^{\top}$, $\xi_i^{\max} = \left(l^{\mathrm{road}}, y^{\mathrm{r,u}} - w_i^{\mathrm{veh}}/2, \psi_i^{\max}, v_i^{\max}\right)^{\top}$, $u_i^{\min} = \left(a_i^{\min}, \delta_i^{\min}\right)^{\top}$, and $u_i^{\max} = \left(a_i^{\max}, \delta_i^{\max}\right)^{\top}$. The lower and upper boundaries of the road are represented by $y^{\mathrm{r,l}}$ and $y^{\mathrm{r,u}}$, respectively. The length of the road is denoted by l^{road}. The width of EV i is given by w_i^{veh}. The lower bounds of the inertial heading angle $\psi_i^{\min}$, speed $v_i^{\min}$, acceleration $a_i^{\min}$, and front steering angle $\delta_i^{\min}$ are considered. We also consider the upper bounds of these states, denoted by $\psi_i^{\max}$ and $v_i^{\max}$, and the control inputs, which are represented by $a_i^{\max}$ and $\delta_i^{\max}$. The values of these parameters are shown in Table 4.1 in Section 4.4.

Collision-Avoidance Constraints

In the following, we explain the calculation of the safe sets $\Xi_{i,k}^{\text{safe},\check{i}}$ in the collision-avoidance constraints (4.4), where ellipse regions approximate the occupied area of one vehicle that other vehicles are not allowed to enter (see [16], [19]).

The non-accessible region around TV $\check{i}$ is given by an inequality constraint

$$d_{\check{i},k} = \frac{\left(\Delta x_{\check{i},k}\right)^2}{s_a^2} + \frac{\left(\Delta y_{\check{i},k}\right)^2}{s_b^2} - 1 \geq 0 \tag{4.11}$$

that defines an ellipse where the center of vehicle $\check{i}$ is the center of the ellipse (see [19]). The size of the ellipse is determined by the semi-major axis s_a and the semi-minor axis s_b. The longitudinal distance and lateral distance between EV i and its TV $\check{i}$ are given by $\Delta x_{\check{i},k}$ and $\Delta y_{\check{i},k}$, respectively, and are defined below:

$$\begin{bmatrix} \Delta x_{\check{i},k} \\ \Delta y_{\check{i},k} \end{bmatrix} = \begin{bmatrix} x_{i,k}^{\text{p}} - \widetilde{x}_{\check{i},k}^{\text{a}} \\ y_{i,k}^{\text{p}} - \widetilde{y}_{\check{i},k}^{\text{a}} \end{bmatrix}. \tag{4.12}$$

The constraint (4.11) is usually overly conservative because, when the ellipse region around the TV $\check{i}$ is larger than the actual vehicle shape, a vehicle might enter the ellipse region without causing a collision. For this reason, we employ the probabilistic chance constraint for collision avoidance that allows vehicles a small probability to enter the safety ellipse of another vehicle:

$$\mathbf{Pr}(d_{\check{i},k} \geq 0) \geq p_i. \tag{4.13}$$

Constraint Tightening

In order to directly solve the SMPC optimal-control problem, we replace the probabilistic chance constraint (4.2f) by a tightened version of $d_{\check{i},k} \geq 0$, where the upper bounds of the tightened constraints depend on the risk parameter p_i and the distribution of the prediction uncertainties ω in the TV models. This allows for replacing the stochastic optimal-control problem with a deterministic optimal-control problem. We adopt the constraint tightening from [16], [19] and summarize it as follows.

From (4.7) in Section 4.3.1, the error between the actual and nominal states of TV $\check{i}$ is $e_{\check{i},k}^{\text{a}} = \xi_{\check{i},k}^{\text{a}} - z_{\check{i},k}^{\text{a}}$. Given (4.12), the constraint (4.11) is linearized around the nominal state $z_{\check{i},k}^{\text{a}}$ of TV $\check{i}$, resulting in

$$d_{\check{i},k} + \nabla d_{\check{i},k} e_{\check{i},k}^{\text{a}} \geq 0 \tag{4.14}$$

where

$$\nabla d_{\check{i},k} = \frac{\partial d_{\check{i},k}}{\partial z_{\check{i},k}^{\text{a}}} = \left(\frac{-2\Delta x_{\check{i},k}}{s_a^2}, \frac{-2\Delta y_{\check{i},k}}{s_b^2}, 0, 0 \right). \tag{4.15}$$

Using inequality (4.14), the probabilistic chance constraint (4.13) is rewritten as

$$\mathbf{Pr}\left(-\nabla d_{\check{i},k} e_{\check{i},k}^{\text{a}} \leq d_{\check{i},k} \right) \geq p_i,\ p_i \in [0.5, 1], k = 1, \ldots, N, \tag{4.16}$$

which can be divided into a deterministic inequality and a probabilistic equation:

$$d_{\check{i},k} \geq \gamma_{\check{i},k} \tag{4.17a}$$

$$\mathbf{Pr}\left(-\nabla d_{\check{i},k} e^{\mathrm{a}}_{\check{i},k} \leq \gamma_{\check{i},k}\right) = p_i,\ p_i \in [0.5,1], k = 1,\ldots,N. \tag{4.17b}$$

Then, according to Theorem 1 in [19], the probabilistic equation in (4.17b) is tightened by choosing $\gamma_{\check{i},k}$ as

$$\gamma_{\check{i},k} = \sqrt{2\nabla d_{\check{i},k} \sum_{\check{i},k} \left(\nabla d_{\check{i},k}\right)^{\top}} \mathrm{erf}^{-1}\left(2p_i - 1\right). \tag{4.18}$$

With the deterministic part of the TV model (4.9a) and the deterministic constraint (4.17a), the SMPC optimal-control problem in expressions (4.2a)–(4.2f) can be transformed into a deterministic problem. Particularly, we introduce the deterministic collision-avoidance constraints for multiple TVs are shown below. With the tightened constraints in expressions (4.17a) and (4.18), the probabilistic chance constraints in (4.4) can be rewritten as the following deterministic expressions:

$$\begin{cases} \begin{cases} d_{i_1,k} \geq \gamma_{i_1,k} \\ \gamma_{i_1,k} = \sqrt{2\nabla d_{i_1,k} \sum_{i_1,k} \left(\nabla d_{i_1,k}\right)^{\top}} \mathrm{erf}^{-1}\left(2p_i - 1\right) \end{cases} \\ \begin{cases} d_{i_2,k} \geq \gamma_{i_2,k} \\ \gamma_{i_2,k} = \sqrt{2\nabla d_{i_2,k} \sum_{i_2,k} \left(\nabla d_{i_2,k}\right)^{\top}} \mathrm{erf}^{-1}\left(2p_i - 1\right) \end{cases} \\ \vdots \\ \begin{cases} d_{i_m,k} \geq \gamma_{i_m,k} \\ \gamma_{i_m,k} = \sqrt{2\nabla d_{i_m,k} \sum_{i_m,k} \left(\nabla d_{i_m,k}\right)^{\top}} \mathrm{erf}^{-1}\left(2p_i - 1\right) \end{cases} \end{cases} \tag{4.19}$$

where $i_1, i_2, ..., i_m \in \mathbb{N}_i$ and $k = 1,\ldots,N$.

4.3.3. Cost Function

In this subsection, we explain how the cost function (4.2b) in the SMPC optimal-control problem is designed to enable the tracking of reference states as well as to minimize control inputs.

For any EV i, the cost function in expression (4.2b) [82] is chosen as

$$V_i\left(\boldsymbol{\xi}_i^{\mathrm{p}}, \boldsymbol{u}_i^{\mathrm{p}}\right) = \sum_{k=0}^{N-1} \|\xi_{i,k}^{\mathrm{p}} - \xi_{i,k}^{\mathrm{ref}}\|_{Q_i}^2 + \|u_{i,k}^{\mathrm{p}}\|_{R_i}^2 + \|\xi_{i,N}^{\mathrm{p}} - \xi_{i,N}^{\mathrm{ref}}\|_{Q_i}^2. \tag{4.20}$$

We define reference states for EV i as $\xi_{i,k}^{\mathrm{ref}}$ and $\xi_{i,N}^{\mathrm{ref}}$ to command EV i to enter or maintain a target lane at a desired velocity for every prediction step k $(k = 1,\ldots,N)$.

The weighting matrices $Q_i \in \mathbb{S}^4$ and $R_i \in \mathbb{S}^2$ are symmetric and positive definite.

4.3.4. Control Algorithm for One Vehicle

We summarize the process of solving the SMPC optimal-control problem by any EV i in Algorithm 2. Note that, in order to simplify the notation, we omitted a symbol for the current time t in the previous sections, when we defined predictions starting from time t. Here, however, in addition to the current time t, we use $t+T$ for the successor time and use $\xi_{k|t}$ and $u_{k|t}$, instead of ξ_k and u_k to describe the states and control inputs at prediction step k ahead of current time t. In simulations, we chose the system dynamic (2.4) as the real dynamics.

Algorithm 2 The SMPC problem for each EV i.

Input: A_i, B_i, p_i, t_0, t_{end}, ξ_0.
Output: $\boldsymbol{u}_i^*$

```
t = t_0
while t < t_end do
    Detect the current states of EV i and its TVs
    Update N_i
    Solve the deterministic SMPC optimal-control problem to find the optimal control input trajectory u*_{i,k|t} (k = 0,1,...,N-1)
    Apply first entry u*_{i,0|t} to real dynamics (2.4) and obtain successor state ξ*_{i,1|t}
    t = t + T
end while
```

4.4. Simulation Results

The performance of the multi-vehicle interactive system was examined via simulations of multiple vehicles on a three-lane highway. The right-most lane is the slow lane. For simplicity, in our simulations, we assumed that all three lanes have the same width and that all vehicles are the same size. The simulation setup, including the parameters of the highway, vehicles, and controller, can be found in Table 4.1. The simulations were executed in MATLAB on a desktop computer with an Intel (R) Core (TM) i3-7100 CPU @ 3.90GHz 3.91 GHz processor. The solving algorithm for the SMPC is based on the NMPC toolbox [86], in which *fmincon* is used as a solver.

We first investigated the effects of the risk parameters on the behaviors of individual SMPC-controlled vehicles in non-interactive systems. Then, we examined how various settings of risk parameters of the SMPC-controlled vehicles influence the performance of interactive systems and provide insight into how to set risk parameters.

4.4.1. The Effects of Risk Parameters on an Individual Vehicle

We studied the effects of risk parameters on the distance between vehicles in a non-interactive system based on a two-vehicle scenario as shown in Figure 4.2. Here, one vehicle is controlled by SMPC, and the other vehicle is non-reactive. The two vehicles start in different lanes with different initial velocities. Vehicle 1 (the non-reactive vehicle) stays in the center lane, and Vehicle 2 (the SMPC-controlled vehicle) merges into the center lane. The simulation lasts 10 seconds. The corresponding initial settings, including the initial states x_0, y_0, ψ_0, and v_0 and reference states y_{ref} and v_{ref} for the vehicles, are shown in Table 4.2.

We define the distance $d_{i,t}^{\text{e}}$ between EV i and its TV $\breve{i}$ at any iteration t/T (current time t) by the evaluation of the collision-avoidance constraint (4.11) along the resulting closed-loop trajectories.

$$d_{i,t}^{\text{e}} = \sqrt{\frac{\left(\Delta x_{\breve{i},t}\right)^2}{s_a^2} + \frac{\left(\Delta y_{\breve{i},t}\right)^2}{s_b^2}} \tag{4.21}$$

where the distance between EV i and its TV $\breve{i}$ at iteration t/T (time t) in the longitudinal direction is denoted as $\Delta x_{\breve{i},t}$, and its lateral counterpart is $\Delta y_{\breve{i},t}$.

Table 4.1.: Parameter Settings

Physical Meaning	**Notation**	**Value**
Length of road	l^{road}	$1500\ \text{m}$
Width of lane	w^{lane}	$5.25\ \text{m}$
Length of vehicle	l^{veh}	$5\ \text{m}$
Width of vehicle	w^{veh}	$2\ \text{m}$
Distance from mass center to front axle	l_f	$2\ \text{m}$
Distance from mass center to rear axle	l_r	$2\ \text{m}$
Lower boundary of road	$y^{\text{r,l}}$	$0\ \text{m}$
Upper boundary of road	$y^{\text{r,u}}$	$15.75\ \text{m}$
Minimum speed	$v^{\min}$	$0\ \text{m}\,\text{s}^{-1}$
Maximum allowable speed	$v^{\max}$	$70\ \text{m}\,\text{s}^{-1}$
Minimum inertial heading	$\psi^{\min}$	$-1.2\ \text{rad}$
Maximum inertial heading	$\psi^{\max}$	$1.2\ \text{rad}$
Minimum acceleration	$a^{\min}$	$-9\ \text{m}\,\text{s}^{-2}$
Maximum acceleration	$a^{\max}$	$6\ \text{m}\,\text{s}^{-2}$
Minimum front steering angle	$\delta^{\min}$	$-0.2\ \text{rad}$
Maximum front steering angle	$\delta^{\max}$	$0.2\ \text{rad}$
Semi-major axis	s_a	$9\ \text{m}$
Semi-minor axis	s_b	$5.5\ \text{m}$
Prediction horizon	N	10
Sampling time	T	$0.2\ \text{s}$
Weighting matrix	Q	$\text{diag}(0,0.5,0.1,1)$
Weighting matrix	R	$\text{diag}(3,5)$

Table 4.2.: Initial settings for a non-interactive two-vehicle scenario.

	x_0	y_0	ψ_0	v_0	y_{ref}	v_{ref}
Vehicle 1	50	7.875	0	27	7.875	27
Vehicle 2	72	2.625	0	24	7.875	30

We investigated how the risk parameters influence the distances between vehicles. The risk parameter determines the probability of collision and, thus, controls the distance between two vehicles. Small risk parameters indicate more-aggressive, less-conservative driving with a higher probability of collision and small distances.

To better visualize the influence of the risk parameters on the distances between vehicles, we chose the distance for risk parameter 0.95 as a baseline and evaluated the deviations between the baseline (minuend) and the resulting distances for each of the other risk parameters 0.70, 0.75, 0.80, 0.85, and 0.90. Each risk parameter setting was simulated 100 times, and in each simulation, the initial states of the vehicles were slightly different.

They were generated from a normal distribution with the initial states (x_0, y_0, ψ_0, v_0), as presented in Table 4.2, as the mean values and a covariance matrix $\text{diag}(0.1, 0.01, 0, 0.01)$. The simulation results are presented in Figure 4.3, and it was confirmed that the greater the risk parameter of the SMPC-controlled vehicle (the more conservative), the smaller the average distance deviations—meaning larger distances between the two vehicles.

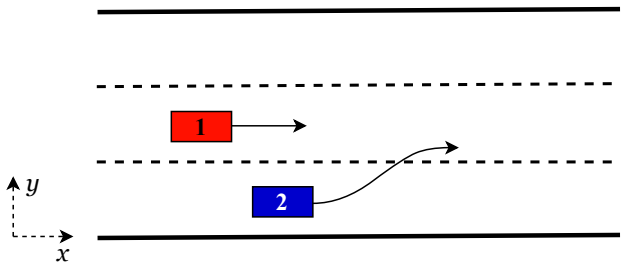


Figure 4.2.: A two-vehicle scenario. There are two vehicles on a three-lane highway. Vehicle 1, in red, is non-reactive in a non-interactive system but reactive in an interactive system and will remain in the center lane. Vehicle 2, in blue, is an SMPC-controlled vehicle, starting in the right, slow lane and later changing into the center lane. The longitudinal and lateral directions are represented by x and y, respectively.

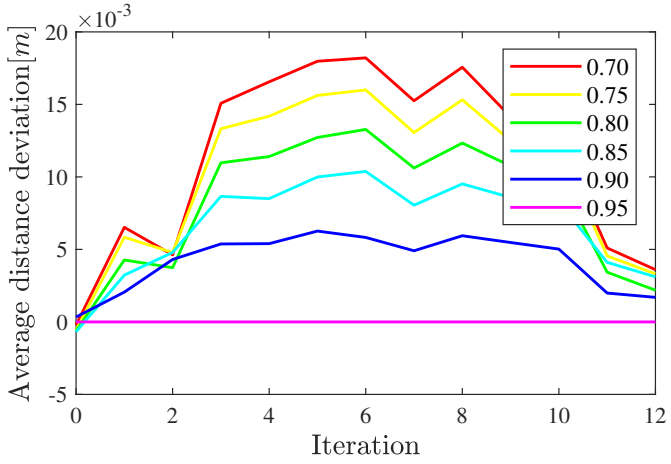


Figure 4.3.: Distance deviations in a non−interactive two−vehicle scenario. The six colored lines represent deviations between the distances for all risk parameters (0.70, 0.75, 0.80, 0.85, 0.90, and 0.95) and the distance for the risk parameter 0.95 during the whole 12 iterations, respectively. The iteration is represented by t/T, where t is the time, and T denotes the sampling time.

4.4.2. The Effects of Risk Parameters on Interactive Systems

In principle, the risk parameter will also determine the distance between vehicles in an interactive system during close interaction. The performance of an individual vehicle depends not only on its own risk parameter but also on the risk parameters of its surrounding vehicles.

The Same vs. Different Risk Parameters

We investigated the state trajectories of two vehicles for different pairs of risk parameters (p_1, p_2), including $(0.70, 0.70)$, $(0.70, 0.95)$, $(0.95, 0.70)$, and $(0.95, 0.95)$, based on the two vehicle scenario in Figure 4.2. Here, both vehicles were reactive and controlled by SMPC. In order to simulate a highly interactive scenario, we slightly adjusted the initial settings of Vehicle 2, as described in Table 4.2, by (1) moving the longitudinal initial position x_0 of Vehicle 2 to 67 m, (2) increasing the initial velocity to 25 ms^{-1}, and (3) decreasing the reference velocity v_{ref} of Vehicle 2 to 27 ms^{-1} as summarized in Table 4.3. We depict the lateral position y of these two vehicles as shown in Figure 4.4.

Table 4.3.: Initial settings for an interactive two-vehicle scenario.

	x_0	y_0	ψ_0	v_0	y_{ref}	v_{ref}
Vehicle 1	50	7.875	0	27	7.875	27
Vehicle 2	67	2.625	0	25	7.875	27

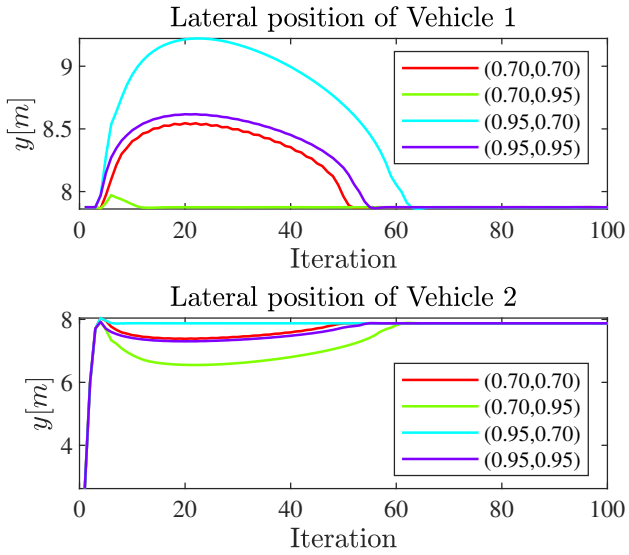


Figure 4.4.: Lateral positions in an interactive two-vehicle scenario. The risk parameter pairs (p_1, p_2) for Vehicle 1 and Vehicle 2 are specified in the legend of the figures.

Figure 4.4 shows that both Vehicle 1 and Vehicle 2 finally reach their target lanes. We first studied the performance for if both vehicles use the same risk parameter by comparing their lateral positions for risk parameter pairs $(0.70, 0.70)$ (red) and $(0.95, 0.95)$ (purple). Both vehicles reach their target lane slightly earlier when the common risk parameter is 0.70. Thus, setting a smaller risk parameter helped the vehicles reach the target lane earlier but not significantly. In total, the resulting trajectories for the risk parameter $(0.70, 0.70)$ did not differ too much from those with $(0.95, 0.95)$.

We next investigated how the vehicles behave if they use different risk parameters. Comparing the trajectories for the risk parameter pair $(0.70, 0.70)$ (red) with that for $(0.95, 0.70)$ (blue), we see that if Vehicle 1 chooses a small risk parameter (driving more aggressively), it only slightly adjusts its behavior to avoid potential collisions before reaching the target lane. Vehicle 2 behaves similarly when Vehicle 1 has a greater risk parameter (driving more conservatively). These results are comparable to those for adjusting the risk parameter of Vehicle 2, which can be found in the comparison of the plots for $(0.70, 0.70)$ (red) and $(0.70, 0.95)$ (green). These results align with the symmetric roles that the two vehicles play in the two-vehicle interactive system, where both vehicles are EVs and treat the other vehicle as a TV.

We summarize the findings for the two-vehicle interactive system as follows:

1. When two vehicles have the same risk parameters:
   - Driving more aggressively can help both to reach the target lane slightly earlier.
   - Changing the risk parameters for all vehicles in the same way does not affect the resulting trajectories significantly and only introduces slightly more or less distance between vehicles.

2. When two vehicles have different risk parameters:
    - The more aggressive that an EV drives, the fewer collision avoidance adjustments to its behavior are required before reaching the target lane.
    - An EV's more-aggressive driving style can contribute to reaching its target lane earlier.
    - A TV's more-conservative driving style can help the EV to reach the target lane earlier with fewer collision avoidance adjustments.

Resolving Conflicts

We now examine the role of the risk parameters in conflict situations. These conflicts were observed in the simulations of the previous subsection where two vehicles could not decide which of them had a higher priority to enter the target lane. This resulted in unnecessarily long lane change durations with oscillations around the target lane. We see that the aggressive vehicle typically dominated the behavior and reached the target lane earlier. When both vehicles had the same risk parameter, target lane and reference velocity, conflict situations often occurred.

We created this kind of conflict by (1) moving Vehicle 2 closer to Vehicle 1 in the longitudinal direction of the initial settings, adjusting the longitudinal initial position x_0 of Vehicle 2, as described in Table 4.3, from 67 to 66 m; and (2) setting the same risk parameter 0.95 for both vehicles. Thus, the vehicles were initially close to each other, shared the same target velocity of 27 $\mathrm{m\,s^{-1}}$, and had the same target lane, the center lane; thus, they competed to occupy the center lane.

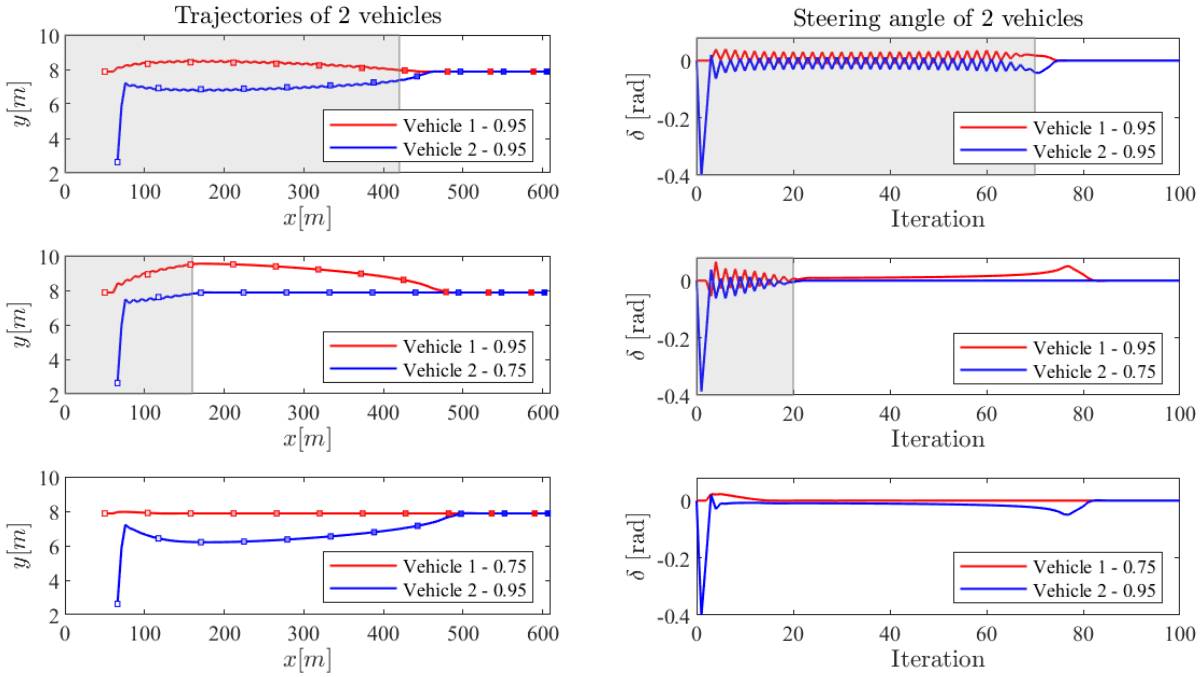
We investigated how the choice of risk parameters affects vehicle dominance by observing the position and the steering angle δ of the vehicles for different risk parameter pairs, including $(0.95, 0.95)$, $(0.95, 0.75)$, and $(0.75, 0.95)$ as shown in Figure 4.5. We mark the time periods where an obvious conflict appears in gray. The oscillating behavior, which is seen in the steering angles in particular, indicates that both vehicles repetitively switched between attempting to approach the target lane and moving away from the target lane to avoid collisions.

Figure 4.5a displays the trajectories of the vehicles for different risk parameter pairs. Figure 4.5b shows the corresponding steering angles. When both vehicles used the same risk parameter 0.95, they remained in conflict until they longitudinally reached around 420 m at approximately iteration 70 and then exited the conflict situation. Reducing the risk parameter of Vehicle 2 from 0.95 to 0.75 helped both vehicles escape from the conflict situation even earlier—at around 160 m in the longitudinal direction and after around 20 iterations. Later, Vehicle 2 occupied the target lane most of the time, playing the dominant role (see the trajectories for $(0.95, 0.75)$). However, if we reduced the risk parameter of Vehicle 1 from 0.95 to 0.75, the conflict situation did not appear anymore, and Vehicle 1 played the dominant role in terms of occupying the target lane (see the trajectories for $(0.75, 0.95)$).

Therefore, we can conclude that (1) reducing the risk parameter of one vehicle in the two-vehicle interactive system shortened or fully eliminates conflict; (2) the vehicle with a smaller risk parameter (more aggressive) tended to be the dominant one; additionally, (3) the same amount of risk parameter reduction for Vehicle 1 and Vehicle 2 had different effects on the conflict situations.

Risk Differences

In the previous discussion in Section 4.4.2, we found that maintaining a difference between the risk parameters of the two vehicles helped to either shorten or completely avoid conflict. However,



(**a**) (**b**)

Figure 4.5.: The trajectories and steering angles of Vehicles 1 and 2 for different risk parameter pairs in an interactive scenario. The gray regions in the plots mark time periods of conflict. In sub-figure (**a**), to display the relative positions of vehicles, we drew the vehicles as small squares every 10 iterations and colored the squares in different shades of red and blue. (**a**) The trajectories of the vehicles. (**b**) The steering angles of the vehicles.

it is also important to know whether the absolute value of the difference matters because this determines how much a vehicle should adjust its behaviors to escape from a conflict situation. Consequently, we decided to further investigate how gradually adjusting the risk parameters of one vehicle affected the resolution of the conflict.

We incrementally increased the risk parameter of Vehicle 1 from 0.75 to 0.95, and the risk parameter of Vehicle 2 remained unchanged, 0.95, resulting in the following risk parameter pairs: $(0.75, 0.95)$, $(0.80, 0.95)$, $(0.85, 0.95)$, $(0.90, 0.95)$, and $(0.95, 0.95)$. We evaluated the effects of these risk parameter pairs employing two metrics, the Distance Deviation (DD) and State Deviation (SD), introduced as follows:

- DD: We consider the Euclidean distance between the centers of the two vehicles (different from the distance definition in Section 4.4.1). The DD is defined as the deviation between the Euclidean distances for any risk parameter pair and the Euclidean distance for the risk parameter pair $(0.75, 0.95)$.

- SD: The deviation between states and reference states, as defined below:

$$\mathbf{err}_{\xi} = \sqrt{\frac{1}{N_{\text{ite}}+1}\sum_{n=0}^{N_{\text{ite}}}\left(\boldsymbol{\xi}_n - \boldsymbol{\xi}_n^{\text{ref}}\right)^2} \tag{4.22}$$

where $\mathbf{err}_\xi$ ($\mathbf{err}_\xi = \left(\mathrm{err}_x, \mathrm{err}_y, \mathrm{err}_\psi, \mathrm{err}_v\right)^\top$) represents the deviation between the real states $\boldsymbol{\xi}_n$ and the corresponding reference states $\boldsymbol{\xi}_n^{\mathrm{ref}}$ during all N_{ite} iterations.

The results for DD and SD are illustrated in Figures 4.6 and 4.7, respectively.

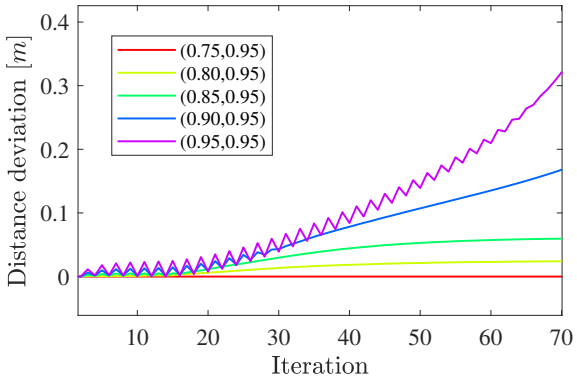


Figure 4.6.: Distance Deviations (DDs) for the vehicles with different pairs of risk parameters, $(0.75, 0.95)$, $(0.80, 0.95)$, $(0.85, 0.95)$, $(0.90, 0.95)$, and $(0.95, 0.95)$ in an interactive scenario.

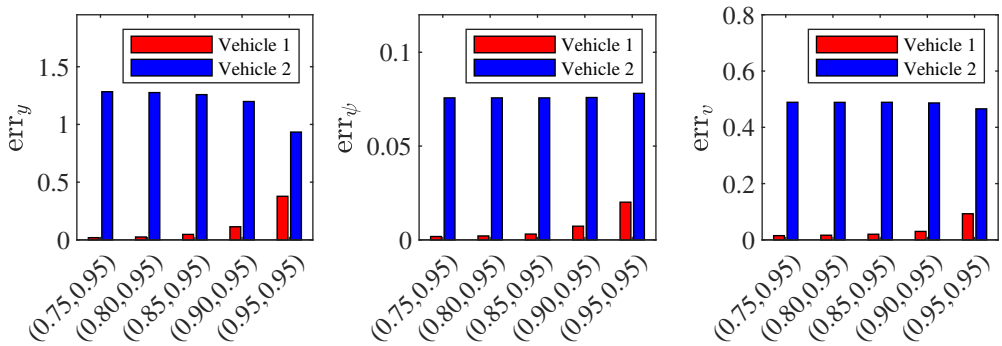


Figure 4.7.: State Deviations (SDs) for the vehicles with different risk parameter pairs, $(0.75, 0.95)$, $(0.80, 0.95)$, $(0.85, 0.95)$, $(0.90, 0.95)$, and $(0.95, 0.95)$ in an interactive scenario.

In Figure 4.6, the oscillations reflect conflict where both vehicles are struggling between reaching/maintaining the common target lane and moving away from the target lane to ensure safety, which causes variations in the distances between them. We conclude from the figure that: (1) the greater the risk parameter of Vehicle 1 (the more conservative), the larger the distance between the two vehicles, which is safer; and (2) a smaller risk parameter of Vehicle 1 can help the two-vehicle interactive system escape from the conflict situation earlier as demonstrated by the results that, for the risk parameter pairs $(0.85, 0.95)$, $(0.90, 0.95)$, and $(0.95, 0.95)$, the conflict situations end roughly after 14, 30, and 68 iterations, respectively.

We show the effect of different pairs of the risk parameters on the SD, including the deviations of the lateral position err_y, inertial heading err_ψ, and velocity err_v, in Figure 4.7. A greater risk parameter of Vehicle 1 (more conservative) causes larger state deviations for Vehicle 1, smaller deviations in the lateral positions and velocities for Vehicle 2, and larger inertial heading deviations for Vehicle 2. Therefore, when Vehicle 1 drives more conservatively, Vehicle 2 can benefit from the conservatism more in terms of reaching the target lane and reference velocity (see the first and

third sub-figures in Figure 4.7). In contrast, this results in larger inertial heading deviations for both vehicles (see the second sub-figure in Figure 4.7) because they are trapped in the conflict situations for a longer time.

The effects of one vehicle's driving style on the two-vehicle interactive system in conflict situations when the other vehicle drives conservatively are summarized as follows:

- The vehicles benefit from the conservative driving style in terms of safety.
- An aggressive driving style can help the two-vehicle interactive system escape conflict situations.
- A vehicle driving more aggressively tends to reach its target lane and reference velocity earlier.

4.5. Conclusions and Future Work

In this chapter (based on our paper [32]), we introduced a Distributed Stochastic Model Predictive Control (DSMPC) framework for a system of vehicles that are coupled through their interactive controllers. Within this framework, each vehicle is controlled by Stochastic Model Predictive Control (SMPC), and each SMPC-controlled vehicle interacts with its TVs, attempting to drive safely at a certain level through the consideration of probabilistic collision-avoidance constraints. Based on this distributed control framework, we studied the effects of risk parameters, which decide vehicles' driving styles, on non-interactive and interactive systems and provide insights into how to set risk parameters in a multi-SMPC-vehicle interactive system.

The simulation in non-interactive systems showed that, when an SMPC-controlled vehicle drives more conservatively, with a greater risk parameter, safety is increased. We found the same results in the simulations in interactive systems. Further, in interactive systems, an aggressive vehicle can reach its driving goals earlier, thus, requiring fewer adjustments to its behaviors. An individual vehicle driving conservatively can also help another vehicle to reach its driving goals earlier. Moreover, one vehicle can also influence the whole system by adjusting its own risk parameter. Vehicles might be trapped in conflict situations; therefore, they cannot decide which one has the higher priority to attain one's driving goals if there are conflicts among the goals. Modifying the risk parameters of one vehicle can help both escape conflict situations; however, the vehicle with a smaller risk parameter tends to dominate the situations.

In our future controller design, incorporating a more realistic prediction of TV's behaviors into the SMPC optimal-control problem will also be considered. In our current SMPC optimal-control problem, any EV assumes that its TVs will stay in their current lanes and maintain their current velocities. This is overly simplified and might cause huge deviations between the TVs' real trajectories and the assumed ones from the perspective of the EV. Therefore, methods that provide more precise predictions of TV behaviors are required. Research into this will be performed in Chapter 6 and also Appendix A.

The results in interactive systems confirmed our *hypotheses* that the behaviors of one vehicle are not only determined by its own control and influenced by other vehicles' behaviors but also can influence the performance of the whole system. These results can be generalized to vehicles that are controlled by other controllers in the future. Additionally, we performed simulations with two vehicles, which is the initial step for investigating multiple interactive systems. In the future, we will research scenarios that are more complicated, e.g., multiple vehicles interacting with the

surrounding vehicles and multiple vehicles having different levels of intelligence/ability to predict and react to other vehicles.

Identification of Interaction-Aware Driving Styles

5.

In a multi-vehicle autonomous driving system, an autonomous vehicle (AV) that is able to identify the driving styles of its nearby AVs can reliably evaluate the risk of collisions and make more reasonable driving decisions. Nevertheless, driving styles for an AV have not been consistently defined in previous literature, although some literature has considered that the driving style is encoded in the trajectories of the AV and can be identified as a cost function, using Maximum Entropy Inverse Reinforcement Learning (ME-IRL) methods. However, an important indicator of the driving style, i.e., how an AV reacts to its nearby AVs, is not fully incorporated in the feature design of previous ME-IRL methods.

In this chapter, we refer to driving style as a cost function of a series of weighted features instead of some semantic meanings like more aggressive and less aggressive. We design novel features to capture the AV's reaction-aware characteristics. Then, we identify the driving styles from the demonstration trajectories using a modified ME-IRL method with our newly proposed features. Here, we generate the demonstration trajectories using Stochastic Model Predictive Control (SMPC) due to its ability to qualitatively change the driving style by simply adjusting the risk parameter. The proposed method is validated using MATLAB simulation and an off-the-shelf experiment. The content of this chapter is published in [34].

5.1. Introduction

The driving style of an Autonomous Vehicle (AV) refers to how it generally achieves its driving goal and interacts with other vehicles, e.g., how to make driving decisions according to the current states, the desired speed, or the collision avoidance requirements [87]. The AV that can predict others' driving styles and incorporate the prediction into its decision-making is considered to be capable of reasonably evaluating and *reacting* to the risk of collisions with other nearby AVs. A reactive AV is expected to make safer and more reasonable driving decisions than those that do not. However, the driving style of an AV has not been consistently defined in the literature. Also, the driving styles in reactive and non-reactive situations may be different, which brings up challenges to its identification. In previous work, the driving style has been represented as a cost function with weighted features [33], [88] and can be learned from demonstration data [40].

Inverse Reinforcement Learning (IRL) that retrieves an unknown reward function from demonstration data has been employed to learn the driving style cost functions [33], [40]–[43]. Among them, [40], [41] learn driving styles based on a stochastic Markov Decision Process (MDP), adopting a probabilistic transition model. However, high-order properties, such as acceleration, can not be incorporated into the feature design with stochastic MDPs, although they are obviously important to determining driving styles. Different from them, in [42], [43], deterministic MDPs are used to model the vehicle dynamic such that accelerations are considered in the feature design.

Instead of modeling the dynamics as deterministic MDP, [33] represents trajectories using time-continuous splines that allow for incorporating acceleration into the feature design. IRL uses a linear combination of the features to capture the characteristics of trajectories [46], [88]. The IRL method aims to find the optimal weights of features and reproduce a trajectory that best mimics the driving style encoded in the demonstration trajectory generated by an expert. However, to the best of our knowledge, these methods have been mainly applied to single-AV cases where the reactions among different AVs were rarely considered. Specifically, how an AV reacts to a nearby AV is not incorporated in the features used to identify the driving style.

Learning the driving style using IRL methods requires demonstration data. Stochastic Model Predictive Control (SMPC) is capable of generating demonstration trajectories that encode the desired driving styles. The driving style of an AV controlled by SMPC depends heavily on the risk parameter in the probabilistic constraint to avoid colliding with obstacles [16], [19], [20], [32], [89]. How the risk parameter qualitatively affects the driving style of an SMPC-controlled vehicle has been described in [32]. A greater risk parameter leads to a more conservative driving style, and vice versa [32]. Therefore, SMPC can get qualitatively aggressive or conservative driving styles by simply adjusting the risk parameter.

5.1.1. Approach Overview

In this chapter, we solve the driving style identification problem for a two-vehicle system. We stand on the position of the ego AV and identify the driving styles of its nearby AV using a Maximum Entropy IRL (ME-IRL) method. Different from the conventional methods used for single-vehicle cases [33], [42], [43], we design four additional features to depict the ego AV's reactions to its nearby AV. Among them, three are active only when the AV is close to the nearby AV. This requires a triggering condition to activate them. The detailed contributions are summarized as follows:

1. We propose four novel features to capture the reaction-aware characteristics of the driving style for a two-vehicle case;

2. We design a triggering condition based on an ellipse index to activate the reaction-aware features.

5.1.2. Chapter Overview

The rest of Chapter 5 is organized as follows. Section 5.2 introduces the the generation of demonstration trajectories using SMPC. In Section 5.3, we present the modified ME-IRL method to identify the driving style from the demonstration trajectory. The simulation studies that validate the efficacy of the proposed method are shown in Section 5.4. Eventually, Section 5.5 concludes Chapter 5.

5.2. Generation of Demonstration Trajectories using SMPC

We use SMPC to generate demonstration trajectories. An Ego Vehicle (EV) that avoids a Target Vehicle (TV) solves the problem introduced in (2.8) of Section 2.2.2 at each time step, as shown

below:

$$
\begin{aligned}
&\min_{\boldsymbol{u}} V(\boldsymbol{\xi},\boldsymbol{u}) \\
\text{s. t. } &\xi_{k+1} = \mathscr{F}(\xi_k,u_k),\ k=0,1,\cdots,N-1, \\
&\xi_k \in \Xi,\ k=1,\cdots,N, \\
&u_k \in \mathscr{U},\ k=0,1,\cdots,N-1, \\
&\mathbf{Pr}(\xi_k \in \Xi_{\text{safe}}) \geq p,\ p\in[0.5,1],\ k=1,2,\cdots,N.
\end{aligned}
$$

We use the linearized and discretized version of the kinematic bicycle model introduced in (2.4) (see Section 2.1.2) as the system model $\xi_{k+1} = \mathscr{F}(\xi_k,u_k)$ to generate EV predictions. The model is shown as follows

$$\xi_{k+1} = \xi_0 + T\mathscr{F}^{\text{c}}(\xi_0,0) + A(\xi_k-\xi_0) + Bu_k, k=0,\ldots,N-1,$$

where the state and control input vectors are $\xi_k = (x_k,y_k,\psi_k,v_k)^\top$ and $u_k = (a_k,\delta_k)^\top$, respectively. Additionally, the linearized, discretized system matrices A and B [39] are given in (2.5). We expect the EV to track reference states ξ_k^{ref} and control inputs u_k of the EV to be small; therefore, cost function $V = \sum_{k=0}^{N-1}(\|\xi_k-\xi_k^{\text{ref}}\|_Q^2 + \|u_k\|_R^2) + \|\xi_N-\xi_N^{\text{ref}}\|_{Q_N}^2$. The weighting matrices are $Q\in\mathbb{R}^{4\times4}$, $R\in\mathbb{R}^{2\times2}$ and $Q_N\in\mathbb{R}^{4\times4}$. The sets Ξ and $\mathscr{U}$ denote the sets of admissible states and the control inputs, respectively, where the road boundaries, physical limitations of the EV and the traffic rules are taken into consideration. The safety constraint $\mathbf{Pr}(\xi_k \in \Xi_{\text{safe}}) \geq p,\ p\in[0.5,1]$ ensures that the EV avoids colliding with the TV. A realization of the safety constraint is shown below.

Safety Constraint with Risk Parameter

The safety constraint ensures that the EV remains outside an ellipse region around the TV with a probability of p. Thus, p is a risk parameter. The center of the ellipse is also the center of the TV. Given the longitudinal distance $\Delta x_k = x_k - x_{\text{TV},k}$ and the lateral distance $\Delta y_k = y_k - y_{\text{TV},k}$ between the EV and TV, the hard constraint that keeps the EV outside the ellipse region is

$$d_k = \frac{{\Delta x_k}^2}{{s_a}^2} + \frac{{\Delta y_k}^2}{{s_b}^2} - 1 \geq 0, \tag{5.2}$$

where the size of the safety ellipse is determined by the length of the semi-major axis s_a and semi-minor axis s_b. $d_k \geq 0$ is one way to realize the hard constraint $\xi_k \in \Xi_k^{\text{safe}}$ in (2.8e). We soften constraint $d_k \geq 0$ employing a probabilistic constraint

$$\mathbf{Pr}(d_k \geq 0) \geq p. \tag{5.3}$$

The safety constraint is active only when the EV is close to the TV and reacts to the TV. The risk parameter dominantly influences the driving style of the EV. The greater p is, the less aggressive the EV is, and vice versa.

The demonstration trajectory $\boldsymbol{r}_{\mathscr{D}}$ is generated in the form of states $\boldsymbol{\xi}$ by SMPC and then re-represented using quintic polynomials to ensure smoothness (see Section 2.4).

5.3. Modified ME-IRL

The driving style is quantified by a cost function represented by a linear combination of features that capture the important characteristics of the trajectories. The ME-IRL method aims to identify the weights of the features that best fit the driving style of the demonstration trajectory and reproduce trajectories that mimic the driving style of the demonstration [33], [42], [43]. The features together with their weights describe the driving style and can be used to measure the similarity between trajectories. How the ME-IRL works was introduced in Section 2.3.

To identify the driving style of an AV in a multi-vehicle system where AVs react to others' behaviors, we modify the ME-IRL method in [33], [42], [43]. The main modifications include four novel reactive features that capture the characteristics while the AV is reacting to nearby AVs. One of them (f_{tiv}) is active during the entire time and the other three (f_{sd}, f_{ed}, f_{id}) are triggered only when the AV is close to the nearby vehicles. The overall framework of our method is shown in Figure 5.1.

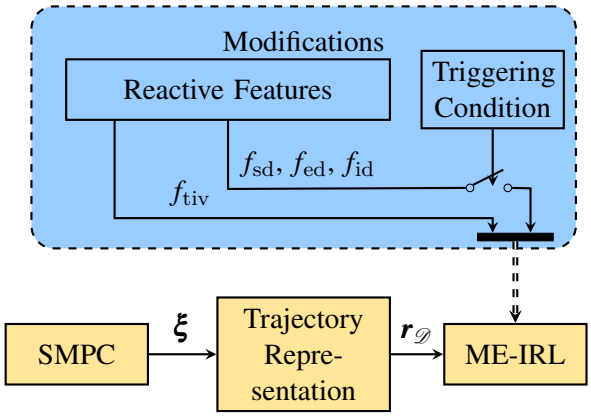


Figure 5.1.: The overall framework of the method.

5.3.1. Novel Features

We introduce features based on a scenario where the controlled vehicle, the EV, avoids a TV, and reacts to the TV when close to it. Apart from the existing features introduced in Section 2.3.1, we propose the following four novel features.

a) The intervehicular time: $f_{\text{tiv}} = \int_t \frac{v^x_{\text{lane}}}{\left| r^x_{\text{TV}}(\tau) - r^x_{\text{EV}}(\tau) \right|} \mathrm{d}\tau$, where v^x_{lane} is the limit velocity of the target lane, which is also the desired speed of the target lane.

b) Start distance: $f_{\text{sd}} = e^{-\left| r^y_{\text{EV}}(t_{\text{trg}}) - r^y_{\text{TV}}(t_{\text{trg}}) \right|}$, where t_{trg} is determined by the triggering condition which will be discussed in Section 5.3.2.

c) End distance: $f_{\text{ed}} = e^{-\left| r^y_{\text{EV}}(t_{\text{trg}}+T_{\text{rct}}) - r^y_{\text{TV}}(t_{\text{trg}}+T_{\text{rct}}) \right|}$, where T_{rct} is a value determined empirically.

d) Integral distance: $f_{\text{id}} = \int_{t_{\text{trg}}}^{t_{\text{trg}}+T_{\text{rct}}} \left| r^y_{\text{EV}}(\tau) - r^y_{\text{EV}}(t_{\text{trg}}) \right| \mathrm{d}\tau$ which is used to capture the changes of the position in the lateral direction during time period $[t_{\text{trg}}, t_{\text{trg}}+T_{\text{rct}}]$.

The features above are important for collision avoidance between two vehicles. The intervehicular time (TIV) [28] is used to measure safety between two vehicles. In general, a greater TIV stands for a safer situation. We require a smaller cost for a safer situation. Therefore, we take the reciprocity of TIV as a feature. The distance-based features f_{sd}, f_{ed}, and f_{id} are proposed to capture the reaction characteristics between the EV and TV in the lateral direction. These three features are active only when the EV is close to the TV. The moment the features become active is referred to as the triggering time t_{trg}. We decide the triggering time t_{trg} using the triggering condition introduced below.

5.3.2. Triggering Condition

The triggering condition for the reactive features f_{sd}, f_{ed}, and f_{id} is based on an ellipse index used to describe the squared elliptical distance [90] between the positions of the EV and the TV, which allows for a more accurate approximation of the physical dimensions of a vehicle, i.e.,

$$s_e = \frac{\Delta x_t{}^2}{l_a{}^2} + \frac{\Delta y_t{}^2}{l_b{}^2} \tag{5.4}$$

where Δx_t and Δy_t are the longitudinal and lateral distances between the EV and TV at time t. Here, the values of parameters l_a and l_b are the same as the s_a and s_b in constraint (5.2). Therefore, $\sqrt{s_e}$ is the elliptical distance between the EV and the TV scaled by l_a and l_b. We set a threshold value λ for the triggering condition

$$s_e < \lambda \tag{5.5}$$

which corresponds to an elliptical region, referred to as a *scaled ellipse*. The reactive features f_{sd}, f_{ed}, and f_{id} are activated when the triggering condition (5.5) is satisfied for the first time. This moment is the triggering time t_{trg}. The reactive features remain active for a duration T_{rct} and become inactive again at time $t_{\mathrm{trg}} + T_{\mathrm{rct}}$.

5.3.3. Feature Scaling

The features including our novel features introduced in this chapter are calculated over different time periods along the trajectory. For example, f_{tiv} is active during the entire time, while f_{sd}, f_{ed}, and f_{id} are only active from t_{trg} to $t_{\mathrm{trg}} + T_{\mathrm{rct}}$. Therefore, the features calculated over a longer time period tend to have greater feature values and have a stronger impact on the cost function $L(\theta, \mathbf{r})$ than those over a shorter time period. To balance the influences of features with different time periods, we scale their values [91] using a matrix $\Omega = \mathrm{diag}(\omega_{ax}, \omega_{ay}, \cdots, \omega_{id}) \in \mathbb{R}^{10\times 10}$, where $\omega_{ax}, \omega_{ay}, \cdots, \omega_{id}$ are empirical scaling coefficients. Thus, the scaled features are $\boldsymbol{f}(\mathbf{r}) = [\omega_{ax} f_{\mathrm{ax}}, \omega_{ay} f_{\mathrm{ay}}, \cdots, \omega_{id} f_{\mathrm{id}}]^\top \in \mathbb{R}^{10}$. In Chapter 5, we set ω_{il}, ω_{el}, ω_{sd}, ω_{ed}, and ω_{id} as 10 and the others (the features are calculated over the whole time period) as 1.

5.3.4. Generation of Control Points

The original demonstration trajectories generated by SMPC are represented in discrete time using the states $\xi_t = [x_t, y_t, \phi_t, v_t]^\top$ of the EV at each sampling time t. However, we need smooth trajectories to be able to use our features that include second-order derivatives. Therefore, we represent the demonstration and the reproduced trajectories using piecewise quintic spline segments to ensure their smoothness. The spline segments are parameterized using control points comprised of the

positions, velocities, and accelerations in the longitudinal and lateral directions. We obtain the positions directly from the state vector. Velocities v_t^x and v_t^y can be calculated employing $v_t^x = v_t \cos\phi$ and $v_t^y = v_t \sin\phi$, respectively. Acceleration a_t^x and a_t^y are approximated by $a_t^x = \frac{v_{t+1}-v_{t-1}}{2T_s}\cos\phi$ and $a_t^y = \frac{v_{t+1}-v_{t-1}}{2T_s}\sin\phi$, respectively, where T_s denotes the time interval between $t-1$ and t. For reproduced trajectories, we simply calculate the velocities and accelerations using the following equations: $v_t^x = \dot{r}_t^x$, $v_t^y = \dot{r}_t^y$, $a_t^x = \ddot{r}_t^x$, and $a_t^y = \ddot{r}_t^y$.

5.4. Simulation Results

We examine the efficacy of the modified ME-IRL method with our newly designed features in simulations. The simulation is run on a laptop with an i7-10875H CPU under 2.30 GHz. The optimization problem in SMPC is solved by employing *fmincon* embedded in the NMPC toolbox [86] in MATLAB.

5.4.1. Simulation Setup

We consider a lane-changing scenario on a three-lane highway, as shown in Figure 5.2. The EV (red) starts in the right lane (bottom lane) and will later move to the center lane and accelerate. The TV (blue) starts and remains in the center lane at a constant velocity 28 m/s. The EV tries to avoid colliding with the TV while moving toward the center lane. The initial states of the EV and TV are $\xi_{\mathrm{EV},0} = [80, 2.625, 0, 25]^\top$ and $\xi_{\mathrm{TV},0} = [60, 7.875, 0, 28]^\top$, respectively. The target velocity of the EV is 30 m/s. We assume that the width of the lanes is the same and the vehicles are the same size. The parameters of the three-lane highway and the vehicles are displayed in Table 5.1.

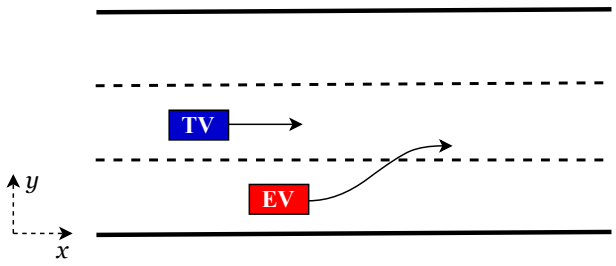


Figure 5.2.: A two-vehicle lane-changing scenario.

Table 5.1.: Parameters of lanes and vehicles

Physical meaning	**Symbol**	**Value**
Width of lane	w^{lane}	5.25 m
Length of vehicle	l^{veh}	5 m
Width of vehicle	w^{veh}	2 m
Distance from vehicle mass center to front axle	l_f	2 m
Distance from vehicle mass center to rear axle	l_r	2 m

We generate the demonstration trajectories of the EV using SMPC, while the trajectory of the TV is specified in advance through hard coding. The SMPC is with a risk parameter $p = 0.7$, a prediction horizon $N = 10$, a sampling time $T_s = 0.2\,\mathrm{s}$, and a total duration $T = 6.2\,\mathrm{s}$ (31 time steps). The boundaries in the constraints are specified as $y \in [l^{\mathrm{veh}}, 3w^{\mathrm{lane}} - l^{\mathrm{veh}}]$, $\phi \in [-0.05, 0.05]\,\mathrm{rad}$, $v \in [0, 70]\,\mathrm{m/s}$, $a \in [-9, 6]\,\mathrm{m/s^2}$ and $\delta \in [-0.05, 0.05]\,\mathrm{rad}$. The safety ellipse is determined by the semi-major axis $s_a = 15\,\mathrm{m}$ and semi-minor axis $s_b = 3\,\mathrm{m}$. The weighting matrices of the cost function are $Q = \mathrm{diag}(10^{-6}, 0.2, 50, 0.2)$, $R = \mathrm{diag}(1, 10)$ and $Q_N = \mathrm{diag}(10^{-6}, 0.2, 50, 0.2)$. The learning termination threshold is $\bar{\varepsilon} = 0.01$.

5.4.2. The Demonstration Trajectories and the Trigger Time

The discrete-time trajectories generated by SMPC are represented using piecewise quintic spline segments to ensure smoothness. These smooth trajectories are the demonstration trajectories in the learning process of the IRL. In this subsection, we show the demonstration trajectories of the EV and the TV, which reflect the reactions of the EV to the TV while avoiding potential collisions. We first illustrate the change of the scaling index of ellipse s_e (or the ellipse index, defined in equation (5.4)) along the longitudinal direction in the top subfigure of Figure 5.3. The demonstration trajectories of the EV and TV with the safety and the scaled ellipses are shown in the bottom subfigure of Figure 5.3, where the ellipses are only displayed at the starting, middle, and ending instants, namely the SMPC time steps 1, 16, and 31, for brevity. The positions of the vehicles at these time steps are also shown as colored rectangles. From the top subfigure, we can see that the ellipse index s_e decreases from 80 m to 170 m since the two vehicles get close to each other during this period (as shown in the bottom figure). Then, from around 170 m, s_e gradually increases since the EV actively avoids the potential collisions with the TV, which indicates the reaction of the EV to the potential collision with the TV. The EV's reaction can also be seen from the bottom subfigure, where the trajectory segment between roughly 155 m and 220 m shows a different curvature than the one before 155 m. The triggering time of the reaction can be determined as the first time when $s_e < \lambda$ is satisfied, i.e., $t_{\mathrm{trg}} = 3\,\mathrm{s}$, where the threshold λ is empirically set to 1.82.

5.4.3. The Reproduced Trajectories

Given the demonstration trajectories of the EV and the TV, we learn the driving style of the EV using the modified ME-IRL. Figure 5.4 compares the learning performance of the ME-IRL methods without (top subfigure) and with (bottom subfigure) our novel features. Specifically, the learning performance is evaluated by the deviation between the trajectories reproduced by the learned driving styles and the demonstration trajectories. Besides, in each subfigure, we illustrate the reproduced trajectories at iterations 1, 11, 21, and the terminal iteration. We can observe that the trajectories reproduced by the driving style with novel features better fit the demonstration trajectory, although the one without novel features is learned faster (26 over 37). Specifically, the ME-IRL without novel features produces larger trajectory gaps than the one with these features, especially between around 150 m and 220 m. This indicates that the conventional ME-IRL can not fully learn the reaction of the EV to the TV, but our modified ME-IRL can.

Similar conclusions can also be drawn from Figure 5.5 which displays the lateral-direction (y-axis) velocity and acceleration of the EV in the learning process using our modified ME-IRL method and the corresponding demonstration data. Here, we only display the lateral direction

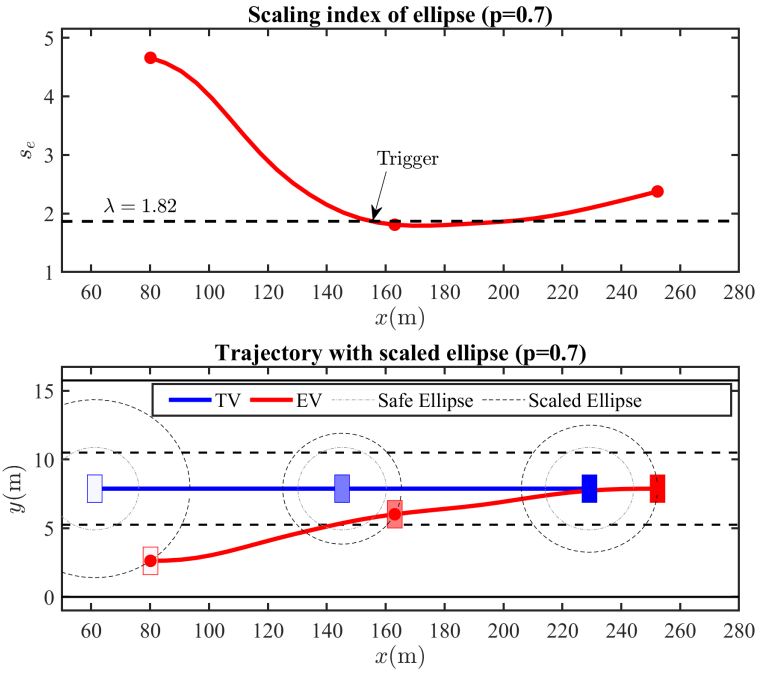


Figure 5.3.: The ellipse index s_e defined in equation (5.4) and the demonstration trajectories of the EV and TV.

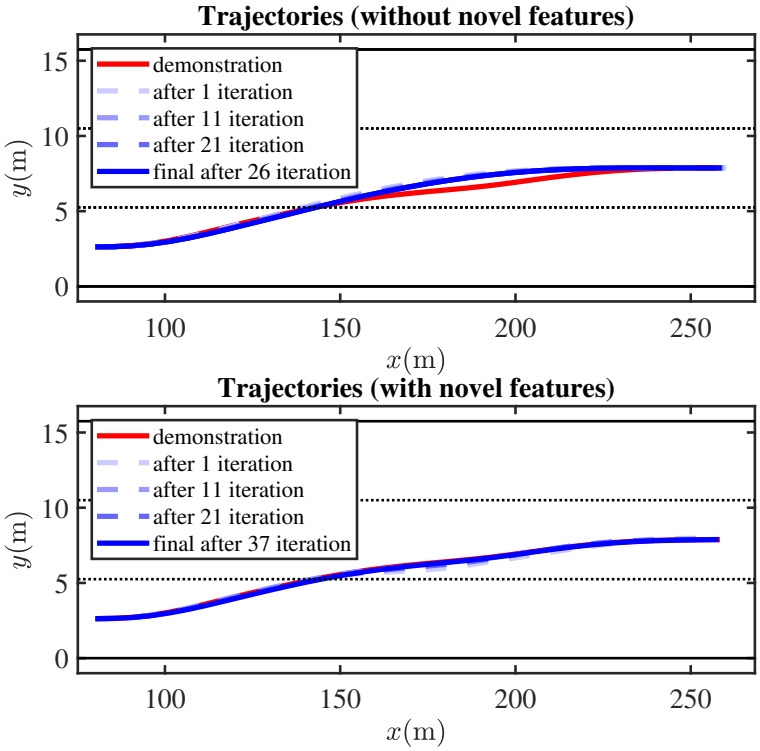


Figure 5.4.: The reproduced trajectories of the EV using ME-IRL without (top) and with (bottom) the novel features, compared with the demonstration data.

since it is much more important than the longitudinal direction for the lane-changing scenario. It can be seen that the velocity and acceleration in the lateral direction converge to those of the demonstration trajectory, which indicates successful learning.

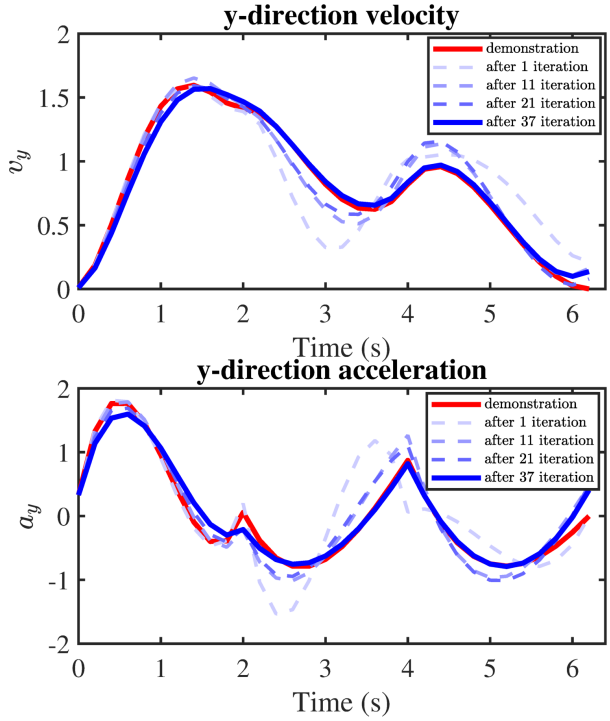


Figure 5.5.: The converging lateral-direction velocity and acceleration of the EV with iterations 1, 11, 21, and 37.

The results in this subsection show the success of using our modified ME-IRL method to learn the reaction of the EV to the TV. The reaction is an important characteristic within the driving style of reactive vehicles. Learning the reaction of a controlled vehicle to other vehicles is the basis for later investigating mutual interactions between vehicles in a multi-vehicle interactive system. Moreover, the weights of the features determine the driving style and are employed to generate trajectories with the same driving styles. However, it is impossible to give a driving style some semantic meanings, such as more aggressive and less aggressive, according to the feature weights.

5.4.4. Simulation experiment in Off-the-shelf Software

To demonstrate the applicability of the proposed method to practical autonomous driving systems, we conduct an experiment in an off-the-shelf simulation environment, the Siemens® Simcenter Prescan Software. The experiment setup is the same as Section 5.4.1. We exhibit EV's optimal trajectory reproduced using our modified ME-IRL method in the experiment. This allows us to compare the reproduced trajectory with the demonstration trajectory. The footage of this experimental study is published in `https://youtu.be/S672tUtHFyY`, where both the bird's eye view from above and the first perspective view from the TV are provided. The experimental results show that the EV's trajectory reproduced using the learned driving style with novel features is very similar to the demonstration trajectory generated in Section 5.4.2. This validates the efficacy of our

method. The successful experiment also demonstrates the applicability of the proposed method in practical autonomous driving systems.

5.5. Conclusions and Future Work

In this chapter, we extend a Maximum Entropy Inverse Reinforcement Learning (ME-IRL) method to identify the driving styles of an Autonomous Vehicle (AV) in a two-vehicle system incorporating the reaction among the vehicles. We propose novel features to capture the reaction-aware characteristics that indicate the driving styles of an AV while it actively avoids colliding with the nearby AV. An ellipse index is proposed to determine the triggering time to activate some reaction-aware features. Simulation in MATLAB and experiment in *Simcenter Prescan* validate the efficacy and applicability of our method. The novel features are designed for a lane-changing scenario.

Accurately predicting the future behaviors of nearby vehicles plays a crucial role in enhancing the safety of AVs. Generally, the driving styles of vehicles tend to be consistent. This consistency creates the potential to predict the behaviors of surrounding vehicles based on their driving styles. Identifying the driving styles of vehicles serves as the initial step for subsequent behavior prediction based on driving styles. This prediction method will later be introduced in Chapter 6. In the next chapter, we will incorporate the learned driving style into the behavior prediction of AVs in multi-vehicle scenarios. Moreover, we use a triggering condition to decide when the three features become active, which matches the characteristics of the reaction that happens only when the two vehicles are close to each other. However, the triggering condition adds additional computation to the late learning process. In Chapter 6, we will research features that capture the characteristics of the reaction but can eliminate the triggering condition.

Additionally, our current demonstration data are generated through MATLAB simulations. In the future, it is worth to testing our methods based on real datasets, such as INTEREATION [92].

Interaction-Aware Prediction of Vehicle Trajectories

6.

In Chapter 4, when a controlled vehicle predicts the behaviors of the surrounding vehicles, it simply assumes that the surrounding vehicles stay in their current lanes and maintain their current velocities. The oversimplified prediction method ignores that the surrounding vehicles might interact with other vehicles or simply adjust their behaviors according to their driving goals. This might lead to substantial deviations between the actual trajectories of the surrounding vehicles and the predicted trajectories. Therefore, prediction methods that can more precisely and reasonably predict the trajectories of the surrounding vehicles, e.g., with the interactions taken into consideration, are required. The driving style of a vehicle remains consistent and is therefore valuable information for predicting its future behaviors. In Chapter 5, we have introduced an MR-IRL-based method that identifies the driving style cost function of a controlled vehicle and further reproduces trajectories that have the same driving style. The driving style of a controlled vehicle includes the characteristic when interacting with another vehicle and is encoded in the demonstration trajectories.

In this chapter, we partially adopt the idea of learning the driving styles of the vehicles using the ME-IRL-based method in Chapter 5 and introduce how to predict the behaviors of surrounding vehicles with the knowledge of their driving styles including interaction characteristics. We examine the efficacy of our interaction-aware prediction method for vehicles in interactive vehicle systems in simulation studies.

6.1. Introduction

To avoid potential collisions with neighboring vehicles, a controlled vehicle must *predict* the future behaviors of the nearby vehicles and *react* accordingly to the predictions. Each controlled vehicle reacts to other vehicles in the environment. Thus, in scenarios involving multiple controlled vehicles, they *interact* with each other. In this chapter, our focus will be on predicting the behaviors of the neighboring vehicles from the perspective of a controlled vehicle in multi-vehicle interactive systems.

The prediction method in Chapter 6 states that any SMPC-controlled vehicle simply assumes that its neighboring vehicles will maintain their current lanes and speeds throughout the prediction horizon [32]. The advantage of this prediction method resides in its high computational efficiency [32]. However, this method ignores that the surrounding vehicles might interact with other vehicles to avoid collisions and achieve their driving goals. This ignorance might result in significant deviations between the actual behaviors and predicted ones and subsequently lead to failures in preventing potential dangers. This motivated us to propose/investigate a prediction method that takes into consideration the interaction between vehicles when predicting the behaviors of the surrounding vehicles.

Many researchers are dedicated to developing interaction-aware behavior predictions and propose numerous valuable methods [26], including physics-based methods [93], [94], classic machine learning-based methods [95]–[103], deep learning-based methods [104]–[129], and reinforcement learning-based methods [130]–[137]. physics-based methods use physical models to generate trajectory predictions [26]. With appropriate physical models, physics-based methods [93], [94] can be employed in various scenarios with minimal computational expenses and rapid execution. Nevertheless, the accuracy of prediction outcomes using such models relies significantly on the inputs and the selection of the model. However, for many vehicle, the inputs of the surrounding vehicles [26] are usually unknown, which causes low accuracy in using models to predict the behaviors of surrounding vehicles. With the exception of physics-based methods [93], [94], all interaction-aware prediction methods are grounded in learning. In comparison to physics-based methods, classic machine learning-based methods can encompass a greater number of factors, resulting in higher accuracy despite an increased computing cost [26]. Most of the classic machine learning-based methods are maneuver-based, predicting trajectories with a predefined maneuver known as a prior. However, the diversity and significant variation in vehicle maneuvers across different scenarios result in poor generalization ability [26]. Deep learning-based methods can achieve relatively high accuracy in prediction and are suitable for more complex environments and a long time horizon. Nonetheless, these methods have drawbacks, e.g., the requirement for a significant amount of training data and a sharp increase in computing costs when additional factors are considered [26]. Reinforcement learning-based methods are likely to generate trajectories with higher accuracy compared to deep learning methods, particularly in longer-time domains. Nevertheless, the majority of these methods tend to be computationally demanding in the retrieval of an expert cost function, making them more suitable for offline calculations rather than online ones. In this chapter, we expect an accurate interaction-aware prediction method that will be processed offline within interactive vehicle systems, e.g., multi-SMPC-vehicle interactive systems introduced in Chapter 4. This expectation is fulfilled through the utilization of a reinforcement learning-based method.

Reinforcement learning-based prediction methods in [130]–[133] employ traditional IRL methods, a branch of Reinforcement Learning (RL), aiming to infer the underlying reward function that guides an expert's behavior. Bhattacharyya *et al.*, [134] employs Generative Adversarial Imitation Learning (GAIL) to generate trajectories that are similar to those of an expert. In [135]–[137], Deep IRL, using deep neural networks to model complex reward functions, is adopted to predict driving behaviors. Both GAIL and Deep IRL are rooted in the principles of IRL while leveraging deep neural networks to model complex functions and representations. Combining with learning networks, [134], [136], [137] have an enhanced capability to extract expert demonstrations and take into consideration a wider array of factors. On the other hand, GAIL and Deep IRL come with extra computational intensity and demand prolonged training periods in contrast to traditional IRL [26]. Generally, prediction methods grounded in traditional IRL can circumvent the aforementioned disadvantages but yield predictions of sufficient quality, particularly when confronted with less intricate behavioral patterns.

6.1.1. Approach Overview

In this chapter, we, standing on the position of a controlled vehicle in multi-vehicle systems, develop an interaction-aware prediction method based on IRL. More specifically, we identify the driving styles of vehicles using IRL [33], [34], [42], [43] (partially adopt the idea in Chapter 5)

and then generate the predicted behaviors according to the identified driving styles. Here, the driving style of a vehicle is defined as a cost function of a series of weighted features including interaction-aware features [34]. Each controlled vehicle considers the predicted behaviors of its surrounding vehicles for collision avoidance, where behaviors are represented using vehicle trajectories. Each controlled vehicle identifies the driving styles of other vehicles according to the multiple demonstration trajectories. To test the efficacy of the prediction method, we apply the prediction method to the SMPC-controlled vehicle within a multi-vehicle interactive system introduced in Chapter 4. We summarize the contributions of this chapter as follows.

1. We propose novel features including interaction-aware features to capture the characteristics of controlled vehicles in two-vehicle interactive systems. We eliminate the triggering condition in reactive features (see Chapter 5) to avoid the additional computation burden on the learning process.

2. We propose a novel idea for demonstrations, segments as demonstrations, that avoids coarseness and lack of details caused by using trajectories with a long temporal scope as demonstrations while enabling a comprehensive understanding of the behaviors of the expert.

3. We generate trajectories with similar driving styles to the surrounding vehicles and treat these trajectories as the predictions of the surrounding vehicles.

4. We incorporate interaction-aware prediction into the problem formulation of SMPC-controlled vehicles.

6.1.2. Chapter Overview

The rest of this chapter is organized as follows. In Section 6.2, we introduce the IRL-based interaction-aware prediction method that is incorporated into the SMPC problem of a vehicle in interactive systems. Section 6.3 demonstrates the efficacy of our interaction-aware prediction method for SMPC-controlled vehicles in interactive systems through simulation studies. We conclude the work introduced in this chapter and outline future directions in Section 6.4.

6.2. Interaction-Aware Prediction of Vehicle Trajectories

The general SMPC optimal-control problem for each vehicle in interactive systems has been introduced in Chapter 4. This section will first introduce the problem formulation of incorporating the IRL-based interaction-aware prediction method into the SMPC problem of a vehicle. Then, we dive deeper into the IRL-based interaction-aware prediction method, presenting the novel features employed to capture the characteristics of the controlled vehicles, the novel demonstrations, the learning process, and the prediction of the trajectories of a Target Vehicle (TV). In the end, we present how exactly we incorporate the prediction of TV into the SMPC optimal-control problem using an interface.

6.2.1. Problem Formulation

As introduced in Chapter 4, for a two-SMPC-vehicle interactive system, each vehicle is an EV and treats the other vehicle as a TV. Each EV solves the following optimal control problem at each time step.

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \sum_{k=0}^{N-1} (\|\xi_k - \xi_k^{\mathrm{ref}}\|_Q^2 + \|u_k\|_R^2) + \|\xi_N - \xi_N^{\mathrm{ref}}\|_{Q_N}^2 \tag{6.1a}$$

$$\text{s. t. } \xi_{k+1} = \mathscr{F}(\xi_k, u_k),\ k = 1, \cdots, N-1, \tag{6.1b}$$

$$\xi_k \in \Xi,\ k = 1, \cdots, N, \tag{6.1c}$$

$$u_k \in \mathscr{U},\ k = 1, \cdots, N-1, \tag{6.1d}$$

$$\Pr\left(\xi_k \in \Xi_k^{\mathrm{safe} \leftarrow \boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}}\right) \geq p,\ k = 1, \cdots, N, \tag{6.1e}$$

where $\xi_k \in \Xi_k^{\mathrm{safe} \leftarrow \boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}}$ ensures that the EV avoids colliding with TV over the prediction horizon. We denote the predicted trajectory of the TV using $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$.

In this chapter, we obtain the prediction of the TV $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$ employing a variant of the IRL-based method in Chapter 5. Specifically, we learn the driving style encoded in the demonstration trajectories of the TV and then reproduce continuous trajectories that have the same driving style; the optimal reproduced trajectory $\boldsymbol{r}_{\mathrm{TV}}^*$ are discretized and then treated as the predicted trajectories of the TV. How we apply the learned driving style to obtain the optimal reproduce trajectory $\boldsymbol{r}_{\mathrm{TV}}^*$ of TV is shown below:

$$\boldsymbol{r}_{\mathrm{TV}}^* = \arg\min_{\boldsymbol{r}_{\mathrm{TV}}} \boldsymbol{\theta}_{\mathrm{TV}}{}^{\top} \boldsymbol{f}(\boldsymbol{r}_{\mathrm{TV}} | \boldsymbol{r}_{\mathrm{IV}}) \tag{6.2}$$

where $\boldsymbol{\theta}_{\mathrm{TV}}$ represents the feature weights learned from the demonstration of the TV using an IRL-based method. $\boldsymbol{f}(\boldsymbol{r}_{\mathrm{TV}} | \boldsymbol{r}_{\mathrm{IV}})$ is the feature vector of the TV. Considering that the EV's future trajectory $\boldsymbol{r}_{\mathrm{IV}}$ will influence the TV's future trajectory $\boldsymbol{r}_{\mathrm{TV}}$, we incorporate this influence into the feature design of the TV, meaning that the mutual interaction between the EV and TV is taken into account in the prediction of the TV. We refer to the EV that is considered in the prediction of the TV as an interactive vehicle (IV) in this chapter.

In this chapter, we borrow the learning and reproducing ideas of the IRL method in Chapter 5 and make two adjustments, shown as follows. We design new features that eliminate the triggering condition in Chapter 5, and consequently can avoid the unnecessary computational burden on the learning process. Moreover, we propose a novel demonstration method that uses segments instead of the entire trajectories as demonstrations. The novel demonstrations enable a comprehensive understanding of the behaviors of the expert while avoiding coarseness and lack of details. The details about the novel IRL method are presented in the following sections.

6.2.2. Features

In this section, we introduce the features used in our learning method in this chapter. The driving style of a vehicle is quantified by a cost function represented by a linear combination of features that capture the important characteristics of the trajectories, such as velocity and acceleration. For a two-vehicle interactive system, how the vehicles interact with each other also reflects their driving styles and therefore should be considered in the feature design. In Chapter 5, we introduced four novel reactive features to capture the interactions between vehicles and a triggering condition to activate three of the reactive features. However, the existence of the triggering condition brings an additional computation burden to the late learning process. Therefore, we eliminate the triggering condition and all the features are active during the learning process in this chapter.

In this chapter, we adopt the x-acceleration feature f_{ax} introduced in Section 2.3.1 and define novel features shown below:

a) x-Jerk: $f_{\mathrm{jx}} = \int_t \left\| \dddot{r}^x(\tau) \right\|^2 \mathrm{d}\tau$. This feature quantifies passenger comfort.

b) y-Speed: $f_{\mathrm{vy}} = \int_t \|\dot{r}^y(\tau)\|^2 \mathrm{d}\tau$ reflects how fast the vehicle moves in the lateral direction in order to change lane.

c) Desired x-Speed: 1) $f_{\mathrm{v1}} = \int_t |v_{\mathrm{des}} - \dot{r}^x(\tau)|^2 \mathrm{d}\tau$ or 2) $f_{\mathrm{v2}} = \int_t |v_{\mathrm{des}} - \dot{r}^x(\tau)| \mathrm{d}\tau$, where the desired speed v_{des} is the limited velocity of the end lane. This feature captures the deviation between the longitudinal speed $\dot{r}^x$ and speed v_{des}.

   Note that, both of the expressions for 'Desired x-Speed' work and we only select one of them during the learning process. This is also suitable for other features with two expressions.

d) Desired Lane: 1) $f_{\mathrm{l1}} = \int_t |l_{\mathrm{des}} - r^y(\tau)|^2 \mathrm{d}\tau$ or 2) $f_{\mathrm{l2}} = \int_t |l_{\mathrm{des}} - r^y(\tau)| \mathrm{d}\tau$ capture the vehicle's ability to track the desired lane l_{des}.

   Features f_{v1}, f_{l1} are in the quadratic form, while f_{v2}, f_{l2} take an absolute value form [33].

e) Safety Level: $f_{\mathrm{sl}} = \int_t \frac{v^2}{\|\boldsymbol{r}(\tau) - \boldsymbol{r}_{\mathrm{IV}}(\tau)\|^2} \mathrm{d}\tau$ captures the safety level, specifically the collision avoidance situation, in an interactive system. Here, $\boldsymbol{r}_{\mathrm{IV}}$ is the future trajectory of the IV that influences the TV.

f) Safe Region: Generally, a convex region that can cover the shape of a vehicle is considered a forbidden region to other vehicles[16]. We choose an elliptic region as the convex region [19], [28], [32], [76]–[78] because it allows a more accurate approximation of the physical dimensions of a vehicle. Our idea is to detect how well the vehicle can stay outside an ellipse region that has the same center as the IV, expressed by 1) $f_{\mathrm{e1}} = \int_t \frac{1}{\frac{(r^x(\tau) - r^x_{\mathrm{IV}}(\tau))^2}{{l_a}^2} + \frac{(r^y(\tau) - r^y_{\mathrm{IV}}(\tau))^2}{{l_b}^2}} \mathrm{d}\tau$ or 2) $f_{\mathrm{e2}} = \int_t \max(0, \lambda - \frac{(r^x(\tau) - r^x_{\mathrm{IV}}(\tau))^2}{{l_a}^2} - \frac{(r^y(\tau) - r^y_{\mathrm{IV}}(\tau))^2}{{l_b}^2}) \mathrm{d}\tau$, where the length of the semi-major and semi-minor axes are represented by l_a and l_b, respectively. λ serves as a threshold for detecting whether the vehicle is safely staying outside the ellipse region of the IV or not.

   In features 'Safety Level' and 'Safety Region', we take the influence of the neighboring vehicle on the vehicle into account. Therefore, these features are referred to as interaction-aware features.

We design two feature expressions for each of the three features 'Desired x-Speed', 'Desired Lane', and 'Safe Region'. Only one feature expression is selected for each feature during the learning process. When selecting feature expressions for 'Desired x-Speed' and 'Desired Lane', we choose either the quadratic form for both features or the absolute value form for them. These cause various feature combinations given below:

$$\boldsymbol{f}_{\mathrm{qua,rec}} = [f_{\mathrm{ax}}, f_{\mathrm{jx}}, f_{\mathrm{vy}}, f_{\mathrm{v1}}, f_{\mathrm{l1}}, f_{\mathrm{sl}}, f_{\mathrm{e1}}]^\top \tag{6.3a}$$

$$\boldsymbol{f}_{\mathrm{abs,rec}} = [f_{\mathrm{ax}}, f_{\mathrm{jx}}, f_{\mathrm{vy}}, f_{\mathrm{v2}}, f_{\mathrm{l2}}, f_{\mathrm{sl}}, f_{\mathrm{e1}}]^\top \tag{6.3b}$$

$$\boldsymbol{f}_{\mathrm{qua,max}} = [f_{\mathrm{ax}}, f_{\mathrm{jx}}, f_{\mathrm{vy}}, f_{\mathrm{v1}}, f_{\mathrm{l1}}, f_{\mathrm{sl}}, f_{\mathrm{e2}}]^\top \tag{6.3c}$$

$$\boldsymbol{f}_{\mathrm{abs,max}} = [f_{\mathrm{ax}}, f_{\mathrm{jx}}, f_{\mathrm{vy}}, f_{\mathrm{v2}}, f_{\mathrm{l2}}, f_{\mathrm{sl}}, f_{\mathrm{e2}}]^\top \tag{6.3d}$$

Any feature combination will serve as $\boldsymbol{f}(\boldsymbol{r}_{\mathrm{TV}}|\boldsymbol{r}_{\mathrm{IV}})$, as shown in (6.2), during the learning and reproducing processes.

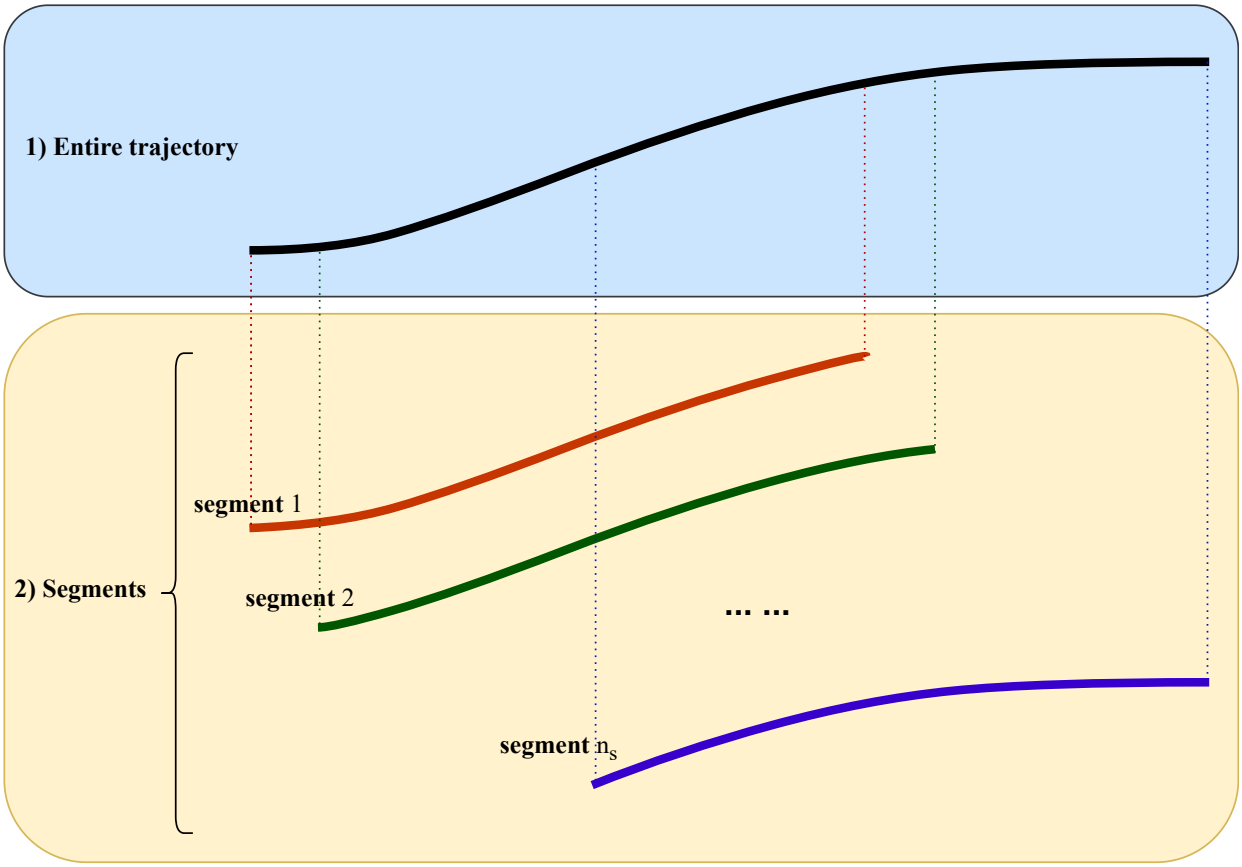


Figure 6.1.: The segments obtained from the entire trajectory.

6.2.3. Demonstration Trajectories

In this section, we introduce the demonstration trajectories employed in the learning process in this chapter. In Chapter 5, the temporal scope of demonstration trajectories is reasonably longer than that of a predicted TV trajectory. Annotating demonstration trajectories with a longer temporal scope allows for a more comprehensive understanding of how expert behaviors evolve over time. However, a longer temporal scope potentially leads to coarser and less detailed captures of the characteristics. Specifically, some characteristics have dominant influences on the feature values in certain stages but only tiny impacts in other stages; therefore, directly using the feature values along an entire trajectory results in overlooking the overwhelming influences on feature values in certain stages. These motivate us to select appropriate demonstrations that enable a comprehensive understanding of the behaviors of the expert while avoiding coarseness and lack of details. Specifically, we adjust the demonstration by segmenting each entire trajectory into multiple shorter segments. In this section, we first introduce how we generate demonstration trajectories in a two-SMPC-vehicle interactive system and then show our adjustments to them in detail. Note that the demonstration trajectories generated using SMPC are not the demonstration trajectories we eventually use in the learning process; therefore, to avoid confusion, we refer to 'demonstration trajectories generated using SMPC' as 'original demonstration trajectories' in this chapter.

Generation of Original Demonstration Trajectories

We generate demonstration trajectories in a two-SMPC-vehicle interactive system where each vehicle is controlled by SMPC and interacts with the other. To obtain the demonstration trajectories of the two SMPC-controlled vehicles, we first set the two-vehicle scenario and then generate the demonstration trajectories for the two vehicles based on the scenario. In order to generalize the demonstration trajectories, we repeatedly generate multiple original demonstration trajectories by adding Gaussian noise to the initial states of the vehicles.

To get an original demonstration trajectory of any SMPC-controlled vehicle (as an EV), we place one SMPC-controlled vehicle and one hard code vehicle (as a TV) whose trajectory is predefined in the two-vehicle scenario and let the SMPC-controlled vehicle react to the hard code vehicle. During generating the original demonstrations, each SMPC-controlled vehicle solves the following problem at each time step:

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \sum_{k=0}^{N-1} (\|\xi_k - \xi_k^{\mathrm{ref}}\|_Q^2 + \|u_k\|_R^2) + \|\xi_N - \xi_N^{\mathrm{ref}}\|_{Q_N}^2 \tag{6.4a}$$

$$\text{s. t. } \xi_{k+1} = \mathscr{F}(\xi_k, u_k),\ k = 1, \cdots, N-1, \tag{6.4b}$$

$$\xi_k \in \Xi,\ k = 1, \cdots, N, \tag{6.4c}$$

$$u_k \in \mathscr{U},\ k = 1, \cdots, N-1, \tag{6.4d}$$

$$\Pr\left(\xi_k \in \Xi_k^{\mathrm{safe} \leftarrow \boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pre-def}}}\right) \geq p,\ k = 1, \cdots, N. \tag{6.4e}$$

where $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pre-def}}$ represents the predefined trajectory of the TV, which is different from the $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$ in problem (6.1). This is the only difference between problems (6.1) and (6.4). We place the trajectories generated by the two SMPC-controlled vehicles in one scenario, resulting in the original demonstration trajectories represented using SMPC states. Each original demonstration trajectory will be re-represented using piecewise quintic spline segments to ensure their smoothness, obtaining trajectory $\boldsymbol{r}_{\mathscr{D}}$.

Segmented Demonstration Trajectories

Different from treating each entire trajectory $\boldsymbol{r}_{\mathscr{D}}$ as a demonstration trajectory in Chapter 5, in this chapter we segment each entire trajectory $\boldsymbol{r}_{\mathscr{D}}$ into multiple shorter segments that partially overlap with the adjacent ones and treat these segments as demonstration trajectories. All the segments together contain information about the entire trajectory, ensuring a comprehensive view.

How we segment the entire trajectory into segments is illustrated in Figure 6.1. Assuming we generate n_d entire trajectories $\boldsymbol{r}_{\mathscr{D}}^i$ $(i = 1, 2, \ldots, n_d)$, we first calculate the mean of these n_d trajectories, obtaining the average trajectory $\tilde{\boldsymbol{r}}_{\mathscr{D}}$, and then segment trajectory $\tilde{\boldsymbol{r}}_{\mathscr{D}}$ into n_s segments $\boldsymbol{s}_{\mathscr{D}}^1, \boldsymbol{s}_{\mathscr{D}}^2, \ldots, \boldsymbol{s}_{\mathscr{D}}^{n_s}$ with different initial points but the same time duration, where 1, 2 and n_s are the serial numbers of the segments. Each segment overlaps with a part of trajectory $\tilde{\boldsymbol{r}}_{\mathscr{D}}$ as well as the adjacent segments. In this chapter, we choose the time interval between the initial points of two adjacent segments the same as the sampling time in the SMPC. Also, the time duration of each segment is the same as the prediction horizon in SMPC.

We learn the driving style from the segment demonstrations and reproduce trajectories with the same driving style. We first discretize the reproduced trajectories and then treat these discretized trajectories as predicted trajectories of the TV of an EV.

Algorithm 3 Learning from segment demonstrations

Input: Original demonstrations $\boldsymbol{r}^i_{\mathscr{D}}$ $(i = 1,2,\ldots,n_d)$ and threshold $\bar{\varepsilon}$ for ending learning process
Output: Optimal feature weights $\boldsymbol{\theta}^*$

Obtain the average trajectory $\tilde{\boldsymbol{r}}_{\mathscr{D}}$ of trajectories $\boldsymbol{r}^i_{\mathscr{D}}$ $(i = 1,2,\ldots,n_d)$
Segment the average trajectory $\tilde{\boldsymbol{r}}_{\mathscr{D}}$ to get segment demonstrations $\boldsymbol{s}^j_{\mathscr{D}}$ $(j = 1,2,\ldots,n_s)$
Compute the empirical average feature $\boldsymbol{f}_{\mathscr{D}}$ using (6.5)
Initialize all elements of $\boldsymbol{\theta}$ to 1, learning rate α to 0.01, learning error $\varepsilon^0_{\boldsymbol{\theta}}$ to 0, the counter of the learning iterations l to 1, and guess of each segment $\boldsymbol{s}^j$ $(j = 1,2,\ldots,n_s)$
while $l \neq 0$ **do**
    Optimize each segment $\boldsymbol{s}^j$ by minimizing cost function $L(\boldsymbol{\theta},\boldsymbol{s}^j) = \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{s}^j)$, obtaining $\boldsymbol{s}^{j\,*}_{\boldsymbol{\theta}}$
    Calculate the approximated gradient $\nabla_{\boldsymbol{\theta}} \approx \frac{1}{n_s}\sum_{j=1}^{n_s} \boldsymbol{f}(\boldsymbol{s}^{j\,*}_{\boldsymbol{\theta}}) - \boldsymbol{f}_{\mathscr{D}}$
    Obtain learning error $\varepsilon^l_{\boldsymbol{\theta}} = \|\nabla_{\boldsymbol{\theta}}\|_2$
    if $\left|\varepsilon^l_{\boldsymbol{\theta}} - \varepsilon^{l-1}_{\boldsymbol{\theta}}\right| < \bar{\varepsilon}$ **then**
        break
    else
        $l = l + 1$
        Update weight vector using updating law $\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha\nabla_{\boldsymbol{\theta}}$
    end if
end while
return $\boldsymbol{\theta}^* = \boldsymbol{\theta}$

6.2.4. Learning of Feature Weights

Due to the novel segment demonstrations, the learning of the feature weights is different from that in Section 2.3.2. Specifically, the empirical feature values of demonstration trajectories, the segments, become

$$\boldsymbol{f}_{\mathscr{D}} = \frac{1}{n_s}\sum_{j=1}^{n_s} \boldsymbol{f}(\boldsymbol{s}^j_{\mathscr{D}}) \tag{6.5}$$

and the approximated gradient $\nabla_{\boldsymbol{\theta}}$ for updating the weight $\boldsymbol{\theta}$ becomes

$$\nabla_{\boldsymbol{\theta}} \approx \frac{1}{n_s}\sum_{j=1}^{n_s} \boldsymbol{f}(\boldsymbol{s}^{j\,*}_{\boldsymbol{\theta}}) - \boldsymbol{f}_{\mathscr{D}} \tag{6.6}$$

where $\boldsymbol{s}^{j\,*}_{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{s}^j} L(\boldsymbol{\theta},\boldsymbol{s}^j)$, where $L(\boldsymbol{\theta},\boldsymbol{s}^j) = \boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{s}^j)$ represents the driving style of the reproduced segment $\boldsymbol{s}^j$ in the learning process. The rest of the learning process is the same as that in Section 2.3.2. We summarize the learning process in Algorithm 3.

At the end of the learning process, we obtain the optimal feature weights for the demonstrations. The optimal weights are later used to reproduce the trajectory of the corresponding vehicles and these reproduced trajectories are discretized and then incorporated into the SMPC formulation as the predicted trajectories of TVs (see $\boldsymbol{\theta}_{\mathrm{TV}}$ in (6.2)).

6.2.5. Prediction of Vehicle Trajectories

In previous subsections, we have introduced how we identify the driving styles of the vehicles in interactive systems. We now present how to employ the identified driving style to predict the trajectories of a TV in the SMPC formulation of a vehicle.

An EV, controlled by SMPC, predicts the trajectories of its TV at each iteration. Consequently, the current positions of the TV and the future trajectory of the IV $\boldsymbol{r}_{\mathrm{IV}}$ are updated at each SMPC iteration. We estimate the future trajectory of the IV $\boldsymbol{r}_{\mathrm{IV}}$ by 1) obtaining the predicted trajectory of the EV by solving its SMPC controller in the previous iteration, denoted as $\boldsymbol{r}_{\mathrm{IV}}^{-}$, and then 2) estimate the current position from the previous one. The reproduced trajectory of the TV $\boldsymbol{r}_{\mathrm{TV}}$ is parameterized by a set of control points $\bar{\boldsymbol{c}} = [\boldsymbol{c}_0^\top \ \boldsymbol{c}_1^\top \ \cdots \ \boldsymbol{c}_{N_s}^\top]^\top \in \mathbb{R}^{6(N_s+1)}$. The initial control point $\boldsymbol{c}_0$ is required to be fixed during the learning process. We obtain $\boldsymbol{c}_0$ using both current state $\xi_{\mathrm{TV},0} = [x_{\mathrm{TV},0}, y_{\mathrm{TV},0}, \psi_{\mathrm{TV},0}, v_{\mathrm{TV},0}]^\top$ at the current iteration and the predicted trajectory from the previous iteration, denoted using $\boldsymbol{r}_{\mathrm{TV}}^{-}$, as shown below

$$\boldsymbol{c}_0 = \begin{bmatrix} [x_{\mathrm{TV},0} & \dot{r}_{\mathrm{TV}}^{x-}(T_s) & \ddot{r}_{\mathrm{TV}}^{x-}(T_s)]^\top \\ [y_{\mathrm{TV},0} & \dot{r}_{\mathrm{TV}}^{y-}(T_s) & \ddot{r}_{\mathrm{TV}}^{y-}(T_s)]^\top \end{bmatrix}, \tag{6.7}$$

where velocities ($\dot{r}_{\mathrm{TV}}^{x-}(T_s)$ and $\dot{r}_{\mathrm{TV}}^{y-}(T_s)$) and accelerations ($\ddot{r}_{\mathrm{TV}}^{x-}(T_s)$ and $\ddot{r}_{\mathrm{TV}}^{y-}(T_s)$) are calculated using the first and second derivative of the trajectory $\boldsymbol{r}_{\mathrm{TV}}^{-}$ at time T_s, respectively. In this chapter, time T_s is set the same as the sampling time in SMPC.

We summarize the process of predicting the trajectory of a TV in the Algorithm 4:

Algorithm 4 Predicting the trajectory of TV

Input: $\boldsymbol{\theta}_{\mathrm{TV}}, \boldsymbol{r}_{\mathrm{TV}}^{-}, \xi_{\mathrm{TV},0}, \boldsymbol{r}_{\mathrm{IV}}^{-}$
Output: $\boldsymbol{r}_{\mathrm{TV}}^{*}$
1: Estimate $\boldsymbol{r}_{\mathrm{IV}}$ based on $\boldsymbol{r}_{\mathrm{IV}}^{-}$
2: Approximate the first control point $\boldsymbol{c}_0$ using $\xi_{\mathrm{TV},0}$ and $\boldsymbol{r}_{\mathrm{IV}}^{-}$ (see (6.7))
3: Initialize the rest of the control points $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{N_s}$
4: Optimize $[\boldsymbol{c}_1^\top \boldsymbol{c}_2^\top \cdots \boldsymbol{c}_{N_s}^\top]^\top$ by minimizing ${\boldsymbol{\theta}_{\mathrm{TV}}}^\top \boldsymbol{f}(\boldsymbol{r}_{\mathrm{TV}} | \boldsymbol{r}_{\mathrm{IV}})$ while letting $\boldsymbol{r}_{\mathrm{IV}}$ and $\boldsymbol{c}_0$ remain fixed
5: Represent $\boldsymbol{r}_{\mathrm{TV}}^{*}$ using $[\boldsymbol{c}_0^\top \boldsymbol{c}_1^\top \cdots \boldsymbol{c}_{N_s}^\top]^\top$

Because we employ a discrete vehicle model in our SMPC formulation (6.1), it is necessary to discretize $\boldsymbol{r}_{\mathrm{TV}}^{*}$ before incorporating it into the SMPC formulation. Therefore, we discretize $\boldsymbol{r}_{\mathrm{TV}}^{*}$ by sampling its data at time series $[0, T_s, \cdots, NT_s]$, obtaining a discrete trajectory $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{discrete}}$, where T_s denotes the sampling time and N represents the prediction horizon in SMPC. The discrete trajectory $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{discrete}}$ is treated as the predicted trajectory of TV, $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}} = [r_{\mathrm{TV},0}^{\mathrm{pred}}, r_{\mathrm{TV},1}^{\mathrm{pred}}, \ldots, r_{\mathrm{TV},N}^{\mathrm{pred}}]^\top$ (see (6.1e)).

6.2.6. Interface between the TV Prediction and SMPC formulation

An EV incorporates the predicted trajectory of the TV into the collision avoidance constraint (6.1e) of its SMPC problem to avoid colliding with the TV. However, due to the presence of uncertainty/error in the TV prediction and the probabilistic chance constraint (6.1e), the SMPC problem (6.1) cannot be solved directly [16]. To handle this, we tighten the probabilistic collision avoidance constraint (6.1e) to an equivalent deterministic version that is dependent on the prediction

error and the risk parameter in SMPC problem (6.1). Similar to that in Section 4.3.2, we adopt the constraint-tightening approach from [16], [19]. The difference from Section 4.3.2 is we do not need to consider the uncertainty propagation along the TV prediction because the uncertainties at different prediction steps along the predicted trajectory are independent. Details are shown below.

We assume that the prediction error $\boldsymbol{e}_{\mathrm{TV}} = [e_{\mathrm{TV},0}, e_{\mathrm{TV},1}, \ldots, e_{\mathrm{TV},N}]^\top$ is subject to a Gaussian distribution with zero mean and covariance matrix $\boldsymbol{\Sigma}^e = [\Sigma_0^e, \Sigma_1^e, \ldots, \Sigma_N^e]^\top$, where $\Sigma_0^e = 0$ and uncertainties $\Sigma_1^e, \Sigma_2^e, \ldots, \Sigma_N^e$ are independent. The predicted trajectory of the TV $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$ is decomposed into two elements: a deterministic component $\tilde{\boldsymbol{r}}_{\mathrm{TV}}$ and an error component $\boldsymbol{e}_{\mathrm{TV}}$

$$\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}} = \tilde{\boldsymbol{r}}_{\mathrm{TV}} + \boldsymbol{e}_{\mathrm{TV}} \tag{6.8}$$

where $\tilde{\boldsymbol{r}}_{\mathrm{TV}} = [\tilde{r}_{\mathrm{TV},0}, \tilde{r}_{\mathrm{TV},1}, \ldots, \tilde{r}_{\mathrm{TV},N}]^\top$.

The collision avoidance constraint $\xi_k \in \Xi_k^{\mathrm{safe} \leftarrow \boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}}$ is realized using an ellipse region outside the TV, as shown below

$$d_k = \frac{\Delta x_k{}^2}{s_a{}^2} + \frac{\Delta y_k{}^2}{s_b{}^2} - 1 \geq 0, \tag{6.9}$$

where the longitudinal distance Δx_k and lateral distance Δy_k between the EV and TV are obtained using

$$\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} x_{\mathrm{EV},k} - \tilde{r}_{\mathrm{TV},k}^x \\ y_{\mathrm{EV},k} - \tilde{r}_{\mathrm{TV},k}^y \end{bmatrix}. \tag{6.10}$$

Therefore, the probabilistic collision avoidance constraint (6.1e) can be rewritten as

$$\mathbf{Pr}(d_k \geq 0) \geq p. \tag{6.11}$$

Given (6.10), constraint (6.9) is linearized around the deterministic position $(\tilde{r}_{\mathrm{TV},k}^x, \tilde{r}_{\mathrm{TV},k}^y)$ of TV, which results in

$$d_k + \nabla d_k e_{\mathrm{TV},k} \geq 0 \tag{6.12}$$

where

$$\nabla d_k = \frac{\partial d_k}{\partial \tilde{r}_{\mathrm{TV},k}} = \left(\frac{-2\Delta x_k}{s_a^2}, \frac{-2\Delta y_k}{s_b^2} \right) \tag{6.13}$$

With (6.12), we rewrite the probabilistic constraint (6.11) as follows

$$\mathbf{Pr}\left(-\nabla d_k e_{\mathrm{TV},k} \leq d_k\right) \geq p,\ p \in [0.5, 1], k = 1, \ldots, N, \tag{6.14}$$

which can be divided into a deterministic inequality

$$d_k \geq \gamma_k \tag{6.15}$$

and a probabilistic equation

$$\mathbf{Pr}\left(-\nabla d_k e_{\mathrm{TV},k} \leq \gamma_k\right) = p,\ p \in [0.5, 1], k = 1, \ldots, N. \tag{6.16}$$

The probabilistic equation (6.16) is tightened by selecting γ_k as

$$\gamma_k = \sqrt{2\nabla d_k \Sigma_k^e (\nabla d_k)^\top}\, \mathrm{erf}^{-1}(2p-1) \tag{6.17}$$

in accord with Theorem 1 in [19]. As a result, the probabilistic (6.1e) is tightened to a deterministic inequality (6.15) and a deterministic equation (6.17).

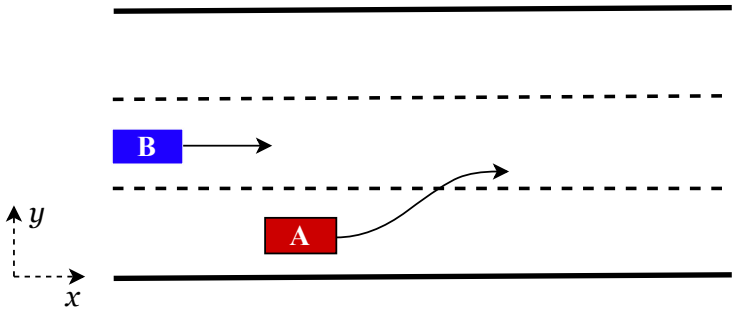


Figure 6.2.: A two-vehicle scenario for generating demonstrations. There are two vehicles, Vehicle A in red and Vehicle B in blue, on a three-lane highway. Vehicle A starts in the right, slow lane and later changes into the center lane. Vehicle B will remain in the center lane.

6.3. Simulation Results

We examine the efficacy of our interaction-aware prediction method for SMPC-controlled vehicles in simulation studies. The simulations are executed on a laptop with processor i7-10875H CPU, 2.30 GHz. The optimization in SMPC is tackled by utilizing the *fmincon* function integrated into the NMPC toolbox in MATLAB, as referenced in [86]. The simulations are based on a two-SMPC-vehicle-interactive system on a three-lane highway. The simulation setup including the parameters of the highway and the two SMPC-controlled vehicles are introduced in Section 6.3.1. The processes of learning the driving styles of the vehicles are exhibited in Section 6.3.2. In the remaining subsections, we investigate the performance of our interaction-aware prediction method for SMPC-controlled vehicles. We first evaluate the performance of the controllers and the prediction when using the four different feature combinations (introduced in Section 6.2.2) in the learning process and provide insights on selecting the optimal feature vector in Section 6.3.3. Then, we demonstrate the efficacy of the interaction-aware features through an ablation study in Section 6.3.4. Subsequently, Section 6.3.5 evaluates the efficacy of our novel segment demonstration trajectories. Specifically, we compare the performances of the controllers and prediction obtained by using segments as demonstrations with the performances when applying the entire trajectories as demonstrations. In Section 6.3.6, we compare our interaction-aware IRL-based TV prediction method with the simple TV prediction method used in Chapter 4; based on the results, we present the strengths and weaknesses of the prediction methods.

6.3.1. Simulation Setup

We introduce our simulation setup in this section. We generate demonstrations of two SMPC-controlled vehicles based on the three-lane highway, as shown in Figure 6.2. For simplicity, we

Table 6.1.: Parameters of the highway and vehicles

Physical meaning	**Notation**	**Value**
Width of lane	w^{lane}	5.25 m
Lower boundary of road	$y^{\text{r,l}}$	0 m
Upper boundary of road	$y^{\text{r,u}}$	15.75 m
Length of vehicle	l^{veh}	5 m
Width of vehicle	w^{veh}	2 m
Distance from mass center to front axle	l_f	2 m
Distance from mass center to rear axle	l_r	2 m

Table 6.2.: Common parameter settings of the two SMPC-controlled vehicles

Physical meaning	**Notation**	**Value**
Minimum speed	$v^{\min}$	0 $\mathrm{m\,s^{-1}}$
Maximum allowable speed	$v^{\max}$	70 $\mathrm{m\,s^{-1}}$
Minimum acceleration	$a^{\min}$	-9 $\mathrm{m\,s^{-2}}$
Maximum acceleration	$a^{\max}$	6 $\mathrm{m\,s^{-2}}$
Minimum orientation angle of vehicle	$\psi^{\min}$	-0.2 rad
Maximum orientation angle of vehicle	$\psi^{\max}$	0.2 rad
Minimum front steering angle	$\delta^{\min}$	-0.1 rad
Maximum front steering angle	$\delta^{\max}$	0.1 rad
Semi-major axis of safe ellipse	s_a	15 m
Semi-minor axis of safe ellipse	s_b	3 m
Prediction horizon	N	10
Sampling time	T_s	0.2 s

assume that the widths of the lanes are the same and the vehicles are the same size. The parameters of the highway and vehicles are summarized in Table 6.1.

As introduced in the 'Generation of Original Demonstration Trajectories' in Section 6.2.3, we obtain the original demonstration trajectory of any individual SMPC-controlled vehicle by placing one SMPC-controlled vehicle and one hard code vehicle with a predefined trajectory in the environment and letting the SMPC controlled vehicle react to the hard code vehicle. To enhance generalization, we generate 5 original demonstrations for each vehicle by adding noise to each value of the initial state vector of the SMPC-controlled vehicle. The noise is represented by random numbers generated from a Gaussian distribution using the 'randn' function in MATLAB.

The two SMPC-controlled vehicles have both shared and distinct parameters. We summarize the shared parameters in Table 6.2 and the distinct ones in Table 6.3. The learning results are used to predict the trajectories of TVs. These TVs are also SMPC-controlled vehicles themselves and they have the same parameters as that in the generation process.

6.3.2. Exhibiting the Process of 'Learning from Segments'

In this section, we exhibit the process of learning the driving style from segments. The feature combination for the learning process is selected from the four feature combinations, e.g.,

Table 6.3.: Parameters of individual vehicles for generating the original demonstration trajectories

Vehicle	Physical meaning	Notation	Value
A	Weighting matrix of states	Q	$\mathrm{diag}(0, 0.25, 5, 0.5)$
	Weighting matrix of control inputs	R	$\mathrm{diag}(0.33, 8)$
	Risk parameter	p	0.9
	Initial states	ξ_0	$[12, 2.625, 0, 25]^\top$
	Reference states	ξ_{ref}	$[1000,\ 7.875,\ 0,\ 30]^\top$
B	Weighting matrix of states	Q	$\mathrm{diag}(0, 50, 20, 0.1)$
	Weighting matrix of control inputs	R	$\mathrm{diag}(0.01, 20)$
	Risk parameter	p	0.9
	Initial states	ξ_0	$[0, 7.875, 0, 25]^\top$
	Reference states	ξ_{ref}	$[1000,\ 7.875,\ 0,\ 25]^\top$

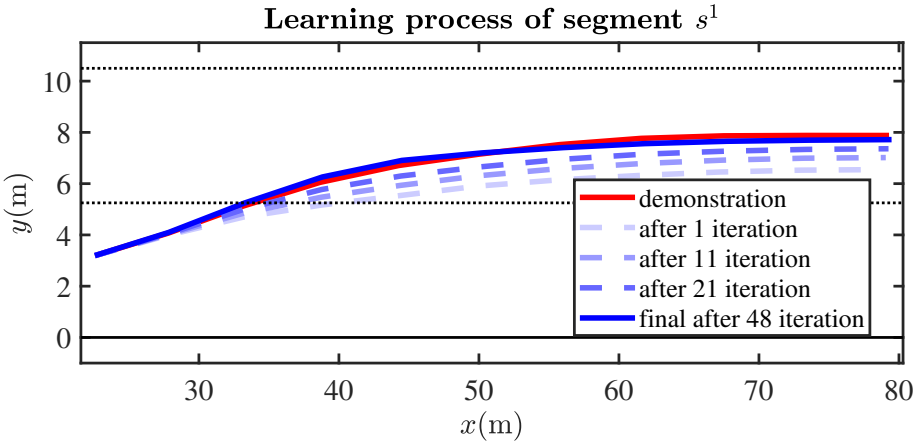
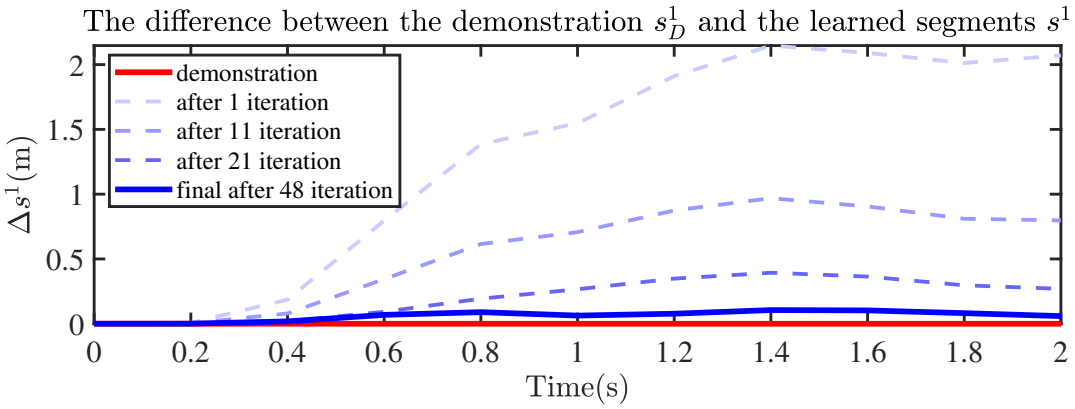
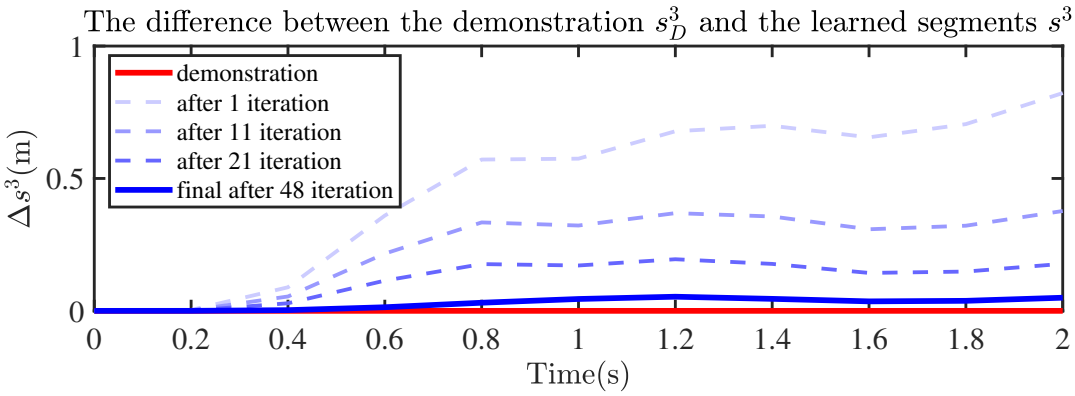


Figure 6.3.: Reproduced trajectories of the first segment $\boldsymbol{s}^1$ of Vehicle A

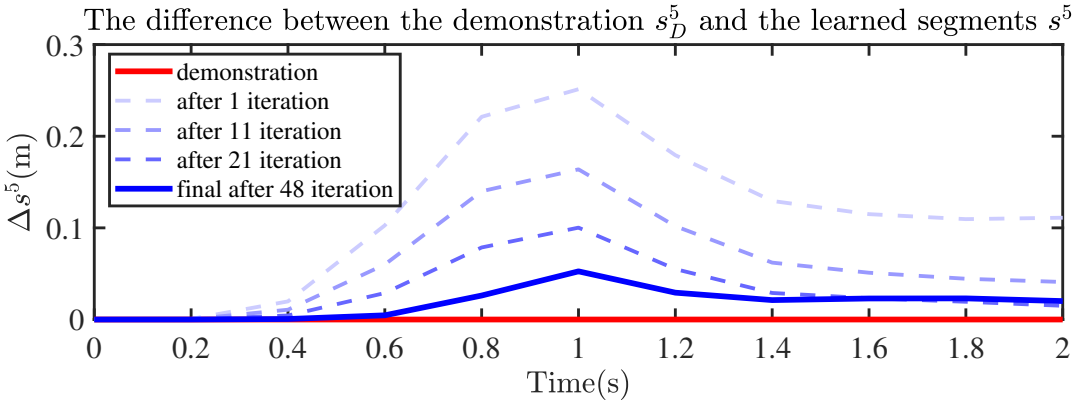
$\boldsymbol{f}_{\mathrm{qua,max}} = [f_{\mathrm{ax}}, f_{\mathrm{jx}}, f_{\mathrm{vy}}, f_{\mathrm{v1}}, f_{\mathrm{l1}}, f_{\mathrm{sl}}, f_{\mathrm{e2}}]^\top$. (In this chapter, the threshold λ in f_{e2} is set to 1.5.) Note that we introduce adjustments regarding feature x-Jerk f_{jx} for learning Vehicle A and Vehicle B. Specifically, when we incorporate all features in $\boldsymbol{f}_{\mathrm{qua,max}}$ in the learning process of Vehicle A, the learned weight for feature x-Jerk f_{jx} is exceptionally small and may even be negative, so we exclude x-Jerk f_{jx} while learning the driving style of Vehicle A. When learning for Vehicle B, the value of f_{jx} is extremely great. Therefore, to bring all feature values within a comparable range, we scale the value of f_{jx} by multiplying it by 0.05 in the learning process for Vehicle B.



(**a**) The difference between the demonstration and reproduced segment $\boldsymbol{s}^1$ of Vehicle A



(**b**) The difference between the demonstration and reproduced segment $\boldsymbol{s}^3$ of Vehicle A



(**c**) The difference between the demonstration and reproduced segment $\boldsymbol{s}^5$ of Vehicle A

Figure 6.4.: The differences/gaps between the demonstration segments and the reproduced segments of Vehicle A during the learning process

To display the process of 'learning from segments', we provide the following information about Vehicle A and Vehicle B, respectively. The information contains the following: 1) the development of learning error $\boldsymbol{\varepsilon_\theta}$ during the overall learning process; 2) the reproduced segments (e.g., $\boldsymbol{s}^1$, $\boldsymbol{s}^3$ and $\boldsymbol{s}^5$) during learning $\boldsymbol{s}^1$; 3) the differences between the demonstrations $\boldsymbol{s}_{\mathscr{D}}^1$, $\boldsymbol{s}_{\mathscr{D}}^3$ and $\boldsymbol{s}_{\mathscr{D}}^5$ and the reproduced segments $\boldsymbol{s}^1$, $\boldsymbol{s}^3$ and $\boldsymbol{s}^5$, respectively.

Learning Process for Vehicle A

We exhibit the learning process of Vehicle A in this subsection. While learning the driving style of Vehicle A, the learning error $\varepsilon_{\boldsymbol{\theta}}$ starts from 16.12, decreases during learning, and eventually converges to 0.13 by the 48th iteration. During the learning process, we reproduce all segments according to the feature weights in each learning iteration, as introduced in step 6 of Algorithm 3. We use segment $\boldsymbol{s}^1$ as an example to exhibit the process of reproducing a segment, as shown in Figure 6.3, where the red trajectory is the demonstration segment, and the dashed blue ones represent the reproduced segments. After the first iteration, the gap between the reproduced segment and the demonstration is relatively big. The reproduced segments incrementally approach the demonstration as learning iterates. The reproduced segment converges to the demonstration segment after 48 iterations. Furthermore, we illustrate how the gaps/differences between a demonstration segment and its corresponding reproduced segments evolve in Figure 6.4. In Figure 6.4a, the gap between the demonstration and the reproduced trajectories of the first segment $\boldsymbol{s}^1$ is significant after 1 iteration, but incrementally shrinks throughout the learning process, and eventually converges to 0. We observe a similar pattern for segment $\boldsymbol{s}^3$ (see Figure 6.4b) and segment $\boldsymbol{s}^5$ (see Figure 6.4c).

Learning Process for Vehicle B

Here, we introduce learning process of Vehicle B. During the process of learning the driving style of Vehicle B, learning error $\varepsilon_{\boldsymbol{\theta}}$ starts from 27.96 and eventually converges to 0.07. Vehicle B moves forward and does not show apparent movement in the lateral direction. Therefore, we illustrate only the longitudinal position of the reproduced trajectories for segment $\boldsymbol{s}^1$ in Figure 6.5. The longitudinal position of the reproduced trajectories incrementally converges to the longitudinal position of the demonstration segment. In Figure 6.6, we also show the difference/gap between the demonstration segments and the reproduced segments of Vehicle B during the learning process. Comparing Figure 6.6 to Figure 6.4, we find that the development of the gap for Vehicle B shows a similar pattern to that for Vehicle A. The only difference is the reproduced trajectories for segment $\boldsymbol{s}^5$ are close to the demonstration even at the beginning of the learning.

To summarize, during the learning process of Vehicle A or Vehicle B, the learning error can gradually converge to a small number; moreover, the reproduced segments can successfully converge to the corresponding demonstration segments.

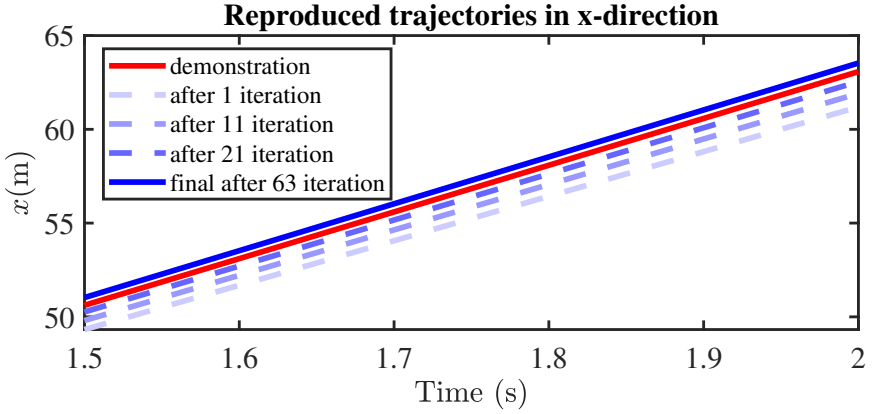
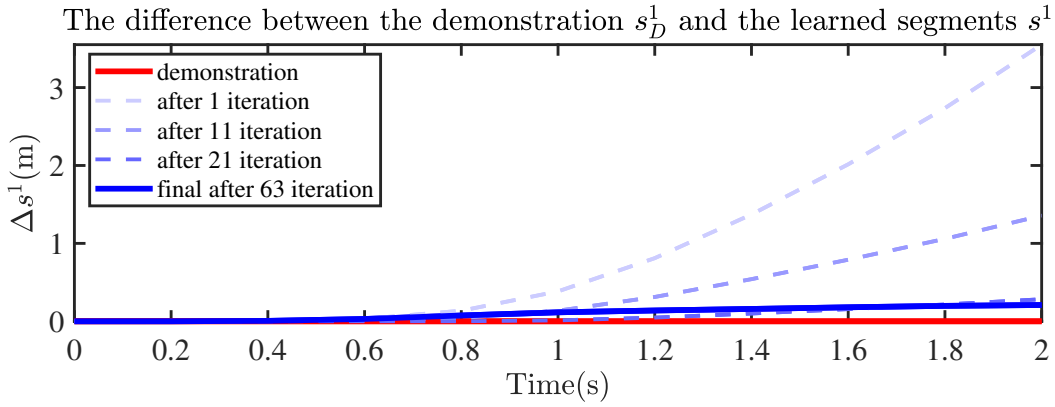
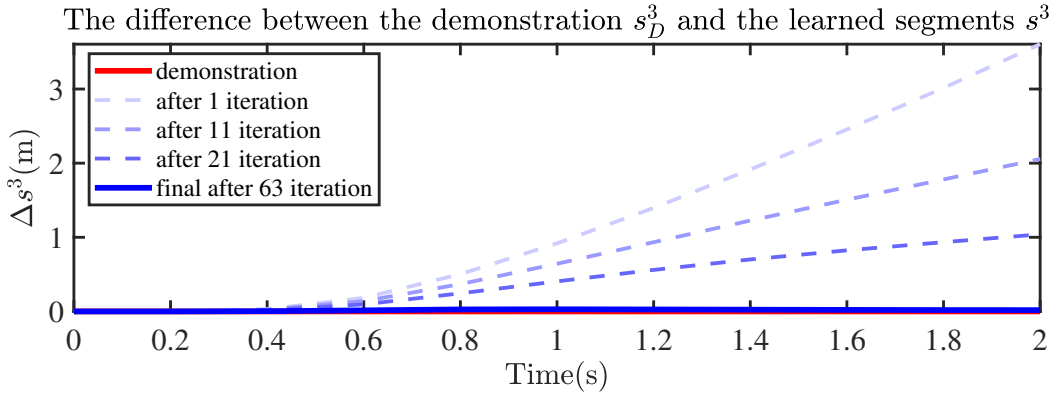


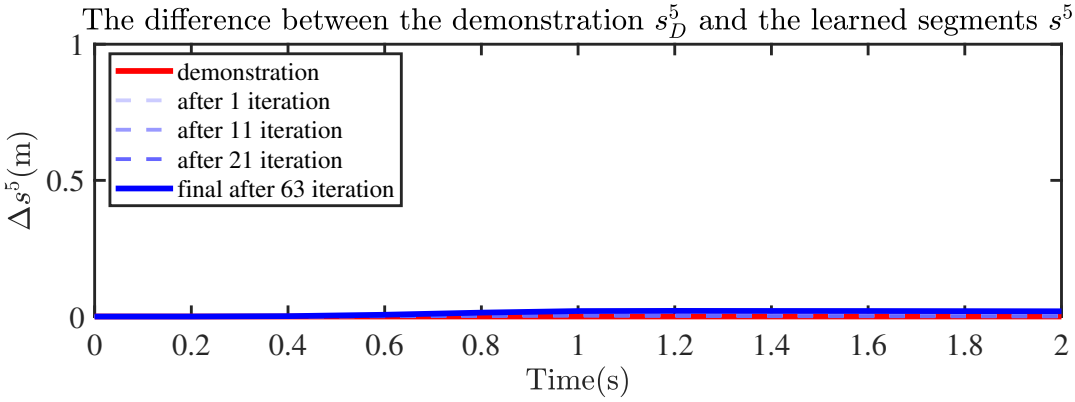
Figure 6.5.: Longitudinal positions of the reproduced trajectories of the first segment $\boldsymbol{s}^1$ of Vehicle B



(**a**) The difference between the demonstration and segment $\boldsymbol{s}^1$ of Vehicle B



(**b**) The difference between the demonstration and segment $\boldsymbol{s}^3$ of Vehicle B



(**c**) The difference between the demonstration and segment $\boldsymbol{s}^5$ of Vehicle B

Figure 6.6.: The differences/gaps between the demonstration segments and the reproduced segments of Vehicle B during the learning process

6.3.3. Evaluation of Different Feature Combinations

As we introduced in Section 6.2.2, we design two feature expressions for each of the three features 'Desired *x*-Speed', 'Desired Lane', and 'Safe Region'. That results in four different feature combinations, including $\boldsymbol{f}_{\text{qua,rec}}$, $\boldsymbol{f}_{\text{abs,rec}}$, $\boldsymbol{f}_{\text{qua,max}}$, $\boldsymbol{f}_{\text{abs,max}}$. We evaluate the performances of the

controllers and the prediction methods achieved when using the four different feature combinations in the learning process and find out the best combination. Specifically, we employ AccEff and SteEff to evaluate the control efforts and RMSE and ADE to evaluate the prediction precision. More details about the evaluation metrics were introduced in Section 2.5.

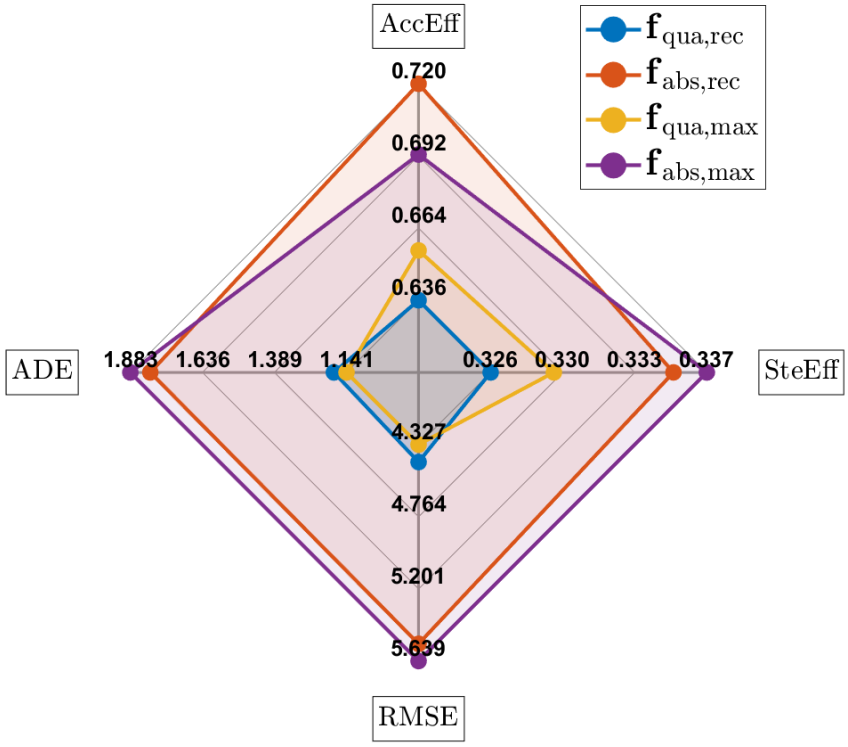


Figure 6.7.: Evaluation of different feature combinations

Table 6.4.: Evaluation values for different feature combinations

Feature	AccEff	SteEff	RMSE	ADE
$f_{qua,rec}$	0.636	0.326	4.433	1.185
$f_{abs,rec}$	0.720	0.335	5.534	1.815
$f_{qua,max}$	0.656	0.329	4.327	1.142
$f_{abs,max}$	0.693	0.337	5.639	1.883

We adopt the various feature combinations in the learning process of our interaction-aware prediction method for SMPC-controlled vehicle and access the values of AccEff, SteEff, RMSE and ADE for different feature combinations. AccEff, SteEff, RMSE and ADE are defined for evaluating the performances of an individual vehicle. However, there are two vehicles in the simulations, and we want to evaluate the performances of the two vehicles. To do this, after obtaining the two values of each evaluation metric for the two vehicles, we sum up each two values and treat the summation of the values as the value of the evaluation metric, as shown below:

$$M_i = M_i^{\mathrm{A}} + M_i^{\mathrm{B}}, i = 1,2,3,4. \tag{6.18}$$

where M_1, M_2, M_3 and M_4 represent the eventual values of the metrics AccEff, SteEff, RMSE and ADE, respectively. M_i^{A} is the value of an evaluation matric for Vehicle A, and M_i^{A} is for Vehicle B.

The strengths and weakness of these feature combinations are displayed using a spider chart shown in Figure 6.7. The specific evaluation values are summarized in Table 6.4. Feature combination $\boldsymbol{f}_{\mathrm{qua,rec}}$ performs best from the controller level, showing the smallest control efforts. The AccEFF value for $\boldsymbol{f}_{\mathrm{qua,rec}}$ is 0.636, which is moderately smaller than 0.656 for $\boldsymbol{f}_{\mathrm{qua,max}}$ but demonstrates substantial advantages compared with 0.693 for $\boldsymbol{f}_{\mathrm{abs,max}}$ and 0.720 for $\boldsymbol{f}_{\mathrm{abs,rec}}$. A similar trend is also found in the SteEff values, $\boldsymbol{f}_{\mathrm{qua,rec}}$ performing moderately better than $\boldsymbol{f}_{\mathrm{qua,max}}$ and significantly better than $\boldsymbol{f}_{\mathrm{abs,rec}}$ and $\boldsymbol{f}_{\mathrm{abs,max}}$. The only difference is that $\boldsymbol{f}_{\mathrm{abs,rec}}$ costs the most control effort for accelerating (0.720 for AccEff) while $\boldsymbol{f}_{\mathrm{abs,max}}$ requires the most control effort for steering (0.337 for SteEff). On the other hand, $\boldsymbol{f}_{\mathrm{qua,max}}$ outperforms other feature combinations in terms of prediction results, evidenced by having the smallest RMSE and ADE values (4.327 and 1.142, respectively). Nevertheless, $\boldsymbol{f}_{\mathrm{qua,rec}}$, with the RMSE value of 4.433 and ADE value of 1.185, exhibits only a marginal inferior performance compared to $\boldsymbol{f}_{\mathrm{qua,max}}$. Similar to the performance at the controller level, feature combinations $\boldsymbol{f}_{\mathrm{abs,rec}}$ and $\boldsymbol{f}_{\mathrm{abs,max}}$ demonstrate suboptimal perform when compared to others, which are indicated by the greater values of RMSE and ADE. In summary, $\boldsymbol{f}_{\mathrm{qua,rec}}$ demonstrates optimal performance at the controller level while $\boldsymbol{f}_{\mathrm{qua,max}}$ excels in terms of prediction methods; meanwhile, $\boldsymbol{f}_{\mathrm{qua,rec}}$ performs relatively good in prediction. In subsequent simulations of this chapter, we will use $\boldsymbol{f}_{\mathrm{qua,rec}}$ as the feature vector for learning.

6.3.4. With vs. Without Interaction-Aware features

In Section 6.2.2, we introduced the features that are employed in the learning process. Among the features, 'Safety Level' and 'Safety Region' are interaction-aware features, which take the influence of the neighboring vehicles on the vehicle into consideration. Interactions occur when the two vehicles are close to each other and try to adjust behaviors, e.g., decelerating and turning slightly to left/right, to avoid potential collisions. In this subsection, we validate the contribution of interaction-aware features to the performances of the controller and the prediction through an ablation study where we compare the results when employing $\boldsymbol{f}_{\mathrm{qua,rec}}$ for learning with the results obtained by using the reduced feature vector $\boldsymbol{f}_{\mathrm{EI}} = [f_{\mathrm{ax}}, f_{\mathrm{jx}}, f_{\mathrm{vy}}, f_{\mathrm{v1}}, f_{\mathrm{l1}}]^\top$ that excludes the interaction-aware features from $\boldsymbol{f}_{\mathrm{qua,rec}}$ for learning. The comparison is based on evaluation metrics AccEff, SteEff, RMSE and ADE, and is visualized using a spider chart shown in Figure 6.8. In Figure 6.8, 'With interaction' represents the results obtained by using feature vector $\boldsymbol{f}_{\mathrm{qua,rec}}$ that includes interaction-aware features, while 'Without interaction' denotes the results when excluding the interaction-aware features from $\boldsymbol{f}_{\mathrm{qua,rec}}$. Figure 6.8 shows that the values of AccEff, SteEff, RMSE, and ADE for the feature vector including interaction-aware features are smaller than that for the feature vector without interaction-aware features, respectively. Therefore, we can draw a conclusion that including interaction-aware features in the learning process leads to reduced control efforts and improved prediction accuracy.

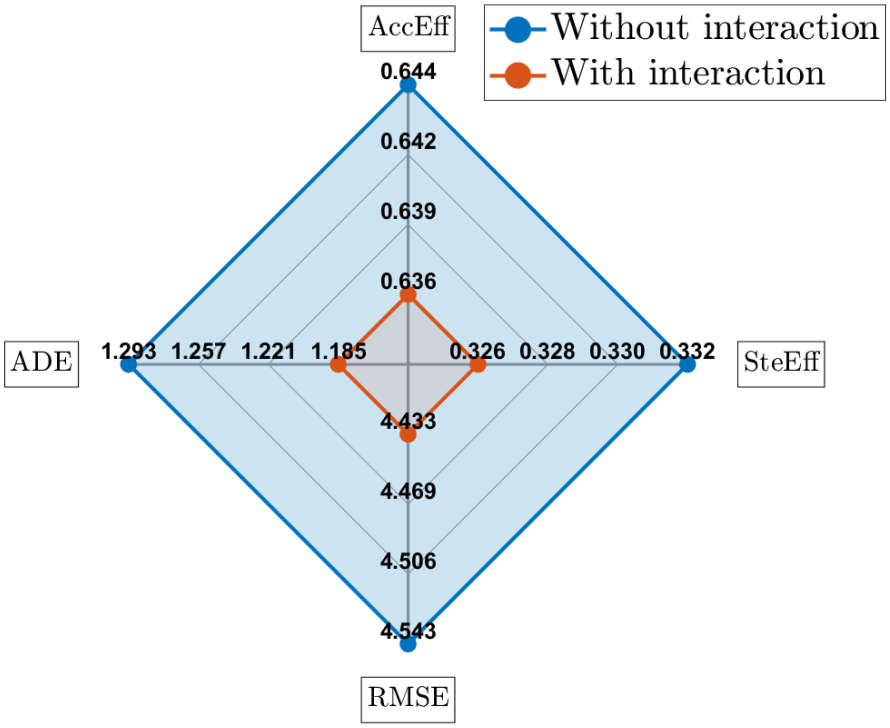


Figure 6.8.: Evaluation results obtained by using feature vectors with and without interaction-aware features for learning.

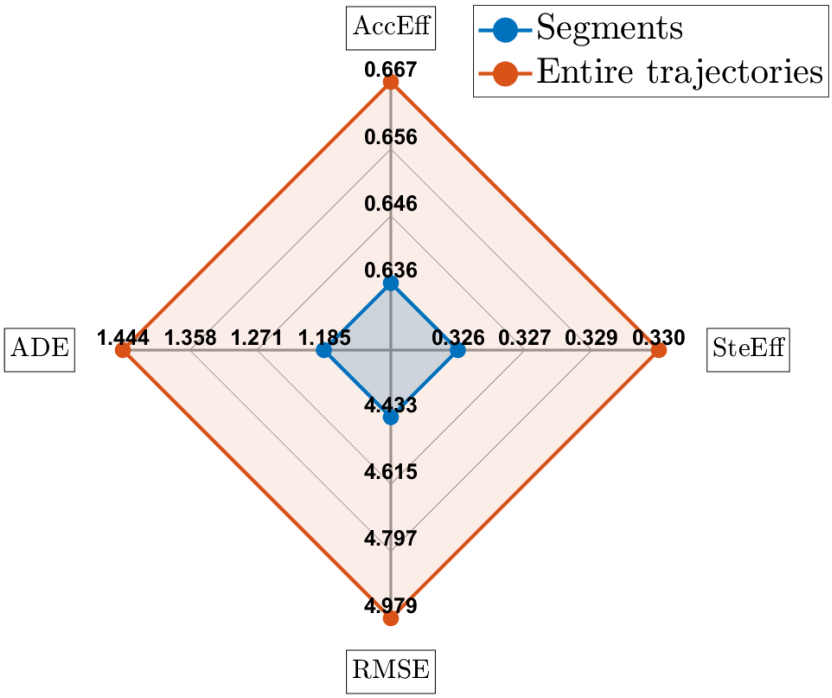


Figure 6.9.: Evaluation of different demonstration methods

6.3.5. Segments vs. Entire Trajectories as Demonstrations

In this subsection, we manifest the efficacy of our novel segment demonstrations. Specifically, we compare the controller performances and prediction results obtained by using the segments as demonstrations with those obtained by using the entire trajectories as demonstrations. The evaluation metrics consist of AccEff, SteEff, RMSE and ADE. We summarize the evaluation values in Table 6.5. To better visualize gaps between the values, we plot them in a spider chart, as shown in Figure 6.9. When we use segments as demonstrations, the resulting control efforts (0.636 for AccEff and 0.326 for SteEff) are slightly less than those for using entire trajectories as demonstrations (0.667 for AccEff and 0.330 for SteEff). Using segments as demonstrations in the learning process leads to increased prediction accuracy, compared to using the entire trajectories as demonstrations. When using segments as demonstrations, we obtain 4.433 for RMSE and 1.185 for ADE. Each of these values is smaller than the corresponding value obtained when using entire trajectories as demonstrations (4.979 for RMSE and 1.444 for ADE).

Table 6.5.: Performances of controllers and predictions when using different demonstration methods

Demonstrations	AccEff	SteEff	RMSE	ADE
Segments	0.636	0.326	4.433	1.185
Entire Trajectories	0.667	0.330	4.979	1.444

To conclude, our novel demonstration method, segmenting the entire trajectories into segments and employing these segments as demonstrations in the learning process, contributes to reducing control efforts and enhancing prediction accuracy (see Figure 6.9).

6.3.6. IRL-based Prediction vs. Simplified Prediction

In this subsection, we compare our interaction-aware IRL-based TV prediction method with the TV prediction method used in Chapter 4. In Chapter 4, an EV simply assumes that its TVs will stay in their current lanes and maintain their current velocities. The prediction method in Chapter 4 is referred to as 'simplified prediction' in this chapter. We evaluate the controller performance and the prediction accuracy of the controlled vehicles obtained when using these two different prediction methods. The evaluation metrics consist of AccEff, SteEff, RMSE and ADE. The evaluation values are summarized in Table 6.6. To intuitively demonstrate the strengths and weaknesses of these prediction methods, we also illustrate the evaluation results in a spider chart shown in Figure 6.10. Using the IRL-based prediction method to predict the behaviors of TVs requires more control efforts than using the simplified prediction. More specifically, the AccEff and SteEff for the IRL-based prediction method are 0.636 and 0.326, respectively. Meanwhile, only 0.458 for AccEff and 0.191 for SteEff are needed when applying the simplified prediction. However, the IRL-based prediction performs better in prediction accuracy compared to the simplified prediction. The RMSE and ADE for the simplified prediction are 6.517 and 5.035, respectively. Each of the values is larger than its corresponding value for the IRL-based prediction, specifically 4.433 for RMSE or 1.185 for ADE. To summarize, the simplified prediction demonstrates superior performance in terms of control efforts, while the IRL-based prediction excels in prediction accuracy.

Table 6.6.: Evaluation of the IRL-based and simplified prediction methods

Prediction method	AccEff	SteEff	RMSE	ADE
IRL-based	0.636	0.326	4.433	1.185
Simplified	0.458	0.191	6.517	5.035

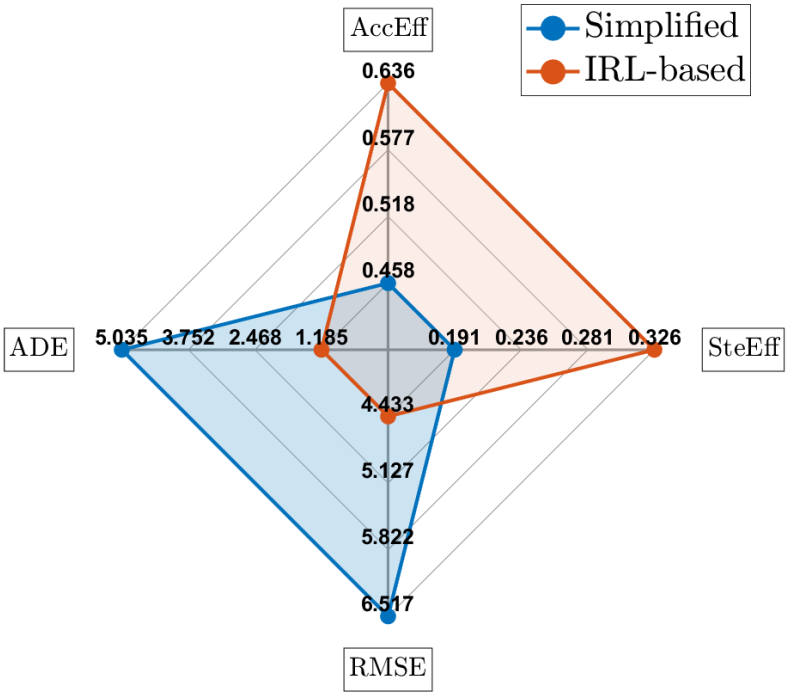


Figure 6.10.: Evaluation of different prediction methods

6.4. Conclusions and Future Work

In this chapter, we developed an interaction-aware prediction method for the SMPC-controlled vehicles in interactive systems (introduced in Chapte 4) to predict the behaviors of surrounding vehicles. To predict the trajectories of vehicles, we first learn the driving style of vehicles using IRL and then reproduce trajectories with similar driving styles to these vehicles, where the driving style is defined as a cost function of a series of weighted features. These reproduced trajectories are discretized and the discretized trajectories are treated as predictions that are incorporated into the collision avoidance constraints along the prediction horizon. The contributions of this chapter are presented as follows. 1) We propose novel features including interaction-aware features to capture the characteristics of SMPC-controlled vehicles in two-vehicle interactive systems. We eliminate the triggering condition in reactive features (see Chapter 5) to avoid the additional computation burden. 2) We introduced a novel concept for demonstrations, segments as demonstrations, that mitigates coarseness and detail deficiency caused by employing trajectories with a long temporal

scope as demonstrations and, meanwhile, enables a comprehensive understanding of the behaviors of the expert. 3) We employed the driving styles of vehicles to predict the behaviors of vehicles; more specifically, trajectories with similar driving styles to the surrounding vehicles are generated, discretized and treated as the predictions of the surrounding vehicles. 4) We incorporated the interaction-aware prediction method into the collision avoidance constraints of vehicles. The efficacy of our interaction-aware prediction method is examined in a two-vehicle interactive scenario in simulation studies.

The simulation results show that our interaction-aware prediction method can successfully take the interactions between vehicles into consideration when generating predictions. Additionally, compared to using the previous demonstration approach in Chapter 5, employing our novel segment demonstrations results in better control performance (lower AccEff and SteEff) and prediction precision (RMSE and ADE). Moreover, incorporating our interaction-aware prediction method into the controller design can achieve better prediction precision compared to incorporating the simplified prediction method in Chapter 4.

In the interaction considered in our feature design, we focus on understanding how the actions of an individual interactive vehicle influence the behaviors of the surrounding vehicles. However, it is also possible that multiple interactive vehicles are influencing the behaviors of the surrounding vehicles simultaneously. In our future work, we will include the influence of multiple interactive vehicles on the behaviors of a vehicle in the feature design. Moreover, we conducted simulations within the simple context, a three-lane highway scenario. In the future, we will research more complicated scenarios including parking, rural driving and urban driving.

7. Conclusions and Outlook

In this thesis, we proposed planning and control approaches for microscopic interactive vehicle systems based on Model Predictive Control (MPC) and especially Stochastic Model Predictive Control (SMPC). In this chapter, we conclude the proposed approaches by chapters, followed by an outlook on potential future research directions in this field.

7.1. Conclusions

After briefly presenting the preliminary technologies that form the foundation for our approaches in this thesis in Chapter 2, we present the approaches and contributions of this thesis in the subsequent four chapters.

In Chapter 3, we introduced a high-level maneuver planning approach for autonomous vehicles controlled by MPC. The maneuver planning approach is comprised of two methods: 1) an existing maneuver generation method [13] that generates multiple feasible maneuvers, and 2) our novel maneuver selection method that chooses the best maneuvers from these feasible ones, considering multiple factors including driving goals, safety, efficiency, and passenger comfort. The maneuver planning approach is integrated with low-level MPC-based trajectory control. Our maneuver selection method enhances the practical applicability of the method in [13] by eliminating the need to explore an excessive number of trajectories.

In Chapter 4, we 1) investigated the interactions among SMPC-controlled vehicles in a microscopic interactive multi-vehicle system designed for a highway environment and 2) provided insights into how to positively influence the interactions to improve the performances of both the individual vehicles and the interactive system. Previous research demonstrated the successful movement of an individual SMPC-controlled vehicle, realizing its driving goals, traffic rules, and collision avoidance requirements in an uncertain environment. However, there was no exploration into whether this success could be extended to an interactive system involving multiple SMPC-controlled vehicles. The journey of this exploration begins with our investigation presented in Chapter 4. To investigate these interactions, we first model the interactive multi-vehicle system employing a Distributed SMPC (DSMPC) framework. In this framework, each vehicle is controlled by SMPC, interacting with surrounding vehicles by observing their current states and predicting future behaviors to avoid potential collisions. In our simulations, we investigate how SMPC risk parameters influence both non-interactive and interactive vehicle-control systems on highways. Based on the results, we provide insights into how to set up risk parameters for SMPC-controlled vehicles in interactive systems.

In Chapter 5, we identified the driving styles including interaction characteristics of controlled vehicles using a Maximum Entropy Inverse Reinforcement Learning (ME-IRL)-based method. Autonomous vehicles' ability to precisely predict the future behaviors of surrounding vehicles is

critical in enhancing safety. Here, it is reasonable to predict the future behaviors of a vehicle based on its driving style due to the general consistency of the style over time. Identifying the driving styles of controlled vehicles serves as the initial step in further prediction that will be further introduced in the conclusion of Chapter 6. The driving style is represented as a cost function of a series of weighted features that capture the characteristics of the demonstration trajectory. The ME-IRL method strives to determine the optimal weights for these features. Our contributions include proposing novel reactive features to capture the interaction characteristics of the vehicles and designing a trigger condition to activate three of the novel features that are active only when the vehicle is close to the nearby vehicle. It is reasonable to use a triggering condition to activate some reactive features, given that the controlled vehicle only reacts to the other vehicle when they are close. However, the existence of the triggering condition will lead to additional computational burden to the later learning process.

Chapter 6 proposed a behavior prediction method for vehicles in microscopic interactive multi-vehicle systems to predict the future behaviors of the surrounding vehicles. This prediction is based on these vehicles' driving styles including interaction characteristics. In Chapter 6, we identify of the driving styles particaly employing the ME-IRL method introduced in Chapter 5. However, We design novel features, including interaction-aware features, to capture the characteristics of vehicles in two-vehicle interactive systems. Here, we eliminate the triggering condition in reactive features (see Chapter 5), which can avoid the additional computation burden on the learning process. Besides, we introduce a novel demonstration concept that accommodates additional details, facilitating a more comprehensive understanding of the experts' behaviors. To predict the behaviors of surrounding vehicles, we produce trajectories that mimic the driving styles of these vehicles. These trajectories are treated as predictions for the behaviors of the surrounding vehicles. Further, we integrate these predictions into SMPC-controlled vehicles within the microscopic interactive multi-vehicle system. Simulation studies exhibit the superior performances achieved by using our novel segment demonstration idea instead of the previous demonstration idea presented in Chapter 5 during the learning process. Compared to the basic prediction method in Chapter 4 where an SMPC-controlled vehicle simply assumes that its surrounding vehicles will move forward without changing their lanes and velocities, the prediction accuracy is improved using our interaction-aware prediction method in Chapter 6.

Overall, we proposed approaches to solve planning and control problems existing in the uncertain driving environment. The uncertainty can result from many factors such as unpredictable behaviors of other road users, sensor limitations, road conditions, map inconsistencies, localization accuracy, and unforeseen emergencies. We address uncertainties caused by dynamic interactions between vehicles and the unpredictable nature of these interactions.

7.2. Outlook

Although the approaches proposed in this thesis provide possible solutions to the challenges in planning and control of autonomous vehicles within the context of microscopic interactive traffic, numerous unresolved issues remain for future research. Below, we discuss several potential research directions.

Improve the maneuver planning method. Our maneuver planning method is executed at each time step, enabling the autonomous vehicle to rapidly adjust its behavior in risky situations. Nevertheless, the high frequency of updating also causes unnecessary computational burden when the

driving environment is mostly safe. Consequently, future efforts should strike a balance between ensuring rapid response when needed and mitigating unnecessary computational overhead. Moreover, our maneuver planning can be integrated with the MPC-based trajectory generation method that executes our high-level maneuver. However, it is still unverified whether our maneuver planning method can smoothly cooperate with other controllers. In the future, it is worth exploring generalizing our maneuver planning method to be able to cooperate with other lower-level controllers.

Explore the applications of our approaches in more complicated environments. We performed simulations with two vehicles in this thesis. In the future, it is worth researching complicated scenarios with multiple vehicles interacting with the surrounding vehicles. Moreover, we selected highway scenarios as the driving environment because the uncomplicated driving structure and limited driving behaviors make highways the most practical setting for the initial study of our approaches. Future work may also focus on exploring the applications of our approaches in diverse environments like urban areas, suburbs, rural regions, and mountainous terrain. One potential challenge to be addressed is each environment requires unique skills and considerations.

Investigate the systems with mixed control strategies. We have investigated a multi-vehicle interactive system with MPC or SMPC as controllers. However, a more realistic scenario where diverse vehicles with different control strategies including human-driven ones coexist is expected in the long term. MPC/SMPC-controlled vehicles' ability to seamlessly collaborate/interact with other diverse vehicles is essential to ensure safe and efficient traffic flow and consequently deserves further investigation.

Try real datasets. In this thesis, we validated the efficacy and applicability of our approaches based on computed-generated data using simulations in MATLAB and experiments in Simcenter Prescan. Computer-generated datasets may lack the authenticity of real-world data, potentially leading to algorithmic biases or oversimplified representations of complex driving scenarios. Therefore, it is important to explore the performances of our approaches also in real datasets, e.g., INTEREATION, that provide authentic representations of real-world driving scenarios.

Extend to macroscopic traffic model when having multiple vehicles. Our current work focuses on a microscopic traffic model, where the dynamics of each traffic participant are individually modeled, and the focus is on studying the traffic characteristics of individual vehicles and analyzing how they interact with each other. Our research goals in this thesis determined our choice of the microscopic traffic model. However, when the number of MPC/SMPC-controlled vehicles in the driving scenarios significantly increases in the future, it is also worth investigating how the SMPC-controlled vehicle influences the macroscopic traffic characteristics, e.g., the intensity, density, and mean speed, of the traffic flow.

Feasibility problems when studying interactions. This thesis focuses on the feasible solutions to MPC/SMPC problems, meaning that the MPC/SMPC vehicles can find control inputs that satisfy all the constraints. However, obtaining feasible solutions is not always possible in our scenario designs, especially when we design scenarios for studying interactions. To study the interactive behaviors of two vehicles, we often have to put the vehicles in potentially conflicting situations, e.g., two vehicles want to occupy a certain road region simultaneously; otherwise, there is no

interactive behavior between vehicles. However, conflicting situations have a higher chance of causing infeasibility problems compared to situations without potential conflicts. Therefore, to further study the interactions, one future research direction could be exploring how to alleviate infeasibility problems in conflicting situations.

Deep Learning for Trajectory Prediction

A.

An MPC-controlled vehicle is required to predict its own trajectories in a finite prediction horizon according to its model. Beyond this, the vehicle should also incorporate the prediction of the trajectory of its nearby vehicles, or target vehicles (TVs) into its decision-making. The conventional trajectory prediction methods, such as the constant-speed-based ones, are too trivial to accurately capture the potential collision risks. In this appendix chapter, we propose a Deep Learning (DL) based trajectory prediction method that uses a Recurrent Neural Network (RNN) for an EV controlled by MPC to predict the future trajectory of a TV based on its historical data. Simulation studies are conducted to showcase the prediction accuracy of the RNN model and the collision-free trajectories generated by the MPC. This content is based on [27].

A.1. Introduction

Model Predictive Control (MPC) has attracted increasing attention in autonomous driving due to its capability of incorporating traffic rules, the physical limitations of vehicles, and the collision avoidance requirements into driving control. MPC iteratively solves an optimization problem and gets a feasible trajectory that is subject to these constraints. An MPC-controlled ego vehicle (EV) is said to be able to interact with a target vehicle (TV) if it can predict the future behaviors of the TV and incorporate the predicted behaviors into its decision-making, such that the risk of potential collisions is avoided. Therefore, predicting the future behaviors of a TV is an important topic to realize risk-aware autonomous driving. Trajectory prediction of an autonomous vehicle is conventionally conducted by assuming a constant speed, i.e., the TV is moving while maintaining its current speed [20], [28], [32]. However, these assumptions ignore the influence of the real-time control inputs of the TV on its future trajectory, especially when it is required to perform a different driving task in a short future horizon. To solve this problem, a more realistic prediction method that does not only consider the current state of the TV but also its historical data should be proposed to achieve precise prediction.

Other than the constant-velocity-based trajectory prediction method, learning-based methods have been used to predict the trajectory of the target vehicles based on their historical trajectories. In [138], deep learning (DL) methods have been successfully used for predicting the behaviors of vehicles. Since the vehicle trajectories can be recognized as the sequences of vehicle positions, recurrent neural network (RNN) is most used due to their capability of handling data sequences. In [139], an RNN model with long-short-term memory units is used for trajectory prediction. In [140], the technology of meta-induction learning is used to incorporate the interaction in a multi-vehicle system. A survey of deep learning methods to solve the vehicle trajectory prediction problem can be referred to in [141]. In this chapter, we use deep learning to predict the TV's trajectory and encode it into the safety constraint of an MPC. As a result, the MPC incorporates the

interaction between the EV and the TV and thus produces risk-aware collision-free motion. To facilitate the interface between deep learning and MPC, calibration of the training data is performed. We also adjust the offset of the predicted TV trajectory to avoid the prediction errors caused by this offset. The rest of the chapter is organized as follows. Sec. A.2 introduces the MPC formulation for autonomous vehicles. Sec. A.3 presents the deep learning-based prediction method. Simulation studies that validate the efficacy of the proposed method are shown in Sec. A.4. We conclude our work and discuss the future work in Sec. A.5

A.2. MPC Incorporating the Predicted TV Trajectory

This section presents the MPC-based motion planning framework incorporating the predicted TV trajectory. A two-vehicle system that contains an EV and a TV is considered. The EV is described using the linearized and discretized kinematic bicycle model [20] introduced in (2.4). Additionally, the EV is controlled by MPC and solves the optimal control problem presented in (2.6) at any current time t.

In this chapter, we define $\xi_k \in \Xi_k^{\text{safe}}$ as an elliptical region around the TV. The center of the ellipse is the geometric center of the TV. The size of the ellipse is sufficiently large to cover the size of the TV. Let x_k^{TV} and y_k^{TV} denote the longitudinal and lateral positions of the predicted TV trajectory at step k. Then, the distance between the vehicles in the two directions at prediction step k are $\Delta x_k = x_k - x_k^{TV}$ and $\Delta y_k = y_k - y_k^{TV}$. Then, the safety set Ξ_k^{safe} is represented as and the safety constraint is

$$\Xi_k^{\text{safe}} = \left\{ (\Delta x_k, \Delta y_k) \,\middle|\, \frac{{\Delta x_k}^2}{a^2} + \frac{{\Delta y_k}^2}{b^2} \geq 1. \right\}. \tag{A.1}$$

We will introduce how to predict the TV trajectory $(x_k^{\text{TV}}, y_k^{\text{TV}})$, $k = 0, 1, \cdots, N$, in the next section.

A.3. Trajectory Prediction

In this section, we present how to predict the lane-changing trajectories of a vehicle using its historical trajectories. We first generate the lane-changing data of a vehicle using a polynomial interpolation method. Then, we use the generated data to train a deep neural network that is applied to predicting the future trajectory of a vehicle based on its historical trajectory.

A.3.1. Data Generation

Predicting future trajectories using historical trajectories renders a regression problem. The primary step is to create the data set used to train a certain prediction model. The data set contains a cluster of historical trajectories of a vehicle as data samples. The ground truth label of each data sample is its corresponding future trajectory. The trajectories are retrieved from a typical type of lane-changing path.

Lane-Changing Path Generation

Lane-changing requires that the vehicle smoothly switches from an original lane to a target lane while maintaining a constant longitudinal velocity v. Therefore, we use piece-wise polynomial

splines to represent the lane-changing path of a vehicle. A typical lane-changing path consists of the following three stages.

- **Preparation stage (I):** the vehicle prepares the lane-changing, moving along the original lane and maintaining a constant speed v. This stage lasts for 2 s.
- **Changing stage (II):** the vehicle changes the lane, starting from the original lane at the original speed v and ending at the target lane at a target speed v. This stage lasts for 4 s.
- **Finishing stage (III):** the vehicle completes the lane-changing, moving along the target lane at speed v. This stage lasts for 2 s.

In this sense, the trajectories of the vehicle at stages I and III are straight lines. In stage II, the trajectory of the vehicle, represented as a sequence of planar coordinates (x,y), is interpolated using a third-order polynomial function

$$y(x) = y_0 + 3(y_T - y_0)\left(\frac{x-x_0}{x_T-x_0}\right)^2 - 2(y_T-y_0)\left(\frac{x-x_0}{x_T-x_0}\right)^3 \tag{A.2}$$

where (x_0, y_0) and (x_T, y_T) are the coordinates of the starting point and the ending point of the vehicle in stage II. The longitudinal coordinate x is sampled at a constant sampling time $\Delta t = 0.1$ s. The generated trajectory is sufficiently smooth with terminal conditions $\dot{x}_0 = \dot{x}_T = v$ and $\dot{y}_0 = \dot{y}_T = 0$.

To ensure the diversity of the data set, we create various lane-changing paths using different velocities v. For both velocities, we take their values from 10 m/s to 40 m/s with a constant increment of 0.1 m/s. Therefore, we ultimately obtain 301 paths with different velocity profiles. Note that the size of the paths, namely the number of sampled coordinates in a path, are different due to different velocities v but the same sampling time Δt.

Segmentation

From the generated lane-changing paths of the vehicle, we create the training and test data. Each sample of the data set is the historical trajectory of the vehicle before a certain time instant, and its ground truth label is the future trajectory starting from this instant. We define that all historical and future trajectories have the same size $M = 30$. In this sense, for every generated lane-changing path, we create a data item by taking a segment that contains $2M$ successive sampled coordinates. The first half of the segment forms a sample of the data, and the other half serves as the ground truth label. The first coordinate of the future trajectory noted as (x_s, y_s), is referred to as the splitting point. For a path sized N, $N > 2M$, we obtain $N - 2M$ segments, i.e., $N - 2M$ labeled data samples.

Calibration

Different data samples have different splitting points which bring different offsets to the longitudinal coordinate of the samples. To eliminate the influence of these offsets on the model training, we subtract x_s from all the longitudinal coordinates of the samples, which is referred to as calibration. After calibration, the splitting points of all data samples have zero longitudinal coordinates $x_s = 0$.

Data Splitting

Having performed segmentation and calibration, we obtain $L = 6622$ data samples. Each sample contains a historical trajectory, with the corresponding future trajectory being its ground truth label. We split the data into a training set and a test set with a ratio of $6:4$. We also randomly shuffle the data to avoid the influence of the continuity of the vehicle motion.

A.3.2. Prediction Model

In this chapter, we use a recurrent neural network (RNN) model to predict future trajectories. The RNN is composed of a sequence input layer, an encoder layer, a latent feature layer, a decoder layer, and an output layer, and is implemented using the MATLAB ®Deep Learning Toolbox. The details of the network structure are introduced as follows.

Sequence Input Layer

This is a typical input layer used to feed the historical trajectories into the RNN. In MATLAB ®, it is created using function *sequenceInputLayer*. The input size is 2, namely the number of planar dimensions. This layer is attached to a normalized layer in the output end.

Encoder Layer

This layer is a gated recurrent unit (GRU) layer used to encode the dependencies between the successive coordinates of the historical trajectories. In MATLAB ®, it is created using function *gruLayer* with layer size 64. This layer is also attached to a normalized layer in the output end.

Latent Feature Layer

This layer is a fully connected layer used to automatically extract the features from the encoded sequential data. It is created using function *fullyConnectedLayer* with layer size 64. Its output passes through a layer of Linear rectification functions (ReLU).

Decoder Layer

Similar to the encoder layer, this layer is also a GRU layer. It is used to decode the sequential features to sequential data that are used to generate the prediction. Its size is set as 128.

Output Layer

This layer is used to map the decoded sequential data to the predicted trajectories. It is constructed by a fully connected layer sized 2 and a regression layer.

A.4. Simulation Studies

In this section, we use a two-lane straight highway scenario to evaluate the MPC incorporating the predicted TV trajectory using deep learning. The scenario considers three parallel horizontal lanes, where the EV and the TV start at the middle and the bottom lanes, respectively. The EV is

required to drive in the middle lane at a constant speed, and the TV needs to change to the middle lane, thus leading to possible collisions. We first evaluate the prediction precision of the trained RNN model. Then, we validate the efficacy of the MPC with predicted TV trajectory.

A.4.1. Evaluation of the Prediction Model

In this subsection, we evaluate the prediction accuracy of the RNN model. The loss function is based on mean squared errors (MSE) between the outputs of the RNN and the ground truth labels of the samples. Specifically, subtraction is performed between them in an element-wise manner. Then, the squared element-wise errors are summed up before being divided by the total number of elements. The training data set that contains 3973 samples is used to train the model. The optimization of the MSE loss is solved using the Adam optimizer [142]. The training is performed on a Thinkpad laptop with Intel(R) Core(TM) i7-10750H CPU at 2.60 GHz. The entire training process takes 30 epochs and 900 iterations with a learning rate of 0.01. We do not need a large number of epochs since predicting trajectories is not a heavy job.

The rooted MSEs (RMSE) as the iteration number increases are illustrated in Fig. A.1. It is noticed that the RMSE decreases as the training proceeds with an ultimate score of 15.92. The RMSE score becomes stable at around iteration 300, which indicates the quick learning speed of the RNN model. This also reflects that trajectory prediction is an easy job for an RNN.

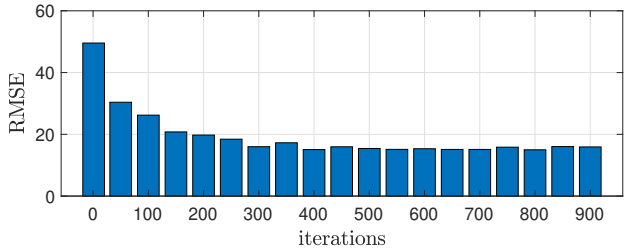


Figure A.1.: The training performance of the RNN model: the convergence of RMSE as the iteration increases.

Then, we test the prediction accuracy of the trained RNN model using the test data set that contains 2649 samples. We calculate the RMSE score for each predicted sample. The RMSE scores of all test samples are shown in the histogram chart in Fig. A.2. It can be seen that the RMSE scores of the prediction vary from 0 to 20. This range is very close to the ultimate training score of the model, 15.92. The overall RMSE of the test is 10.91. This indicates the accuracy of the trained RNN model for trajectory prediction.

A.4.2. Incorporating the Prediction Model to MPC

We use the predicted TV trajectory provided by the trained RNN model to generate the TV future trajectory $(x_k^{\mathrm{TV}}, y_k^{\mathrm{TV}})$, $k = 0, 1, \cdots, N$, and encode it to the safe set (A.1). Note that the starting point of the predicted TV trajectory may not be aligned with the current position of the TV due to the prediction error of the RNN model. This misalignment, however, can be eliminated by adding an offset to the predicted trajectory such that its starting point matches the current position of the TV.

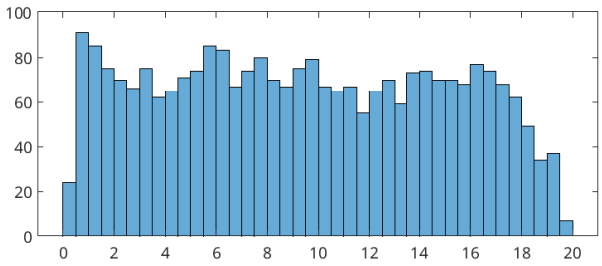


Figure A.2.: The training performance of the RNN model: the convergence of RMSE as the iteration increases.

The initial state of the EV is $\boldsymbol{\xi}_0^{\mathrm{EV}} = [28, 7.875, 0, 20]^\intercal$. The EV is intended to reach a reference speed 20 m/s. Then, we design an MPC for the EV as (2.6) incorporating the interaction safety constraint (A.1). The parameters of the MPC are set as $N = 10$, $T = 0.2\,\mathrm{s}$, $y \in [l^{\mathrm{veh}}, 3w^{\mathrm{lane}} - l^{\mathrm{veh}}]$, $\psi \in [-1.2, 1.2]\,\mathrm{rad}$, $v \in [0, 70]\,\mathrm{m/s}$, $a \in [-9, 6]\,\mathrm{m/s^2}$ and $\delta \in [-0.52, 0.52]\,\mathrm{rad}$. The parameters of the elliptical safety set Ξ_k^{safe} are determined as $a = 7\,\mathrm{m}$ and $b = 2.2\,\mathrm{m}$. The weighting matrices of the cost function are $Q = \mathrm{diag}(0, 0.1, 0.001, 1)$, $R = \mathrm{diag}(3, 0.5)$ and $S = \mathrm{diag}(0, 0.1, 0.001, 1)$. The initial state of the TV is $\boldsymbol{\xi}_0^{\mathrm{TV}} = [36, 2.625, 0, 18]^\intercal$. The trajectory of the TV is generated using a similar MPC to the EV but with a reference speed 20 m/s, the center lane as the target lane, and without incorporating the safety constraint (A.1).

The generated trajectories of the EV and the TV within 11 simulation steps are shown in Fig. A.3, in blue and red, respectively. Their positions in the simulation steps 1, 4, 7, 11 are displayed as colored squares, shadow to dark in the order of time. The predicted TV trajectories at all simulation steps are also shown in the figure as dotted gray lines. From Fig. A.3, we can see that the EV decelerates to avoid potential collisions with the TV as the TV starts to change its lane and then block the way of the EV in the center lane. When the TV finishes the lane-changing and accelerates for its own control target, the EV also starts to accelerate to reach its desired speed since it realizes that the risk of potential collisions is mitigated. This indicates that the EV is able to recognize the TV's lane-changing intention and be aware of the risk of collisions before the TV reaches the center lane. Therefore, the capability of the proposed MPC to mitigate the upcoming risks is addressed. Besides, from the figure, we can also see that the predicted TV trajectories become more and more consistent with the ground truth TV trajectory. This is because more data in the historical trajectory leads to higher prediction precision.

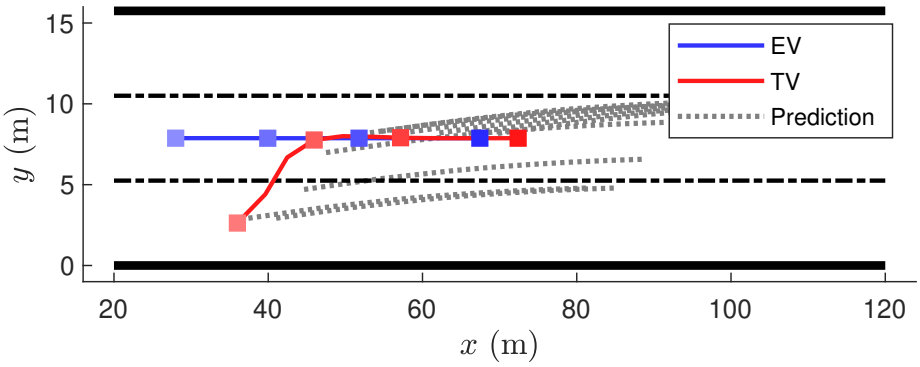


Figure A.3.: A two-vehicle lane-changing scenario.

A.5. Conclusion

In this chapter, we use deep learning to predict the trajectory of the TV and incorporate it into the safety constraint of the MPC of the EV. As a result, the EV is able to incorporate the risk of collisions with the TV into its motion planning such that it can make reasonable and risk-aware decisions in autonomous driving. Technical points such as data calibration and offset compensation are used to ensure that the MPC incorporates the more realistic predicted trajectory. An interesting case that is not investigated in this chapter is that both vehicles can interact with each other, which renders a two-player game. Our method in this chapter can be extended to this case in future work.

Notation

Acronyms and Abbreviations

MPC	model predictive control
RMPC	robust model predictive control
SMPC	stochastic model predictive control
DMPC	distributed model predictive control
DSMPC	distributed stochastic model predictive control
MDP	markov decision process
RL	reinforcement learning
IRL	inverse reinforcement learning
ME-IRL	maximum entropy inverse reinforcement learning
AV	autonomous vehicle
EV	ego vehicle
TV	target vehicle
AccEff	acceleration effort
SteEff	steering effort
RMSE	root mean squared error
ADE	average displacement error
TTC	time to collision
TIV	the inter-vehicular time
DE	decelerating
CS	maintaining current speed
AC	accelerating
LCL	changing to the left lane
LK	keep moving in the current lane

LCR	changing to the right lane
DD	distance deviation
SD	state deviation
DL	deep learning
RNN	recurrent neural network
GAIL	generative adversarial imitation learning

Parameters

ξ_k	state vector at time k
x_k	longitudinal position at time k
y_k	lateral position at time k
$v_{x,k}$	longitudinal velocity at time k
$v_{y,k}$	lateral velocity at time k
u_k	control input vector at time k
$v_{x,k}$	longitudinal acceleration at time k
$v_{y,k}$	lateral acceleration at time k
$\boldsymbol{\xi}$	states along the prediction horizon
$\boldsymbol{u}$	control inputs along the prediction horizon
ξ	state vector in the continuous time model
u	control input vector in the continuous time model
x	longitudinal position in the continuous time model
y	lateral position in the continuous time model
ψ	inertial heading in the continuous time model
v	vehicle velocity in the continuous time model
a	acceleration in the continuous time model
δ	front steering angle in the continuous time model
T	sampling time

A	system matrix
B	system matrix
l_f	front axle of the vehicle
l_r	rear axle of the vehicle
β	angle of the vehicle with respect to the longitudinal axis of the road
N	prediction horizon
Ξ	set of states
$\mathscr{U}$	set of control inpits
Ξ_{safe}	safety set
p	risk parameter
$\mathbf{Pr}(*)$	probability of occurrence of event $*$
f_{ax}	feature of x-acceleration
f_{ay}	feature of y-acceleration
f_{v}	feature of desired velocity
f_{l}	feature of desired lane
f_{il}	feature of initial lane
f_{el}	feature of end lane
$\boldsymbol{r}$	a trajectory
$\boldsymbol{f}$	feature vector
r^x	position in the longitudinal direction
r^y	position in the lateral direction
v^x_{des}	desired velocity in the longitudinal direction
l_{des}	desired lane
l_{initial}	initial lane of the EV
t_{turn}	time when the EV remains in the initial lane l_{initial}
l_{target}	target lane of the EV
t_{end}	ending time
$\boldsymbol{\theta}$	weight vector

$\boldsymbol{\theta}^*$	optimal weight vector
θ_m	weight of a specific feature m
$\boldsymbol{r}_{\mathscr{D}}$	demonstration trajectory
$\boldsymbol{r}_{\boldsymbol{\theta}}$	a trajectory can be reproduced by a learned weight vector $\boldsymbol{\theta}$
$\boldsymbol{r}_{\boldsymbol{\theta}^*}$	optimal reproduced trajectory
$\nabla_{\boldsymbol{\theta}}$	gradient for the update of $\boldsymbol{\theta}$
α	learning rate
$\varepsilon^i_{\boldsymbol{\theta}}$	learning error at iteration i with weight $\boldsymbol{\theta}$
$\bar{\varepsilon}$	predefined threshold for learning
$\boldsymbol{c}_j$	control input at time t_j
$\bar{\boldsymbol{c}}$	set of control points
N_s	number of spline segments
$\boldsymbol{s}_j$	piecewise quintic spline for time interval $[t_j, t_{j+1}]$,
r^x_j	position in the longitudinal direction at interval $[t_j, t_{j+1}]$
v^x_j	velocity in the longitudinal direction at interval $[t_j, t_{j+1}]$
a^x_j	acceleration in the longitudinal direction at interval $[t_j, t_{j+1}]$
r^y_j	position in the lateral direction at interval $[t_j, t_{j+1}]$
v^y_j	velocity in the lateral direction at interval $[t_j, t_{j+1}]$
a^y_j	acceleration in the lateral direction at interval $[t_j, t_{j+1}]$
Δa	nominal ranges of the acceleration
$\Delta\delta$	nominal ranges of the steering angle
$\boldsymbol{r}_{\mathrm{p}}$	predicted trajectory
$\boldsymbol{r}_{\mathrm{r}}$	real trajectory
$r_{\mathrm{p}k}$	position on the predicted trajectory $\boldsymbol{r}_{\mathrm{p}}$ at time step k
$r_{\mathrm{r}k}$	position on the real trajectory $\boldsymbol{r}_{\mathrm{r}}$ at time step k
K	number of time steps along the entire trajectory
$\Delta x_{\mathrm{EV,TV}}$	relative distance between the EV and the TV in the longitudinal direction
x_{EV}	longitudinal position of the EV

x_{TV}	longitudinal position of the TV
$\Delta v_{\mathrm{EV,TV}}$	relative velocity between the EV and the TV
v_{EV}	longitudinal velocity of the EV
v_{TV}	longitudinal velocity of the TV
v^*	velocity of the vehicle behind
w_{veh}	width of the vehicle
w_{lane}	width of each lane
s_a	major axis of the ellipse
s_b	minor axis of the ellipse
Q	weighting matrix
R	weighting matrix
Q_N	weighting matrix
q_1	one element in weighting matrix Q
q_2	one element in weighting matrix Q
r_1	one element in weighting matrix R
r_2	one element in weighting matrix R
r_3	one element in weighting matrix R
r_4	one element in weighting matrix R
$q_{N,1}$	one element in weighting matrix Q_N
$q_{N,2}$	one element in weighting matrix Q_N
$q_{N,3}$	one element in weighting matrix Q_N
$q_{N,4}$	one element in weighting matrix Q_N
ξ_k^{ref}	reference states at time k
$\mathbb{G}$	graph
$\mathbb{V}$	set of nodes
$\mathbb{E}$	set of edges
$\mathscr{A}$	adjacency matrix of a graph
$\mathbb{N}_i$	set consisting of the neighbors of vehicle i

$\mathbb{O}_i$	set of all vehicles that identify vehicle i as a neighbor
$\boldsymbol{\xi}_i^{\mathrm{p}}$	predicted state of vehicle i along the prediction horizon
$\boldsymbol{u}_i^{\mathrm{p}}$	predicted control input of vehicle i
$\boldsymbol{\xi}_i^{\mathrm{a}}$	assumed states of vehicle i along the prediction horizon
$\boldsymbol{u}_i^{\mathrm{a}}$	assumed control inputs of vehicle i along the prediction horizon
$\boldsymbol{\xi}_i^{*}$	optimal states of vehicle i along the prediction horizon
$\boldsymbol{u}_i^{*}$	optimal control inputs of vehicle i along the prediction horizon
$\boldsymbol{\xi}_{\mathrm{TV}}^{\mathrm{a}}$	assumed states of TV along the prediction horizon
$\boldsymbol{u}_{\mathrm{TV}}^{\mathrm{a}}$	assumed control inputs of TV along the prediction horizon
p_i	risk parameter of vehicle i
$\xi_{\check{i},k}^{\mathrm{a}}$	assumed trajectory of TV $\check{i}$ at prediction step k
$u_{\check{i},k}^{\mathrm{a}}$	assumed control inputs of TV $\check{i}$ at prediction step k
$\omega_{\mathrm{TV},k}^{\mathrm{a}}$	uncertainty in the prediction of TV behaviors at prediction step k
$\omega_{\check{i},k}^{\mathrm{a}}$	uncertainty in the prediction of TV $\check{i}$ behaviors at prediction step k
$z_{\check{i},k}^{\mathrm{a}}$	nominal states of TV $\check{i}$
$e_{\check{i},k}^{\mathrm{a}}$	errors in predicting the behaviors of TV $\check{i}$ at prediction step k
$\sum_{\check{i},k}$	covariance matrix for prediction error $e_{\check{i},k}^{\mathrm{a}}$
$\sum_{\omega_i^{\mathrm{a}}}$	covariance matrix for the uncertainty in the prediction
$\mathbf{err}_{\xi}$	deviation between the real states and the reference states during all iterations
f_{tiv}	feature of the intervehicular time
f_{sd}	feature of start distance
f_{ed}	feature of end distance
f_{id}	Integral distance
t_{trg}	time when the trigger condition is triggered
T_{trg}	time duration for the trigger condition
v_{lane}^{x}	limit velocity of the target lane
l_a	a parameter for the trigger condition

l_b	a parameter for the trigger condition
s_e	square of the elliptical distance between the EV and the TV
$\boldsymbol{f}_{\mathrm{qua,rec}}$	a feature combination
$\boldsymbol{f}_{\mathrm{abs,rec}}$	a feature combination
$\boldsymbol{f}_{\mathrm{qua,max}}$	a feature combination
$\boldsymbol{f}_{\mathrm{abs,max}}$	a feature combination
f_{jx}	feature of x-Jerk
f_{vy}	feature of y-Speed
f_{v1}	feature of Desired x-Speed
f_{v2}	feature of desired x-Speed
f_{l1}	feature of desired Lane)
f_{l2}	feature of desired Lane)
f_{sl}	feature of safety level
f_{e1}	feature of safe region
f_{e2}	feature of safe region
v_{des}	limited velocity of the end lane
l_{des}	desired lane
$\boldsymbol{r}_{\mathrm{IV}}$	future trajectory of the interactive vehicle
$\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pre-def}}$	predefined trajectory of the TV
$\boldsymbol{r}_{\mathscr{D}}^{i}$	an original demonstrations ($i = 1,2,\ldots,n_d$)
n_d	number of the original demonstration trajectories
$\tilde{\boldsymbol{r}}_{\mathscr{D}}$	average trajectory of all original demonstration trajectories
$\boldsymbol{s}_{\mathscr{D}}^{i}$	segment demonstration ($i = 1,\cdots,n_s$)
n_s	number of the segments as demonstrations
$\boldsymbol{f}_{\mathscr{D}}$	feature values of the segment demonstrations
$\boldsymbol{r}_{\mathrm{IV}}^{-}$	trajectory of the EV obtained in the previous SMPC iteration
$\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{discrete}}$	a discrete trajectory of TV
$\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$	predicted trajectory of the TV

$\tilde{\boldsymbol{r}}_{\mathrm{TV}}$	a deterministic component of trajectory $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$
$\boldsymbol{e}_{\mathrm{TV}}$	an error component of trajectory $\boldsymbol{r}_{\mathrm{TV}}^{\mathrm{pred}}$
$\boldsymbol{\Sigma}^{e}$	covariance matrix of $\boldsymbol{e}_{\mathrm{TV}}$
$d_{\breve{\imath},k}$	variable in safety constraint of EV i to avoid vehicle $\breve{\imath}$ at prediction step k
$d^{\mathrm{e}}_{\breve{\imath},t}$	elliptical distance between EV i and its TV $\breve{\imath}$ at any iteration t/T (current time t)

Sets and Operators

$\mathbb{R}$	set of all real numbers
$\mathbb{N}$	set of all natural numbers
$\mathbb{R}^{+}$	set of all positive real numbers
$\mathbb{N}^{+}$	set of all positive natural numbers
$\mathbb{R}^{m}$	set of all column vectors with m elements
$\mathbb{R}^{m\times n}$	set of all $m \times n$ matrices whose elements are real values
$\mathbb{S}$	set of symmetric and positive definite matrices
$\mathbb{S}^{m}$	set of symmetric matrices of order m
$\dot{a}$	derivative of a
$\ddot{a}$	second-order derivative of a
$\lvert a\rvert$	absolute value of a
$\lVert a\rVert$	2-norm of vector a
$\lVert a\rVert_2$	2-norm of vector a
$[0;1)$	right open interval
$[0;1]$	closed interval
$\mathrm{diag}(\cdots)$	diagonal matrix

Functions

$\mathscr{F}^{\mathrm{c}}$	nonlinear continuous kinematic bicycle model
$\mathscr{F}$	system dynamic model to generate predictions in MPC
$\mathscr{F}^{\mathrm{p}}$	system dynamic of an EV
$\mathscr{F}^{\mathrm{a}}$	system dynamic of a TV from the perspective of an EV
V	cost function in MPC
L	cost function that represents the driving style
erf^{-1}	inverse error function

List of Figures

List of Tables

Bibliography

[1] K. Yang, X. Tang, J. Li, *et al.*, "Uncertainties in onboard algorithms for autonomous vehicles: Challenges, mitigation, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 8963–8987, 2023.

[2] E. Marti, M. A. De Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[3] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, 2023.

[4] L. Chen, S. Teng, B. Li, *et al.*, "Milestones in autonomous driving and intelligent vehicles—part ii: Perception and planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 10, pp. 6401–6415, 2023.

[5] S. Teng, X. Hu, P. Deng, *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Transactions on Intelligent Vehicles*, 2023.

[6] L. Chen, Y. Li, C. Huang, *et al.*, "Milestones in autonomous driving and intelligent vehicles—part i: Control, computing system design, communication, hd map, testing, and human behaviors," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 9, pp. 5831–5847, 2023.

[7] L. Chen, Y. Li, C. Huang, *et al.*, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1046–1056, 2022.

[8] C. Menéndez-Romero, F. Winkler, C. Dornhege, and W. Burgard, "Maneuver planning for highly automated vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1458–1464.

[9] C. Chen, A. Gaschler, M. Rickert, and A. Knoll, "Task planning for highly automated driving," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 940–945.

[10] D. Ferguson, T. Howard, and M. Likhachev, "Motion planning in urban environments," *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 1149–1154, Jan. 2008.

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

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

[13] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouvelière, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 589–606, 2010.

[14] D. Wang and F. Qi, "Trajectory planning for a four-wheel-steering vehicle," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, IEEE, vol. 4, 2001, pp. 3320–3325.

[15] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[16] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *12th International Symposium on Advanced Vehicle Control*, 2014, pp. 712–719.

[17] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller, "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 811–817.

[18] 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.

[19] 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 *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 1317–1324.

[20] 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.

[21] T. A. N. Heirung, J. A. Paulson, J. O'Leary, and A. Mesbah, "Stochastic model predictive control—how does it work?" *Computers & Chemical Engineering*, vol. 114, pp. 158–170, 2018.

[22] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng, "Explicit use of probabilistic distributions in linear predictive control," *Automatica*, vol. 46, no. 10, pp. 1719–1724, 2010.

[23] J. Maroto, E. Delso, J. Felez, and J. M. Cabanellas, "Real-time traffic simulation with a microscopic model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 513–527, 2006.

[24] J. L. Zambrano-Martinez, C. T. Calafate, D. Soler, J.-C. Cano, and P. Manzoni, "Modeling and characterization of traffic flows in urban environments," *Sensors*, vol. 18, no. 7, p. 2020, 2018.

[25] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[26] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[27] N. Dang, Z. Zhang, J. Liu, M. Leibold, and M. Buss, "Incorporating target vehicle trajectories predicted by deep learning into model predictive controlled vehicles," *arXiv preprint arXiv:2310.02843*, 2023.

[28] N. Dang, T. Brüdigam, M. Leibold, and M. Buss, "Combining event-based maneuver selection and mpc based trajectory generation in autonomous driving," *Electronics*, vol. 11, no. 10, p. 1518, 2022.

[29] L. Dai, Y. Xia, Y. Gao, B. Kouvaritakis, and M. Cannon, "Cooperative distributed stochastic mpc for systems with state estimation and coupled probabilistic constraints," *Automatica*, vol. 61, pp. 89–96, 2015.

[30] L. Dai, Y. Xia, Y. Gao, and M. Cannon, "Distributed stochastic mpc of linear systems with additive uncertainty and coupled probabilistic constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3474–3481, 2016.

[31] G. Zhao and S. Yang, "Distributed stochastic mpc for linear systems with probabilistic constraints and quantisation," *IET Control Theory & Applications*, vol. 14, no. 3, pp. 396–404, 2020.

[32] 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.

[33] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 2641–2646.

[34] N. Dang, T. Shi, Z. Zhang, W. Jin, M. Leibold, and M. Buss, "Identifying reaction-aware driving styles of stochastic model predictive controlled vehicles by inverse reinforcement learning," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2023, pp. 2887–2892.

[35] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Dynamic systems and control conference*, vol. 44175, 2010, pp. 265–272.

[36] J. Levinson, J. Askeland, J. Becker, *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE intelligent vehicles symposium (IV)*, IEEE, 2011, pp. 163–168.

[37] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *16th International IEEE conference on intelligent transportation systems (ITSC 2013)*, IEEE, 2013, pp. 2335–2340.

[38] 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.

[39] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving: Extended version," *arXiv preprint arXiv:2009.09381*, 2020.

[40] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[41] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Aaai*, Chicago, IL, USA, vol. 8, 2008, pp. 1433–1438.

[42] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ser. ICML'12, Edinburgh, Scotland: Omnipress, 2012, pp. 475–482, ISBN: 9781450312851.

[43] H. Kretzschmar, M. Kuderer, and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents," in *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2014, pp. 4015–4020.

[44] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5355–5362, 2020.

[45] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of operations research*, vol. 153, pp. 235–256, 2007.

[46] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation.," in *Robotics: science and systems*, 2012.

[47] C. Sprunk, B. Lau, P. Pfaffz, and W. Burgard, "Online generation of kinodynamic trajectories for non-circular omnidirectional robots," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 72–77.

[48] A. Simon and J. C. Becker, "Vehicle guidance for an autonomous vehicle," in *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No. 99TH8383)*, IEEE, 1999, pp. 429–434.

[49] M. Bartyś and B. Hryniewicki, "The trade-off between the controller effort and control quality on example of an electro-pneumatic final control element," in *Actuators*, MDPI, vol. 8, 2019, p. 23.

[50] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)," *Geoscientific model development discussions*, vol. 7, no. 1, pp. 1525–1534, 2014.

[51] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection," *Neural networks*, vol. 108, pp. 466–478, 2018.

[52] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.

[53] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1662–1668.

[54] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[55] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1643–1656, 2014.

[56] A. Piazzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic g/sup 2/-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.

[57] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 510–515.

[58] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2010.

[59] E. Dijkstra, "A note on two problems in connexion with graphs," English, *Numerische Mathematik*, vol. 1, pp. 269–271, 1959, ISSN: 0029-599X.

[60] J. Bohren, T. Foote, J. Keller, *et al.*, "Little ben: The ben franklin racing team's entry in the 2007 darpa urban challenge," *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, pp. 231–255, 2009.

[61] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[62] Q. Wang, B. Ayalew, and T. Weiskircher, "Predictive maneuver planning for an autonomous vehicle in public highway traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1303–1315, 2019.

[63] D. Kogan and R. M. Murray, "Optimization-based navigation for the darpa grand challenge," 2006.

[64] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[65] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha — a local, continuous method," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 450–457.

[66] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *Ieee access*, vol. 2, pp. 56–77, 2014.

[67] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on control systems technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[68] J. Yoon and C. D. Crane, "Path planning for unmanned ground vehicle in urban parking area," in *2011 11th International Conference on Control, Automation and Systems*, 2011, pp. 887–892.

[69] M. Rufli and R. Siegwart, "On the application of the d* search algorithm to time-based planning on lattice graphs," in *Proc. of The 4th European Conference on Mobile Robotics (ECMR)*, Eidgenössische Technische Hochschule Zürich, 2009, pp. 105–110.

[70] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2-4, pp. 190–216, 2010.

[71] J. Bhargav, J. Betz, H. Zehng, and R. Mangharam, "Deriving spatial policies for overtaking maneuvers with autonomous vehicles," in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, IEEE, 2022, pp. 859–864.

[72] A. Musa, M. Pipicelli, M. Spano, *et al.*, "A review of model predictive controls applied to advanced driver-assistance systems," *Energies*, vol. 14, no. 23, p. 7974, 2021.

[73] J. C. Hayward, "Near-miss determination through use of a scale of danger," 1972.

[74] S. D. Pendleton, H. Andersen, X. Du, *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

[75] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.

[76] 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.

[77] C. Jewison, R. S. Erwin, and A. Saenz-Otero, "Model predictive control with ellipsoid obstacle constraints for spacecraft rendezvous," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 257–262, 2015.

[78] A. Schimpe and F. Diermeyer, "Steer with me: A predictive, potential field-based control approach for semi-autonomous, teleoperated road vehicles," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, pp. 1–6.

[79] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[80] B. Yang and C. Monterola, "A simple distributed algorithm for lightless intersection control based on non-linear interactions between vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2017, pp. 1–6.

[81] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[82] W. B. Dunbar and D. S. Caveney, "Distributed receding horizon control of vehicle platoons: Stability and string stability," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, 2011.

[83] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2016.

[84] P. Liu, A. Kurt, and U. Ozguner, "Distributed model predictive control for cooperative and flexible vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1115–1128, 2018.

[85] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Constraint-tightening and stability in stochastic model predictive control," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3165–3177, 2016.

[86] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.

[87] L. Eboli, G. Mazzulla, and G. Pungillo, "How drivers' characteristics can affect driving style," *Transportation research procedia*, vol. 27, pp. 945–952, 2017.

[88] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2111–2117.

[89] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian process-based stochastic model predictive control for overtaking in autonomous racing," *ArXiv*, vol. abs/2105.12236, 2021.

[90] I. Karafyllis, D. Theodosis, and M. Papageorgiou, "Lyapunov-based two-dimensional cruise control of autonomous vehicles on lane-free roads," *Automatica*, vol. 145, p. 110 517, 2022, ISSN: 0005-1098.

[91] M. M. Ahsan, M. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021.

[92] W. Zhan, L. Sun, D. Wang, *et al.*, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088 [cs, eess]*, Sep. 2019.

[93] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer, "Imm object tracking for high dynamic driving maneuvers," in *IEEE Intelligent Vehicles Symposium, 2004*, IEEE, 2004, pp. 825–830.

[94] V. Lefkopoulos, M. Menner, A. Domahidi, and M. N. Zeilinger, "Interaction-aware motion prediction for autonomous driving: A multiple model kalman filtering scheme," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 80–87, 2020.

[95] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds. in 2010 ieee," in *RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803.

[96] Y. Guo, V. V. Kalidindi, M. Arief, *et al.*, "Modeling multi-vehicle interaction scenarios using gaussian random field," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 3974–3980.

[97] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.

[98] S. Zhang, Y. Zhi, R. He, and J. Li, "Research on traffic vehicle behavior prediction method based on game theory and hmm," *IEEE Access*, vol. 8, pp. 30 210–30 222, 2020.

[99] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, 2015.

[100] 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, 2015.

[101] G. He, X. Li, Y. Lv, B. Gao, and H. Chen, "Probabilistic intention prediction and trajectory generation based on dynamic bayesian networks," in *2019 Chinese Automation Congress (CAC)*, IEEE, 2019, pp. 2646–2651.

[102] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification," *Electronics*, vol. 8, no. 1, p. 40, 2019.

[103] Y. Li, X.-Y. Lu, J. Wang, and K. Li, "Pedestrian trajectory prediction combining probabilistic reasoning and sequence learning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 461–474, 2020.

[104] W. Ding and S. Shen, "Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 9610–9616.

[105] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified lstm models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38 287–38 296, 2019.

[106] W. Ding, J. Chen, and S. Shen, "Predicting vehicle behaviors over an extended horizon using behavior interaction network," in *2019 international conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 8634–8640.

[107] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 1441–1446.

[108] X. Li, Y. Liu, K. Wang, and F.-Y. Wang, "A recurrent attention and interaction model for pedestrian trajectory prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1361–1370, 2020.

[109] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, "Graph neural networks for modelling traffic participant interaction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 695–701.

[110] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 3960–3966.

[111] X. Li, X. Ying, and M. C. Chuah, "Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," *arXiv preprint arXiv:1907.07792*, 2019.

[112] H. Jeon, J. Choi, and D. Kum, "Scale-net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 2095–2102.

[113] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 424–14 432.

[114] R. Chandra, T. Guan, S. Panuganti, *et al.*, "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4882–4890, 2020.

[115] Z. Zhao, H. Fang, Z. Jin, and Q. Qiu, "Gisnet: Graph-based information sharing network for vehicle trajectory prediction," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–7.

[116] J. Gao, C. Sun, H. Zhao, *et al.*, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[117] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 532–539.

[118] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6272–6281.

[119] L. Zhang, Q. She, and P. Guo, "Stochastic trajectory prediction with social graph network," *arXiv preprint arXiv:1907.10233*, 2019.

[120] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *Conference on Robot Learning*, PMLR, 2022, pp. 1035–1045.

[121] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE intelligent vehicles symposium (IV)*, IEEE, 2018, pp. 1179–1184.

[122] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 336–345.

[123] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8454–8462.

[124] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.

[125] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 6150–6156.

[126] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1349–1358.

[127] T. Zhao, Y. Xu, M. Monfort, *et al.*, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 126–12 134.

[128] Y. Wang, S. Zhao, R. Zhang, X. Cheng, and L. Yang, "Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 236–248, 2020.

[129] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, Springer, 2020, pp. 624–641.

[130] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2111–2117.

[131] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5355–5362, 2020.

[132] Y. Xu, T. Zhao, C. Baker, Y. Zhao, and Y. N. Wu, "Learning trajectory prediction with continuous inverse optimal control via langevin sampling of energy-based models," *arXiv preprint arXiv:1904.05453*, 2019.

[133] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 8, pp. 10 239–10 251, 2021.

[134] R. Bhattacharyya, B. Wulfe, D. J. Phillips, *et al.*, "Modeling human driving behavior through generative adversarial imitation learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 2874–2887, 2022.

[135] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," *arXiv preprint arXiv:1612.03653*, 2016.

[136] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1073–1087, 2017.

[137] C. Jung and D. H. Shim, "Incorporating multi-context into the traversability map for urban autonomous driving using deep inverse reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1662–1669, 2021.

[138] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2022.

[139] L. Hou, L. Xin, S. E. Li, B. Cheng, and W. Wang, "Interactive trajectory prediction of surrounding road users for autonomous driving using structural-lstm network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4615–4625, 2019.

[140] C. Dong, Y. Chen, and J. M. Dolan, "Interactive trajectory prediction for autonomous driving via recurrent meta induction neural network," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1212–1217.

[141] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[142] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.