ABSTRACT

Title of Dissertation: SAFE NAVIGATION OF AUTONOMOUS VEHICLES IN STRUCTURED MIXED-TRAFFIC ENVIRONMENTS

Muhammad Faizan Tariq
Doctor of Philosophy, 2023

Dissertation Directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

The primary driving force behind autonomous vehicle (AV) research is the prospect of enhancing road safety by preventing accidents caused by human errors. To that end, it seems rather improbable that AVs will replace all human-driven vehicles in the near future. The more plausible scenario is that AVs will gradually be introduced on public roads and highways in the presence of human-driven vehicles, leading to mixed-traffic scenarios. In addition to the existing challenges associated with autonomous driving stemming from various uncertainty factors associated with sensing, prediction, control, and computation, these situations pose further difficulties pertaining to the variability in human driving patterns. Therefore, to ensure widespread public acceptance of AVs, it is crucial to develop planning and decision-making algorithms, while benefiting from modern sensing, computation, and control methods, that can deliver safe, efficient, and reliable performance in mixed-traffic situations.

Considering the need to cater to the behavior variability of human drivers, we address the joint decision-making and motion planning problem in structured environments with a multi-timescale navigation architecture. Specifically, we design algorithms for commonly encountered highway driving scenarios that require effective real-time decision-making, reliable motion prediction of on-road entities, behavior consideration of on-road agents, and attention to safety as well as passenger comfort. The specific problems addressed in this dissertation include bidirectional highway overtaking, highway maneuvering in traffic, and crash mitigation on highways.

In the proposed framework, we first identify and exploit the different timescales involved in the navigation architecture. Then, we propose algorithmic modules while pursuing systematic complexity (data and computation) reduction at different timescales to gain immediate performance improvements in inference and action/response delay minimization. This leads to real-time situation assessment, computation, and action/control, allowing us to satisfy some of the key requirements for autonomous driving algorithms. Notably, the algorithms proposed in this dissertation ensure that the safety of the overall system is a fundamental constraint built into the system. Distinctive features of the proposed approaches include real-time operation capability, consideration for behavior variability of on-road agents, modularity in design, and optimality with respect to various metrics.

The algorithms developed and implemented as part of this dissertation fundamentally rely upon the application of optimization techniques in a receding horizon fashion which allows for optimality in performance while explicitly accounting for actuation limits, vehicle dynamics, and safety. Even though the modularity of the proposed navigation framework allows for the incorpo-

ration of modern prediction and control methods, we develop various prediction modules for the trajectory prediction of on-road agents. We further benefit from risk evaluation methodologies to ensure robustness to behavior variability of human drivers on the road and handle collision-prone situations.

In the spirit of emulating real-world situations, we place special emphasis on utilizing realistic driving simulations that capture the effects of communication delays between different modules, limitations in computation resources, and randomization of scenarios. For the developed algorithms, we evaluate the performance in comparative singular case studies as well as randomized Monte Carlo simulations with respect to several metrics to assess the efficacy of the developed methods.

SAFE NAVIGATION OF AUTONOMOUS VEHICLES IN STRUCTURED MIXED-TRAFFIC ENVIRONMENTS

by

Muhammad Faizan Tariq

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor John Baras, Chair/Advisor
Professor Eyad Abed
Professor Dinesh Manocha
Professor William Levine
Professor Pratap Tokekar

Dedication

To my family.

Acknowledgments

I would like to express my sincerest gratitude to my advisor, Professor John S. Baras, for giving me the opportunity to work on some of the most exciting problems in the research area of my interest while allowing me the freedom to carve out my own research path. It has been a privilege working under his guidance and supervision. His unbridled passion and relentless work ethic toward facing modern research problems head-on have been truly inspirational. His penchant for examining ideas and notions well beyond their surface value has been infectious and molded me into the researcher that I am today. I am also immensely thankful to Mrs. Kimberly Edwards for her ever-present support with any and all administrative issues that I encountered during my time at the University of Maryland (UMD). I would also like to acknowledge the Office of Naval Research (ONR) for partially supporting my dissertation with grant N00014-17-1-2622.

I am incredibly grateful to Honda Research Institute (HRI), US, for the instrumental role it played in the final stretch of my Ph.D. journey. My supervisors, Dr. Sangjae Bae and Dr. David Isele, were a continual source of support, guidance, advice, and resources. Their gentle demeanor coupled with endless words of encouragement paved the way for me to push beyond my limits and meet my research goals.

I would also like to thank my Ph.D. committee members - Professor Eyad Abed, Professor Dinesh Manocha, Professor William Levine and Professor Pratap Tokekar - for dedicating their

time to support my research endeavors.

I am also deeply thankful to my peers in Prof. Baras' research group, who have shaped and inspired me through discussions, presentations, and collaborations. My co-authors, Christos and Nilesh, helped me kickstart my research journey and have remained a constant part of it thereafter through their support, mentorship, and, most importantly, their friendship.

Last but not least, my dissertation work would not have been possible without the people who supported me mentally, physically, and emotionally throughout the process. Therefore, I would be remiss if I did not acknowledge my family back home for their countless sacrifices, endless prayers, and unwavering support throughout this Ph.D. journey. I also owe a great debt of gratitude to my family here in the U.S. for being an ever-present source of support, advice, encouragement, and comfort through the trials and tribulations of life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

5G	5th Generation Mobile Network
AD	Autonomous Driving
ADAS	Advanced Driver Assistance System
AEB	Autonomous Emergency Braking
AV	Autonomous Vehicle
CARLA	Car Learning to Act (Open-Source Simulator)
CasADi	Computer Algebra Systems aided Algorithmic Differentiation
CAV	Connected Autonomous Vehicle
COM	(Ego Vehicle's) Center of Mass
CRM	Coherent Risk Metric
CSI	Crash Severity Index
CVaR	Conditional Value-at-Risk
DARPA	Defense Advanced Research Projects Agency
DM	Distortion Metric
EA*	Extended-Astar (algorithm)
FIR	Finite Impulse Response
HDV	Human-Driven Vehicle
HJ	Hamilton Jacobi reachability
HSL	Harwell Subroutine Library
IDM	Intelligent Driver Model
IEEE	Institute of Electrical and Electronics Engineers
IPOPT	Interior Point Optimizer
LiDAR	Light Detection and Ranging
LP	Linear Programming

LSTM	Long-Short-Term Memory
MAIS	Maximum Abbreviated Injury Severity
MIP	Mixed Integer Program
MILP	Mixed Integer Linear Program
MI-MPC	Mixed Integer Model Predictive Controller
MIQP	Mixed Integer Quadratic Program
MOBIL	Minimizing Overall Braking Induced by Lane change
MP	Motion Planner
MPC	Model Predictive Control
NNMPC	Neural Networks integrated Model Predictive Control
No-Change	No lane-change model
ODD	Operational Design Domain
POM	Predictive Occupancy Map
RCMS	Risk-Aware Crash Mitigation System
RMOP	Risk-Aware Mixed-Integer Optimization-based Planner
RNN	Recurrent Neural Network
ROS	Robot Operating System
RRT	Rapidly exploring Random Tree
SAE	Society of Automotive Engineers
SGAN	Social Generative Adversarial Network
Sim2Real	Simulation-to-Real
SLAM	Simultaneous Localization and Mapping
SLAS	Speed and Lane Advisory System
SUMO	Simulation of Urban Mobility Software
TraCI	(SUMO's) Traffic Controller Interface
TTC	Time-to-Collision
TTCE	Time-to-Closest-Encounter
VaR	Value-at-Risk
V2V	Vehicle-to-Vehicle (Communication)
V2X	Vehicle-to-Everything (Communication)

Chapter 1: Introduction

Ever since the turn of the century, interest in automated vehicle technology has been growing, motivated by the idea of having safer roads by eliminating accidents caused by human error. This interest culminated in four cars completing the 60-mile-long urban environment route in the 2007 DARPA Urban Challenge [2]. That feat demonstrated the viability of this particular area of research and encouraged numerous vehicle manufacturers to direct their attention toward this ever-growing trend. Since then, with all the computational and intellectual resources pooled together, the research community as a whole has been edging ever closer to achieving Level 4 or 5 autonomy [3] for vehicles on public roads. With the existing Advanced driver-assistance (ADAS) [4] systems in modern vehicles, we have already accomplished Level 1 to 2 autonomy [3] in which significant control authority has already been handed over to automated systems, even though the human is still an integral part of the system, and the one in command. In pursuit of achieving the ultimate goal of having a ubiquitous presence of fully autonomous vehicles (beyond Level 4) on public roads, the research community first needs to carefully address the problems that lie on the path to making that dream a reality. See [3] for a detailed description of the Society of Automotive Engineers' (SAE) categorization of the five levels of autonomy of vehicles.

A major hurdle preventing the massive deployment of autonomous vehicle technology is

the imminent coexistence of autonomous vehicles (AVs) and human-driven vehicles (HDVs) on public roads, leading to mixed-traffic situations. With the gradual introduction of AVs alongside the existing human-driven vehicles (HDVs), a multitude of challenges is forecasted for the near future, particularly in the domains of decision-making, planning, and control, collectively referred to as *navigation*. Specifically, any navigation algorithm deployed on the AV needs to be cognizant of the varying driving patterns characteristics of human drivers [5]. This entails considering factors such as varying speeds, driving styles, responses to unexpected events, and conveying as well as understanding intentions implicitly. Therefore, the algorithms implemented on AVs must be capable of adapting and interacting effectively with HDVs to ensure smooth and safe maneuvering in mixed-traffic scenarios. By addressing these challenges head-on, we can pave the way for the successful and harmonious integration of AVs into the existing transportation landscape.

Inspired by how human decision-making works [6], we approach the autonomous vehicle navigation problem with various variants of a multi-time-scale predictive control approach with consideration for human drivers' behavior. In his theory of 'thinking fast and slow,' Kahneman [6] proposes a human decision-making model composed of two systems: System 1 which is fast, automatic, frequent, and unconscious; and System 2 which is slow, effortful, infrequent, and conscious. The two systems work harmoniously with complex communication, computation, and information interchange allowing humans to perform navigation tasks effortlessly. This concept is well-illustrated by an example from John Doyle's work on speed-accuracy tradeoff [7]. Imagine a mountain biker descending a trail: The biker's System 2, which involves conscious and deliberate decision-making based on advanced warning via visual information, sets high-level goals for

navigating the trail while System 1, which operates unconsciously and reflexively, handles disturbances and minor obstacles while accounting for the dynamics of the rider and bike. Therefore, the coordinated operation of these two systems enables the biker to successfully *navigate* the mountain trail. Motivated by this, we propose our multi-time-scale navigation architecture, outlined in Section 1.1 below.

1.1 Algorithmic Pipeline

Applying the principles of human decision-making to the autonomous vehicle navigation task, it becomes readily apparent that the decision-making, planning, and control subtasks require multiple subsystems working harmoniously at different time scales and model fidelities to achieve safe and efficient driving. Therefore, we utilize a multi-time-scale navigation stack and an overall modular algorithmic pipeline that allows for the integration of various external modules to meet autonomous driving requirements.

Figure 1.1 illustrates the algorithmic pipeline of the proposed navigation stack in reference to the various existing algorithmic modules deployed on an autonomous vehicle. Our self-contained navigation stack, as outlined by the dotted rectangle in Figure 1.1, is composed of behavior planning, motion planning, and vehicle control modules. The behavior planner handles high-level decision-making to meet the problem-specific design objectives while accounting for the behavior of human-driven vehicles whereas the motion planner determines a trajectory based on the goals set by the behavior planner. The two planning modules typically incorporate trajectory predictors to model and predict the behavior of human-driven vehicles. The controller, on the other hand,

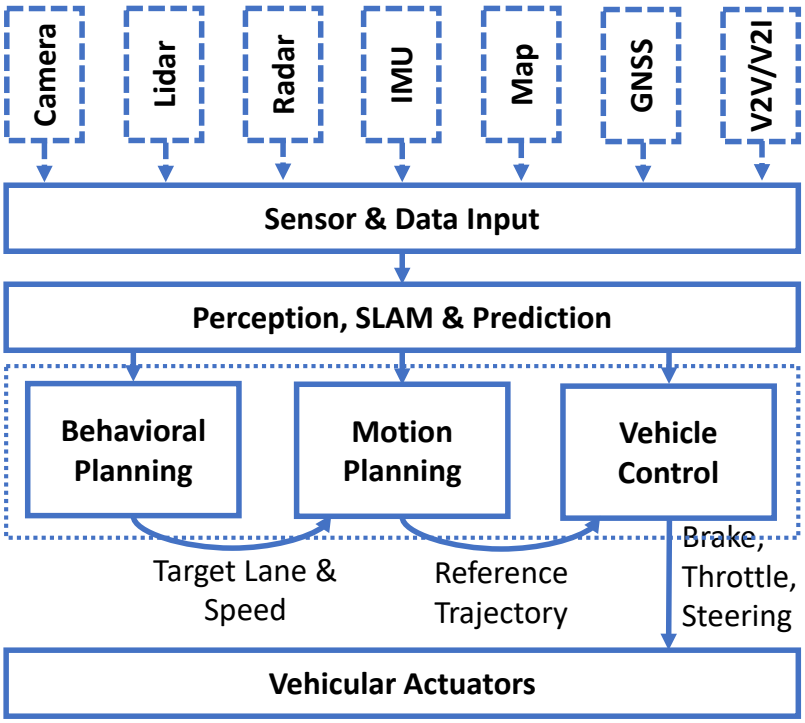

Figure 1.1: **Algorithmic Pipeline**. Given the geographical information of the surroundings (lane boundaries and center line trajectories) and the relative positions and velocities of neighboring agents, the behavioral planning module computes the target lane and speed reference. The center line trajectory of the target lane coupled with the target speed is then passed to the motion planner that generates a reference trajectory as a function of time-dependent longitudinal and lateral displacements. The vehicle controllers take in the reference trajectory to determine throttle and steering commands to track the trajectory accordingly.

generates actuation commands to accurately follow the trajectory generated by the motion planner, considering sensor uncertainties and actuator disturbances. The taxonomy of the various components of the navigation stack is borrowed from [8].

As for the data flow between the various algorithmic modules, the raw sensory input data is first processed by the Perception and Simultaneous Localization and Mapping (SLAM) modules to place the autonomous vehicle relative to the various environmental entities in a unified frame of reference. This information is then transmitted to the navigation stack. The output of the

navigation module is then passed down further in terms of actuation commands (brake, throttle, and steering) to the actuators. In terms of computational tradeoffs, the behavior planning layer operates at a low frequency with a long planning horizon ($> 15s$) while employing a low-fidelity dynamical model; the motion planning layer operates at a moderate frequency with a moderate planning horizon ($\sim 4s$) while employing a moderate-fidelity dynamical model; and the vehicle control layer operates at a high frequency with short control horizon ($\sim 2s$) while employing a high-fidelity dynamical model.

Given the abundance of methods available to address control and actuation tasks [9–15], our focus in this dissertation is on the algorithm development for behavior and motion planning modules, while leveraging existing control modules from the literature.

1.2 Trajectory Prediction

An essential consideration for the behavior and motion planning modules is the accurate and reliable motion prediction of on-road agents. Here, any moving entity present within the traffic environment qualifies as an on-road agent as long as its motion somehow affects the decision-making process of the autonomous ego vehicle. With the ever-growing corpus on motion prediction methods [16], it is imperative to design modular *planning* algorithms that have the flexibility to incorporate modern *prediction* algorithms.

As for our proposed approaches, we develop various prediction algorithms with varying levels of sophistication, dependent on the task at hand. However, as alluded to in Section 1.1, a key component of our algorithmic pipeline is the modularity in its design. This allows for the

integration of modern prediction algorithms that may be better suited for the problem at hand.

1.3 Challenges

Even though AV navigation research has made significant strides in recent years [8], there still are several challenges that need to be addressed before AVs can gain a significant presence on public roads [3]. The ability to navigate in the presence of human-driven, especially volatile, vehicles is particularly difficult because the decision of an AV, in regards to behavior and motion planning, at any moment is highly dependent on the behavior of neighboring vehicles. The key considerations in the decision-making and planning process include reliable situation assessment as well as timely decision-making. A misinformed or mistimed decision, in the worst case, will lead to a fatal road accident or, at the very least, earn the wrath and mistrust of other drivers. On the contrary, a sensible well-timed maneuver can gain improvements in travel time, passenger comfort, and safety for not just the ego vehicle but the surrounding vehicles.

Reliable situation assessment involves reasonable trajectory prediction of the surrounding vehicles. This is typically challenging because the future trajectory of a vehicle is highly contingent upon the intrinsic behavior variability of human drivers, as outlined in Section 1.2. As for timely decision-making, it requires the algorithms onboard an autonomous vehicle to operate in real-time so that they are able to quickly adjust to the rapidly evolving dynamic situation. Therefore, some of the key requirements for an AV decision-making system are embodied by the need for safety, which is aided by adequate situation assessment, as well as real-time operation.

The algorithms presented in this dissertation prioritize the safety of the overall system as

a fundamental requirement. Notable characteristics of our proposed methods include real-time operability, accounting for the variability in on-road agent behavior, modularity in design, and optimization with respect to some metrics.

1.4 Contributions

Considering the core tenets of AV technology, we tackle various closely related problems that highlight the need for safety in the presence of varying driver behavior as well as the importance of real-time operation. To start off, we first address the autonomous vehicle overtaking problem in a bidirectional traffic flow setting. In this problem, the autonomous ego vehicle is required to perform overtaking maneuvers on a two-lane bidirectional highway by safely moving into the adjacent lane while avoiding oncoming traffic. Here, maintaining safety takes precedence over improving performance, i.e. minimizing travel time, due to the potential fatality of a head-on collision. For this problem, we also explore the benefits of cooperative sensing in the decision-making process.

Moving on, we address the maneuvering problem in traffic on a multi-lane highway that highlights the importance of foresight in behavior planning as well as behavior consideration of on-road agents. This scenario emphasizes the importance of incentivizing performance while influencing the behavior of other vehicles. However, safety still remains the ultimate priority for the ego vehicle. Unlike the bidirectional overtaking problem, however, being overly defensive in this scenario can be disadvantageous as it may encourage aggressive human-driven vehicles to cut in front, potentially compromising safety.

Finally, we address the crash mitigation problem where an AV has to perform evasive maneuvers in high-risk situations and minimize the severity of a collision if a crash is inevitable. Contrary to the scenarios discussed earlier, this problem is not tied to any specific use case but is applicable to any generic situation where an AV finds itself needing to perform evasive maneuvers in order to ideally prevent a collision or minimize its severity if it is truly unavoidable.

Bidirectional Overtaking

In Chapter 2, we tackle the autonomous vehicle overtaking problem in a bidirectional mixed-traffic setting using a non-cooperative as well as cooperative approach. In the first part of the chapter (Section 2.3), we address the non-cooperative case by designing a model predictive controller that maximizes the ego vehicle's velocity while prioritizing safety and accounting for driver comfort. Our controller does not assume full knowledge of the environment and utilizes a realistic sensing and occlusion model instead. In regards to safety, we define variable safety margins as functions of user-defined vehicle-specific attributes. Finally, the complexity of the receding horizon optimal control problem is reduced by introducing a binary decision variable to approximate the integrated lateral dynamics of the ego vehicle, thus decoupling the longitudinal and lateral components of the dynamical model. As a result, the proposed approach is able to operate in a limited sensing range while accounting for occlusion, is capable of active perception, is able to retract the overtake decision through the receding horizon approach, is robust to variations in sensory input and driving behaviors of external agents, and reduces to a mixed-integer optimization problem with linear constraints yielding low computational complexity. We evaluate the efficacy of the proposed

approach in a widely used simulation environment. These contributions are published in [17].

In the second part of the chapter (Section 2.4), we shift our focus to the cooperative case where we combine the benefits of cooperation-driven communication-based control systems with the safety guarantees of an optimization-based controller in the domain of bidirectional mixed traffic overtaking. The main contribution of this work involves the implementation of a vehicle-to-vehicle communication-based multi-agent strategy for autonomous bidirectional overtaking which reduces the impact of blind spots created by sensor occlusion. In this regard, we also propose a velocity-tracking method for trajectory prediction and an improved sensor occlusion model. Our cooperative method is then tested using the realistic SUMO traffic simulation system, and the results and comparisons are presented. We show that this method leads to improved performance with less risky maneuvers, higher overall throughput, and a high success rate for overtaking attempts. Further tests are also carried out to show the impact of the penetration level of connected autonomous vehicles and lane density on the performance of our method. These contributions are published in [18].

Highway Maneuvering

In Chapter 3, we address the highway maneuvering problem in the presence of HDVs with both the interaction and risk *agnostic* as well as *aware* approaches. The primary focus in the first part of the chapter (Section 3.2) is on foresight in decision-making, without any interaction and risk considerations. Therefore, we introduce a real-time optimization-based behavior planning module that combines the advantages of optimization-based methods with high computational efficiency.

This is achieved by utilizing a binary representation of the decoupled lane indicator dynamics and implementing algorithmic modifications for improved numerical computations. This method ensures optimality in terms of travel time and comfort while also providing safety guarantees and real-time applicability for long planning horizons. Additionally, its modular design allows for the integration of external trajectory prediction modules. By incorporating strategic decision-making while fulfilling the core algorithmic requirements, our proposed method achieves long-term performance benefits, as validated in simulation studies. These contributions are published in [19].

In the second part of the chapter (Section 3.3), we enhance the formulation by incorporating interaction and risk considerations into the previously developed behavior planning framework. To handle the inherent behavioral uncertainty of on-road agents, we utilize an interaction-aware trajectory prediction model based on a Recurrent Neural Network (RNN) architecture. The volatility in behavior is quantified using the Conditional Value-at-Risk (CVaR) metric, while slack variables are introduced to ensure solution feasibility. Additionally, a comprehensive case study and extensive Monte-Carlo testing are conducted using a realistic simulator to evaluate the effectiveness of our enhanced method.

Crash Mitigation

In Chapter 4, we propose a novel Risk-Aware Crash Mitigation System (RCMS) that comprises two key components: an activation mechanism (Section 4.4.1) and a modular trajectory generation module (Section 4.4.2). The activation mechanism plays a crucial role in achieving a

seamless transition between RCMS and Motion Planning (MP) systems, as these systems possess inherently different objectives. To facilitate this transition, the activation mechanism integrates instantaneous and predictive collision risk evaluations within a hysteresis band. This approach ensures a smooth handoff between the two systems, enhancing passenger comfort and safety. The trajectory generation component, on the other hand, focuses on minimizing situational risk by leveraging a smooth functional profile for risk evaluation. This component considers various constraints, such as actuation limits, dynamical constraints, and road limitations, to generate trajectories that effectively mitigate situational risk. By considering these factors, we aim to enhance the overall safety and efficiency of the autonomous vehicle navigation system, especially in collision-prone scenarios. To validate the effectiveness of our proposed approach, we conduct simulations involving high-risk situations.

1.5 Notation

Throughout this dissertation, we denote the set of integers by $\mathbb{Z}$, the set of non-negative integers by $\mathbb{N}$, and the set of real numbers by $\mathbb{R}$. For some $a, c \in \mathbb{Z}$ and $a < c$, we write $\mathbb{Z}_{[a,c]} = \{b \in \mathbb{Z} \mid a \leq b \leq c\}$. For some $d, f \in \mathbb{N}$ and $d < f$, we write $\mathbb{N}_{[d,f]} = \{e \in \mathbb{N} \mid d \leq e \leq f\}$. For some $g, i \in \mathbb{R}$ and $g < i$, we write $\mathbb{R}_{[g,i]} = \{h \in \mathbb{R} \mid g \leq h \leq i\}$. Whenever $<, \leq, =, \geq, >$ are applied to vectors, they are interpreted element-wise.

Chapter 2: Bidirectional Overtaking

With the advent of autonomous vehicles on public roads imminent in the near future, it is only a matter of time before autonomous vehicles will be sharing public roads with human-driven vehicles. Therefore, it is imperative to place special emphasis on addressing safety concerns in scenarios pertaining to mixed-traffic settings that are comprised of human-driven and autonomous vehicles. The overtaking problem in a bidirectional mixed-traffic setting is a prime example that highlights the difficulties pertaining to the variations in human driving patterns [5, 20]. While fundamentally similar to the lane-changing problem [21], the overtaking problem has added complexity due to the presence of oncoming traffic, which increases the chances of head-on collisions. Moreover, the safety of overtaking maneuvers is highly contingent upon uncertainty factors such as road conditions, measurement accuracy, sensor occlusion, human driving behavior, etc., making it a challenging problem to address.

In this chapter, we address the problem of autonomous vehicle overtaking in a bidirectional mixed-traffic setting using a receding horizon optimization-based approach for the non-cooperative as well as the cooperative case. In the first part of this chapter, we address the non-cooperative case, where no autonomous agents are assumed to be present in the vicinity of the autonomous ego vehicle. Therefore, the ego vehicle relies solely on its onboard sensors as the primary source of

input information. In the second part, we extend our analysis by incorporating cooperative sensing and a refined trajectory prediction model to assess performance improvements in the presence of varying densities of autonomous agents in the environment.

In the first part, we propose a mixed-integer model predictive controller that maximizes the ego vehicle's velocity while prioritizing safety and accounting for driver comfort. The proposed approach: (i) operates in a limited sensing range while accounting for occlusion; (ii) is able to retract the overtake decision through a receding horizon approach; (iii) is robust to the variations in sensory input and driving behaviors of external agents due to behavior-dependent safety margins; and (iv) reduces to a mixed-integer optimization problem with linear constraints, yielding low computational complexity. We further demonstrate the behavior of the proposed approach in a realistic traffic simulation environment.

In the second part, we utilize the communication capabilities of connected autonomous vehicles (CAVs) to design a cooperative communication-based approach that utilizes the information shared between CAVs to reduce the effects of sensor occlusion while benefiting from the local velocity prediction based on past tracking data. Our control framework aims to perform overtaking maneuvers with the objective of maximizing velocity while prioritizing safety and passenger comfort. Our method is also capable of reactively adjusting its plan to dynamic changes in the environment. The performance of the proposed approach is verified using realistic traffic simulations under varying CAV penetration levels.

This chapter is structured as follows: Section 2.1 provides an overview of the problem; Section 2.2 summarizes the existing methods available in the literature to solve the problem; Section

2.3 expounds on the non-cooperative case; Section 2.4 details the cooperative case; Section 2.5 summarizes our contributions while providing concluding remarks; and, Section 5.2 expands on the future outlook. As for the non-cooperative case, Section 2.3.1 formalizes the problem; Section 2.3.2 details the various components of the proposed approach and highlights its analytical as well as implementation details; and Section 2.3.3 analyzes the performance of the proposed approach by running a series of tests and providing a discussion on the results. As for the cooperative case, Section 2.4.1 details the problem description; Section 2.4.2 discusses the proposed approach while highlighting its implementation and analytical details; and Section 2.4.3 evaluates the performance of the proposed approach by running a series of tests and providing a discussion on the results.

2.1 Overview

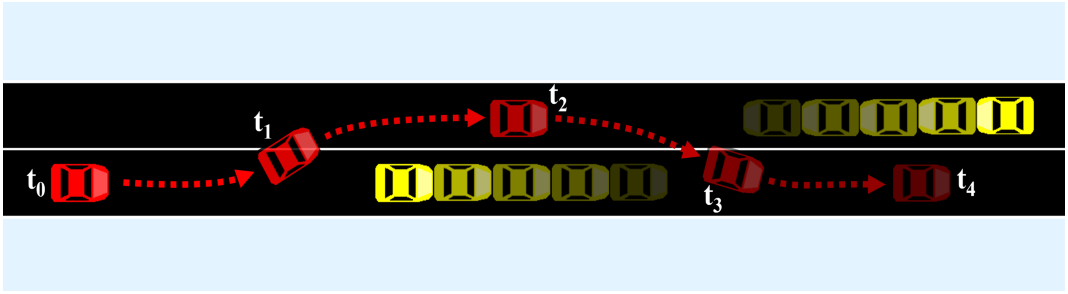

Figure 2.1: **Bidirectional Overtaking Scenario.** The ego vehicle is depicted in red while the leading and oncoming vehicles are both shown in yellow. The progression of time is shown with a lighter shade. Dotted arrows show the ego vehicle's trajectory while the $t_0 < t_1 < t_2 < t_3 < t_4$ markers represent the progression of time.

In the bidirectional overtaking problem, an ego vehicle is required to overtake the leading vehicle(s) by moving into the adjacent lane while avoiding oncoming traffic. To motivate this problem, consider the scenario presented in Figure 2.1. This scenario involves three types of vehicles:

an autonomous ego vehicle that carries out the overtaking maneuver, vehicle(s) traveling ahead in the same lane as the ego vehicle, and the approaching vehicle(s) in the oncoming lane. No restriction is placed on the class of non-ego vehicles such that they can be human-driven or autonomous. The goal of the ego vehicle is to travel at its maximum safe velocity and, when required, safely overtake the vehicle(s) traveling ahead before returning to its original lane. Safety encompasses respecting ego vehicle's design (state, control, and actuation) limits, real-time operation, maintaining safe distances to the neighboring vehicles, and obeying traffic rules (staying within lanes and not exceeding speed limits). The typical vehicle trajectories in a simple three-vehicle scenario are depicted in Figure 2.1, with the ego vehicle depicted in red and the lead vehicle and oncoming vehicle both depicted in yellow.

2.2 Literature Review

The approaches available in the literature to address the overtaking problem from a single agent's perspective can broadly be classified into sampling, learning, and optimization-based methods. The sampling-based approaches typically involve sampling feasible trajectories from a reachable safe set and are, therefore, able to incorporate non-holonomic constraints and safety guarantees [22]. Typically, single-query methods, in particular the different variants of RRT, are preferred over multi-query methods, e.g., roadmap-based methods, due to the faster execution time and their ability to incorporate non-holonomic constraints [23]. However, the overall driving experience is often uncomfortable due to the concatenation of individual trajectories, and the asymptotic optimality guarantees do not translate to real-world implementability in complex driving scenarios

due to high sample complexity [23].

The learning-based methods are among some of the most popular in the literature. Due to a lack of a standardized dataset for the problem at hand, these approaches consist mainly of variations of reinforcement learning techniques trained in simulated environments [24]. Despite their seemingly good performance in simulations, the real-world implementation of these approaches raises a number of concerns (Sim2Real gap), such as the need for large amounts of training data, exploration of unsafe behaviors, general inability to handle edge cases, and most importantly, the lack of explainability and safety guarantees as a consequence of utilizing neural networks as function approximators [25].

The optimization-based approaches have been a standard in solving problems pertaining to autonomous vehicle planning and control. In the class of optimization-based approaches, potential-field-based methods [26] yield decent collision avoidance performance but are unable to accommodate vehicle dynamics. To accommodate the vehicle dynamics into the problem formulation, the most prominent approaches in this area revolve around optimal control methods which are able to handle collision avoidance constraints while providing performance guarantees. However, they do so at the cost of high computational complexity, since they require solving an optimization problem on a functional space over a large time window, where the integration of non-linear dynamical models typically yields the additional bottleneck of non-convex constraints [27]. An efficient trade-off between performance guarantees and computational complexity is the Model Predictive Control (MPC) approach [13]. Many MPC variants have been proposed in the literature, such as Stochastic-MPC [28] and Robust-MPC [29], that allow for uncertainty considerations

in the system dynamical model. However, these methods have not yet been applied to the bidirectional traffic flow setting. A promising exception in this regard is the recent work of Sulejmani et al. in [30], where a Stochastic-MPC method is applied in parallel with a Bayesian network to predict the trajectories of human-driven vehicles. This approach, however, would require retraining the network when used in different environments. Finally, stochastic control formulations that explicitly account for the behavior of human-driven vehicles have been explored in [31,32]. These methods incorporate stochastic models to represent the behavior of human-driven vehicles and derive optimal control laws based on a stochastic formulation. However, despite their rigorous formulation, these methods rely on simplifying assumptions and encounter significant computational complexity, rendering them impractical for real-time applications.

From the multi-agent perspective, the use of communication among CAVs in order to improve the overall efficiency and safety of many complex traffic conditions such as highway merging [33] and traffic shock wave dissipation [34] have also been explored in the literature. This cooperation-focused approach has multiple benefits in a bidirectional overtaking scenario. Some of these benefits have been studied in [35] and [36], where probabilistic-driven approaches have been applied to cooperative single-direction overtaking and lane changing, respectively.

2.3 Non-Cooperative Overtaking

In the first part of this chapter, we present a planning and control framework for the non-cooperative case, where we do not assume the presence of any other autonomous agents near the autonomous ego vehicle, hence, limiting the source of input information for the ego vehicle

to its onboard sensors. Then, the focus is on developing a model predictive controller (MPC) that maximizes the velocity of the ego vehicle while considering safety and driver comfort. The proposed method incorporates realistic sensing and occlusion models, customizable safety margins based on vehicle-specific attributes, and a binary decision variable to simplify the computational complexity of the control problem.

2.3.1 Problem Formulation

In this section, we detail the road model, the dynamics model for the ego and the non-ego vehicles, and the sensing model for the ego vehicle that will be used for the remainder of this work.

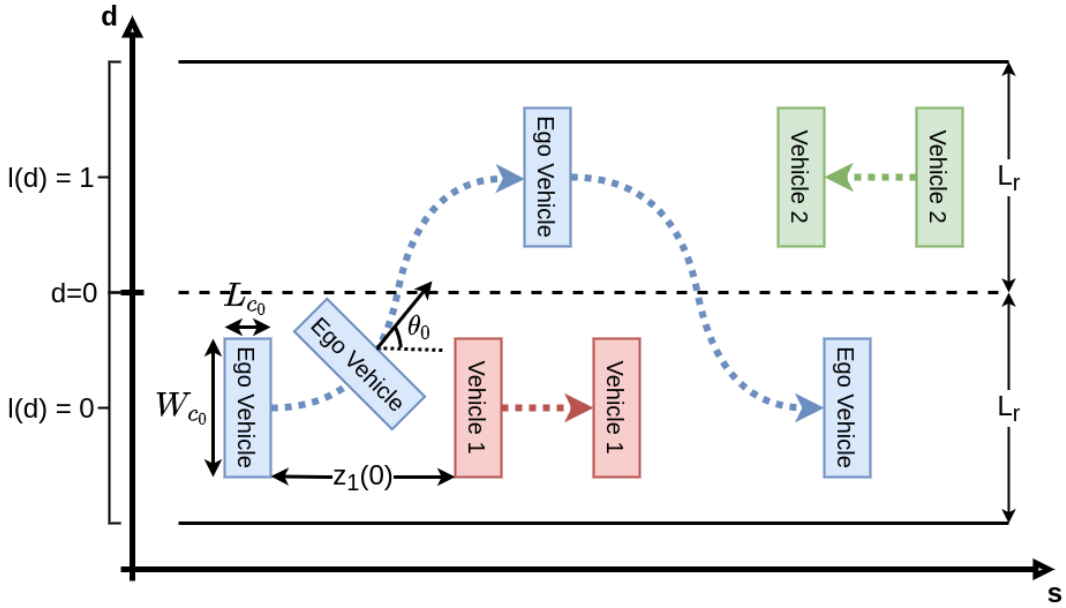

Figure 2.2: **Bidirectional Overtaking Problem Formulation.** On a two-lane bidirectional highway, the ego vehicle (shown in blue) overtakes the lead vehicle (shown in red) while avoiding the oncoming vehicle (shown in green). The trajectories of vehicles over the course of the ego vehicle's overtaking maneuver are shown by the dashed arrows in appropriate colors. The variables introduced in the formulation are also labeled appropriately.

2.3.1.1 Road Model

The width of each of the lanes is denoted by L_r and the speed limit by V_l. Let the global Frenet coordinate frame be centered at the left end of the lane divider. The Frenet coordinate frame comprises two variables: longitudinal displacement (s) and lateral displacement (d). The binary variable l distinguishes between the two lanes such that $l(d) = \mathbb{1}_{d \geq 0}$. All introduced variables are depicted in Figure 2.2.

2.3.1.2 Vehicle Dynamics

Let $t \in \mathbb{R}_{\geq 0}$ denote the time variable and $\mathbb{V}(t) = \{x_i(t) \in \mathbb{X} \mid i \in \mathbb{N}_{[0,n]}\}$ denote the set of states of the vehicles present in the environment at time instant t. The state x_i is comprised of longitudinal displacement (s_i), lateral displacement (d_i) and heading angle (θ_i), represented compactly as:

$$x_i(t) = [s_i(t), d_i(t), \theta_i(t)]^\top \in \mathbb{X} \tag{2.1}$$

where

$$\mathbb{X} = \{z \in \mathbb{R}^3 \mid [-\infty, -L_r, -\pi]^\top \leq z \leq [\infty, L_r, \pi]^\top\}$$

represents the physical limits on the state variables. The index $i = 0$ identifies the ego vehicle while $i \in \mathbb{N}_{[1,n_0]}$ and $i \in \mathbb{N}_{[n_0+1,n]}$ denote the vehicles, excluding the ego vehicle, traveling in lanes $l(d(t)) = 0$ and $l(d(t)) = 1$, respectively. We define $n_1 = |\mathbb{N}_{[n_0+1,n]}| = n - n_0$. For simplicity, and without affecting any safety guarantees (Section 2.3.2.3), the vehicles are physically modeled as rectangles with the length and width of vehicle i defined as L_{c_i} and W_{c_i}, respectively, as shown

in Figure 2.2.

Ego Vehicle Dynamics Model: With the sampling period denoted by T_s, the discretized dynamics of the ego vehicle are modeled by the non-holonomic unicycle (*Dubins*) model, as follows:

$$s_0(k+1) = s_0(k) + v_0(k) \cdot \cos(\theta_0(k)) \cdot T_s \tag{2.2}$$

$$d_0(k+1) = d_0(k) + v_0(k) \cdot \sin(\theta_0(k)) \cdot T_s \tag{2.3}$$

$$\theta_0(k+1) = \theta_0(k) + \omega_0(k) \cdot T_s \tag{2.4}$$

where $s_0(0) = 0$, $d_0(0) = -{}^{L_r}/_2$, $\theta_0(0) = 0$, and $k \in \mathbb{N}_{\geq 0}$. Here, the longitudinal velocity (v_0) and angular velocity (ω_0) are the control variables, represented compactly as:

$$u_0(k) = [v_0(k), \omega_0(k)]^\top \in \mathbb{U}(u_0(k-1), T_s) \tag{2.5}$$

where

$$u_0(0) = [0, 0]^\top, \text{ and}$$

$$\mathbb{U}(a, T_s) = \{z \in \mathbb{R}^2 \mid [0, \omega_{min}]^\top \leq z \leq [V_{max}, \omega_{max}]^\top,$$

$$[A_{min}, \alpha_{min}]^\top \leq \frac{z - a}{Ts} \leq [A_{max}, \alpha_{max}]^\top\}.$$

Here, $\mathbb{U}(u_0(k-1), T_s)$ represents the physical limits on control inputs at time instant k. V_{max} denotes the maximum reachable linear velocity of the ego vehicle while ω_{min} and ω_{max} denote the minimum and maximum reachable angular velocity of the ego vehicle. The linear velocity, $v_0(k)$, has a lower bound of 0 because reversing behavior is not permitted on a highway. Moreover,

bounds are placed on the actuation limits by defining the admissible linear and angular accelerations. Here, A_{min} and A_{max} correspond respectively to the maximum linear deceleration and acceleration while α_{min} and α_{max} correspond respectively to the maximum angular deceleration and acceleration of the ego vehicle.

Observed Vehicles' Dynamics Model: The dynamics of the observed vehicles are modeled using the linearized unicycle (*Dubins*) model around the trajectory defined by $\theta(k) = 0$, as follows:

$$\begin{aligned} s_i(k+1) &= s_i(k) + v_i(k) \cdot T_p + \tilde{n}_{v_i}(k) \\ d_i(k+1) &= d_i(0) \\ \theta_i(k+1) &= \theta_i(0) \end{aligned} \tag{2.6}$$

for all $i \in \mathbb{N}_{[1,n]}$, where $v_i(k) \in \mathbb{R}_{[0,V_{max}]}$, $\tilde{n}_{v_i}(k) \sim \mathcal{N}(0,\, \sigma_v^2)$, $d_i(0) = \begin{cases} -\frac{L_r}{2}, & i \in \mathbb{N}_{[1,n_0]} \\ \frac{L_r}{2}, & i \in \mathbb{N}_{[n_0+1,n]} \end{cases}$, and $\theta_i(0) = 0$. Here, T_p corresponds to the observation sampling period and $\tilde{n}_{v_i}$ corresponds to the noise in the dynamics model. Notice that the observation sampling period is denoted by a separate variable T_p to indicate that it need not be the same as the discretization sampling period T_s, introduced in Section 2.3.1.2. This distinction will play an important role in Section 2.3.2.1.

2.3.1.3 Sensing Model

Measurement variables: Utilizing data from the onboard sensor suite, state-of-the-art sensor fusion algorithms [37] are able to obtain the relative displacement of vehicles present in the vicinity

of the ego vehicle with a high degree of accuracy. Based on the availability of sensor data, we formulate the problem in terms of the relative displacements $z_i(k)$ of the set of observed vehicles $\{i : x_i \in \mathbb{O}\}$, defined in Section 2.3.1.3. The noisy observation measurements $\tilde{z}_i(k)$ are defined in a moving coordinate system centered at the ego-vehicle at any time instant k as follows:

$$\tilde{z}_i(k) = z_i(k) + \tilde{n}_{s_i} = s_i(k) - s_0(k) + \tilde{n}_{s_i} \tag{2.7}$$

where $\tilde{n}_{s_i} \sim \mathcal{N}(0, \sigma_s^2)$, and $\tilde{z}_i(k) \in \mathbb{R}_{[-L_s, L_s]}$. Here, L_s corresponds to the measurement range of the ego vehicle's on-board sensor suite and $\tilde{n}_{s_i}$ corresponds to the noise in the measurement variables. The additive uncorrelated Gaussian noise model for the measurement noise is justified by the experimental results that show good real-world performance for 3D LiDAR data under this modeling assumption [38].

Occlusion: To account for occlusion, we model the limited sensor visibility in the adjacent lane in the presence of a leading vehicle. This is done by introducing an alternate measurement range $L_o < L_s$, shown in Figure 2.3, that defines the maximum longitudinal displacement at which a vehicle can be observed in the adjacent lane when a leading vehicle is present within the measurement range L_s. In practice, L_o can be a function of headway (distance to the leading vehicle), but for the scope of this work, it is assumed to be constant. We take the value of L_o as the worst-case measurement range for the adjacent lane i.e. the visibility range in the adjacent lane when the leading vehicle is traveling at the minimum allowable safety margin L_{0_1} (see Section 2.3.2.2).

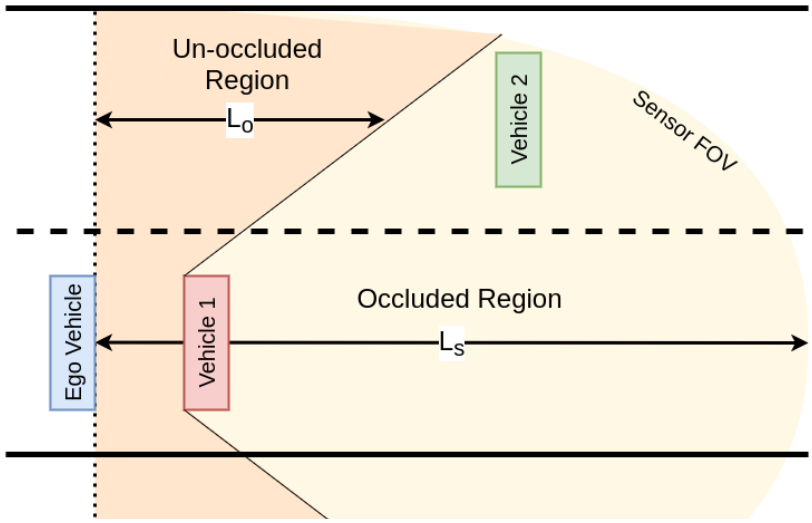

Figure 2.3: **Occlusion Model**. Due to the proximity of the leading vehicle to the ego vehicle, the ego vehicle's visibility range is restricted such that the oncoming vehicle, which would have been otherwise visible, is now hidden.

Then, the ego vehicle's observation state set is defined as:

$$\mathbb{O}_0(k) = \min_{0 \leq z_i(k) \leq L_s} \{x_i(k) \in \mathbb{V}(k) \setminus \{x_0(k)\} \mid (l(d_i(k)) = l(d_0(k))\} \tag{2.8a}$$

$$\mathbb{O}_1(k) = \{x_i(k) \in \mathbb{V}(k) \setminus \{x_0(k)\} \mid (l(d_i(k)) \neq l(d_0(k)) \cap (|z_i(k)| \leq L_o)\} \tag{2.8b}$$

$$\mathbb{O}(k) = \mathbb{O}_0(k) \cup \mathbb{O}_1(k) \tag{2.8c}$$

At any time instant k, the set of states of the observed vehicles is denoted by $\mathbb{O}(k)$, as shown in (2.8c). For the vehicles traveling in the same lane as itself, the ego vehicle is able to observe the vehicle in its direct line of sight given that it falls within the measurement range L_s, as shown in (2.8a). As for the vehicles traveling in the adjacent lane, the ego vehicle is able to observe all the vehicles in the un-occluded region, defined by L_o, as shown in (2.8b).

2.3.2 Methodology

In this section, we describe a state estimation mechanism based on Kalman filtering, define the main optimal control problem, and prove the existence of a feasible solution that respects all the defined safety constraints.

2.3.2.1 State Estimation

The ego vehicle estimates the relative longitudinal displacement $\hat{s}_i(k)$ of the vehicles with states in the observation state set $\mathbb{O}(k)$ based on the longitudinal dynamics model (2.6) and the measured longitudinal displacement (2.7). Let the estimated relative longitudinal displacement and the estimated covariance at time instant k for the vehicle i be defined as $\hat{s}_i(k)$ and $\hat{\Sigma}_i(k)$ respectively. Then, for all $k > 0$ and $i \in \mathbb{N}_{[1,n]}$ such that $x_i \in \mathbb{O}(k)$, $\hat{s}_i(k)$ and $\hat{\Sigma}_i(k)$ can be estimated using a Kalman filter [39] as follows:

$$\tilde{u}_i(k) = \frac{\hat{s}_i(k-1) - \hat{s}_i(k-2)}{T_p} \tag{2.9}$$

$$\tilde{s}_i(k) = A_i(k) \cdot \hat{s}_i(k-1) + B_i(k) \cdot \tilde{u}_i(k) - (s_0(k) - s_0(k-1)) \tag{2.10}$$

$$\tilde{\Sigma}_i(k) = A_i(k) \cdot \hat{\Sigma}_i(k-1) \cdot A_i^\top(k) + R_i(k) \tag{2.11}$$

$$K_i(k) = \frac{\tilde{\Sigma}_i(k)}{\tilde{\Sigma}_i(k) + Q_i(k)} \tag{2.12}$$

$$\hat{s}_i(k) = \tilde{s}_i(k) + K_i(k) \cdot (\tilde{z}_i(k) - \tilde{s}_i(k)) \tag{2.13}$$

$$\hat{\Sigma}_i(k) = (1 - K_i(k)) \cdot \tilde{\Sigma}_i(k) \tag{2.14}$$

where $\hat{s}_i(q) = \tilde{z}_i(q)\ \forall q \in \{0,1\}$, $\hat{\Sigma}_i(0) = R_i(k) = \sigma_{v_i}^2$, $Q_i(k) = \sigma_{s_i}^2$, $A_i(k) = 1$, $B_i(k) = \begin{cases} T_p, & i \in \mathbb{N}_{[1,n_0]} \\ -T_p, & i \in \mathbb{N}_{[n_0+1,n]} \end{cases}$. Since the estimator sampling time T_p is independent of the controller sampling time T_s, the estimator can be run at a much higher frequency as compared to the controller because the linear recursive updates of the estimator have low computational complexity. This difference in frequency updates is a known property of dynamic observer design that helps establish practical convergence of the estimated observations to the actual ones.

2.3.2.2 Optimal Control Problem

The optimal control objective is to determine a sequence of velocity commands that would enable the ego vehicle to maximize its velocity while respecting its dynamics, actuator limits, and safety constraints. In a highway overtaking scenario, where the road curvature is small, it is reasonable to assume that the longitudinal and lateral dynamics of the ego vehicle are decoupled [40]. Moreover, given an adequate longitudinal safety margin, the existing low-level vehicle controllers are able to efficiently perform lane change maneuvers while abiding by lateral dynamical constraints [41]. Under these premises, we introduce a hierarchical controller architecture composed of a high-level central controller and a low-level lateral controller. The central controller handles the high-level decision-making based on estimated longitudinal measurements and provides commands to the lateral controller that governs the low-level lateral movements.

Central controller: The central controller makes use of a binary decision variable $\mathcal{D}(k)$ to abstract out the lateral dynamics. This allows for a decoupling of longitudinal and lateral dynamics and

controllers, which in turn leads to an overall reduction in computational complexity, as evident in the timing statistics provided in Section 2.3.3.

Let the decision to overtake at any time instant k be denoted by $\mathcal{D}(k) \in \{0, 1\}$. Here, $\mathcal{D}(k) = 1$ corresponds to the decision to travel in the adjacent lane ($l(d(k)) = 1$) while $\mathcal{D}(k) = 0$ represents the decision to travel in the original lane ($l(d(k)) = 0$). With the lateral component of the dynamics model in Section 2.3.1.2 represented by the binary decision variable, the decoupled longitudinal dynamics model linearized around the trajectory $\theta(k) = 0$ is posed as follows:

$$\bar{x}(k+1) = \bar{x}(k) + \bar{u}(k) \cdot T_s \tag{2.15}$$

where $\bar{u}(k) \in \bar{\mathbb{U}}(\bar{u}(k-1), T_s) = \{z \in \mathbb{R}_{[0,V_l]} \mid A_{min} \leq \frac{z-\bar{u}(k-1)}{Ts} \leq A_{max}\}$, $\bar{x}(0) \in \mathbb{R}_{[-\infty,\infty]}$, and $\bar{u}(0) \in \mathbb{R}_{[0,V_l]}$. Here, $\bar{x}(k)$ and $\bar{u}(k)$ correspond to the longitudinal displacement and velocity, respectively, while $\bar{\mathbb{U}}(\bar{u}(k-1), T_s)$ encompasses the actuator limits.

The central controller is posed as a mixed-integer MPC (MI-MPC) that outputs the binary decision $\mathcal{D}(k+1)$ and the control input $v_0(k+1)$, at time instant k. The goal is to maximize the velocity of the ego vehicle, minimize the time spent in the adjacent lane and penalize abrupt changes in velocity while satisfying vehicle limits and safety constraints. The optimization problem is

defined as follows:

$$\min_{\substack{\bar{u}^k(1),\cdots,\bar{u}^k(H);\\ \mathcal{D}^k(1),\cdots,\mathcal{D}^k(H)}} \quad \sum_{j=1}^{H} \left[-\gamma_1 \cdot \bar{u}^k(j) + \gamma_2 \cdot \mathcal{D}^k(j) + \gamma_3 \cdot (\bar{u}^k(j) - \bar{u}^k(j-1))^2\right] \tag{2.16}$$

$$\text{s.t.} \quad \bar{x}^k(0) = 0 \tag{2.17}$$

$$\bar{u}^k(0) = \bar{u}(k) \tag{2.18}$$

$$\forall j \in \{1, \cdots, H\}:$$

$$\bar{x}^k(j+1) = \bar{x}^k(j) + \bar{u}^k(j) \cdot T_s \tag{2.19}$$

$$\bar{x}^k(j) \in \mathbb{S}(j) \tag{2.20}$$

$$\bar{u}^k(j) \in \bar{\mathbb{U}}(\bar{u}^k(j-1), T_s) \tag{2.21}$$

$$\mathcal{D}^k(j) \in \{0, 1\} \tag{2.22}$$

In the formulated optimization objective (2.16), the tradeoff parameters γ_1, γ_2, and γ_3 govern the tradeoff between maximizing velocity, minimizing time spent in the adjacent lane, and minimizing abrupt changes in velocity between consecutive time steps. Increasing parameter γ_1 yields a more aggressive behavior with a higher emphasis placed on achieving maximum velocity at the expense of driver comfort while parameters γ_2 and γ_3 emphasize driver comfort by reducing lane and velocity changes, respectively, while sacrificing velocity gains. The output of the optimization at time instant k is $\{\bar{u}_*^k(1), \cdots, \bar{u}_*^k(H), \mathcal{D}_*^k(1), \cdots, \mathcal{D}_*^k(H)\}$ which is applied in a receding horizon fashion.

Safety Constraints: At any planning instant j, the ego vehicle needs to maintain vehicle-dependent longitudinal safety margins to the vehicles traveling in its lane. This is compactly represented as follows:

$$\mathbb{S}_0(j) = \{z \in \mathbb{R} \mid (1 - \mathcal{D}^k(j)) \cdot (|\hat{s}_i(j) - z| - (L_{c_i} + L_{sm_i}(k))) \geq 0, \forall i \in \mathbb{N}_{[1,n_0]} \ni x_i \in \mathbb{O}(k)\} \tag{2.23}$$

$$\mathbb{S}_1(j) = \{z \in \mathbb{R} \mid \mathcal{D}^k(j) \cdot (|\hat{s}_i(j) - z| - (L_{c_i} + L_{sm_i}(k))) \geq 0, \forall i \in \mathbb{N}_{[n_0+1,n]} \ni x_i \in \mathbb{O}(k)\} \tag{2.24}$$

$$\mathbb{S}(j) = \mathbb{S}_0(j) \cup \mathbb{S}_1(j) \tag{2.25}$$

Here, the safe set $\mathbb{S}(j)$ represents the set of longitudinal coordinates deemed safe for the ego vehicle to be in at planning instant j and $L_{sm_i}(k)$ corresponds to the longitudinal safety margin that the ego vehicle needs to maintain from vehicle i for the entirety of the planning horizon at time instant k, which is defined as follows:

$$L_{sm_i}(k) = L_{0_i} + \frac{L_{v_i}}{V_l} \cdot \tilde{u}_i(k) + \frac{L_{a_i}}{A_{max}} \cdot \frac{|\tilde{u}_i(k) - \tilde{u}_i(k-1)|}{T_s} + \mathbb{1}_{i \in \mathbb{N}_{[n_0+1,n]}} \cdot \frac{L_{l_i}}{V_l} \cdot (\bar{u}(k) + \tilde{u}_i(k)) \tag{2.26}$$

The safety margin $L_{sm_i}(k) \in \mathbb{R}_{>0}$ for vehicle i at time instant k depends on its estimated control input (longitudinal velocity, Section 2.3.2.1); the change in its estimated control input between sampling time steps (longitudinal acceleration); and the summation of its estimated control input with the ego vehicle's own control input (relative longitudinal velocity) for the oncoming vehicles. Here, $L_{0_i} \in \mathbb{R}_{>0}$ corresponds to the minimum nominal safety margin that needs to be maintained

regardless of the behavior of vehicle i while $L_{v_i} \in \mathbb{R}_{>0}$, $L_{a_i} \in \mathbb{R}_{>0}$ and $L_{l_i} \in \mathbb{R}_{>0}$ correspond respectively to the multiplicative factors associated with velocity, acceleration and lane of vehicle i. The safety margin parameters are all vehicle dependent since different types of vehicles (cars, trucks, bikes, etc.) require different nominal safety margins and multiplicative factors. The modification of the safety margin with the behavior of the corresponding vehicle allows the controller to perform optimally regardless of varying driver patterns, as validated in Section 2.3.3.2.

Remark 1. *In contrast to the existing work [31, 32], the proposed approach estimates the driving behavior of a vehicle from observations rather than having it as a user-defined parameter, which results in a reactive control strategy. This allows for a change in estimated behavior over time, which is necessary since other vehicles are expected to react to the ego vehicle's actions.*

Remark 2. *The safety margins (L_{0_i}, L_{v_i}, L_{a_i} and L_{l_i}) are fine-tuned empirically based on the ego vehicle's dynamical constraints and the operational design domain (ODD) specifications, allowing for varying level of defensiveness or aggressiveness, as demonstrated in Section 2.3.3.2.*

Remark 3. *The decision-making system does not explicitly account for the time required by the lateral controller to execute its desired decision. This is by design since the ego vehicle may be able to observe the initially occluded vehicle(s) by nudging into the adjacent lane and potentially retracting its decision thereafter without having to move all the way to the center of the adjacent lane. Moreover, this also allows for the decision-making system to be made completely independent of the choice of the lateral controller, as long as L_{0_i} has been tuned appropriately (i.e. chosen large enough to accommodate the longitudinal distance covered while changing lanes) for the controller at hand.*

Safety constraint implementation: The safety constraints posed in (2.23) and (2.24) do not adhere to the standard form of a mixed integer quadratic program [42]. In order to convert the constraints into the standard linear form, the big-M method [43] is applied as follows:

$$(2.23) \iff \begin{cases} (\hat{s}_i(j) - z + M \cdot a(j) - (L_{c_i} + L_{sm_i}(k))) + N_1 \cdot \mathcal{D}^k(j) \geq 0 & \text{(2.27a)} \\ -(\hat{s}_i(j) - z - M \cdot (1 - a(j)) + (L_{c_i} + L_{sm_i}(k))) + N_1 \cdot \mathcal{D}^k(j) \geq 0 & \text{(2.27b)} \end{cases}$$

$$(2.24) \iff \begin{cases} (\hat{s}_i(j) - z + M \cdot b(j) - (L_{c_i} + L_{sm_i}(k))) + N_2 \cdot (1 - \mathcal{D}^k(j)) \geq 0 & \text{(2.28a)} \\ -(\hat{s}_i(j) - z - M \cdot (1 - b(j)) + (L_{c_i} + L_{sm_i}(k))) + N_2 \cdot (1 - \mathcal{D}^k(j)) \geq 0 & \text{(2.28b)} \end{cases}$$

where $M, N_1, N_2 \gg 0$, and $a(k), b(k) \in \{0, 1\}$. The constants N_1 and N_2 allow for automatic satisfaction of the inactive constraints out of (2.23) and (2.24), based on the value of $\mathcal{D}^k(j)$, thus removing the quadratic terms. The constant M, in conjunction with the boolean variables $a(k)$ and $b(k)$ which are responsible for accommodating the sign of the absolute value term, allows for the transformation of the absolute value constraint into two linear ones. Therefore, the linear constraints (2.27a)-(2.28b) are used as safety constraints in the implementation of the central controller.

Lateral Controller: Our approach can work in conjunction with any lane-changing model and lateral controller found in the literature [44–46], given that the low-level controller does not significantly alter the longitudinal dynamics. An example of such a decoupled lateral controller is provided in [28]. In this work, without loss of generality, we make use of a simple yet sufficiently effective lateral controller (see Section 2.3.3) represented by the filtering of the decision signal

$\mathcal{D}(k)$ by a moving average (FIR) low-pass filter with a given window size N, as follows:

$$d_0(k) = \frac{L_r}{N} \sum_{n=0}^{N-1} \mathcal{D}(k-n) - \frac{L_r}{2}. \tag{2.29}$$

2.3.2.3 Feasibility Analysis

Theorem 1. *Suppose that* $L_o > \dfrac{2V_{max}^2}{-A_{min}} + L_{sm}^{max}$, *where* L_{sm}^{max} *is the maximum possible safety margin, based on the chosen parameters, and* $\forall i \in \mathbb{N}_{[0,n]}$, $\tilde{n}_{v_i}(k) = \tilde{n}_{s_i}(k) = 0$, $\tilde{u}_i(k) = v_i(k) = v_i(0)$. *If* $s_i(0)$ *is such that the problem (Section 2.3.2.2) is initially feasible at* $k = 0$, *with* $\mathcal{D}_0^*(1) = 0$, *then under the assumption of self-preserving agents [47], it remains feasible for all* $k \geq 0$.

Proof. Given the conditions of *Theorem 1*, a feasible solution that holds for all $k > 0$, in the presence of a leading vehicle, is as follows:

$$\bar{u}^k(j) = \begin{cases} \tilde{u}_a(k), & \Delta u(j) \geq A_{min} \cdot Ts \\ & \Delta u(j) \leq A_{max} \cdot Ts \\ \bar{u}^k(j) + A_{max} \cdot Ts, & \Delta u(j) \geq A_{max} \cdot Ts \\ \bar{u}^k(j) - A_{min} \cdot Ts, & \Delta u(j) \leq A_{min} \cdot Ts \end{cases} \tag{2.30}$$

$$\mathcal{D}^k(j) = 0 \tag{2.31}$$

for all $j \in \{1, \cdots, H\}$, where $\Delta u(j) = \tilde{u}_a(k) - \bar{u}^k(j)$, $a \in \mathbb{N}_{[1,n]}$ such that $x_a(0) \in \mathbb{O}_0(0)$. □

Remark 4. *The conditions in Theorem 1 are required for completeness purposes since there exist*

scenarios such as ill-posed initialization or relying on faulty sensors, resulting in $|\tilde{n}_{s_i}(k)| \gg 0$ *(measurements dominated by noise) or* $L_s \approx 0$ *(no visibility), that may lead to infeasibility. An example of an ill-posed initialization that leads to infeasibility is as follows: Let* $s_i(0)$ *be such that* $\mathcal{D}_0^*(1) = 1$ *(initial decision to move to adjacent lane), coupled with* $n_1 \to \infty$ *(high-density traffic in original lane) and* $n_0 > 0$ *(at least one vehicle in adjacent lane). In this case, the problem stays infeasible and unavoidably results in a collision. However, for appropriately defined realistic constants, the problem with well-tuned hyperparameters is expected to stay feasible at all times.*

2.3.3 Experimental Results

The performance of the proposed approach is evaluated in a bidirectional closed-loop road simulation which facilitates continuous rerouting of traffic, enabling us to evaluate the long-term behavior of the approach. This is an important consideration, especially in the case of high-density traffic, since a long-duration simulation can reveal corner cases that may not appear in a short-term simulation of a single overtake maneuver that is often utilized in existing works [31, 32, 44].

The simulation is implemented in the SUMO environment [48], depicted in Figure 2.4. The Gurobi Optimizer (Version 9.1.1) [49] is used to solve the mixed-integer quadratic programming (MIQP) problem posed in Section 2.3.2.2, at every sampled time instant T_s. The controller communicates with the simulator using SUMO's built-in traffic controller interface (TraCI). The simulation and optimization algorithms are both implemented on a personal computer equipped with an Intel i7-4710HQ CPU with 16GB of RAM running Ubuntu 18.04. We note that the average time required for each optimization step is $11.246\,ms$ with a standard deviation of $0.233\,ms$. For

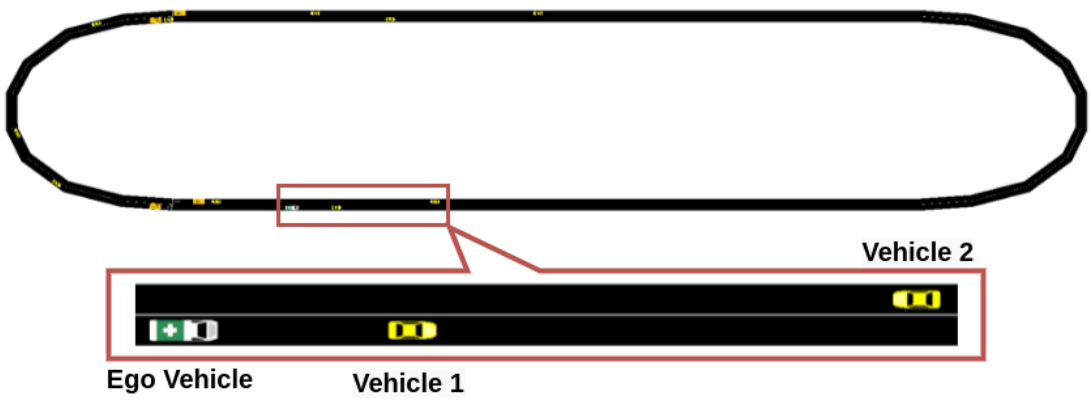

Figure 2.4: **SUMO Simulation Environment**. The experiments are carried out on a closed bidirectional highway loop that allows for the continuous rerouting of vehicles, enabling long-term behavior evaluation. On the road, the ego vehicle is depicted in green while the HDVs are shown in yellow.

reference, the reaction time of a human driver is $2.3\,s$ [50], and a vehicle with a speed of $60\,m/h$ covers $0.2682\,m$ in $10\,ms$. This suggests that the proposed approach is suitable for real-time use in real-world scenarios. Note that the running time of about 10ms is achieved on an average personal computer and can be further reduced by the use of dedicated hardware.

2.3.3.1 Simulated Trajectories

The behavior of the ego vehicle during an overtake scenario involving five vehicles is illustrated in Figure 2.5. The simulation and the controller parameters are outlined in Table 2.1. The ego vehicle can initially detect two vehicles in its sensing range: a leading vehicle in the same lane (Vehicle 1) and an oncoming vehicle in the adjacent lane (Vehicle 3). Note that the initially occluded (depicted in a lighter shade) Vehicle 2 comes into the ego vehicle's sensing range after initiating the overtaking maneuver. After successfully overtaking Vehicle 1, the ego vehicle is able to sense another leading vehicle (Vehicle 2) while still having Vehicle 3 in its sensing range. After

Simulation Parameters	**Value**
Simulation step size	100 ms
Simulation duration	1 hour
Road length	1 km
Road speed limit (V_l)	20 m/s
Other vehicles' speed ($v_i(k) : \forall i \neq 0, \forall k$)	10 m/s
Vehicles' length ($L_{ci} : \forall i$)	5 m
Vehicles' width ($W_{ci} : \forall i$)	2.16 m
Controller Parameters	**Value**
Controller sampling time (T_s)	500 ms
Maximum acceleration (A_{max})	6 m/s^2
Maximum deceleration (A_{min})	-9 m/s^2
Maximum velocity (V_{max})	30 m/s
Normal sensing range (L_s)	150 m
Occluded sensing range (L_o)	75 m
Planning horizon (H)	10 s
Safety Margin Parameters ($[L_{0_i}, L_{v_i}, L_{a_i}, L_{l_i}]$)	[10, 5, 5, 10]

Table 2.1: Simulation & Controller Parameters for Non-Cooperative Overtaking

waiting for Vehicle 3 to pass, the ego vehicle starts another overtake maneuver. After initiating the maneuver, an additional oncoming vehicle in the adjacent lane (Vehicle 4) enters the ego vehicle's sensing range (un-occluded region), and the overtake decision is retracted.

This result highlights the importance of active perception in autonomous driving. That is, for a safe controller to be used in real-world settings, perception, and control cannot be assumed independent. Rather, taking an action, e.g., starting the process of overtaking, can result in better sensory input which should be used to alter the decision of the vehicle online. One important detail, which is often overlooked, is that the action that leads to new observations, such as the one presented above, must be safe. This is, therefore, different from the way reinforcement learning algorithms are trained [25] and is availed to us by the MPC approach, detailed in Section 2.3.2.2.

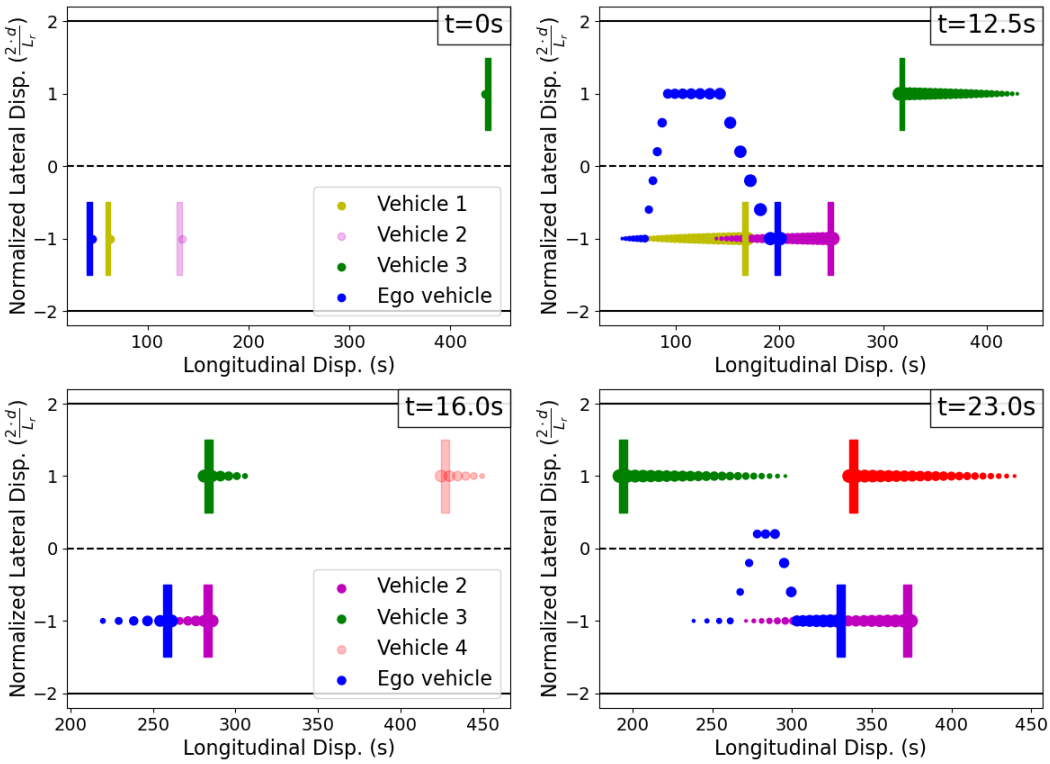

Figure 2.5: **Vehicle Trajectories**. For the ego vehicle (shown in blue), the leading vehicles are Vehicle 1 (shown in yellow) and Vehicle 2 (shown in purple), while the oncoming vehicles are Vehicle 3 (shown in green) and Vehicle 4 (shown in red). The lighter shade for a vehicle at a given time instant alludes to its invisibility from the ego vehicle's perspective due to occlusion. The ego vehicle successfully completes the overtake of Vehicle 1 at $t = 12.5s$ but fails to overtake Vehicle 2 at $t = 23.0s$ due to the initially occluded Vehicle 4.

2.3.3.2 Performance Metrics

We analyze the performance of the proposed approach by considering the effects of different parameters on the following performance metrics, computed over a long-term simulation in the closed loop environment shown in Figure 2.4:

(i) Average velocity M_v;

(ii) Average change in velocity M_j;

(iii) Time spent in the adjacent lane M_t;

(iv) Number of overtake maneuvers initiated M_o;

(v) Success percentage of overtake maneuvers M_s.

The goal of the ego vehicle is to maximize the *average velocity* while minimizing the *time spent in the adjacent lane* and *average change in velocity between time steps*, which is used as a metric to evaluate driver comfort. The *number of overtake maneuvers initiated*, and their *success percentage*, are utilized as measures of aggressiveness of the ego vehicle.

High-level Optimization Parameters: The behavior of the ego vehicle depends on the tradeoff parameters γ_1, γ_2, and γ_3 in the objective function (2.16). The effects of parameters γ_1 and γ_3 on the metrics M_v, M_s, and M_t are illustrated in Figure 2.6. The parameter γ_1 governs the weightage of the algorithm on maximizing ego vehicle's velocity. Increasing γ_1 up to a certain threshold yields an increase in average velocity and overtaking success. Beyond this point, however, further increase in the value of γ_1 results in overly aggressive behavior, where the ego vehicle tries to initiate an overtake maneuver at every chance it gets, resulting in a decrease in overtaking success, an increase in the time spent in the adjacent lane, and a decrease in the average velocity. In contrast to γ_1, the parameter γ_3 reflects the driver's comfort by restricting abrupt changes in velocity. As expected, increasing γ_3 results in lower average velocity, lower overtaking success, and greater time spent in the adjacent lane.

HDVs' Level of Aggressiveness: HDVs' driving imperfections can be modeled by the $Sigma(\sigma)$ parameter in SUMO's Krauss Car-following model [51], and when coupled with varying accelera-

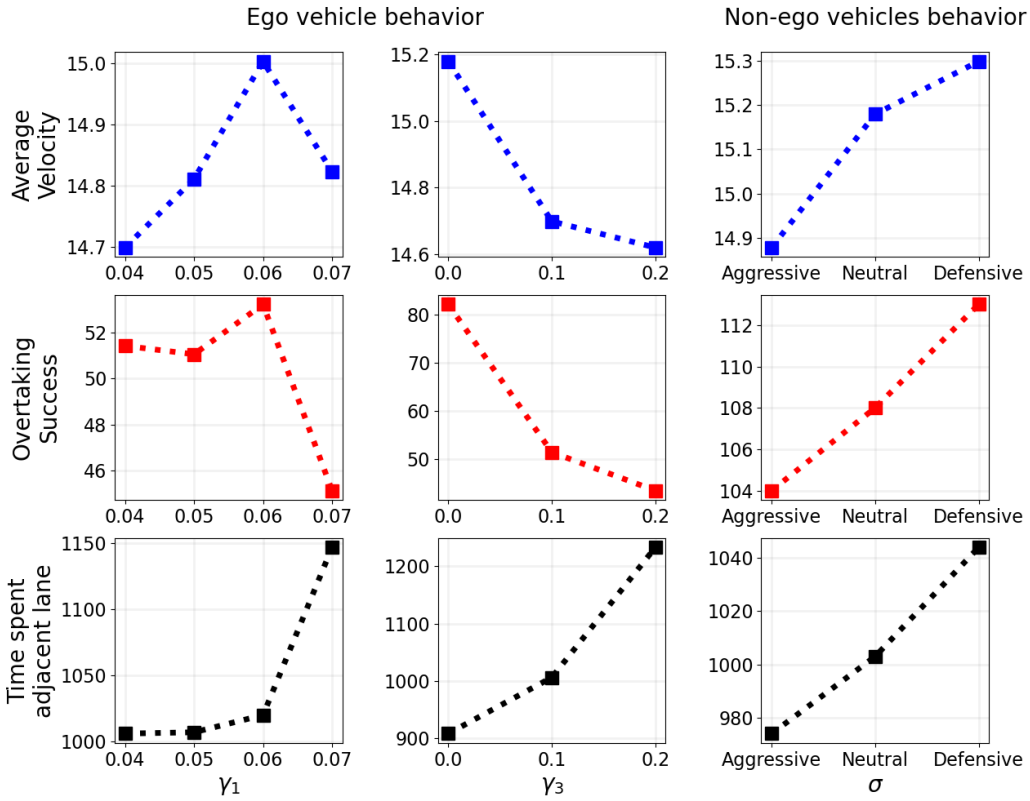

Figure 2.6: **Algorithm Performance w.r.t. Varying Behaviors of Ego and Non-ego Vehicles**. The y-axes for the top, center, and bottom plots correspond respectively to average velocity, overtaking success, and time spent in the adjacent lane. The x-axes for the left, center, and right plots correspond respectively to γ_1, γ_3, and σ. γ_1 and γ_3 control the ego vehicle's behavior while σ governs the non-ego vehicles' behavior.

tion and deceleration limits, it gives rise to varying levels of aggressiveness of the non-ego vehicles, as depicted in Figure 2.6. In response to aggressive behavior from non-ego vehicles, the ego vehicle attempts fewer overtake maneuvers while spending less time overall in the adjacent lane, resulting in a lower average velocity. In response to defensive behavior from non-ego vehicles, the ego vehicle attempts more overtake maneuvers while spending more time in the adjacent lane, resulting in a higher average velocity.

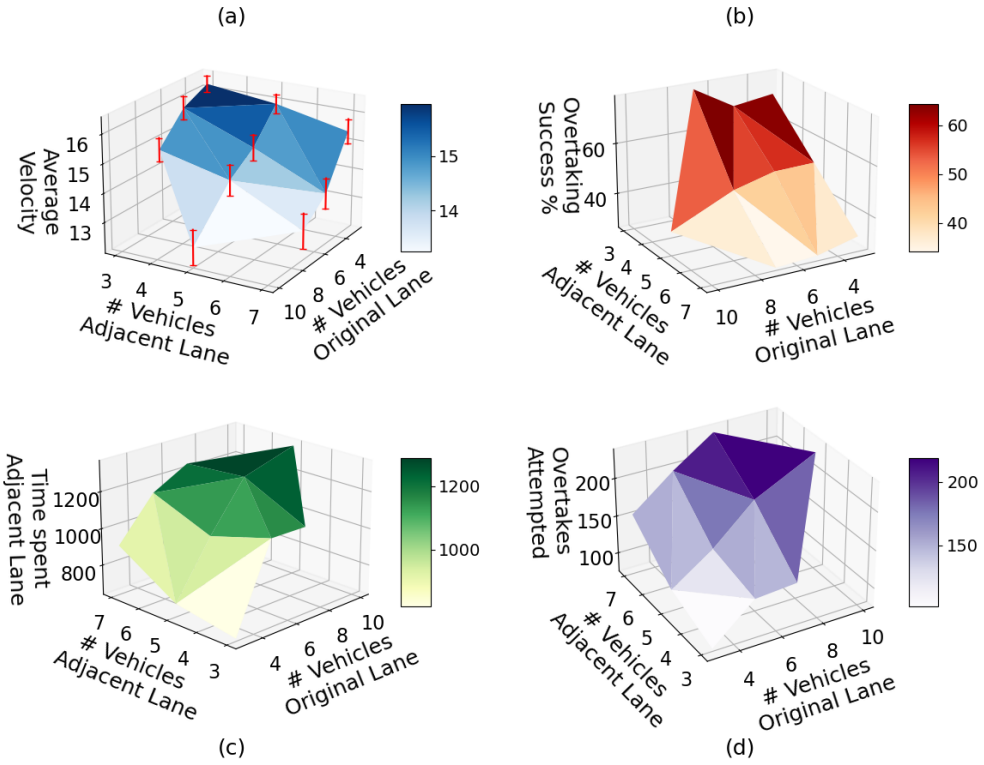

Figure 2.7: **Algorithm Performance w.r.t. Varying Traffic Density**. The z-axes for the top-left (a), top-right (b), bottom-left (c), and bottom-right (d) plots correspond respectively to average velocity, overtaking success, time spent in the adjacent lane, and overtakes attempted. The x-axes and y-axes correspond respectively to the number of vehicles in original and adjacent lanes while the color bars are shown only for better visualization of the gradient. The error bars in the top-left plot (a) represent the variations in velocity.

Traffic density: Finally, the behavior of the algorithm in varying traffic densities is summarized in Figure 2.7. As the number of vehicles in both lanes decreases, the average velocity increases, and the average velocity fluctuation, represented by the red error bars in Figure 2.7(a), decreases. These trends seem more pronounced in regards to the vehicles in the original lane because, with a lower number of leading vehicles, the ego vehicle has to execute fewer overtake maneuvers. With decreasing traffic density on the road, the time spent in the adjacent lane decreases (Figure 2.7(c)), the overtake attempts decrease (Figure 2.7(d)), and their success rate increases (Figure 2.7(b)).

Note that none of the experiments resulted in a collision.

2.3.3.3 Comparison with Existing Methods

The discussion on the performance of overtaking methods found in the literature often ends with showing that a single overtake maneuver can be successfully performed [13, 24, 31, 32]. In contrast, we have demonstrated the feasibility of the proposed approach, without assuming global knowledge, in theory and through simulations. In addition, we have further analyzed the performance of the proposed approach by considering different performance metrics, as explained in Section 2.3.3.

2.4 Cooperative Overtaking

The promise of increased safety, efficiency, and ease of access are the key motivations in the development and introduction of connected autonomous vehicles (CAVs) into modern road networks. Apart from the behavior unpredictability of the human-driven vehicles (HDVs), which was effectively addressed in Section 2.3, another complication involved in the bidirectional overtaking problem is the low visibility caused by sensor occlusion - a situation in which neighboring vehicles block the line-of-sight view of vehicles ahead and in other lanes. Due to this, there are several blind spots in an autonomous vehicle's perception of its environment. This then leads to either overly conservative behavior which reduces efficiency or highly risky maneuvers which may lead to increased collisions. The solution to this problem involves finding ways in which to bridge these gaps and fill in the missing information in a CAVs field of vision. To this effect, we explore the possibility of using vehicle-to-vehicle (V2V) communication between CAVs to share local information about neighboring tracked vehicles. The onboard sensor suite on CAVs is capable of detecting and tracking the dynamics of their immediate neighboring vehicles. This information can then be shared with other CAVs within the communication range. This means that in addition to its own onboard sensors, CAVs can collect information about other on-road vehicles and obstacles by communicating with CAVs downstream of their location.

Now, in the second part, we extend our previously discussed planning and control framework to the cooperative case, where we benefit from the information interchange between autonomous agents using V2V communications to mitigate the challenges posed by sensor occlusion. We do this by incorporating an improved trajectory prediction model and cooperative sensing to leverage

the collective information obtained from other vehicles, thus enriching the perception capabilities of our system. We then evaluate the benefits of the enhanced receding horizon framework in the presence of varying degrees of CAVs within the environment.

2.4.1 Problem Description

In this section, we model the road infrastructure used, the characteristics of CAVs and HDVs, and the proposed sensor occlusion model. Throughout the remainder of this chapter, $\mathbb{V}(k)$ will denote the set of all the vehicles in the simulation at time k, and $\mathbb{A}(k)$ will denote the set of all the CAVs in the simulation at time k.

2.4.1.1 Road Model

The physical road structure is modeled as a long continuous road segment with two adjacent lanes, each with traffic flow in opposite directions. We set the length of this road to 2 km, which is a parameter that can be modified. We denote the speed limit of the road segment as $\bar{v}$ and compute distances in the Frenet coordinate system; the distance along the road is defined as the longitudinal displacement and the distance perpendicular to the road is defined as lateral displacement. It is also possible to change the density of vehicles on this road segment in either direction as necessary. These vehicles take the form of either CAVs or HDVs, and we can change the ratio of CAVs to HDVs (CAV penetration level) in the simulation. A section of this simulated road is shown in Figure 2.8, in which lane structure, HDVs, CAVs, and CAV sensor ranges are highlighted.

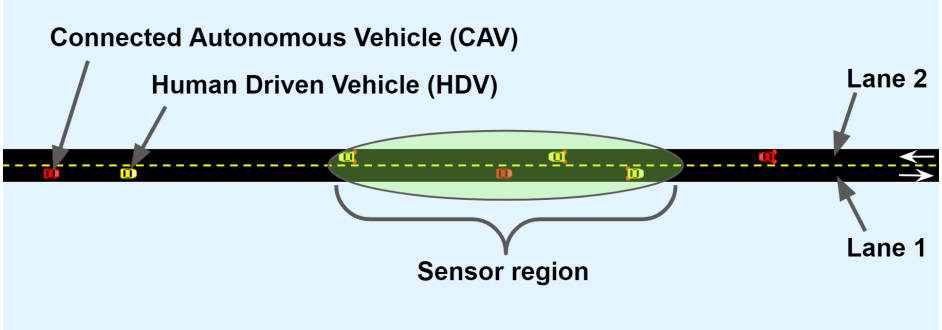

Figure 2.8: **Road Model**. The clearly demarcated lanes on the road section separate the traffic flow in opposite directions. The CAVs and HDVs are depicted in red and yellow respectively while a CAV's sensing region is highlighted in green.

2.4.1.2 Vehicle Dynamics

Two different types of vehicles are used in this research: Human-driven vehicles that are modeled in Section 2.4.1.5, and CAVs that are modeled in the ensuing discussion. We assume that each CAV has onboard a low-level local controller c_i which is capable of computing the necessary throttle and braking actuation commands in order to execute high-level velocity goals as well as compute the steering commands that control the vehicles' lateral motion in order to keep the vehicle in lane. Therefore, for the proposed high-level controller, we find it unnecessary to consider the highly non-linear dynamics of real-world vehicles. This allows the i^{th} vehicle to be modeled as a point object moving along the center of the lane according to the non-linear differential equation:

$$\dot{s_i} = f(t, s_i, c_i), \qquad s_i(t_i^0) = s_i^0 \tag{2.32}$$

where t_i^0 is the initial time at which the i^{th} vehicle enters the road segment. Therefore, we can define the high-level discretized and linearized longitudinal vehicle dynamics by the following

velocity control scheme:

$$\begin{aligned} s_i(k) &= s_i(k-1) + \frac{v_i(k-1) + v_i(k)}{2} \cdot T_s \\ v_i(k) &= u_i(k) \end{aligned} \tag{2.33}$$

where T_s denotes the sampling time while $s_i(k)$, $v_i(k)$, and $u_i(k)$ respectively denote the longitudinal displacement, velocity, and applied control of each vehicle i for $i \in \mathbb{A}(k) = \{1, \ldots, n(k)\}$. Here, $\mathbb{A}(k)$ represents the set of CAVs on the modeled highway stretch at time instant k, and the total number of CAVs is denoted by $n(k)$. It is important to note that the velocity control requested by the high-level controller should be reachable by the low-level vehicle controller c_i in the system (2.32). Additionally, as the control applied by the high-level controller is independent of the lane the CAV is in, we also assume that the lane-changing procedures are handled by a separate lane-change controller. The input to the lane-changing controller would be a high-level goal indicating which lane to change into and what time to begin the lane change. For the overtaking case, the desired maneuver can be seen as consisting of two sequential lane changes at calculated times.

At a time instant k, each CAV i is assigned an integer variable $l_i(k) \in \{0, 1\}$ which denotes the lane it is currently on (here, 0 and 1 represent the original lane and oncoming lane respectively). It also has a vehicle length L_i^e parameter and has bounds on its maximum linear acceleration A_i^{max} and maximum deceleration A_i^{min} capabilities. Therefore, the CAVs are defined by the following state vectors:

$$X_i(k) = [s_i(k), v_i(k), l_i(k), L_i^e, A_i^{max}, A_i^{min}]^{\mathrm{T}} \tag{2.34}$$

Remark 5. *For CAV $i \in \mathbb{A}(k)$, we define the sets $\mathbb{V}_0^i(k)$ and $\mathbb{V}_1^i(k)$ as the set of all the vehicles in*

the CAV's original ($l_i(k)$=0) and oncoming ($l_i(k)$=1) lanes respectively.

2.4.1.3 Sensing Model

In terms of the sensing capabilities of the CAVs, each CAV is assumed to have the minimum required onboard sensing capability to detect the positions and velocities of surrounding vehicles within a realistic sensing range. Each CAV is assumed to be capable of tracking the positions of up to five adjacent non-occluded vehicles surrounding it. These vehicles involve the ego vehicle's leader and follower in its own lane as well as a maximum of three vehicles in the oncoming lane. In practice, the actual number of vehicles tracked may be lower due to sensor occlusion and the density of vehicles on the road. A scenario where the ego vehicle is tracking four surrounding vehicles is highlighted in Figure 2.9.

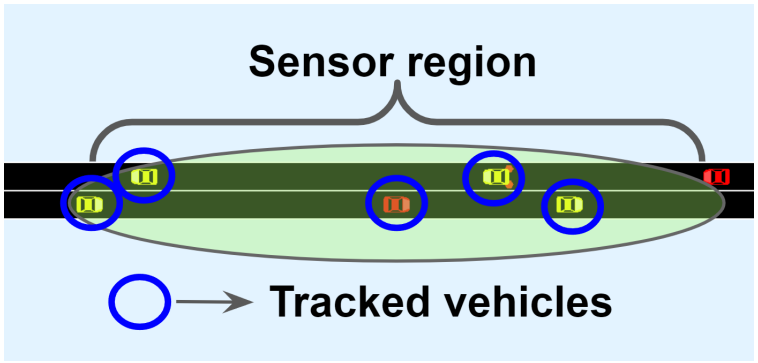

Figure 2.9: **Sensing Model**. The sensing region of a CAV (in red) is highlighted in green, while the tracked vehicles (in yellow) are identified by the enclosing blue circles.

This assumption of tracking surrounding vehicles is justified by the fact that modern CAVs have an advanced sensor suite that allows them to track the relative displacement of nearby line-of-sight obstacles/vehicles with very high accuracy. This data can then be used to compute the instantaneous velocities of surrounding vehicles with a low margin of error. The main features of

a neighboring vehicle collected by each CAV are the vehicle's longitudinal position, current lane, current velocity, and a memory of past velocities.

In addition to the onboard sensor suite, the other important source of information available to the ego vehicle is obtained in the form of V2V communication with other CAVs. Once each CAV uses its onboard sensors to collect the features of neighboring vehicles, it can then communicate this information with other CAVs within its communication range. Therefore, even if a vehicle is occluded to the ego vehicle, as long as this vehicle is visible to a downstream CAV, the ego vehicle can collect the information it needs for safe and effective trajectory planning. When considering the V2V communication capabilities of each vehicle, we assume that the CAVs communicate using a combination of IEEE 802.11p and 5G networks. Additionally, in this research, we do not consider the impact of network delay and packet loss during transmission. Therefore, we assume that vehicles within a realistic communication range of each other can share information in real-time.

2.4.1.4 Occlusion Model

With the sensor occlusion model formulation, we aim to capture the effects of neighboring vehicles on the visibility range of a CAV. If there are no neighboring/blocking vehicles close to the ego vehicle, the visibility region is assumed to extend up to a fixed maximum value L_s, defined as the sensor range. However, not all vehicles within this range are visible to the CAV due to occlusion. The vehicle immediately ahead of the CAV will block the vehicles further up. Additionally, depending on the proximity of the leading vehicle to the ego vehicle, visibility will also be reduced

in the oncoming lane. The resultant visible regions are shown in Figure 2.10.

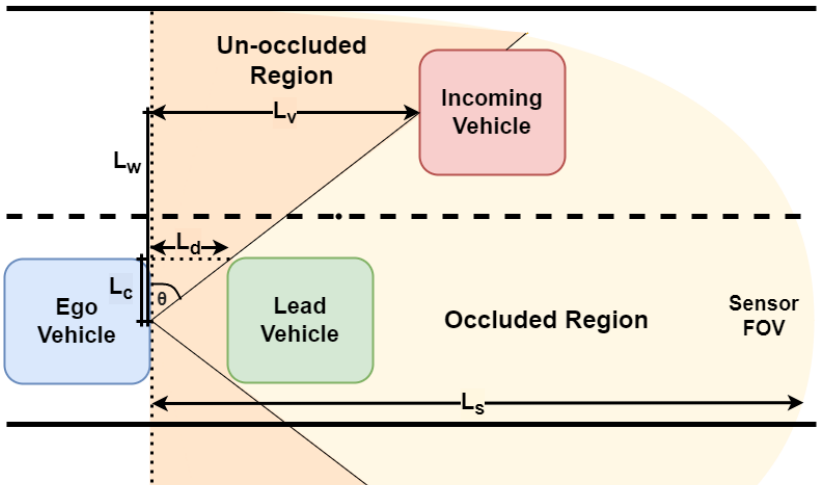

Figure 2.10: **Enhanced Occlusion Model**. Due to the presence of the leading vehicle, the ego vehicle's visibility range is restricted such that the oncoming vehicle falls within the occluded region. This occluded region is characterized by the proximity of the leading vehicle to the ego vehicle.

Here, $L_d^i(k)$, L_c, L_w and $L_v^i(k)$ represent the distance gap to the leading vehicle, half the average width of a vehicle, the lane width, and the resulting un-occluded visible range in the adjacent lane respectively. These variables are connected as in (2.37).

$$z_p^i(k) = s_p(k) - s_i(k), \quad (p, i \in \mathbb{V}(k);\ p \neq i) \tag{2.35}$$

$$L_d^i(k) = \min_{p \in \mathbb{V}(k)} \{\{z_p^i(k), L_s\} \mid l_p(k) = l_i(k)\} \tag{2.36}$$

$$\begin{aligned} tan\theta &= \frac{L_d^i(k)}{L_c} = \frac{L_v^i(k)}{L_w} \\ L_v^i(k) &= \frac{L_d^i(k) \cdot L_w}{L_c} \end{aligned} \tag{2.37}$$

Given that L_c and L_w do not change with time, we can compute the instantaneous adjacent

lane visibility range as a function of the distance gap to the leading vehicle. Vehicles present in this visibility region can be tracked by the ego vehicle and are added to the observation state of the ego vehicle.

$$\mathbb{O}_i^0(k) = \{p \in \mathbb{V}(k) \mid l_p(k) = l_i(k), 0 \leq z_p^i(k) \leq L_d^i(k)\} \tag{2.38a}$$

$$\mathbb{O}_i^1(k) = \{p \in \mathbb{V}(k) \mid l_p(k) \neq l_i(k), 0 \leq z_p^i(k) \leq L_v(k)\} \tag{2.38b}$$

$$\mathbb{O}_i(k) = \mathbb{O}_i^0(k) \cup \mathbb{O}_i^1(k) \tag{2.38c}$$

For an ego CAV $i \in \mathbb{A}(k)$, the set of observed vehicles (2.38c) at time instant k, is denoted by $\mathbb{O}_i(k)$. The CAV i can observe leading vehicles traveling ahead in its same lane (2.38a) as long as the leading vehicle is within the measurement range L_s. The CAV i can observe vehicles traveling in its adjacent lane (2.38b), within the un-occluded visible region, $L_v(k)$ (2.37).

2.4.1.5 Traffic Model

The process of modeling human driving behavior in simulation usually involves two separate models: A car-following model is used to compute the safe following velocity of a vehicle, considering its dynamic constraints and its interactions with the leading vehicle, while a lane change model is used to determine when to change lane and the parameters necessary for a safe lane change maneuver. While there are many car-following models such as the Krauss model [51] and the Intelligent Driver Model (IDM) [52], we opt to use the Krauss model for its accuracy and simplicity. This model computes the safe following speed $v_s(t)$ by considering the impact of speed limits $\bar{v}$, vehicle acceleration capabilities a_{max}, the vehicle deceleration profile $b(v(t))$, distance

gap $\Delta s(t)$ and speed $v_l(t)$ of the leading vehicle, time step Δt and driver reaction time τ_r as shown in equation (2.39). The final output speed provided to the vehicle will then have a zero mean Gaussian noise added to it to model the imperfections in human driving. For the lane change model we use the Erdmann [53] model, which allows for the tuning of each vehicle's lane-changing behavior.

$$v_s(t) = \min \left\{ \bar{v}, v(t) + a_{max}\Delta t, v_l(t) + \frac{\Delta s(t) - v_l(t)\tau_r}{\frac{v(t)}{b(v(t))} + \tau_r} \right\} \tag{2.39}$$

2.4.2 Methods and Procedures

In this section, we describe the state estimation algorithm to estimate the states of the observed vehicles, the prediction model to generate the predicted future trajectories for those vehicles, and the optimal control algorithm that is responsible for the decision-making aspects of each of the CAVs.

2.4.2.1 State Prediction

In the literature, it has been shown that an arctangent function is a good representation of the acceleration model of a car. Therefore, we utilize a piecewise linear approximation of the arctangent function, to model the velocity profile of an observed vehicle. In order to predict the future velocity profile of a vehicle p at time step k, we perform linear regression with mean-squared error on the previously observed velocity data points to obtain the slope ($\bar{m}_p^k$) parameter. We utilize the estimated parameter to project the velocity into the future for a given number of steps, defined as H_a. For the remaining duration of the prediction horizon, we assume the velocity to remain

constant. This is compactly represented in (2.40).

$$\hat{v}_p^k(0) = v_p(k)$$
$$\hat{v}_p^k(j) = \begin{cases} \min\{\bar{v}, \hat{v}_p^k(j-1) + \bar{m}_p^k \cdot j\}, & 0 < j \leq H_a \\ \hat{v}_p^k(j-1), & H_a < j \leq H_p \end{cases} \tag{2.40}$$

Here, $\hat{v}_p^k(j)$ corresponds to the predicted velocity j time steps into the future starting at time instant k of an observed vehicle $p \in \mathbb{O}_i(k)$, H_a corresponds to the acceleration horizon and H_p corresponds to the prediction horizon. Figure 2.11 shows the comparison between (2.40) and the arctangent function.

Each CAV then estimates the relative longitudinal displacement $\hat{s}_p^k(j)$ of the observed vehicles $p \in \mathbb{O}_i(k)$, (with $\hat{s}_p^k(0) = s_p(k)$), using the computed predicted velocities $\hat{v}_p^k(j)$ as follows:

$$\hat{s}_p^k(j) = \hat{s}_p^k(j-1) + \frac{\hat{v}_p^k(j-1) + \hat{v}_p^k(j)}{2} \cdot T_s \tag{2.41}$$

2.4.2.2 State Aggregation

Based on the state prediction model described in 2.4.2.1, we know that each CAV can track a group of vehicles in its 'line of sight' as discussed in 2.4.1.3. Note that the vehicles that can be tracked by a CAV also depend on the sensor occlusion status of the CAV as shown in 2.4.1.4. This tracked vehicle information, including predicted velocities and positions, can then be shared with other CAVs within V2V communication range. Therefore, the ego CAV $i \in \mathbb{A}(k)$ obtains

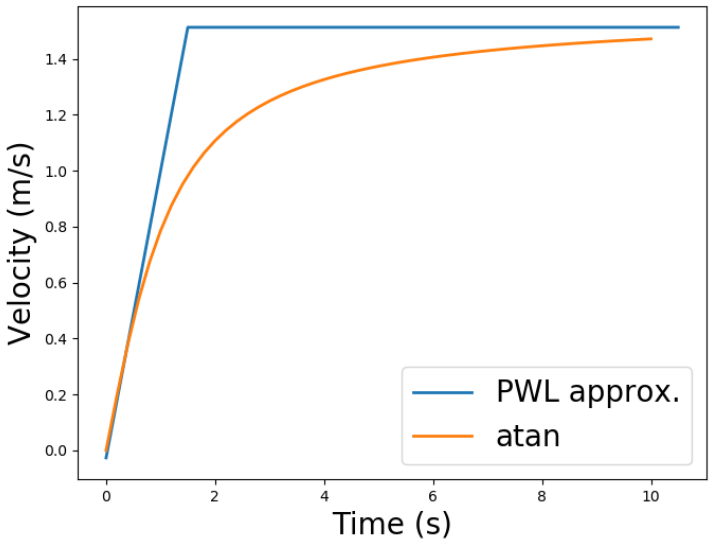

Figure 2.11: **Piecewise Linear Approximation of the Arctangent Function**. The arctangent function is shown in red while its piecewise linear approximation, used for our velocity prediction, is shown in blue.

information about its surrounding vehicles from two sources, its own on-board sensor systems ($\mathbb{O}_i(k)$) and the information communicated to it from other CAVs ($\mathbb{C}_i(k)$).

However, it is important that we do not overload the optimization-based controller with unnecessary information, which may lead to increased computation times. We introduce a sorting step to select which surrounding vehicles are most important for the overtaking problem. These vehicles (set $\mathbb{I}_i(k)$) that are pertinent to the decision-making process of CAV i, can include up to three leading vehicles and one following vehicle on the ego vehicle's own lane as well as up to two leading vehicles and one following vehicle in the oncoming lane. Therefore, the predicted future states of up to six vehicles (set $\mathbb{W}_i(k)$) can be provided to the MI-MPC overtaking controller. The next step is to populate $\mathbb{W}_i(k)$ based on the information availability of vehicles in set $\mathbb{I}_i(k)$.

$$\mathbb{W}_i(k) = \mathbb{I}_i(k) \cap (\mathbb{O}_i(k) \cup \mathbb{C}_i(k)) \tag{2.42}$$

The MI-MPC controller of CAV i will then be provided the predicted states of the vehicles in $\mathbb{W}_i(k)$.

2.4.2.3 Optimal Model Predictive Control

The proposed optimal controller is responsible for computing the sequence of velocity and lane change commands which would allow the ego CAV to maximize its velocity while respecting system dynamics and safety constraints. In this formulation, we also assume that the longitudinal and lateral dynamics of the ego CAV are decoupled [40], as justified by the lower road curvatures present in highway overtaking scenarios and the capability of low-level lane change controllers to perform maneuvers while respecting lateral dynamic constraints [41]. Note that this proposed optimal controller is present on every CAV $i \in \mathbb{A}(k)$ and they all perform their own computations independently.

Objective function: Each CAV $i \in \mathbb{A}(k)$ has an optimal controller, formulated as a mixed-integer model-predictive optimal control problem (MI-MPC), that provides, at any time instant k, the control input $u_i(k+1)$, and the binary overtaking decision $\mathcal{D}_i(k+1)$.

The objective function is formulated as a maximization of the velocity of the ego CAV while minimizing the time spent in the oncoming lane and minimizing abrupt changes in velocity. This is defined as follows:

$$\min_{\substack{u_i^k(1),\cdots,u_i^k(H);\\ \mathcal{D}_i^k(1),\cdots,\mathcal{D}_i^k(H)}} \quad \sum_{j=1}^{H} -\gamma_1 \cdot u_i^k(j) + \gamma_2 \cdot \mathcal{D}_i^k(j) + \gamma_3 \cdot (u_i^k(j) - u_i^k(j-1))^2 \tag{2.43}$$

Here, H denotes the planning horizon. The proposed optimization objective (2.43) contains three trade-off parameters γ_1, γ_2 and γ_3, which handle the trade-off between maximizing velocity, minimizing time spent in the oncoming lane, and minimizing sudden changes in velocity. Increasing γ_1 leads to increased focus on velocity maximization which results in more aggressive overtaking behaviors and a less comfortable experience for passengers. Increasing γ_2 and γ_3 on the other hand, leads to a reduction in risky overtakes while improving the comfort of the passengers, at the cost of increased travel time.

Dynamic constraints: The dynamic constraints ensure that the optimization controller generates reachable controls. At time instant k, the initial values of position, velocity control, and lane control are set as $s_i^k(0) = s_i(k)$, $u_i^k(0) = v_i(k)$, and $\mathcal{D}_i^k(0) = l_i(k)$ respectively. Next, the longitudinal command velocity $u_i^k(j)$, for all $j \in \{1, \ldots, H\}$, is bounded by the speed limit:

$$0 \leq u_i^k(j) \leq \bar{v} \tag{2.44}$$

as well as the acceleration capabilities of each vehicle:

$$A_i^{min} \cdot T_s \leq u_i^k(j) - u_i^k(j-1) \leq A_i^{max} \cdot T_s \tag{2.45}$$

where T_s, as referenced previously, is the time resolution for our computations. Regarding the lateral movement, we confine the time-dependent binary decision variable $\mathcal{D}_i^k(j)$ to $\{0, 1\}$. More specifically, $\mathcal{D}_i^k(j) = 1$ corresponds to the decision to travel in the adjacent lane while $\mathcal{D}_i^k(j) = 0$ represents the decision to travel in the original lane. A difference in the current lane and the binary decision variable (i.e. $\mathcal{D}_i^k(1) \neq l_i(k)$) will trigger a lane-change maneuver at time instant k. We find that this simplification leads to a significant reduction in computational complexity.

Safety constraints: We next introduce the constraints responsible for the prevention of rear-end and lateral collisions. The longitudinal positions ($s_i(k)$) of the CAV are computed with respect to the decoupled longitudinal dynamics model given in equation (2.33).

At any future time step j starting from the time instant k, the ego CAV needs to maintain a safe longitudinal distance to all the known vehicles traveling in its current lane. This requirement can be defined as:

$$(1 - \mathcal{D}_i^k(j)) \cdot (|\hat{s}_p^k(j) - s_i^k(j)| - (L_p^e + M_{s_p}^i(j))) \geq 0, \forall p \in \{\mathbb{V}_0^i(k) \cap \mathbb{W}_i(k)\} \tag{2.46}$$

$$\mathcal{D}_i^k(j) \cdot (|\hat{s}_q^k(j) - s_i^k(j)| - (L_q^e + M_{s_q}^i(j))) \geq 0, \forall q \in \{\mathbb{V}_1^i(k) \cap \mathbb{W}_i(k)\} \tag{2.47}$$

where $\mathbb{V}_0^i(k)$ and $\mathbb{V}_1^i(k)$ correspond respectively to the original lane and oncoming lane vehicle set for the CAV i. Here, equations (2.46) and (2.47) represent the collision prevention constraints in the original and the oncoming lanes respectively. These constraints need to be checked for all time instances in the planning horizon ($\forall j \in \{1, \cdots, H\}$). We define $M_{s_p}^i(j)$, the longitudinal safety margin that the ego CAV $i \in \mathbb{A}(k)$ needs to maintain from the vehicle $p \in \mathbb{V}(k)$ at time instant k,

as follows:

$$M^i_{s_p}(j) = M_{0_p} + \frac{M_{v_p}}{\bar{v}}\hat{v}^k_p(j) + \frac{M_{a_p}}{A^{max}_i} \cdot \frac{|\hat{v}^k_p(j) - \hat{v}^k_p(j-1)|}{T_s} + \mathbb{1}_{\mathbb{V}^i_1(k)}(p) \cdot \frac{M_{l_p}}{\bar{v}} \cdot (v^k_i(j) + \hat{v}^k_p(j)) \quad (2.48)$$

The safety margin $M^i_{s_p}(j) \in \mathbb{R}_{>0}$ for vehicle p depends on the longitudinal velocity, longitudinal acceleration, and the relative longitudinal velocity if p is in the oncoming lane of CAV i ($p \in \mathbb{V}^i_1(k)$). Here, $M_{0_p} \in \mathbb{R}_{>0}$ represents the nominal safety gap that will be maintained regardless of vehicle p's driving behavior. Additionally, $M_{v_p} \in \mathbb{R}_{>0}$, $M_{a_p} \in \mathbb{R}_{>0}$ and $M_{l_p} \in \mathbb{R}_{>0}$ correspond respectively to the multiplicative factors associated with the velocity, acceleration, and lane of vehicle p.

To simplify the optimization process, we pose the safety constraints in the standard linear form, using multiple applications of the big-M method [43]. The optimization constraint (2.46) is then converted to (2.49) and (2.50), while (2.47) is converted to (2.51) and (2.52). This converts the optimization problem into a standard mixed integer quadratic program [42], which is computationally efficient to solve.

$$(\hat{s}^k_p(j) - s^k_i(j) + N_0 \cdot a(j) - (L^e_p + M^i_{s_p}(j))) + N_1 \cdot \mathcal{D}^k_i(j) \geq 0 \quad (2.49)$$

$$-(\hat{s}^k_p(j) - s^k_i(j) - N_0 \cdot (1 - a(j)) + (L^e_p + M^i_{s_p}(j))) + N_1 \cdot \mathcal{D}^k_i(j) \geq 0 \quad (2.50)$$

$$(\hat{s}^k_q(j) - s^k_i(j) + N_0 \cdot b(j) - (L^e_q + M^i_{s_q}(j))) + N_2 \cdot (1 - \mathcal{D}^k_i(j)) \geq 0 \quad (2.51)$$

$$-(\hat{s}^k_q(j) - s^k_i(j) - N_0 \cdot (1 - b(j)) + (L^e_q + M^i_{s_q}(j))) + N_2 \cdot (1 - \mathcal{D}^k_i(j)) \geq 0 \quad (2.52)$$

Here, these constraints are repeated $\forall p \in (\mathbb{V}_0^i(k) \cap \mathbb{W}_i(k))$, $\forall q \in (\mathbb{V}_1^i(k) \cap \mathbb{W}_i(k))$ and $\forall\ j \in \{1, \ldots, H\}$. Also, $N_0, N_1, N_2 \gg 0$, and $a(j), b(j) \in \{0, 1\}$. The constants N_1 and N_2 allow for automatic satisfaction of the inactive constraints based on the value of $\mathcal{D}_i^k(j)$. The constant N_0, in conjunction with the boolean variables $a(j)$ and $b(j)$ allows for the transformation of the absolute relative distance constraints in (2.46) and (2.47) into linear constraints.

The constraints given by equations (2.44), (2.45), (2.49), (2.50), (2.51) and (2.52) along with the objective function in (2.43), form the optimization problem for optimal CAV longitudinal command velocity and lane changing decision computation. The output of the optimization problem at time instant k is $\{u_i^{k*}(1), \cdots, u_i^{k*}(H), \mathcal{D}_i^{k*}(1), \cdots, \mathcal{D}_i^{k*}(H)\}$ which is applied in a receding horizon fashion, i.e., $u_i(k+1) = u_i^{k*}(1)$, and $\mathcal{D}_i(k+1) = \mathcal{D}_i^{k*}(1)$.

Remark 6. *The behavior of the algorithm can be modified by altering the safety margin (M_{0_p}, M_{v_p}, M_{a_p} and M_{l_p}) parameters. In our tests, these parameters are tuned empirically to guarantee safe overtaking behavior whenever overtaking attempts are carried out. Additionally, the proposed optimal controller does not explicitly account for the time taken by the low-level lateral controller to execute its desired maneuver. We note that this simplification does not introduce further safety concerns, given that L_{0_i} has been tuned appropriately (e.g. chosen large enough), and that the ego CAV has the ability to retract a lane-changing decision without having to move all the way to the center of the adjacent lane, which is achieved by the receding horizon framework of MPC.*

Low-level lateral controller: The proposed framework allows for the incorporation of any lane-changing model and low-level lateral controller found in the literature [45, 46], as long as this controller does not result in significant changes to the longitudinal dynamics of the system. An

example of such a decoupled lateral controller is provided in [28].

2.4.3 Experimental Setup and Results

The performance of the proposed approach is evaluated on a bidirectional highway road segment simulation, implemented on the SUMO [48] traffic simulation platform. The simulation setup for the highway segment is shown in Figure 2.12. The controller communicates with the simulator using the TraCI traffic controller interface. All simulations and control algorithms are run on a personal computer with an Intel i7-8750H CPU and 32GB of RAM.

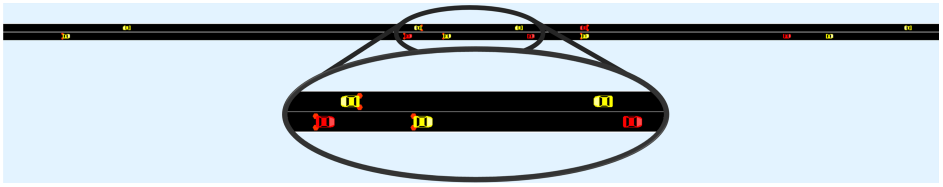

Figure 2.12: **SUMO Simulation Environment**. The experiments are carried out on a 2 km-long bidirectional highway. This short highway segment highlights the presence of AVs, depicted in red, and HDVs, shown in yellow, on the road.

The length of the bidirectional highway segment simulated is 2 km long. The vehicles used were a mix of CAVs and HDVs in varying proportions (CAV penetration levels).

2.4.3.1 Performance Evaluation

The performance of our proposed communication-based overtaking algorithm is evaluated according to its ability to avoid collisions, perform successful overtakes in order to maximize throughput, and minimize the amount of failed overtake attempts and risky maneuvers.

Figure 2.13 shows the trajectories taken by 11 vehicles, 2 of which are CAVs following our control algorithm. The trajectories of these 2 CAVs are marked in red with the color switching to

Simulation Parameters	**Value**
Simulation step size	100 ms
Simulation duration	1 hour
Road length	2 km
Road speed limit ($\bar{v}$)	20 m/s
Average HDV speed	10 m/s
Controller Parameters	**Value**
Controller sampling time (T_s)	500 ms
Maximum acceleration (A^{max})	4 m/s^2
Maximum deceleration (A^{min})	-9 m/s^2
Maximum velocity (V^{max})	20 m/s
CAV sensing range (L_s)	150 m
Planning horizon (H)	10 s
Safety Margin Parameters ($[L_{0_i}, L_{v_i}, L_{a_i}, L_{l_i}]$)	[10, 5, 5, 10]

Table 2.2: Simulation & Controller Parameters for Cooperative Overtaking

green whenever these CAVs move into the oncoming lane. The blue trajectories having positive gradients depict the HDVs traveling in the same direction and lane as the CAVs. The purple trajectories having negative gradients depict the HDVs traveling in the opposite direction to the CAVs in the oncoming lane. To avoid confusion, we selected a section of the simulation in which no oncoming lane CAVs were present. Overtakes occur whenever a CAV trajectory crosses a same-lane HDV denoted in blue. Figure 2.13 shows four such successful overtakes. Note that at the point of overtaking, the trajectories of the CAVs should be green indicating that the CAV is in the oncoming lane. As such collisions with CAVs occur only if a blue trajectory intersects a red trajectory (collision in the original lane) or if a purple trajectory intersects a green trajectory (collision in the oncoming lane). We do not observe any of these conditions which shows us that our algorithm allows for collision-free successful overtaking with the minimum time spent in the oncoming lane.

Additionally, we note that the average time required for each optimization computation step

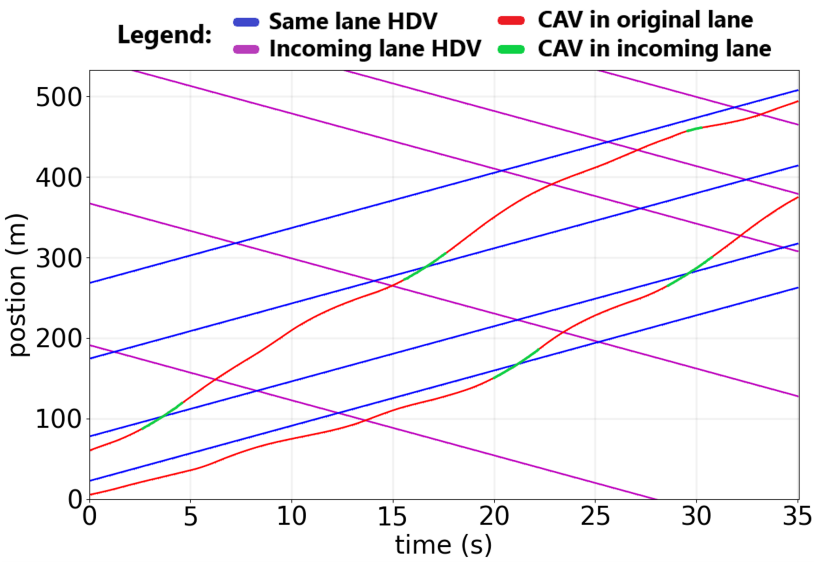

Figure 2.13: **Vehicle Trajectories**. 2 CAVs (red and green trajectories) are attempting to overtake in the presence of 9 HDVs (blue and purple trajectories). Blue (purple) trajectories are HDVs traveling in the same (adjacent) lane as the CAVs. Overtakes occur whenever a CAV trajectory (red) crosses the same lane HDV trajectory (blue). The green color signifies that a CAV has moved into the adjacent lane.

for an ego CAV is $32.36\,ms$ with a standard deviation of $13.19\,ms$. As the controller operates with a time step of $500ms$, this provides excess margins to ensure real-time operation capability.

2.4.3.2 Comparative Analysis

In order to highlight the benefits of the proposed communication-based (Cooperation) approach, we compare its performance with an approach that does not use any inter-vehicle communication (Single Agent) and an approach that assumes the CAV has global knowledge (Global Info) about its surrounding. In the Single Agent approach, CAVs cannot communicate with each other and must rely on their own sensor information for decision-making. In the Global Info case, CAVs make the unrealistic assumption of having access to the states of all its neighboring vehicles irrespective of their sensor occlusion status. We utilize two key metrics to compare the perfor-

mances of these three algorithms: The number of overtakes attempted per CAV and the success ratio of completed overtakes to the attempted overtakes. Here, the attempted overtakes track the number of times the CAV moves into the oncoming lane to try to perform an overtaking maneuver. Overtaking success, on the other hand, tracks what percentage of these overtaking attempts actually lead to a successful overtaking maneuver. The performance of the three algorithms in terms of these two evaluation metrics with varying traffic flow conditions is depicted in Figure 2.14 and Figure 2.15. Here, the flow rate represents the vehicles per minute entering the simulation. As the average velocities of vehicles remain generally constant, the flow rate is also directly proportional to the density of vehicles in the simulation.

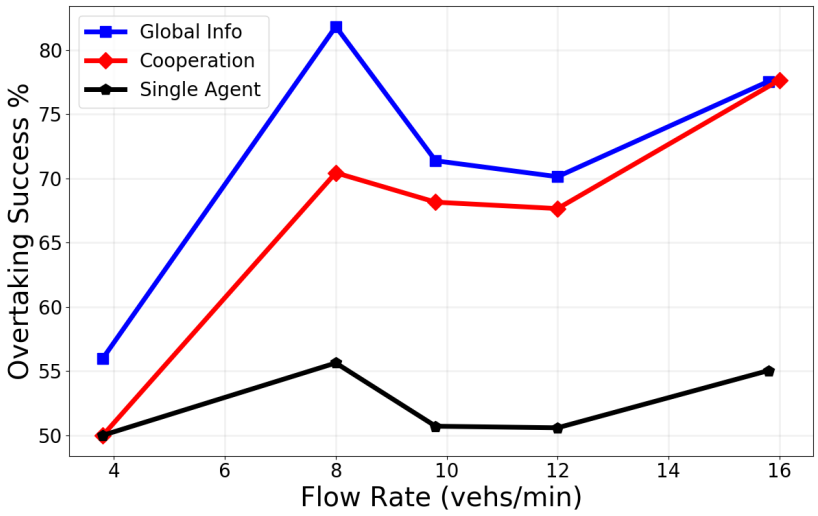

Figure 2.14: **Overtaking Success w.r.t. Varying Flow Rate**. With the overtaking success % on the y-axis and the flow rate in vehicles/minute on the x-axis, the plots show a comparison between the Global Info (in blue), Cooperation (in red), and Single Agent (in black) strategies.

From Figure 2.14, we observe that the Cooperation method as expected performs significantly better than the Single Agent method. This difference in performance is highlighted further with increasing levels of input flow rate. This is due to the fact that with an increasing input flow

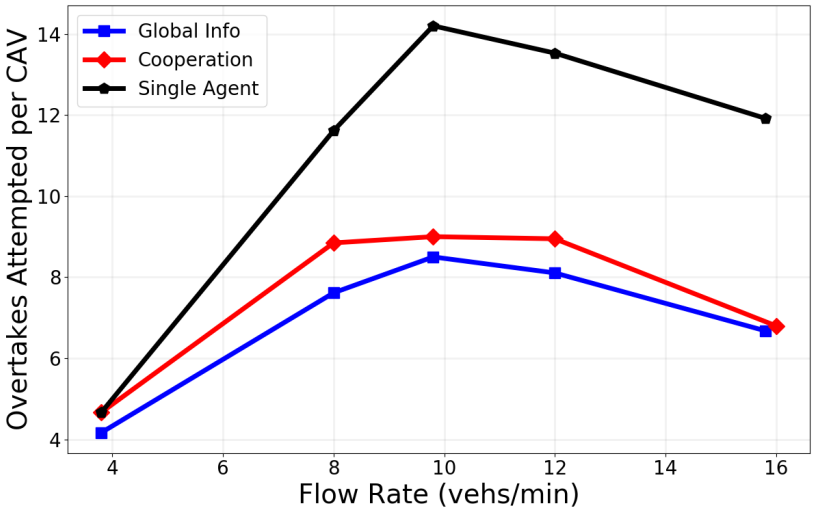

Figure 2.15: **Overtakes Attempted w.r.t. Varying Flow Rate**. With the overtakes attempted per CAV on the y-axis and the flow rate in vehicles/minute on the x-axis, the plots show a comparison between the Global Info (in blue), Cooperation (in red), and Single Agent (in black) strategies.

rate, there is a higher probability that some of the vehicles within the communication range of the ego CAV are other CAVs that can share information about their neighboring vehicles. This provides the ego CAV with more information about its surroundings which leads to less risky overtaking attempts and more overtaking successes. As expected, the unrealistic Global Info method outperforms our Cooperation method. However, we observe that this difference reduces with increasing input flow rate for the reason mentioned earlier i.e. the increased number of CAVs in the communication range. From Figure 2.15, we observe that up to a certain input flow rate, all methods show an increase in overtake attempts since at higher densities, there are more HDVs to overtake. However, beyond a certain level of flow rate, we see an overall drop in overtaking attempts as there are fewer overtaking opportunities due to increased vehicle density. We also find that our method is able to gather enough information about its surroundings using communication, resulting in very low levels of risky overtakes attempted as witnessed by its comparative

performance to the Single Agent and Global Info strategies in Figure 2.15.

2.4.3.3 Impact of CAV Penetration Levels

For our cooperative control strategy, we find that the CAV penetration level plays a significant role in performance output.

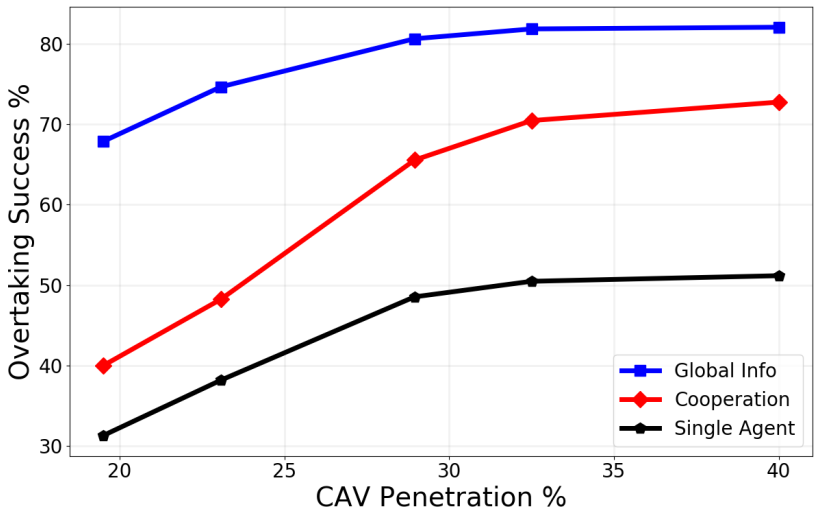

Figure 2.16: **Overtaking Success w.r.t. CAV Penetration Level**. With the overtaking success % on the y-axis and the CAV penetration % on the x-axis, the plots show a comparison between the Global Info (in blue), Cooperation (in red), and Single Agent (in black) strategies.

Figure 2.16 shows an increase in the successful overtakes for the Cooperation method with an increasing CAV penetration level. This is due to the increase in the number of CAVs in the environment which enables greater information interchange between the CAVs (i.e. increased CAV penetration leads to increased information available to each CAV for decision-making).

In Figure 2.16 and 2.17, we observe that the overall performance of our Cooperation algorithm lies in between that of the Global Info and the Single Agent methods. On average, we achieve around 40% improvement over the Single Agent method in terms of successful overtakes

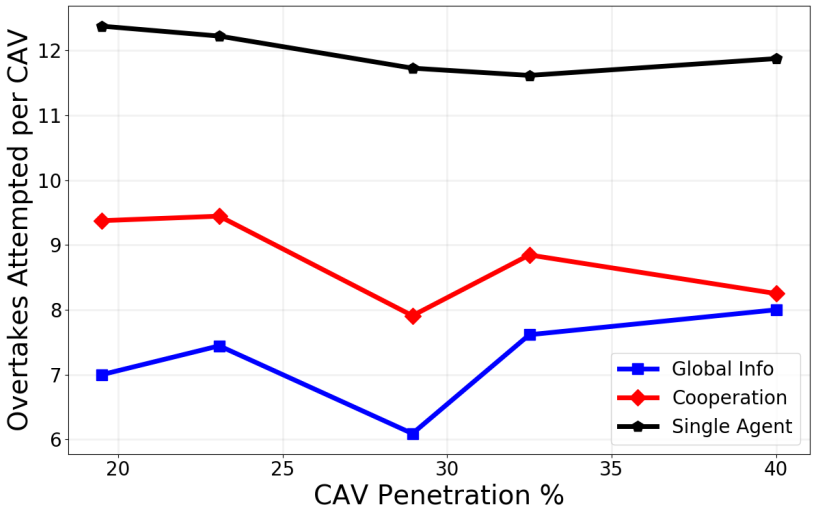

Figure 2.17: **Overtakes Attempted w.r.t. CAV Penetration Level**. With the overtakes attempted per CAV on the y-axis and the CAV penetration % on the x-axis, the plots show a comparison between the Global Info (in blue), Cooperation (in red), and Single Agent (in black) strategies.

and reduced unnecessary overtaking attempts. We also find that as the CAV penetration increases, our method's performance approaches that of the Global Info case.

2.5 Conclusion

In this chapter, we address the autonomous vehicle overtaking problem in a bidirectional mixed-traffic setting using a receding horizon optimization-based approach for the non-cooperative and cooperative cases. The first part of the chapter addresses the non-cooperative case where no autonomous agent is assumed to be present in the vicinity of the ego vehicle. As a result, the ego vehicle relies entirely on its onboard sensors for input information. In the second part, the analysis is extended by incorporating cooperative sensing and an improved trajectory prediction model. The performance enhancements are then evaluated in different densities of autonomous agents present in the environment.

In the first part, we develop a novel mixed-integer model predictive controller, having low computational complexity to perform autonomous overtaking while prioritizing safety. The ability to retract the overtaking decision after initiating the maneuver is gained through the introduction of a binary decision variable. The intrinsic behavior variability of the observed vehicles is accounted for by the modification of vehicle-dependent safety margins. The operational capability of the method in dense traffic is achieved through explicit modeling of limited sensing range and occlusion. Finally, the performance of the controller in diverse settings is verified through simulations in the SUMO environment.

In the second part, we propose an extension for the method developed in the first part for CAVs through a V2V communication-based cooperative control strategy. With this enhancement, the CAVs share information with each other, allowing them to overcome blind spots in sensing and perform safer overtaking maneuvers. We couple the capabilities of V2V information sharing for traffic state estimation with the mixed-integer model predictive controller capable of computing safe overtaking trajectories. We also perform explicit modeling of limited sensor ranges and blind spots caused by sensor occlusion to allow for realistic dense traffic performance tests of our approach. The performance of the proposed approach is evaluated using the SUMO platform and we demonstrate that the enhanced method is capable of achieving a much higher percentage of successful overtakes while reducing the amount of risky unnecessary overtaking attempts when compared to a single agent method with no communication between agents.

Future work for the non-cooperative case entails the implementation of the proposed approach on a small-scale physical setup to evaluate the robustness properties of the controller and

the need to incorporate different noise models to enhance real-world performance. As for the co-operative case, the benefits of collaborative decision-making among CAVs can be explored for the proposed framework.

Chapter 3: Highway Maneuvering

Building on the discussion of safe navigation of autonomous vehicles in the presence of human-driven vehicles, we consider another commonly faced driving situation that highlights the complexities arising from the unpredictable nature of human drivers. In a highway driving situation, a vehicle needs to maneuver around traffic by choosing a series of appropriate lanes and velocities to minimize its travel time, ensure safety and achieve better visibility, while accounting for the behavior of neighboring vehicles. Although seemingly straightforward, lane changing is considered to be one of the riskiest driving behaviors since it is highly contingent upon multi-modal trajectory predictions of neighboring vehicles and requires timely decision-making [54]. It is further influenced by a number of uncertainty factors such as road conditions, measurement accuracy, and a long tail of behavioral uncertainty of on-road agents. However, if executed efficiently, lane changing coupled with speed adjustment can yield a significant improvement in minimizing overall travel time while ensuring passenger comfort [55].

Modern vehicles, such as a Tesla, are already able to perform lane changes on highways [56]. However, the algorithms deployed on these vehicles tend to have short time planning horizons and are generally very conservative so they are ineffective in dense traffic situations. Previous studies have addressed the general lane-changing problem through deterministic rule-based approaches

such as MOBIL [57]. However, these methods tend to be short-sighted, or greedy, at times due to their rigid nature, as demonstrated in [55]. Specifically, Bae et. al. outlined an improved graph-based *risk-aware* planning model that showed a significant reduction in overall travel time. However, the formulation was limited to constant speed planning, which limited its ability to assess the advantages of speed adjustment in long-term planning.

As demonstrated in Section 3.2.4, it may be necessary at times to sacrifice short-term benefits to gain long-term performance improvements. In such a scenario, an approach with speed adjustment coupled with a long planning horizon has the foresight to deliver significantly better results. Moreover, the inclusion of speed adjustment in the decision-making process inhibits the risk of incurring trajectory infeasibility as the environmental conditions may prevent the ego vehicle from traveling at a constant reference speed and the low-level planner may be unable to handle such a discrepancy. Therefore, in Section 3.2, we propose a low complexity receding horizon optimization-based approach that outputs the lane change maneuvers coupled with speed adjustments for long planning horizons ($> 15s$) while guaranteeing safety. The long-horizon strategic decision-making gives the ego vehicle the ability to proactively anticipate and handle challenging driving situations.

The foresighted optimization-based behavior planning framework developed in Section 3.2 optimizes travel time and passenger comfort in determining the future speed and lane profiles but does not explicitly account for the behavioral uncertainty of on-road agents. This uncertainty tends to have a long tail in its distribution that needs to be effectively handled by planning methods [58]. Therefore, Section 3.3 builds on the framework presented in Section 3.2 to address *long-term*

behavior planning under such uncertainty by incorporating effective risk evaluation to quantify the 'severity' of the consequences of decision-making under this uncertainty.

This chapter is structured as follows: Section 3.1 summarizes the existing methods available in the literature to address the highway maneuvering problem; Section 3.2 details our foresighted optimization-based interaction and risk agnostic approach; Section 3.3 details the interaction and risk aware extension; and, Section 3.4 recaps the main artifacts of this chapter. As for the interaction and risk agnostic case, Section 3.2.1 provides an overview of the problem; Section 3.2.2 formalizes the problem; Section 3.2.3 details the various components of the proposed approach while highlighting the implementation and analytical details; and, Section 3.2.4 analyzes the performance of the proposed approach by running a series of comparative tests. As for the interaction and risk aware case, Section 3.3.1 provides an overview of the problem; Section 3.3.2 details the problem formulation; Section 3.3.3 outlines the interaction-aware prediction model and the risk evaluation method employed in our approach; and, Section 3.3.4 details the experimental setup, followed by a qualitative and quantitative comparative analysis with respect to the state-of-the-art behavior planning methods.

3.1 Literature Review

In prior research, target lane and speed determination for lane-changing maneuvers have primarily been examined through the lens of motion planning methods [59]. In general, the various motion planning techniques proposed in the literature can be classified into five main categories: rule-based, sampling-based, learning-based, optimal control related, and interaction-aware

approaches. Since the rule-based and graph-based methods have already been elaborated upon in the preliminary discussion, we focus on the remaining methods in the ensuing discussion.

In the context of sampling-based approaches, single-query methods, particularly the various forms of RRT, are generally favored over multi-query methods, such as roadmap-based methods, due to their faster computation time and the capability to handle non-holonomic constraints [23]. Despite their abilities to incorporate safety guarantees by sampling feasible trajectories from a reachable safe set [22], and perform risk evaluation through chance constraints [60], these methods can often yield an uncomfortable driving experience due to stringing together of individual trajectories. Additionally, the asymptotic optimality guarantees provided by these methods do not necessarily translate to practical success in complex driving situations due to their high sample complexity [23].

In the extensive body of literature on learning-based methods, different variations of Reinforcement Learning techniques have been identified as the preferred approach for AV navigation by the research community [24, 25, 61–64]. While these methods have shown promising results in simulations, their implementation in real-world settings raises concerns such as the need for a large amount of training data, exploration of unsafe behaviors during training, and difficulties in handling edge cases. These approaches primarily rely on using neural networks as function approximators, which results in low computational complexity but also presents limitations in terms of interpretability and safety guarantees.

As for the interaction-aware planning methods, the game-theoretic approaches [65–68] have gained significant prominence. Due to the inherent complexity of highway driving scenarios, char-

acterized by the presence of various traffic participants with diverse and often unknown objectives, leading to conflicting situations, these approaches tend to handle on-road negotiations through an explicit understanding and modeling of the intent of on-road agents while effectively indicating their own intentions. Despite significant research efforts toward designing human behavior models, a common limitation in this line of research is the high computational complexity associated with the inherent uncertainty associated with human behaviors. As a result, these methods tend to be computationally intensive and typically plan over a limited time horizon, leading to conservative behavior and an inability to change lanes in highly uncertain situations.

Finally, in terms of optimization-based approaches, the extensions of optimal control methods have been extensively studied. Unlike the potential-field-based approaches [26], which offer satisfactory collision avoidance performance but lack the ability to integrate vehicle dynamics, the optimal control methods [27], specifically the derivatives of the Model Predictive Control (MPC) approach [13, 29, 41, 69], provide excellent collision avoidance performance while incorporating vehicle dynamics. Typically, these methods are used at the lower layers of the navigation architecture (See Section 1.1) due to their ability to incorporate high-fidelity dynamics. However, this improved performance comes at a cost of increased computational complexity, primarily due to the non-convex constraints introduced by the integration of non-linear dynamical models and collision avoidance constraints. This results in a restricted planning horizon, limited to only a few seconds. Our proposed architecture also falls within the realm of optimization-based methods, but we pursue systematic complexity reduction in order to distribute the computational load of this paradigm. Specifically, we utilize a multi-timescale architecture that allows us to isolate the long

and short-horizon planning modules. In this work, our focus is on long-term planning, which is responsible for lane and speed advisory, while the short-term planning module, which generates a comfortable and safe trajectory, is taken from our previous work [70].

3.2 Interaction & Risk Agnostic Planning

This section proposes a hierarchical autonomous vehicle navigation architecture composed of a high-level speed and lane advisory system (SLAS) coupled with low-level trajectory generation and trajectory following modules. Specifically, we target a multi-lane highway driving scenario where an autonomous ego vehicle navigates in traffic. We propose a novel receding horizon mixed-integer optimization-based method for SLAS with the objective of minimizing travel time while accounting for passenger comfort. We further incorporate various modifications in the proposed approach to improve the overall computational efficiency and achieve real-time performance. We demonstrate the efficacy of the proposed approach in contrast to the existing methods, when applied in conjunction with state-of-the-art trajectory generation and trajectory following frameworks, in a CARLA simulation environment.

3.2.1 Overview

To motivate the problem, consider the scenario presented in Figure 3.1. Based on the predicted motion (shown in a lighter shade) of the neighboring vehicles (shown in orange), the ego vehicle (shown in blue) may decide to either change lanes left in an attempt to minimize its travel time or slow down in the current lane to maintain safety. However, it would be imprudent for the

ego vehicle to risk changing lanes right and consequently get stuck behind a slow-moving vehicle even though there is presently a greater headway. This simple scenario highlights the importance of foresight and long planning horizon in strategic decision-making for autonomous vehicles.

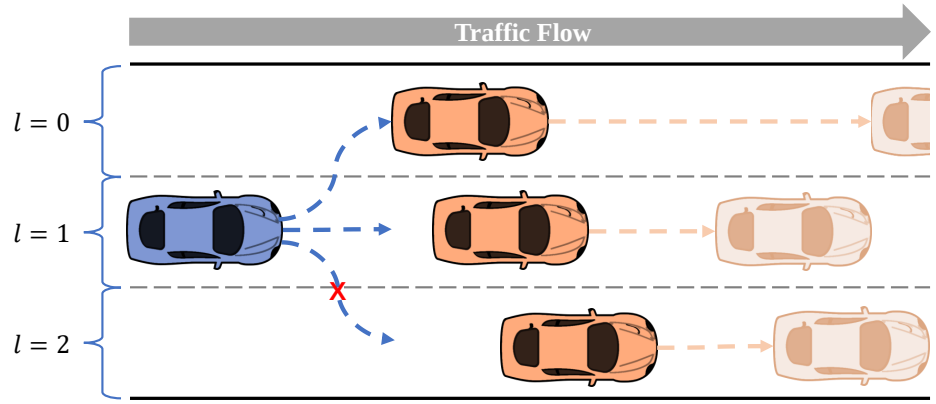

Figure 3.1: **Motivational Example**. With a slow-moving vehicle ahead, the ego vehicle (in blue) may decide to either change lanes to the fast-moving lane (left) to minimize travel time or adjust its speed without changing lanes to preserve safety but it would be unwise for it to switch to the slow-moving lane (right) as that would not benefit travel time or safety.

Contribution

The key requirements for the algorithmic design of an autonomous vehicle include real-time operation, safety guarantees, optimality with respect to some metric(s), and accounting for the behavior variability of on-road agents. Considering these requirements, we propose an optimization-based behavioral planning framework that enables autonomous vehicle maneuvering on multi-lane highways. While having the benefits of optimization-based approaches, our method achieves a low computational complexity by employing a binary representation of the decoupled lane indicator dynamics in lieu of lateral dynamics and utilizing algorithmic modifications to aid numerical computations. Specifically, our method provides:

- optimality with respect to travel time and comfort;
- safety and feasibility guarantees;
- real-time applicability for a long planning horizon; and
- modularity in design, which enables the integration of external trajectory prediction modules.

The proposed method fills in the research gap by meeting all the key algorithmic requirements while simultaneously gaining the foresight to make strategic decisions that yield long-term performance benefits, as verified in Section 3.2.4.

3.2.2 Problem Formulation

In this section, we present the algorithmic pipeline and formalize the road, observation, and vehicle dynamics models that will be utilized in the subsequent sections.

3.2.2.1 Algorithmic Pipeline

Our main focus in this Section is the development of the behavior planning module, highlighted as SLAS in Figure 3.2. SLAS outputs the target lane and reference speed which are utilized by the motion planning module to generate a reference trajectory for the ego vehicle. The vehicle controllers compute the throttle and steering commands to track the trajectory accordingly. Further details regarding the different modules can be found in Section 1.1.

For the motion planning module, we adopt the Neural Networks integrated Model Predictive Control (NNMPC) [70] due to its ability to accommodate the behaviors of neighboring vehicles

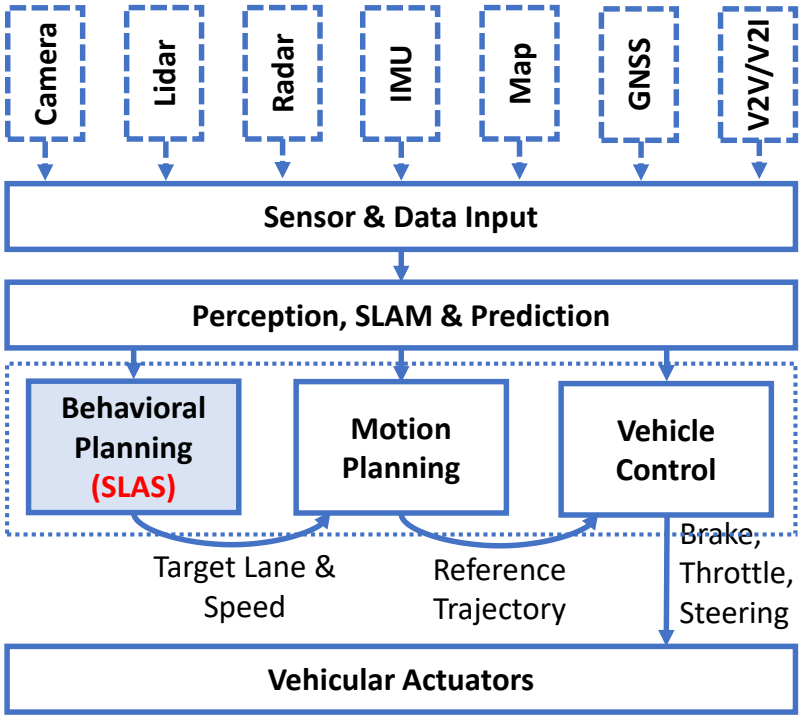

Figure 3.2: **Algorithmic Pipeline**. Based on the localization information from the Perception, and Simultaneous Localization and Mapping (SLAM) modules, our behavior planning framework, SLAS, outputs the target lane and speed to the motion planner, NNMPC, which generates a reference trajectory for the vehicle control module.

in the trajectory generation process. In our approach, we assume that the perception (of other vehicles) and the localization (of the ego vehicle) are known without any uncertainty, for simplicity, but the modular architecture avails us the ability to integrate any perception or SLAM module in the overall framework.

3.2.2.2 Road Model

The physical road structure is modeled as a continuous multi-lane highway with negligible curvature and unidirectional traffic flow. The lanes on the highway are clearly demarcated and at any given time k, the number of available lanes for the vehicles to travel on is denoted by $N_l(k)$

while the road speed limit is denoted by V_l. Therefore, the set of lanes available for traveling at a given time instant k is denoted by $\mathbb{L}(k) = \mathbb{Z}_{[0,N_l(k)]}$. We work with the Frenet coordinate system where the distance along the road is denoted by the longitudinal displacement (s) and the distance perpendicular to the road is defined by the lateral displacement (d). Each lane is assigned a lane indicator variable l. The leftmost lane, with respect to the direction of traffic flow, is assigned a value of $l = 0$ while each subsequent lane is assigned an increasing integer value for l, as depicted in Figure 3.1.

3.2.2.3 Vehicle Model

Since we aim to have real-time computations for a long planning horizon ($> 15s$), we model the vehicle dynamics with a linearized decoupled dynamical system. For the highway driving scenario, where the road curvature is typically small, it is reasonable to assume a decoupling between the lateral and the longitudinal dynamics [40], especially for the behavior planning layer. Therefore, we utilize a linear constant acceleration model for the longitudinal dynamics and abstract out the lateral dynamics with a lane indicator variable. For the lane change dynamics, we use a moving average filter coupled with a rounding function to model the time required by the ego vehicle to change lanes. This is compactly represented as:

$$s_0(k) = s_0(k-1) + \frac{v_0(k-1) + v_0(k)}{2} \cdot T_s \tag{3.1}$$

$$l_0(k) = \left\lfloor \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{L}(k-i) + \frac{1}{2} \right\rfloor \tag{3.2}$$

where $s_0(k)$, $v_0(k)$, $l_0(k)$ and $\mathcal{L}(k)$ denote the ego vehicle's longitudinal displacement, speed, lane indicator, and target lane, respectively, at time instant k; the subscript $_i$ indexes the vehicles on the road with $_0$ being reserved for the ego vehicle; T_s denotes the discretization time step; and, N corresponds to the number of time steps required to change lane. The state ($x_0(k)$) and control input ($u_0(k)$) to the system at time instant k are defined as:

$$x_0(k) = \begin{bmatrix} s_0(k) & l_0(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R} \times \mathbb{L}(k) \tag{3.3}$$

$$u_0(k) = \begin{bmatrix} v_0(k) & \mathcal{L}(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}_{[0,V_m]} \times \mathbb{L}(k) \tag{3.4}$$

where V_m denotes the maximum speed of the ego vehicle.

3.2.2.4 Observation Model

For practical considerations, we restrict the ego vehicle's visibility range to the sensory perception limit, denoted by R_v. Then, the set of vehicles in ego vehicle's visibility range at time instant k, represented by $\mathbb{O}(k)$, is defined as:

$$\mathbb{O}(k) = \{i \in \mathbb{Z}_{>0} \mid |s_i(k) - s_0(k)| \leq R_v\} \tag{3.5}$$

where $s_i(k)$ corresponds to the longitudinal displacement of the observed vehicle.

Remark 7. *For the multi-lane highway driving scenario, occlusion does not play a prominent role so we do not account for it in the existing formulation. However, the proposed framework can easily accommodate occlusion and measurement uncertainties since the receding horizon approach bases*

its decision on the most up-to-date information available at any given time, as demonstrated in Section 2.3.1.3.

3.2.3 Methodology

In this section, we describe the prediction model to generate the predicted future trajectories of observed vehicles and present a discussion on the proposed receding horizon optimization-based behavioral planning module.

3.2.3.1 Trajectory Prediction

Reliable behavior and trajectory prediction of other traffic participants are crucial for the safe maneuvering of autonomous vehicles. The algorithm proposed in Section 3.2.3.2 is able to incorporate any generic prediction module available in the literature [16] as long as it can provide a deterministic predicted future trajectory for a given vehicle. In this work, we formulate a low-complexity prediction model that highlights the flexibility and efficiency of our proposed approach.

For an observed vehicle $i \in \mathbb{O}(k)$, the future speed profile is predicted using a piece-wise linear function while the lane profile is assumed to stay constant for the duration of the prediction horizon. At a given time step k, the estimated acceleration ($\bar{a}_i^k$) and the estimated speed ($\bar{v}_i^k$) parameters are obtained through linear regression with mean-squared error on the past $o_i^k > 1$ speed observations. Based on the estimated parameters, we predict the future speed and longitudinal

displacement as follows:

$$\hat{v}_i^k(j) = \begin{cases} \bar{v}_i^k, & j = 0 \\ \hat{v}_i^k(j-1) + \bar{a}_i^k \cdot T_s, & 0 < j \leq H_a \\ \hat{v}_i^k(j-1), & j > H_a \end{cases} \tag{3.6}$$

$$\hat{s}_i^k(j) = \begin{cases} s_i(k), & j = 0 \\ \hat{s}_i^k(j-1) + \frac{T_s}{2} \cdot (\hat{v}_i^k(j-1) + \hat{v}_i^k(j)), & j > 0. \end{cases} \tag{3.7}$$

Here, H_a corresponds to the acceleration horizon while $\hat{v}_i^k(j)$ and $\hat{s}_i^k(j)$ respectively represent the predicted speed and longitudinal displacement for vehicle i, j time steps into the future starting from the current time instant k.

Remark 8. *Due to the modular nature of the proposed framework, the behavior planning module detailed in Section 3.2.3.2 can work with advanced maneuver-based (e.g. Markov Chain [71]) and interaction-based (e.g. Social Generative Adversarial Networks [72]) trajectory prediction modules, allowing for interactive maneuvering behaviors.*

3.2.3.2 Speed and Lane Advisory System

The goal of our behavior planning module, Speed and Lane Advisory System, abbreviated as SLAS, is to determine a sequence of speed and lane change commands that would enable the ego vehicle to maximize its speed, thus minimizing the travel time, while accounting for driver comfort and abiding by its dynamical, actuator, and safety limits. The output of this module is a

relatively smooth speed and lane change profile which is then passed on to a motion planner. It is necessary to incorporate the dynamical and actuator limits in the behavioral planning module so as not to provide the motion planner with goals that are not reachable and jeopardize the safety of the overall system as a result.

In the subsequent discussion, we provide a formulation of the optimization problem for SLAS, highlight the modifications necessary to improve the computational complexity, and present safety and feasibility analyses.

Optimization Problem with Integer Constraints: SLAS is posed as an optimization problem, with the objective of maximizing speed while minimizing frequent lane changes and abrupt changes in speed. The output of SLAS, at time instant k, is the control input $u_0(k+1)$, as defined in (3.4).

The optimization problem is formulated as follows:

$$\min_{\substack{v^k(1),\cdots,v^k(H);\\ \mathcal{L}^k(1),\cdots,\mathcal{L}^k(H)}} \sum_{j=1}^{H} [-\gamma_1 \cdot v^k(j) + \gamma_2 \cdot (\mathcal{L}^k(j) - \mathcal{L}^k(j-1))^2 + \gamma_3 \cdot (v^k(j) - v^k(j-1))^2] \tag{3.8}$$

$$\text{s.t.} \quad s^k(0) = 0 \tag{3.9}$$

$$v^k(0) = v_0(k) \tag{3.10}$$

$$\mathcal{L}^k(p) = l_0(k), \quad \forall p \in \mathbb{Z}_{[-N+1,0]} \tag{3.11}$$

$$\forall j \in \mathbb{Z}_{[1,H]} :$$

$$v^k(j) \in \mathbb{R}_{[0,V_l]} \tag{3.12}$$

$$\frac{v^k(j) - v^k(j-1)}{Ts} \in \mathbb{R}_{[A_{\min},A_{\max}]} \tag{3.13}$$

$$s^k(j) = s^k(j-1) + \frac{v^k(j-1) + v^k(j)}{2} \cdot T_s \tag{3.14}$$

$$\mathcal{L}^k(j) \in \mathbb{L}(k) \tag{3.15}$$

$$\mathcal{L}^k(j) - \mathcal{L}^k(j-1) \in \mathbb{Z}_{[-1,1]} \tag{3.16}$$

$$l^k(j) = \left\lfloor \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{L}^k(j-i) + \frac{1}{2} \right\rfloor \tag{3.17}$$

$$\min_{i \in \mathbb{A}(k)} \{|\hat{s}_i^k(j) - s^k(j)|\} \geq L_i^s(j), \tag{3.18}$$

$$\mathbb{A}(k) = \{a \in \mathbb{O}(k) \mid l^k(j) = l_a(k)\}.$$

Objective Function: In the formulation above, the optimization variables are the ego vehicle's speed ($v^k(j)$) and target lane ($\mathcal{L}^k(j)$), j step into the future, starting from the time instant k. Here, H corresponds to the planning horizon. The scalarization parameters γ_1, γ_2, and γ_3 in the objective function (3.8) account for a relative tradeoff between maximizing speed, minimizing lane changes,

and minimizing abrupt changes in speed respectively. Increasing γ_1 yields a more aggressive behavior with the priority placed on maximizing speed while γ_2 and γ_3 combine to place an emphasis on maximizing passenger comfort by reducing lane and speed changes respectively.

Dynamical Constraints: These constraints are put in place to ensure the dynamical feasibility of the solution. The constraints (3.9), (3.10), and (3.11) serve to initialize the longitudinal displacement, speed, and target lane respectively for the optimizer, based on the values observed at time instant k. The constraints (3.12) and (3.13) bound the ego vehicle's speed by the speed limit and the acceleration limits of the vehicle respectively. The ego vehicle's speed is then used to calculate the projected longitudinal displacement in (3.14).

The target lane values at any planning step (j) are restricted to the set of reachable values by (3.15), (3.16) and (3.17). Here, (3.15) restricts the target lane to the set of available lanes ($\mathbb{L}(k)$), (3.16) ensures that the lane change, if needed, is made to the adjacent lane only and (3.17) models the time steps (N) required for a lane change. The flooring function can easily be transformed into a couple of linear constraints by the introduction of an auxiliary integer variable, as shown in the Appendix. Finally, $l^k(j)$ is merely the internal representation of the lane the ego vehicle is projected to travel on at planning step j.

Safety Constraint: The safety constraint (3.18) ensures that the ego vehicle maintains a minimum safe distance ($L_i^s(j)$) to the nearest vehicle i, in its projected lane of travel ($l^k(j)$), at planning instant j. We borrow the definition of this safe distance from [73], where the authors provide a formalization based on the clause from Vienna Convention on Road Traffic that states that "*A vehicle [...] shall keep at a sufficient distance [...] to avoid collision if the vehicle in front should*

suddenly slow down or stop." Furthermore, the absolute value constraint can be decomposed into linear constraints by the application of the big-M method and the introduction of an auxiliary variable, as shown in the Appendix.

Remark 9. *The proposed formulation can accommodate an arbitrary number of lanes at any given time instant k. This means that if at any given time, the number of available lanes for traveling either increases or decreases, the proposed formulation will still continue to hold. This is an important consideration since many a time on highways, some lanes are blocked due to various unanticipated situations such as a road accident, roadwork, narrowing of road, etc.*

Computational Complexity Reduction: This section details the optimization problem reformulation with binary variables, optimization warm start technique, and lazy constraint implementation, all of which combine to improve the computational complexity of our SLAS module.

Binary Variables: The proposed formulation in Section 3.2.3.2 has relatively high computation complexity (computation time of $\sim 2s$ in the worst case scenario - slow moving traffic blocking all the lanes) due to the integer decision variables yielding a mixed-integer optimization problem [74]. To circumvent the computational overload, we reformulate the problem with binary variables that replace the integer variables, as follows:

$$\forall i \in \mathbb{L}(k), \forall j \in \mathbb{Z}_{[1,H]} : \tilde{\mathcal{L}}^k(i,j) \in \{0,1\} \tag{3.19}$$

where the $\tilde{\mathcal{L}}^k(i,j)$ represents the modified target lane variable, indexed by the lane (i) as well as the planning step (j) and $\tilde{\mathcal{L}}^k(a,b) = 1$ represents the choice of lane $a \in \mathbb{L}$ as the target lane at

planning step $b \in \mathbb{Z}_{[1,H]}$. Then, some of the constraints from the SLAS formulation in Section 3.2.3.2 are modified as follows:

$$(3.11) \rightarrow \tilde{\mathcal{L}}^k(l_0(k), 0) = 1 \tag{3.20}$$

$$(3.15) \rightarrow \sum_{i \in \mathbb{L}(k)} \tilde{\mathcal{L}}^k(i, j) = 1, \forall j \in \mathbb{Z}_{[1,H]} \tag{3.21}$$

$$(3.16) \rightarrow \tilde{\mathcal{L}}^k(a, j) = 1 \implies \sum_{b \in \mathbb{B}(k)} \tilde{\mathcal{L}}^k(b, j) = 1, \tag{3.22}$$

$$\mathbb{B}(k) = \mathbb{Z}_{[a-1,a+1]} \cap \mathbb{L}(k)$$

$$(3.18) \rightarrow \min_{i \in \mathbb{A}(k)} \{|\hat{s}_i^k(j) - s^k(j)|\} \geq \tilde{L}_i^s(j), \tag{3.23}$$

$$\mathbb{A}(k) = \{a \in \mathbb{O}(k) \mid \tilde{\mathcal{L}}^k(l_a(k), j) = 1\}.$$

Here, (3.20) initializes the target lane, (3.21) restricts the target lane at any planning step to the set of available lanes, (3.22) restricts the lane change between consecutive planning steps to the adjacent lanes, and (3.23) represents the augmented safety constraint. The implication ($\implies$) in (3.22) can easily be transformed into a linear constraint (see Appendix). The augmented minimum safety distance ($\tilde{L}_i^s(j)$) incorporates the time required to execute the lane change maneuver (N) from (3.17) into the following unified safety constraint:

$$\tilde{L}_i^s(j) = L_i^s(j) + \gamma_d(\delta^k(j)) \cdot (v^k(j) \cdot N \cdot T_s) \tag{3.24}$$

$$\gamma_d(\delta^k(j)) = \gamma_4 \cdot \frac{2|\delta^k(j)|}{L_l} \tag{3.25}$$

$$\delta^k(j) = \min \left\{ \delta(k) + \frac{L_l \cdot j}{N}, \frac{L_l}{2} \right\} \tag{3.26}$$

where L_l is the width of the lanes (see Figure 3.1), $\delta(k)$ is the signed lateral deviation of the ego vehicle from the **previous** target lane's boundary at time step k, and $\gamma_d(\delta^k(j))$ is the dynamic cost of deviation from the previous target lane ($\mathcal{L}(k-1)$). Moreover, in the cost function (3.8), we take $\mathcal{L}^k(0) = \mathcal{L}(k-1)$. These costs are introduced to prevent the swerving (canceling of lane switch before completion) behavior unless absolutely necessary (for safety purposes).

Remark 10. *Since the ego vehicle is considered to have changed lane once it crosses a lane boundary, the deviation $\delta^k(j)$ is considered from the lane boundary instead of the center of the target lane to maintain the continuity of $\gamma_d(\delta^k(j))$ with respect to the lateral displacement of the ego vehicle. Specifically, $\delta^k(j) > 0$ if the ego vehicle has crossed the previous target lane boundary and 0 otherwise. This is an important consideration since a discontinuity in $\gamma_d(\delta^k(j))$, upon completion of lane change, may lead to infeasibility.*

Remark 11. *The swerving behavior is suppressed but not completely eliminated with a hard constraint since such behavior is necessary at times to react to the environment's unpredictability. This reactive strategy, which is a distinctive feature of our approach, avails the algorithm the ability to proactively 'change its mind' in case something unanticipated happens in the environment that can jeopardize safety.*

Optimization Warm Start: To aid the optimizer in finding an initial feasible solution, we provide the solution from the previous time step as a reference. Formally, $\{v^{k-1}(2), \cdots, v^{k-1}(H), \mathcal{L}^{k-1}(2), \cdots, \mathcal{L}^k(H)\}$ is provided as a reference for $\{v^k(1), \cdots, v^k(H-1), \mathcal{L}^k(1), \cdots, \mathcal{L}^k(H-1)\}$. This doesn't imply that the solution from time step $k-1$ will hold exactly at time step k, owing to the unmodeled disturbances, but providing this reference aids the optimizer in finding an initial

feasible solution in the vicinity of the reference solution. This observation is rooted in the premise that the solution for the long planning horizon is not expected to change significantly between time steps, given the sampling time is not too large and the predicted behavior of on-road agents does not alter significantly.

It is also worth pointing out that the priority here is quickly finding a feasible solution that obeys the safety constraints and actuator limits, and recursively improving it, rather than excessively iterating to reach a global optimum. In our experiments, it was observed that a suboptimal solution was qualitatively not significantly different from the optimal one. Therefore, we utilize the cutting planes method for optimization [75], which first looks for a feasible solution using our provided reference, and then recursively updates it until either the globally minimal solution is found or the time limit is reached.

Lazy Constraints: To further enhance the computational efficiency, we introduce a lazy implementation of the lane-changing constraints (3.22). It was observed in our experiments that a feasible solution without the lane changing constraints (3.22) can be found several orders of magnitude ($\sim 10\times$) quicker than if we include these constraints so we decided to have a lazy implementation for them. With a lazy implementation [76], the solver finds a set of feasible solutions without the inclusion of these constraints and then determines the feasibility of those solutions from the reduced problem with respect to the lazy constraints.

Feasibility: By an argument similar to the one presented in Section 2.3.2.3, it is a relatively straightforward proof for recursive feasibility of the problem, i.e. the optimization problem will continue to stay feasible, if initially feasible, with the trivial solution being matching the speed of

the leading vehicle and not changing lanes.

3.2.4 Results

In this section, we detail our experimental setup, demonstrate the performance of SLAS, and report a qualitative as well as quantitative comparative analysis. The baselines in our comparative analysis are set to: Extended-Astar (EA*) [55], MOBIL [57], and no lane-change model (No-Change).

3.2.4.1 Experimental Setup

The implementation setup, depicted in Figure 3.3, is composed of the CARLA simulator (Version 0.9.11) [77], SLAS module (Section 3.2.3.2), and the planner and controller module [70]. To solve the optimization problem for SLAS, we use Gurobi Optimizer (Version 9.1.1) [49]. The simulations are performed on a computer equipped with an Intel Xeon(R) CPU E5-2643 v4 @ 3.40GHz × 12 and NVIDIA Titan XP, running Ubuntu 20.04 LTS. On average, the time required for each optimization step is $\sim 0.096s$, while the maximum time limit for the optimizer is set to $0.2s$, indicating the strong potential for real-time applicability.

3.2.4.2 Case Study

Figure 3.4 illustrates the test case scenario for our comparative analysis. The scenario is composed of a highway segment with four lanes and the rightmost lane reserved for merging vehicles. The ego vehicle is initialized to follow a slow-moving vehicle in lane 1 and has even

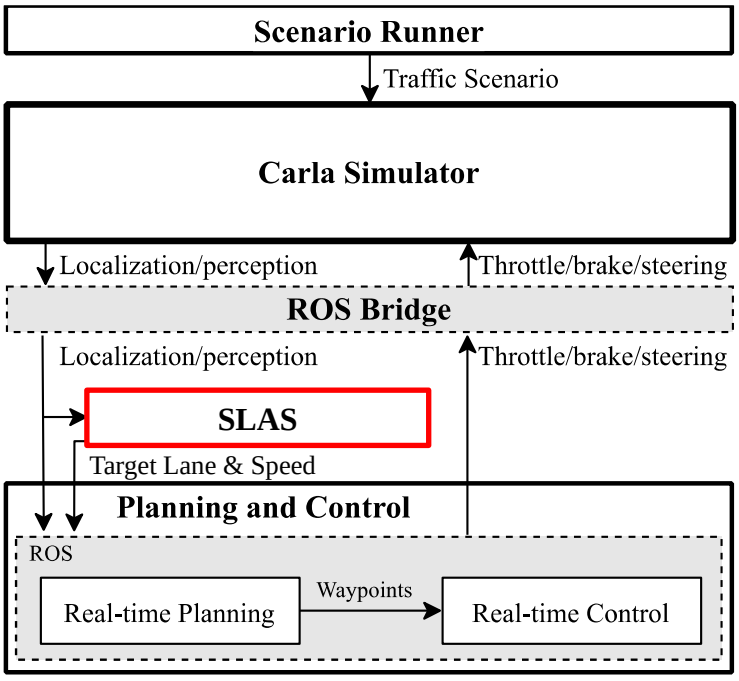

Figure 3.3: **Simulation Setup**. Scenario Runner sets up the scenario for the CARLA Simulator, which then communicates with the SLAS and the Planning and Control ROS (Robot Operating System) nodes through the ROS bridge node.

slower-moving traffic to its right in lane 2. Thus, the only option for it, in order to minimize travel time, is to switch left to lane 0 with faster-moving traffic and greater headway. Once it moves to lane 0 and overtakes the slow-moving vehicle in lane 1, it has two options: either to keep traveling in lane 0 without making any lane change decisions until getting close to the lead vehicle or proactively exploiting the gap in lane 2 to switch to lane 3 in anticipation of traffic buildup in lanes 1 and 2. A strategic decision-maker with foresight will choose to take the latter option and make the decision proactively for a greater overall benefit.

The evaluation metrics for the comparative analysis include travel time, lateral displacement, headway, and distance to the closest vehicle. As for the simulation parameters, the simulation step size is set to $0.05s$ (simulation frequency of $20Hz$); the velocities of vehicles in lanes 0, 1, and

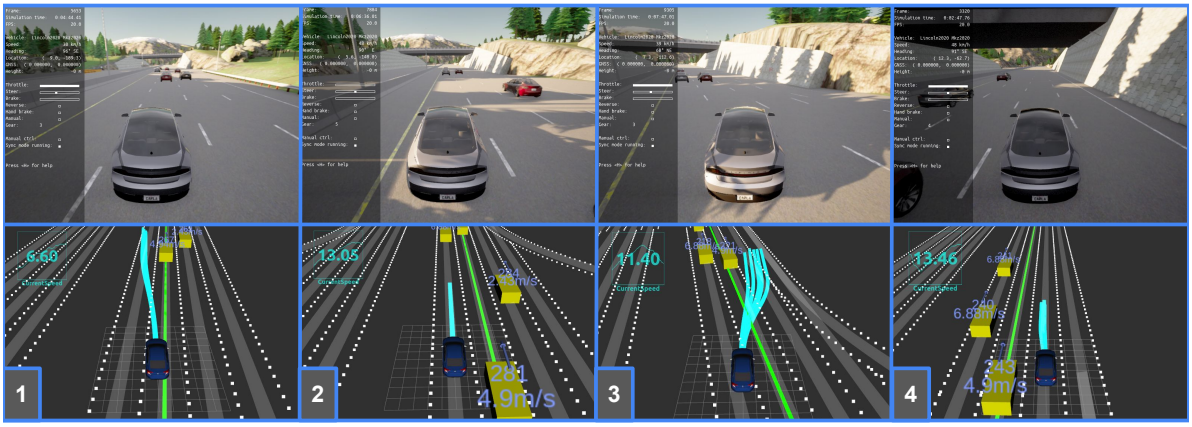

Figure 3.4: **Testing Scenario**. In this scenario with three lanes - lane 0 (left), lane 1 (center), and lane 2 (right) - the expected motion of the ego vehicle, over the course of the simulation, is shown with numbered frames. The rightmost lane (lane 3) is reserved for merging traffic so it is not utilized in our simulation.

2 are set to 8, 5, and 2 m/s respectively while the speed limit V_l is set to $15m/s$; the length of the highway patch is set to $350m$ while the width between the lanes is set to $3.5m$; and the sensor visibility range is set to $R_v = 50m$. The parameters for SLAS are set as follows: $T_s = 0.4s$, $H = 40$, $N = 3$, $A_{min} = -5m/s^2$, $A_{max} = 3.5m/s^2$, $\gamma_1 = 1$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.01$. The values of these parameters can be tuned to yield an aggressive or defensive behavior of the algorithm.

Travel Time: The left plot in Figure 3.5 depicts the travel time as a function of longitudinal displacement for the four algorithms. As seen in the plot, our method (SLAS) maintains a lower overall travel time as compared to the other methods. Quantitatively speaking, SLAS outperforms EA*, MOBIL, and No-change methods by 12.72%, 23.52%, and 54.34% respectively in terms of the time required to complete the simulation scenario. This shows that our method's foresight compensates for its apparent conservativeness arising from the need to preserve passenger comfort.

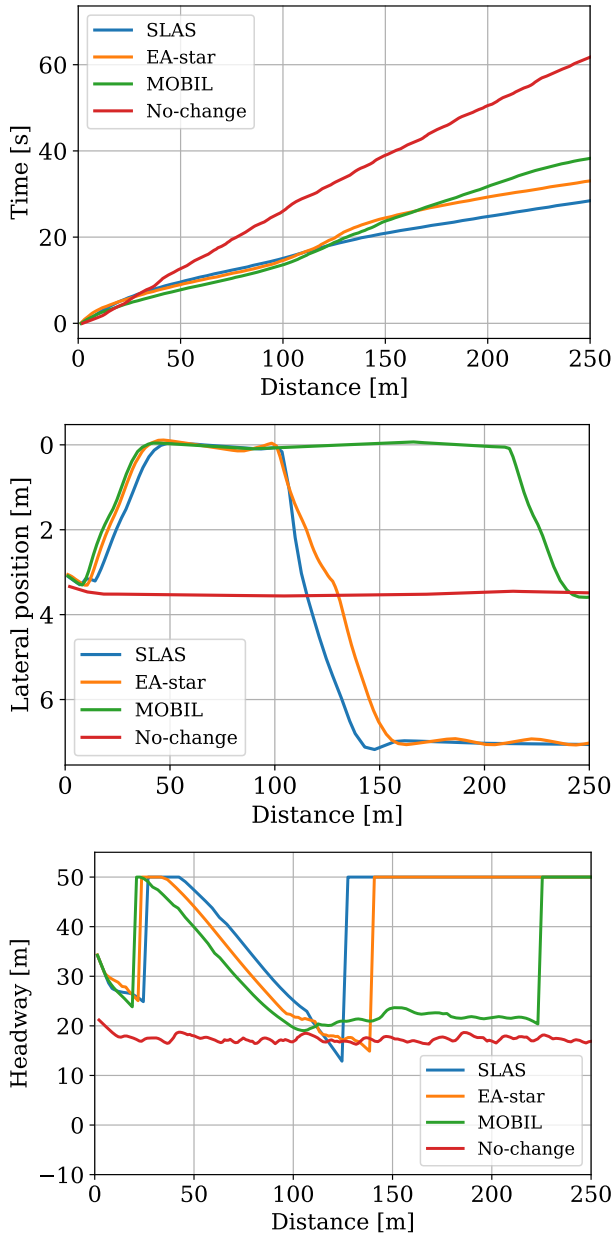

Figure 3.5: **Case Study Results**. Top: Travel time comparison. Center: Lane choice (lateral position) comparison. The center lines of lanes 0 (left), 1 (center), and 2 (right) have fixed lateral displacements of $0m$, $3.5m$, and $7m$ respectively. Bottom: Headway comparison. With no leading vehicle, the headway is restricted by the visibility range of $50m$.

Lateral Displacement: To identify the differences in lane-changing behaviors between the four approaches, the relationship between lateral and longitudinal displacements over the course of the simulation is highlighted in the center plot of Figure 3.5. In the plot, the lateral displacement of 0 corresponds to the center of lane 0 while the center of each following lane is $3.5m$ away. Comparing the performance of the four algorithms, we see SLAS and EA* showing relatively similar performances, resulting from proactive decision-making. In contrast, since MOBIL only assesses the advantage of switching to the adjacent lanes, it is unable to see the benefit of proactively switching to lane 2. This explains why EA* and SLAS start outperforming MOBIL in terms of travel time (left plot) at around the 130 [m] mark for longitudinal displacement.

As for a direct comparison between SLAS and EA*, the benefits of having a speed advisory system become apparent in this center plot. Due to speed control, SLAS is able to constantly maintain a greater headway (right plot) without having to brake significantly upon getting too close to the lead vehicle. This results in a smooth lateral displacement profile which allows the vehicle to change lanes with a minimal jerk (quantitative analysis to follow in Section 3.2.4.3) and deliver better overall timing performance (left plot).

Headway: The right plot in Figure 3.5 shows the headway maintained by the ego vehicle over the course of the simulation. In accordance with our prior discussion, MOBIL cruises behind the front vehicle, maintaining a relatively low headway until sufficient space in the adjacent lane is found to perform the lane-change maneuver. On the other hand, EA* and SLAS show a comparable headway trajectory, however, SLAS maintains a greater headway throughout and achieves the maximum headway prior to EA*. Quantitatively, SLAS maintains on average 9.43%, 36.57%,

and 113.17% more headway than the EA*, MOBIL, and No-change approaches respectively. This strong performance by SLAS can be attributed to its incorporation of safety guarantees coupled with its consideration for passenger comfort.

Distance to closest vehicle: Finally, we compare the distance that the ego vehicle maintains from the closest vehicle throughout the simulation. On average, SLAS maintains 9.28%, 32.01%, and 22.84% more distance in comparison to EA*, MOBIL, and No-change approaches respectively. These numbers are a testament to the strength of our approach resulting from the consideration of a long planning horizon coupled with speed control.

3.2.4.3 Monte Carlo Simulations

Model	**Comp. Time**	**Brake**	**Brake Jerk**	**Thr.**	**Thr. Jerk**	**Ang. Acc.**	**Ang. Jerk**
Average							
SLAS	27.84	-0.46	-0.45	0.74	0.69	2.21	5.04
EAstar	27.23	-0.56	-0.47	0.83	0.77	2.83	6.69
MOBIL	28.06	-0.62	-0.49	0.86	0.79	2.43	5.62
Standard Deviation							
SLAS	1.77	0.25	0.09	0.08	0.15	0.73	1.89
EAstar	2.85	0.38	0.15	0.11	0.21	1.48	3.93
MOBIL	3.82	0.42	0.18	0.11	0.23	0.77	2.05

Table 3.1: Monte Carlo Simulation Results for Interaction and Risk Agnostic Planning

To demonstrate the long-term performance of the three approaches (SLAS, EA*, and MOBIL), we run a series of Monte Carlo simulations on scenarios with randomized initial positions (within a range of $8m$) and velocities (within ranges of 8, 5, and 2 m/s assigned to each of the three lanes randomly) of traffic participants. The result from 50 simulations is presented in Table

3.1.

In this table, the columns represent the different evaluation metrics, the rows identify the three algorithms, and the values highlighted in green represent the best result with respect to each evaluation metric. The evaluation metrics, going from left to right in the table, are completion time (s), brake ($\mathbb{R}_{[-1,0]}$), brake jerk ($\mathbb{R}_{[-1,0]}$), throttle ($\mathbb{R}_{[0,1]}$), throttle jerk ($\mathbb{R}_{[0,1]}$), angular acceleration ($^\circ/s^2$) and angular jerk ($^\circ/s^3$). Apart from completion time, the remaining metrics, based on the commands passed to the vehicular actuators (Figure 3.2), are used to model passenger comfort. In terms of average performance, SLAS greatly outperforms the other methods when it comes to passenger comfort since it explicitly accounts for comfort in the formulation. However, it does so at a cost of slightly reduced performance in regards to travel time, when compared to EA*, since SLAS tries to strike a balance between minimizing travel time and maximizing passenger comfort. SLAS also secures the lowest standard deviation, for each of the evaluation metrics, when compared to the other methods, which points to the consistency in its long-term performance.

3.3 Interaction & Risk Aware Planning

In this section, we develop a risk-aware mixed-integer optimization-based planning (*RMOP*) framework for long-horizon speed and lane advisory on multi-lane highways. The key to successful maneuvering on highways is precise long-term risk assessment, so we leverage Conditional Value-at-Risk (CVaR) as a risk evaluation metric to systematically handle behavioral uncertainties of on-road agents. The planner is formulated as a mixed-integer program with the objective to optimize travel time and passenger comfort while maintaining safety. We show that the problem can be reformulated as a mixed-integer linear program (MILP) that allows for enhanced computational efficiency. With this reformulation, *RMOP* achieves an operation frequency of over 10 Hz for a planning horizon of 16 seconds while integrating an interaction-aware prediction model. Finally, the simulation studies in a CARLA simulation environment show that *RMOP* consistently outperforms the state-of-the-art planning methods in terms of travel time, passenger comfort, and safety, highlighting its strong potential for enhancing autonomous driving systems.

3.3.1 Overview

To motivate the problem, consider the scenario presented in Figure 3.6. Based on the predicted trajectories (shown in a lighter shade) of the neighboring vehicles (shown in orange), the ego vehicle (shown in blue) has to decide on its speed and lane change maneuvers for the near future. A greedy, or short-sighted, agent may decide to exploit the available headway in the right lane but will soon find itself stuck behind an even slower vehicle. A risk-agnostic foresighted agent may decide to switch to the left lane to minimize travel time but will end up adversely affecting

passenger comfort, or even jeopardizing safety, while overlooking the erratic behavior of the leading driver. A risk-aware foresighted agent, on the other hand, will choose to decrease its speed and patiently travel behind the slow-moving vehicle, until a safe option opens up that will give it the ability to maximize its speed while ensuring safety. This simple scenario highlights the benefit of foresight and risk evaluation in the decision-making process for an AV.

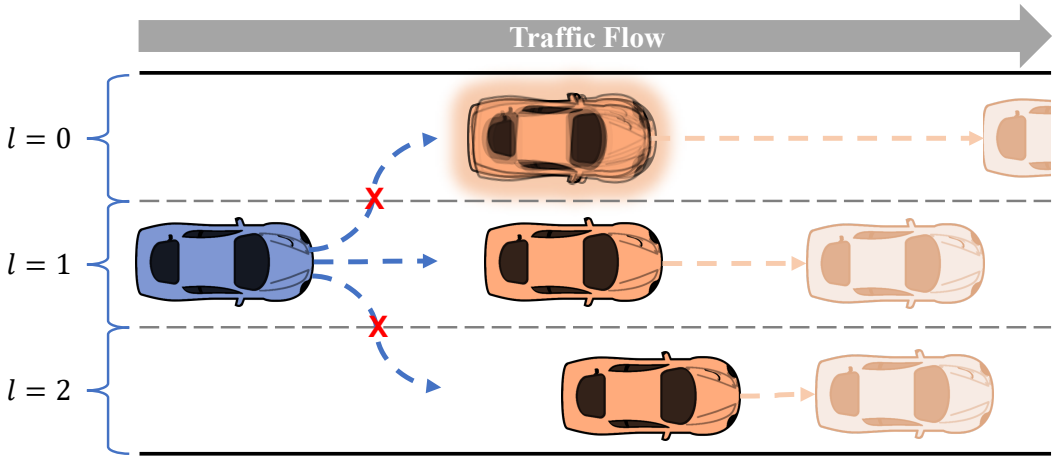

Figure 3.6: **Motivational Example**. On a multi-lane highway with traffic flow to the right, the ego vehicle (in blue) is stuck behind a slow vehicle. The lane on the right has greater headway but an even slower vehicle while the lane on the left has a fast but highly volatile (high-risk) vehicle so the prudent choice for a foresighted risk-aware agent is to continue traveling behind the slow-moving vehicle until better options become available.

Contribution

The key requirements for an AV navigation system are embodied by safe, comfortable, and efficient maneuvers, especially in the presence of volatile on-road agents. These high-level requirements, in turn, impose some key constraints on the trajectories generated by the navigation stack. Specifically, the generated trajectories need to be: (i) safe, in terms of avoiding collisions and adhering to vehicular actuation limits; (ii) kinodynamically feasible so that they can be tracked by

lower-level modules; (iii) optimal with respect to certain metrics such as risk, travel time, comfort, etc.; (iv) real-time computable to enable quick reactions to the rapidly changing environment; and, (v) adaptable to various scenarios while catering to changes in environmental and vehicle-specific parameters.

With the aim of satisfying these requirements, we introduce an optimization-based high-level behavior planning module into a modular navigation stack, which is responsible for determining target lane and speed commands for long-term planning ($> 15s$) on a multi-lane highway. The proposed method combines the previously discussed advantages of optimization-based approaches with low computational complexity, achieved through the use of binary representation of decoupled lane change dynamics. Moreover, various reformulations and algorithmic modifications are performed to facilitate numerical computations while the solution feasibility is ensured through the introduction of slack variables. In order to handle the intrinsic behavioral uncertainty of on-road agents, an interaction-aware trajectory prediction model based on a Recurrent Neural Network (RNN) architecture is deployed, and the risk associated with the behavior volatility is quantified through Conditional Value-at-Risk (CVaR) metric. Finally, a case study and in-depth Monte-Carlo testing are performed on the CARLA simulator to assess the efficacy of the proposed method.

3.3.2 Problem Formulation

We utilize the same algorithmic pipeline as well as the road, observations, and vehicular dynamics models as discussed in the prior Section 3.2.2 but for the sake of completeness, we recap them in this section while providing some additional details pertinent to the interaction and risk

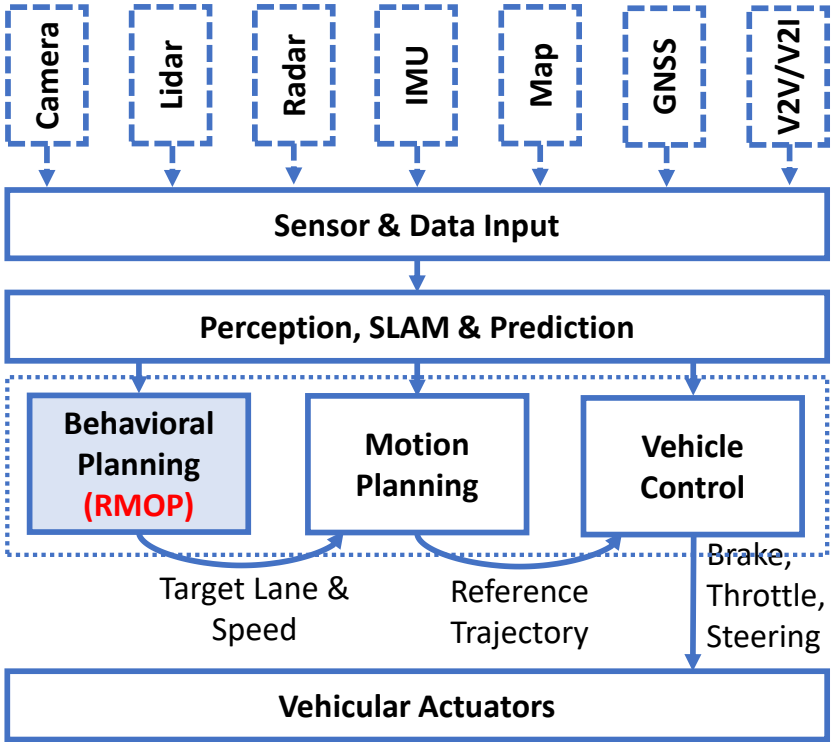

Figure 3.7: **Algorithmic Pipeline**. The raw sensory input is processed by the Perception, and Simultaneous Localization and Mapping (SLAM) modules to locate the AV within the environment. The navigation stack (outlined by the dotted rectangle), composed of our behavioral planning module, RMOP, as well as the motion planning module, NNMPC, and the vehicle control module, uses this information to control the AV through actuation commands (brake, throttle, and steering).

aware approach.

3.3.2.1 Algorithmic Pipeline

Figure 3.7 illustrates the data flow between the various algorithmic modules on board an AV. The nomenclature for the different modules constituting the navigation stack (outlined by the dotted rectangle) is taken from [8]. The modular architecture allows for the integration of external modules without disrupting the entirety of the system. More details regarding this framework can be found in Section 1.1. In this work, the goal is to develop a risk-aware behavior planning module,

referenced as *RMOP* in Figure 3.7. As for the motion planning module, we utilize the Neural Networks integrated Model Predictive Control (NNMPC) [70], as outlined in Section 3.2.2.1, due to its ability to consider the actions of neighboring vehicles during trajectory generation.

3.3.2.2 Road Model

In this work, we consider a clearly marked multi-lane one-way highway with minimal curvature, as depicted in Figure 3.6. At any given time instant k, the road speed limit is denoted by $V_r(k)$ while the number of lanes available for traveling is denoted by $N_r(k)$. Then, we define the set $\mathbb{L}(k) = \mathbb{Z}_{[0,N_r(k)]}$ that represents the set of lanes available to travel on at time instant k. Moreover, we also assign a unique identifier $l \in \mathbb{L}(k)$ to each of the lanes such that the leftmost lane, relative to the traffic flow direction, is assigned a value of $l = 0$ while each subsequent lane is assigned an increasing integer value for l, as shown in Figure 3.6.

For the planning subsystem, we utilize the Frenet coordinate system where the distance along the road is denoted by the longitudinal coordinate (s) and the distance perpendicular to the road is defined by the lateral coordinate (d). The lane identifier (l) can be obtained from the lateral coordinate (d) through a transformation denoted by $\xi \colon d \mapsto l$. Furthermore, any point (x, y) in a global Cartesian coordinate system can be transformed into the Frenet coordinate system, w.r.t the length-parameterized road curve, by the transformation:

$$\varphi \colon (x, y) \mapsto (\varphi_s, \varphi_d), \tag{3.27}$$

where the longitudinal and lateral displacements after the transformation are denoted by φ_s and φ_d

respectively [1]. The details of this transformation can be found in the Appendix of this chapter.

3.3.2.3 Observation Model

The visibility range of the ego vehicle is restricted by the sensory perception limit, denoted by R_v. Therefore, the set of vehicles visible to the ego vehicle at time instant k is given by:

$$\mathbb{O}(k) = \left\{ i \in \mathbb{Z}_{>0} \;\middle|\; |s_i(k) - s_0(k)| \leq R_v \right\} \tag{3.28}$$

where $s_i(k)$ is the longitudinal displacement of vehicle i present in the environment.

Remark 12. *For the sake of brevity, we henceforth do not show an explicit dependence of the environmental variables on the current time instant k such that $V_r(k) \to V_r$, $N_r(k) \to N_r$, $\mathbb{L}(k) \to \mathbb{L}$, and $\mathbb{O}(k) \to \mathbb{O}$. However, such a dependence should always be assumed unless specified explicitly.*

Remark 13. *Although omitted in this work to keep the formulation simple and concise, partial information resulting from occlusion or sensory limitation, can be handled by the proposed receding horizon framework (Section 3.3.3.2) since the decision is based on the recently available information, as demonstrated in Section 2.3.1.3.*

Remark 14. *There is no restriction placed on the mode of incoming data so the proposed framework (Section 3.3.3.2) can easily incorporate modern V2X communication methods, as demonstrated in Section 2.4.*

3.3.2.4 Vehicle Model

For the behavior planning layer, we utilize the linear decoupled dynamical model from Section 3.2.2.3 whose usage is justified by the minimal road curvature assumption from Section 3.3.2.2, which is typical for most highways [40]. The state ($x_0(k)$) and control input ($u_0(k)$) to the system at time instant k are:

$$x_0(k) = \begin{bmatrix} s_0(k) & l_0(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R} \times \mathbb{L} \tag{3.29}$$

$$u_0(k) = \begin{bmatrix} v_0(k) & \mathcal{L}(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}_{[0,V_0]} \times \mathbb{L} \tag{3.30}$$

where $s_0(k)$, $l_0(k)$, $v_0(k)$, $\mathcal{L}(k)$ and V_0 respectively denote the ego vehicle's longitudinal displacement, lane, speed, target lane, and maximum speed at time instant k; the subscript $_i$ indexes the vehicles on the road with $_0$ being reserved for the ego vehicle. Then, the dynamical model is formalized as:

$$s_0(k) = s_0(k-1) + \frac{v_0(k-1) + v_0(k)}{2} \cdot T_s \tag{3.31}$$

$$l_0(k) = \left\lfloor \frac{1}{N} \sum_{n=0}^{N-1} \mathcal{L}(k-n) + \frac{1}{2} \right\rfloor \tag{3.32}$$

where T_s denotes the discretization time step and, N, chosen based on the empirical data [28], corresponds to the number of time steps required for a typical lane change. Here, the longitudinal dynamics are represented by a linear acceleration model while the lane change dynamics are represented by a low-pass filter coupled with a rounding function.

3.3.3 Methodology

This section details the interaction-aware prediction model, receding horizon optimization-based behavioral planning module, and risk evaluation method to model the behavioral uncertainty of on-road agents.

3.3.3.1 Trajectory Prediction

Our proposed behavior planning framework (see Section 3.3.3.2) is able to accommodate any prediction algorithm that outputs deterministic predicted trajectories for the vehicles under consideration. For the scope of this work, however, we propose our own prediction module that has low computational complexity and delivers a robust and reliable performance when coupled with our behavior planning framework, as demonstrated in Section 3.3.4.2.

In our approach, we integrate two distinct prediction models, one specifically designed for short-term predictions that consider interactions with other entities, and another for long-term predictions that focus on independent motion forecasting. We make a distinction between short-term and long-term predictions because the uncertainty over an agent's expected position as well as its behavior grows into the future so the interactions cannot be accurately predicted well into the future. For this reason, we employ a modified constant acceleration model for long-term predictions as it will be futile to use a high-fidelity prediction model for distant future predictions.

For the short-term prediction, we use Social Generative Adversarial Networks (SGAN) [72], a Recurrent Neural Network (RNN) scheme, that captures the sequential interactions between on-road agents. This is an important consideration since each driver on the road is expected to react

to the actions of other agents on the road. SGAN is composed of a generator and a discriminator, each of which is made up of long-short-term memory (LSTM) networks. The input to SGAN is a series of vehicle positions of each interacting agent for the duration of the observation horizon and the output is a series of predicted future positions of each agent for the duration of the prediction horizon. The details on the internal workings of SGAN can be found in [72]. We use a previously trained SGAN model from [70] for our short-term predictions.

Formally, the functional mapping of the SGAN model $\phi(k)$ at time instant k is given by:

$$\phi(k)\colon \{(\bar{x}_i(a), \bar{y}_i(a))\} \mapsto \{(\hat{x}_i(b), \hat{y}_i(b))\} \tag{3.33}$$

where for each vehicle $i \in \mathbb{O}(k)$, $(\bar{x}_i(a), \bar{y}_i(a))$ corresponds to the past measurements while $(\hat{x}_i(b), \hat{y}_i(b))$ corresponds to the future predicted positions. Here, $a \in \{k - H_o + 1, \cdots, k\}$, where H_o is the observation horizon, and $b \in \{k + 1, \cdots, k + H_s\}$, where H_s is the short-term prediction horizon.

For the long-term prediction, we use a constant speed model with the speed estimate obtained from the output of the SGAN model. The short-term and the long-term prediction modules are

combined to generate the overall prediction algorithm as follows:

$$\hat{s}_i^k(j) = \begin{cases} \bar{s}_i(k), & j = 0 \\ \varphi_s(\hat{x}_i(k+j), \hat{y}_i(k+j)), & 0 < j \leq H_s \\ \hat{s}_i^k(j-1) + {}^1\!/\!_C \cdot (\hat{s}_i^k(H_s) \\ \quad - \hat{s}_i^k(H_s - C)), & j > H_s \end{cases} \tag{3.34}$$

$$\hat{l}_i^k(j) = \begin{cases} \bar{l}_i(k), & j = 0 \\ \xi(\varphi_d(\hat{x}_i(k+j), \hat{y}_i(k+j))), & 0 < j \leq H_s \\ \hat{l}_i^k(j-1), & j > H_s \end{cases} \tag{3.35}$$

Here, $\hat{l}_i^k(j)$ and $\hat{s}_i^k(j)$ respectively represent the predicted lane and longitudinal displacement for vehicle i, j time steps into the future starting from the current time instant k while $C \ll H_s$ is the averaging factor introduced to reduce noise in the speed estimate.

Remark 15. *Our modular algorithmic framework allows for the integration of external prediction and control modules availing our approach the flexibility to benefit from modern prediction and control paradigms.*

Remark 16. *The value of the hyperparameter H_s is determined empirically based on the experimental data [28].*

3.3.3.2 Behavior Planning Module

The behavior planning module, referred to as *RMOP* in Figure 3.7, outputs the speed and target lane commands for the lower-level navigation modules to follow. The main consideration while deciding on these high-level commands is to minimize travel time while prioritizing safety and comfort. We opt for an optimization-based framework that allows for the integration of risk-aware safety and dynamical constraints while accounting for actuation limits so as not to output commands that the lower-level modules are unable to follow. Specifically, we adopt a receding horizon optimization framework that consists of repeatedly solving a constrained optimization problem over a moving planning horizon, using the most recent observed data, and employing prediction and dynamical models, while only taking immediate action.

We now formalize the receding horizon optimization problem that embodies *RMOP*, discuss the modifications necessary to ensure feasibility while improving run-time complexity, and outline our risk evaluation methods.

Mixed-Integer Optimization Problem: The main objective of *RMOP* is to minimize travel time while maximizing passenger comfort. We model passenger discomfort by two main metrics: abrupt changes in speed, and successive lane changes. Therefore, in order to maximize passenger comfort, we seek to minimize these metrics corresponding to passenger discomfort whereas the travel time is minimized by maximizing overall speed. This essentially results in a multi-objective optimization problem with the goal to perform an effective tradeoff between multiple competing objectives.

In order to facilitate the formulation of the optimization problem, we represent the set of discrete planning time steps by $\mathbb{H} = Z_{[1,H]}$, where H is the planning horizon, while k corresponds to the current time instant at which the optimization problem is being solved.

Objective Function: The mathematical description of the objective function is as follows:

$$\min_{\mathcal{C}^k(j,n,i)} \sum_{j \in \mathbb{H}} \big[\gamma_v \cdot |v^k(j) - V_r| + \gamma_a \cdot |v^k(j) - v^k(j-1)| \\ + \gamma_l \sum_{n \in \mathbb{L}} |I^k(n,j) - I^k(n,j-1)| + \gamma_s \sum_{i \in \mathbb{O}} \mu^k(i,j)\big] \tag{3.36}$$

where $\mathcal{C}^k(j,n,i) = \{v^k(j), I^k(n,j), \mu^k(i,j)\}$, $\forall j \in \mathbb{H}$, $n \in \mathbb{L}$, and $i \in \mathbb{O}$ constitute the set of optimization variables while $\gamma_v > 0$, $\gamma_a > 0$ and $\gamma_l > 0$ are the scalarization parameters that respectively perform a tradeoff between maximizing speed (represented by $v^k(j)$), minimizing abrupt changes in speed, and minimizing lane changes (represented by the binary lane indicator variable $I^k(n,j)$). Specifically, γ_v governs the aggressiveness of the algorithm by prioritizing speed maximization, or equivalently, travel time minimization, whereas γ_a and γ_l combine to place an emphasis on maximizing passenger comfort by reducing speed and lane changes respectively. The scalarization parameter $\gamma_s \gg 0$ is responsible for minimizing the value of the slack variables (represented by $\mu^k(i,j)$).

Another peculiar artifact of the objective function is the inclusion of the non-differentiable absolute value function as opposed to the differentiable quadratic function in Section 3.2.3.2. This is because each absolute value function can be decomposed into a linear function coupled with two linear constraints, with the introduction of an auxiliary variable. This results in the overall problem

being a Mixed-Integer Linear Program (MILP) as opposed to a Mixed-Integer Quadratic Program (MIQP), and MILP tends to have better runtime complexity as compared to a MIQP, when the dimension of the problem is similar [74]. The mathematical details can be found in the Appendix.

Constraints: The constraints for the receding horizon optimization problem can be categorized as follows: initialization, actuation limits, system dynamics, and safety.

Initialization: The optimization problem at time instant k is initialized as follows:

$$s^k(0) = 0 \tag{3.37}$$

$$v^k(0) = v_0(k) \tag{3.38}$$

$$I^k(n,0) = \begin{cases} 1, & n = l_0(k) \\ 0, & o.w. \end{cases}, \ \forall n \in \mathbb{L} \tag{3.39}$$

Here, the ego vehicle's longitudinal displacement $(s^k(0))$ is initialized to 0 in (3.37) in order to have a moving frame of reference centered at the ego vehicle for every instance of the optimization problem while the speed $(v^k(0))$ and lane values are initialized based on the observed values in (3.38) and (3.39). We introduce a binary lane indicator variable $I^k(i,0)$ in lieu of the integer-valued lane variable $l_0(k)$ in order to speed up the optimization problem, as demonstrated in Section 3.2.3.2.

Actuation Limits: The actuation limits, which restrict outputs to agree with the physical limitations of the vehicle and environment, are defined $\forall j \in \mathbb{H}$ as follows:

$$v^k(j) \in \mathbb{R}_{[0,V_r]} \tag{3.40}$$

$$\frac{v^k(j) - v^k(j-1)}{Ts} \in \mathbb{R}_{[A_{\min}, A_{\max}]} \tag{3.41}$$

$$I^k(n,j) \in \{0,1\},\ \forall n \in \mathbb{L} \tag{3.42}$$

$$\sum_{n \in \mathbb{L}} I^k(n,j) = 1 \tag{3.43}$$

$$I^k(a, j-1) = 1 \implies \sum_{b \in \mathbb{B}} I^k(b,j) = 1, \quad \forall a \in \mathbb{L},\ \mathbb{B} = \mathbb{Z}_{[a-1,a+1]} \cap \mathbb{L} \tag{3.44}$$

Here, (3.40) and (3.41) ensure that the speed output conforms to the road speed limit and the vehicular acceleration limits respectively. As for the lane output, (3.42) defines the auxiliary *binary* lane indicator variable such that $I^k(n,j) = 1$ embodies 'selecting' lane n to travel on at planning step j while (3.43) and (3.44) impose well-posedness conditions on $I^k(n,j)$. Specifically, (3.43) ensures that *only one* lane is chosen to travel on at any planning step j while (3.44) ensures that lane change between consecutive time steps, if needed, is restricted to the set of available *adjacent* lanes, denoted by $\mathbb{B}$. The implication ($\implies$) in (3.44) can be transformed into a linear constraint with the help of the big-M method, as shown in the Appendix.

System Dynamics: The dynamical constraints, that allow for the prediction of future longitudinal positions of the ego vehicle, are defined $\forall j \in \mathbb{H}$ as follows:

$$s^k(j) = s^k(j-1) + \frac{v^k(j-1) + v^k(j)}{2} \cdot T_s \tag{3.45}$$

This essentially is the same decoupled linear acceleration dynamical model described in Section 3.3.2.4.

Remark 17. *The lane change dynamics are omitted from the constraints because, as per the analysis in Section 3.2.3.2, it is possible to reformulate the problem to have single-step lane change dynamics while augmenting the safety margin (3.48) to incorporate the distance covered while changing lanes. This results in a significant decrease in the runtime complexity.*

Safety: The safety constraints, which ensure that the ego vehicle maintains an adequate distance to each of the surrounding vehicles for $\forall j \in \mathbb{H}$, are defined as follows:

$$I^k(c, j) = 1 \implies \mathbb{S}(c, j), \ \forall c \in \mathbb{L} \tag{3.46}$$

$$\mathbb{S}(c, j) = \{|\hat{s}_i^k(j) - s^k(j)| + \mu(i, j) \geq M_i(j) \mid i \in \mathbb{O},$$
$$\hat{l}_i^k(j) = c\} \tag{3.47}$$

$$M_i(j) = \widetilde{M}_i(j) + M_i^d(j) + \gamma_r \cdot \rho_i^k \tag{3.48}$$

Here, (3.46) ensures that *if* lane c is 'selected' to travel on at planning step j, the set of constraints defined by $\mathbb{S}(c, j)$ is activated while (3.47) provides the definition for this safety constraint set.

Specifically, (3.47) ensures that the ego vehicle maintains an adequate vehicle-dependent safety margin ($M_i(j)$) to each of the vehicles traveling in lane c at planning step j. The implication constraint in (3.46) and the absolute value constraint in (3.47) can both be transformed into linear constraints, again with the help of the big-M method, as demonstrated in the Appendix.

Remark 18. *The decomposition of the absolute value constraint in (3.47) into a couple of linear constraints allows for the definition of* distinct *forward and rear safety margins, as indicated in the Appendix. In order not to be overly conservative, we define a smaller rear-end safety margin, which agrees with our real-life observations [47].*

Furthermore, in order to deal with risk-sensitive scenarios, where some vehicles may behave erratically, or even adversarially, it is necessary to introduce slack variables $\mu^k(i,j)$ to keep the problem feasible at all times. In volatile situations, the assumptions required for feasibility guarantees in Section 2.3.2.3 fail to hold, so we resort to having slackness in our safety constraints that allow us to take 'reasonable' actions even in the case of unanticipated behaviors of on-road agents.

Remark 19. *To ensure that the slack variables are 'activated' only when absolutely necessary, they are penalized heavily in the objective function by setting the associated parameter value* $\gamma_s \gg \max\{\gamma_v, \gamma_a, \gamma_l, \gamma_r, \gamma_d\}$.

The vehicle-dependent safety margin ($M_i(j)$), for vehicle i at planning step j, is composed of three distinctive parts: nominal safety margin, lateral dynamics augmentation, and risk margin. Firstly, the definition of the nominal safety margin $\widetilde{M}_i(j)$ is borrowed from [73] where Pek et. al. formalize it based on the clause from Vienna Convention on Road Traffic that states that "*A vehicle [...] shall keep at a sufficient distance [...] to avoid collision if the vehicle in front*

should suddenly slow down or stop." Secondly, we have the augmentation term $M_i^d(j)$ that subsumes the lane change dynamics, as referred to in *Remark 17*. This term incorporates the distance covered while changing lanes while incorporating a dynamic cost of deviation from the previous lane output that prevents unnecessary cancellation of initiated lane change maneuvers. Further details regarding this augmentation term can be found in Section 3.2.3.2. Finally, ρ_i^k corresponds to the risk associated with vehicle i at time instant k while γ_r is the risk-aversion factor with higher values of γ_r leading to more risk-averse (conservative) behavior. The risk quantification method for ρ_i^k is detailed in Section 3.3.3.3.

Output: The output of the optimization problem instance at time instant k is denoted by the set $\mathcal{C}_*^k(j,n,i) = \{v_*^k(j), I_*^k(n,j), \mu_*(i,j)\}$, $\forall j \in \mathbb{H}$, $n \in \mathbb{L}$, and $i \in \mathbb{O}$. From this optimized set, we extract the target control commands for the duration of the planning horizon (H) as follows:

$$u_*^k(j) = \begin{bmatrix} v_*^k(j) & \mathcal{L}_*^k(j) \end{bmatrix}^{\mathrm{T}}, \quad \forall j \in \mathbb{H} \tag{3.49}$$

where the target lane output $\mathcal{L}_*^k(j)$ is obtained from the optimal lane indicator variable $I_*^k(n,j)$ by the following relation:

$$\mathcal{L}_*^k(j) = \sum_{n \in \mathbb{L}} n \cdot I_*^k(n,j) \tag{3.50}$$

This constitutes a valid output since (3.43) ensures that only one lane is selected at any planning step j.

Finally, the target control commands are passed down to the motion planning layer in a receding horizon fashion, meaning that only the immediate control command ($u_*^k(1)$) is passed

down at time instant k and the problem is reinstantiated and solved with updated measurements and observations at time instant $k+1$.

Remark 20. *As per Remark 12, the set $\mathbb{L}$ is allowed to be time-dependent, and RMOP is compatible with such a definition. This allows our behavior planning module to accommodate an arbitrary number of lanes such that if the number of lanes is projected to change, be it through lane closure, narrowing road, etc., our framework can make timely decisions to mitigate the effects of such unanticipated events.*

Remark 21. *In order to suppress the swerving behavior, i.e. cancellation of a lane change maneuver before completion, $I^k(n,0)$ term in (only) the objective function (3.36) is replaced by $\tilde{I}^k(n,0) = \mathbb{1}_{\mathcal{L}_*^{k-1}(1)}(n)$, $\forall n \in \mathbb{L}$. This facilitates the completion of initiated lane change maneuvers unless a cancellation is necessary to maintain safety in case of unanticipated events.*

3.3.3.3 Risk Evaluation

Risk can intuitively be defined as the likelihood and severity of the damage that the ego vehicle may suffer in the future [16]. Formally, it is captured by *Definition 1*.

Definition 1 (Risk)**.** *For a random variable $\mathcal{X}$ defined on the probability space $(\Sigma, \mathcal{F}, \mathbb{P})$ with Σ, $\mathcal{F}$, and $\mathbb{P}$ corresponding to the sample space, σ-algebra, and probability measure over $\mathcal{F}$ respectively, risk (ρ) is defined as [58]: $\rho\colon \mathcal{X} \to \mathbb{R}$.*

Therefore, risk is basically a functional mapping of a random variable to a real number. However, owing to the generality of this concept, it is natural to expect a number of different mathematical models to represent risk. In [58], Majumdar et. al. provide a comprehensive overview of

some of the most common risk metrics found in the Robotics literature while arguing that the only 'sensible' metrics for Robotics applications are the ones that fall under the category of *distortion metrics (DMs)* [78, 79], a subset of *coherent risk metrics (CRMs)* [80]. Such a categorization is based on a set of axioms proposed in the Finance literature, where the concept of risk originated. Based on the arguments presented in [58], we opt not to use commonly used risk metrics such as Mean-Variance or Entropic Risk Measure as they are not CRMs. We decide instead to go with *Conditional Value-at-Risk (CVaR)* as our preferred risk metric since it is a valid DM and therefore enjoys the associated axiomatic properties, making it an appropriate metric for our application. Intuitively, $CVaR_\alpha(\mathcal{X})$ is the expected value of the random variable $\mathcal{X}$'s distribution in its upper $(1-\alpha)$-tail. Formally, it is defined as follows:

Definition 2 (Conditional Value-at-Risk (CVaR))**.** *Let $\mathcal{X}$ be a bounded-mean random variable i.e. $\mathbb{E}[|\mathcal{X}|] < \infty$. Let, also, the cumulative distribution function be defined as $F(x) = \mathbb{P}(\mathcal{X} \leq x)$. Then, the* Value-at-Risk (VaR) *at confidence level $\alpha \in (0,1)$, denoted by VaR_α, is defined as the $(1-\alpha)$-quantile of $\mathcal{X}$ i.e. $VaR_\alpha(\mathcal{X}) = \inf\{x \mid F(x) \geq \alpha\}$. The* Conditional Value-at-Risk (CVaR) *at confidence level $\alpha \in (0,1)$, which measures the expected value of $\mathcal{X}$ in the $(1-\alpha)$-tail, is defined as [81]:*

$$CVaR_\alpha(\mathcal{X}) = \inf_{w \in \mathbb{R}} \mathbb{E}\left\{ w + \frac{[\mathcal{X} - w]^+}{1-\alpha} \right\} \tag{3.51}$$

where $(z)^+ = \max\{z, 0\}$ represents the positive part of z, and $\mathbb{E}\colon \mathcal{X} \to \mathbb{R}$ is the Expectation operator.

In our work, we evaluate the overall risk associated with a vehicle as a convex combination of the risk associated with its longitudinal and lateral motions. The overall risk for vehicle i is

therefore given by:

$$\rho_i^k = \beta \cdot \text{CVaR}_{\alpha_s}(\mathcal{A}_i^k) + (1-\beta) \cdot \text{CVaR}_{\alpha_d}(\mathcal{W}_i^k) \tag{3.52}$$

where $\mathcal{A}_i^k$ and $\mathcal{W}_i^k$ respectively correspond to the *absolute* linear acceleration and *absolute* angular speed random variables for vehicle i at time instant k while $\beta \in (0,1)$ serves as the tradeoff (or normalizing) factor between the two risk modalities. The CVaR confidence levels for longitudinal and lateral risk are denoted by α_s and α_d respectively, which are tuned based on the empirical data distribution.

Remark 22. *We evaluate CVaR for the stochastic processes $\mathcal{A}_i^k$ and $\mathcal{W}_i^k$, indexed by time instant k, using the empirical estimator from [82], where the authors derive rigorous concentration bounds for the estimator.*

Example: To delineate the effects of risk evaluation on the decision-making strategy, we refer back to the motivational example presented in Figure 3.6. We simulate this example with and without risk evaluation, and present the result in Figure 3.8.

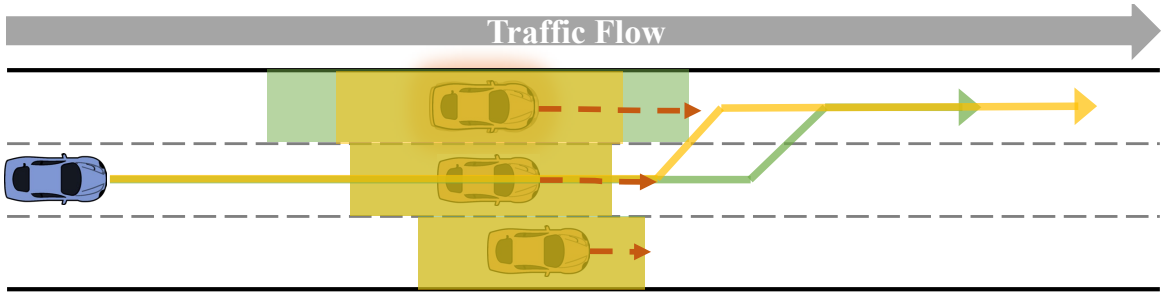

Figure 3.8: **Risk Evaluation for Behavior Planning**. The ego vehicle (in blue) waits for a longer time under a risk-sensitive strategy (shown with the green trajectory) before changing to the fast lane, as compared to the risk-agnostic strategy (shown with the yellow trajectory).

In Figure 3.8, the length of the dashed red arrows represents the mean speeds of the vehicles while the green and yellow arrows represent the planned trajectory with and without risk

evaluation, respectively. The yellow rectangles represent the risk-agnostic safety margin while the green rectangle represents the risk-sensitive safety margin. The vehicle on the left lane has speed $v_0 \sim \mathcal{N}(\mu_v, \sigma_v^2)$ while the remaining two vehicles are going at constant speeds, hence, the overlapping of green and yellow rectangles.

The risk-sensitive strategy demonstrates a more defensive behavior, due to the augmented safety margin, where the ego vehicle takes longer to switch to the fast-moving lane which allows the faster-moving volatile vehicle to move further. Contrariwise, the risk-agnostic approach is more aggressive, which may jeopardize safety in high-risk scenarios. Although the overall travel distance, depicted by the length of the trajectories, seems better for the aggressive risk-agnostic strategy, it becomes apparent in Section 3.3.4.3 that a risk-sensitive strategy can help improve travel time in highly volatile situations.

3.3.3.4 Computational Complexity Reduction

In order to speed up the optimization process, the warm-start technique, based on the solution from the previous time step, is implemented in order to provide an initial reference for the optimizer, while also having a lazy implementation for the lane change constraints (3.44). The justifications and details of these algorithmic modifications can be found in Section 3.2.3.2.

3.3.4 Results

This section details the experimental setup, followed by a qualitative and quantitative comparative analysis with respect to the state-of-the-art behavior planning methods. Specifically,

RMOP is compared against the Speed and Lane Advisory System (SLAS) from Section 3.2, Extended-Astar (EA*) [55], MOBIL [57], and no lane-change (No-Change) methods in a single-scenario as well as Monte-Carlo simulation studies.

3.3.4.1 Experimental Setup

The experimental setup, illustrated in Figure 3.9, consists of the CARLA simulator (Version 0.9.11) [77], the *RMOP* module (Section 3.3.3.2), and the planner and controller modules [70]. We utilize the Gurobi Optimizer (Version 10.0.1) [49] to solve the optimization problem for RMOP. The simulations are carried out on a computer with an AMD Ryzen 7 5800h × 16 and NVIDIA GeForce RTX 3080, running Ubuntu 20.04 LTS. On average, the optimization process takes $0.08s$ per step, and the maximum time limit for the optimizer is set to $0.1s$, demonstrating the potential for real-time applicability.

3.3.4.2 Case Study

Figure 3.10 depicts the test case scenario that is used to perform a comparative analysis, in terms of various evaluation metrics between *RMOP*, SLAS, EA*, MOBIL, and no lane-change methods. The scenario is composed of a four-lane highway segment with lanes labeled as $0-3$ relative to the direction of traffic flow, and the rightmost lane (lane 3) is reserved for merging vehicles so only lanes $0-2$ are available for traveling. The ego vehicle is initialized to move behind a slow vehicle in lane 1 so in order to minimize travel time, it needs to switch to lane 0, which has faster-moving traffic and greater headway, as shown in Frame 1. Once the ego vehicle

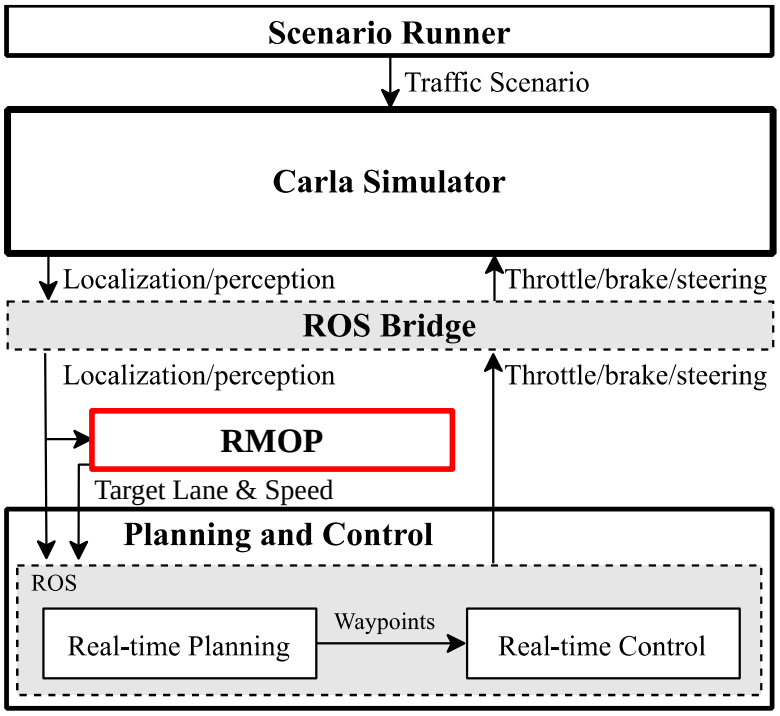

Figure 3.9: **Implementation Setup**. The simulation scenario, generated by the Scenario Runner, is passed on to the CARLA Simulator, which communicates with the *RMOP* and the Planning and Control ROS (Robot Operating System) nodes, via the ROS bridge node, at a frequency of 10 Hz.

moves to lane 0, the vehicle traveling ahead suddenly stops, resulting in a *high-risk situation*. The ego vehicle has to react to this situation appropriately and move back to lane 1 as soon as possible, as shown in Frame 2. Once the ego vehicle is back in lane 1, it is following another slow-moving vehicle, with no headway in any of the adjacent lanes, so it has to resort to following the slow-moving vehicle until a space to overtake opens up in any of the adjacent lanes. As soon as it passes the stopped vehicle in lane 0, the ego vehicle is able to perform an overtake maneuver by exploiting the available headway in lane 0, as shown in Frame 3. However, as soon as the ego vehicle speeds up in lane 0, after the overtake, another vehicle from lane 1 suddenly swerves into lane 0 ahead of the ego vehicle, resulting in another *high-risk situation*. The ego vehicle has to promptly react to this situation and switch back to lane 1, as soon as it notices this unfavorable development, as

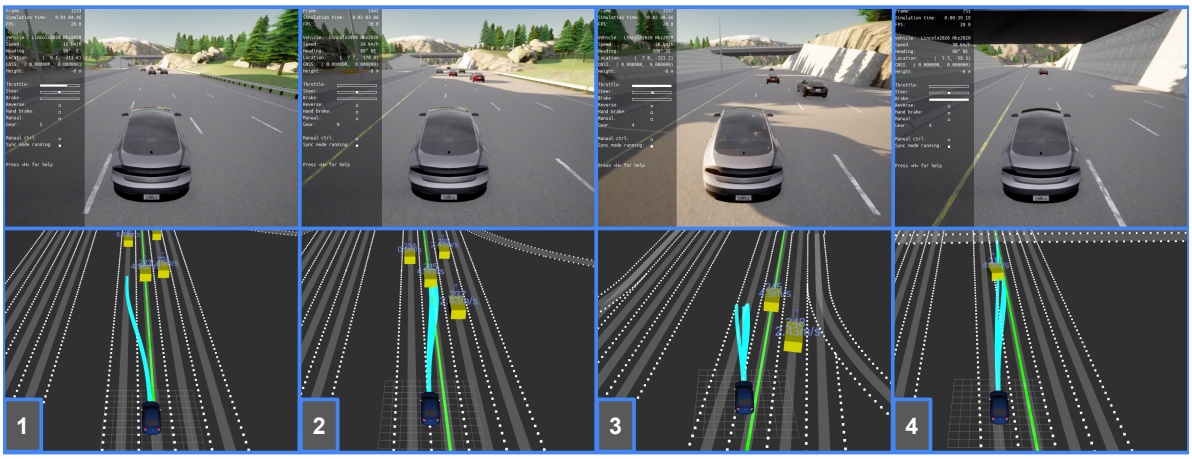

Figure 3.10: **Test Scenario for the Case Study**. On a stretch of multi-lane highway, lanes 0 (left), 1 (center), and 2 (right) are available for traveling while lane 3 (right-most) is reserved for merging traffic. The expected motion of the ego vehicle during the simulation is depicted through the numbered frames.

shown in Frame 4.

The comparative analysis evaluation metrics consist of travel time, lateral displacement, headway, and distance to the closest vehicle, all with respect to the longitudinal displacement of the center of mass of the ego vehicle (COM).

Simulation parameters include a simulation step size of $0.05s$ (simulation frequency of $20Hz$); speeds of vehicles in lanes 0, 1, and 2 set to 8, 5, and 2 m/s respectively, with a speed limit of $15m/s$ (V_r); highway segment length of $350m$, with lane width of $3.5m$; and, a sensor visibility range of $R_v = 50m$. The *RMOP* parameters are set as follows: $T_s = 0.4s$, $H = 40$, $N = 3$, $A_{min} = -5m/s^2$, $A_{max} = 3.5m/s^2$, $\gamma_1 = 0.5$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.01$. These parameters can be adjusted to achieve aggressive or defensive behavior of the algorithm.

Travel Time: The left plot of Figure 3.11 illustrates the travel time of the ego vehicle relative to the longitudinal displacement of COM. *RMOP* starts off slower as compared to the other methods

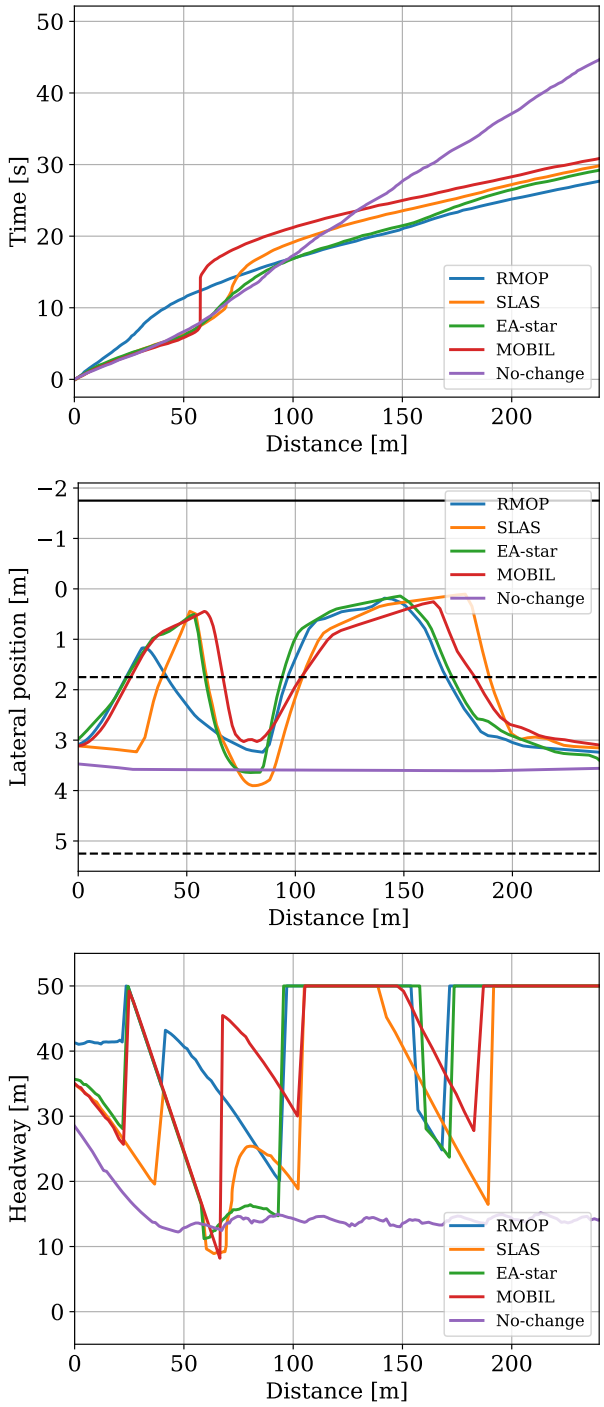

Figure 3.11: **Case Study Results**. The x-axes represent the longitudinal displacement of the center of mass of the ego vehicle (COM) while the y-axes are defined as follows. Top: Travel time; Center: Lateral displacement of COM - The solid horizontal line depicts the road boundary while the dashed horizontal lines represent the lane boundaries; Bottom: Headway - Restricted to the visibility range of $50m$, when no leading vehicle is present.

due to its emphasis on passenger comfort but as soon as the high-risk situation of car stopping develops, it is able to recognize and react to it earliest, allowing it to make up for its slow start. Due to its ability to perceive risk, it doesn't have to slow down significantly and this, in turn, allows it to achieve minimum overall time. Quantitatively, *RMOP* has an improvement of 7.20%, 5.48%, 10.48% and 41.16% against SLAS, EA*, MOBIL and No-change methods respectively.

Lateral Displacement: The center plot of Figure 3.11 outlines the lateral and longitudinal displacements of COM. In the plot, the horizontal solid represents the left road boundary while the dashed horizontal lines represent the lane boundaries. It is evident from the plot that *RMOP* is able to perceive the risk of stopping vehicle earliest, and switches back to lane 1 at around $30m$ longitudinal displacement mark, without even fully merging into lane 0. This allows it to maintain its speed without much adjustment, as indicated previously. As for the response to swerving vehicle, *RMOP* is again able to recognize the situation earliest at around the $150m$ longitudinal displacement mark and react accordingly to maintain a smooth lane change profile, with EA* coming in as a close second due to its risk-sensitive formulation.

Headway: The right plot of Figure 3.11 depicts the ego vehicle's headway, which is restricted to a maximum value of its visibility range of $50m$, with respect to the longitudinal displacement of COM. *RMOP* continues to maintain a greater headway for almost the entirety of the simulation, in comparison to the other methods due to its proactive risk assessment and decision-making. Specifically, compared to SLAS, EA*, MOBIL, and No-change methods respectively, *RMOP* has an improvement of 125.73%, 79.54%, 145.22%, and 64.44% in terms of *minimum headway*, and

an improvement of 38.60%, 22.90%, 29.56%, and 176.36% in terms of the *average headway* over the course of the simulation.

Distance to closest vehicle: Finally, we provide a quantitative analysis of the distance between the ego vehicle and its nearest neighboring vehicle throughout the simulation. On average, *RMOP* maintained a greater distance to its closest neighbor as compared to SLAS, EA*, MOBIL, and No-change methods by 63.75%, 47.97%, 58.14%, and 55.77% respectively. This data makes a strong case for the quality of *RMOP* in maintaining safety and preserving passenger comfort even in the presence of volatile on-road agents.

3.3.4.3 Monte Carlo Simulations

To have an in-depth understanding of the performance of *RMOP*, SLAS, EA*, and MOBIL approaches in high-risk situations, a series of 100 Monte-Carlo simulations are conducted on randomized scenarios. The randomization is performed with respect to the baseline scenario outlined in Figure 3.10. The initial positions of the vehicles are allowed to vary within a range of $8m$ while the mean speeds of vehicles in the three lanes are allowed to randomly vary within ranges of $8m/s$, $5m/s$, and $3m/s$. Moreover, in order to develop high-risk scenarios, each vehicle is assigned one of the following four behaviors at random: travel at the constant mean speed, have a variation around the mean speed between time steps (jerk), stop at a random location, and swerve into an adjacent lane at a random location. The results from these simulations are outlined in Table 3.2.

In Table 3.2, the columns represent the various evaluation metrics while the rows correspond to the specific algorithms under evaluation. The cells highlighted in green indicate the optimal

Model	Time	Succ. %	Coll. %	Time-out %	Brake Avg	Thr. Avg	Acc. Max	Brake Jerk Avg	Thr. Jerk Avg	Jerk Max	Ang. Acc. Avg	Ang. Acc. Max	Ang. Jerk Avg	Ang. Jerk Max
Average														
RMOP	29.7	99	1	0	-0.27	0.67	1.94	-0.43	0.53	6.87	1.41	15.71	2.76	31.73
SLAS	30.16	90	7	3	-0.67	0.8	2.13	-0.53	0.63	5.02	1.98	23.37	4.04	51.71
EAstar	29.74	92	7	1	-0.8	0.91	2.34	-0.58	0.76	7.92	1.99	20.37	3.92	40.9
MOBIL	31.47	79	7	14	-0.72	0.9	2.39	-0.58	0.74	6.79	1.71	19.75	3.39	41.04
Standard Deviation														
RMOP	3.72	0.1	0.1	0	0.33	0.06	0.32	0.12	0.16	5.39	0.95	13.37	1.84	25.45
SLAS	3.84	0.3	0.26	0.17	0.38	0.09	0.32	0.13	0.16	3.89	1.28	17.35	2.75	42.87
EAstar	6.53	0.27	0.26	0.1	0.5	0.12	0.44	0.26	0.3	5.4	0.95	10.28	1.94	22.09
MOBIL	8.79	0.41	0.25	0.35	0.45	0.12	0.34	0.26	0.33	4.93	1.08	13.27	2.34	32.37

Table 3.2: Monte Carlo Simulation Results for Interaction and Risk Aware Planning

performance for each metric. The evaluation metrics listed in the table, ordered from left to right, are as follows: completion time (s); success ($\%$); collision ($\%$); timeout ($\%$); average braking ($\mathbb{R}_{[-1,0]}$); average throttling ($\mathbb{R}_{[0,1]}$); maximum acceleration (m/s^2); average braking jerk ($\mathbb{R}_{[-1,0]}$); average throttling jerk ($\mathbb{R}_{[0,1]}$); maximum jerk (m/s^3); maximum angular acceleration (rad$/s^2$); average angular jerk (rad$/s^3$); and, maximum angular jerk (rad$/s^3$). Here, collision ($\%$), time-out ($\%$), and success ($\%$) correspond respectively to the percentage of simulation runs in which the ego vehicle got into a collision with another vehicle, was unable to reach the end of the highway segment within the allocated time of $80s$, and was able to reach the end of the highway segment without any collisions. Apart from these percentages, and completion time, which corresponds to the time required by the ego vehicle for a 'successful' run, the remaining metrics are utilized to quantify passenger comfort.

In terms of the average results depicted in Table 3.2, *RMOP* outperforms the other methods in every single metric apart from maximum jerk, which is a testament to our method's efficacy in simultaneously optimizing for travel time and passenger comfort, especially in high-risk situations.

As for maximum jerk, *RMOP* has a worse performance compared to the risk-agnostic methods (SLAS and MOBIL) because it has to perform aggressive maneuvers at times to avoid collisions and get out of tricky situations. The risk-agnostic methods have higher collision and time-out percentages just because they are unable to deal with high-risk situations and end up getting into a collision or getting stuck behind a stopped vehicle indefinitely. One interesting fact observed from these results is that even though the inclusion of risk may apparently add a level of defensiveness to the algorithm, it actually yields travel time improvements over the course of the simulation, because the ability to anticipate and react to high-risk situations, with a certain level of foresight, allows the vehicle to take timely actions to preserve safety and minimize overall travel time.

As for the standard deviation results, which serve to identify a level of consistency in the long-term performance over different simulation runs, *RMOP* again outperforms other methods, especially in averaged metrics over the course of each run. It underperforms some of the methods in terms of the maximization metrics, which can again be explained based on the need to perform evasive safety maneuvers in high-risk situations. Therefore, *RMOP* consistently and reliably demonstrates superior performance compared to the other methods, especially in high-risk situations.

Remark 23. *The modularity in* RMOP*'s design enables it to be integrated into any existing system where a long-horizon speed and lane advisory can be beneficial. While the simulation studies reported improvements in various travel time and passenger comfort related metrics for a specific system architecture, with fixed motion planner and controller modules, we expect the improvements to be sustained for other similar systems, settings, and modules.*

3.4 Conclusion

In this chapter, we address the multi-lane highway maneuvering scenario first with an interaction and risk agnostic approach and then incorporate interaction and risk evaluation strategies into the formulation. In the first part of this chapter, we detail our interaction and risk agnostic approach. Specifically, we propose a novel behavior planning module, that outputs strategic target lane and reference speed commands, and incorporate it with a state-of-the-art motion planning and control framework. We formulate the approach as a receding horizon mixed integer optimization with the goal of minimizing travel time while accounting for passenger comfort for a long planning horizon. In order to reduce the computational overload, we reformulate the problem by replacing integer variables with binary ones and further incorporate various modifications to aid numerical computations. We also carry out a detailed comparative analysis to demonstrate the performance of our approach on the CARLA simulator.

In the second part, we detail our enhanced interaction and risk aware approach. Specifically, we present an efficient risk-aware behavior planning framework that is integrated into the navigation architecture detailed in the first part of the chapter. As before, we formulate the approach as a receding horizon mixed-integer program but the new formulation is reformulated to yield a mixed-integer linear program (MILP) to further alleviate the computational burden. In order to have a precise risk assessment, we utilize Conditional Value-at-Risk (CVaR) as a risk evaluation metric, while integrating an interaction-aware trajectory prediction algorithm, to systematically handle behavioral uncertainties of on-road agents. To assess the merits of our proposed architecture, we conduct a comprehensive comparative analysis against various state-of-the-art planning

frameworks as well as our interaction and risk agnostic framework on the CARLA simulator.

Future work in this area will explore the incorporation of delays and uncertainty measures in the perception and localization modules to evaluate the robustness properties of the proposed approach before testing it on a physical vehicle in real-world scenarios.

3.5 Appendix

3.5.1 Frenet Coordinate System

In order to transform the position of a vehicle from Cartesian frame (x, y), to Frenet frame (s, d), we first need a reference curve (center line) - which can be a lane boundary, reference trajectory, etc. in the autonomous driving case - with respect to which we perform the transformation. This reference is typically given as a parametric curve, such as a spline, as shown by $s(t)$ in Figure 3.12 [1]. Based on this parametric curve representation, we can redefine the vector $\vec{x}$ defined in the

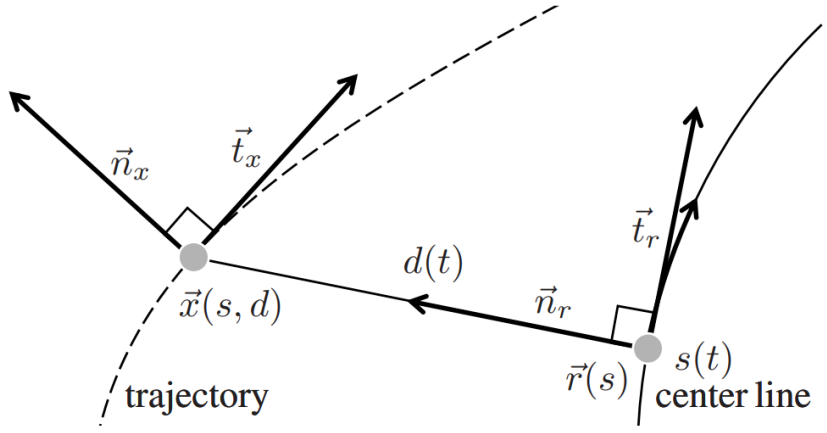

Figure 3.12: **Frenet Frame Transformation** [1]. The representation of vector $\vec{x}$ in the Frenet coordinate system w.r.t. the reference curve, denoted by $s(t)$, is given by $\vec{x}(s, d)$.

Cartesian coordinate system (x, y), for the generic case, as follows:

$$\vec{x}(s(t), d(t), l(t)) = \vec{r}(s(t)) + l(t) \cdot \vec{t_r} + d(t) \cdot \vec{n_r} \tag{3.53}$$

To simplify this notation, t is chosen such that $\vec{r}(s(t))$ is the *closest point* on the curve $s(t)$ to the point defined by the vector $\vec{x}$. In this case, the tangential component zeros out since the point is orthogonal to the curve and we are only left with the longitudinal and lateral components as follows:

$$\vec{x}(s, d) = \vec{r}(s(t)) + d(t) \cdot \vec{n_r} \mid_{t=t^*} \tag{3.54}$$

where

$$t^* = \{t \mid \min_t \|\vec{x}(s(t)) - \vec{r}(s(t))\|\}.$$

This is effectively the Frenet frame transformation, as indicated in Figure 3.12.

3.5.2 Flooring Constraint

For $y \in \mathbb{Z}$ and $x \in \mathbb{R}$, the constraint $y = \lfloor x \rfloor$ can be represented by the following linear constraints:

$$y \leq x, \quad y + 1 \geq x + \epsilon$$

where $\epsilon > 0$ accounts for the feasibility tolerance.

3.5.3 Implication Constraint

For $a, b \in \{0, 1\}$, the constraint $(a = 1) \implies (b = 1)$ can be represented with a linear constraint as follows:

$$b + M \cdot (1 - a) \geq 1 - \epsilon$$

where $M \gg 0$ (big-M) and $\epsilon > 0$ accounts for numerical errors (chosen to be 0.1 in our implementations).

3.5.4 Absolute Value Constraint

For $\Delta s, L_s \in \mathbb{R}$, the constraint $|\Delta s| - L_s \geq 0$ can be represented as:

$$\Delta s \geq L_s \vee \Delta s \leq -L_s.$$

This can further be generalized, as done in our implementation, to have different forward and rear safety margins as:

$$\Delta s \geq L_s^f \vee \Delta s \leq -L_s^r$$

where L_s^f and L_s^r are forward and rear safety margins respectively. This can be represented with the following linear constraints:

$$\Delta s + M \cdot c - L_s^f \geq 0$$

$$-\Delta s + M \cdot (1 - c) - L_s^r \geq 0$$

where $M \gg 0$ (big-M) and $c \in \{0, 1\}$ is responsible for making a choice between the two constraints.

3.5.5 Absolute Value Cost

For $x(j) \in \mathbb{R}$, $y(j) \in \mathbb{R}$, and $j \in \mathbb{Z}_{[1,H]}$, the cost function:

$$\min_{x(j)} \sum_{j} |x(j) - y(j)|$$

can be decomposed into the following linear program:

$$\begin{aligned} \min_{a(j)} \quad & \sum_{j} a(j) \\ \text{s.t.} \quad & x(j) - y(j) \leq a(j) \\ & y(j) - x(j) \leq a(j) \end{aligned}$$

where $a(j) \in \mathbb{R}$ is the auxiliary optimization variable.

Chapter 4: Crash Mitigation

Many modern vehicles are already equipped with collision warning and braking systems that help to reduce the number and severity of rear-end collisions [83]. These systems typically utilize Autonomous Emergency Braking (AEB) systems [84, 85] to provide collision warnings and perform partial or full braking based on the longitudinal distance from the leading vehicle. However, in terms of planning and control, these systems are limited strictly to braking behaviors. This restricts their ability to perform complex maneuvers involving an intricate interplay between throttling, braking, and steering commands, while considering vehicle dynamics, in order to handle collision-prone situations.

This chapter looks at developing a more complete steering and acceleration control system capable of reducing the number and severity of collisions in a wider class of situations. Such a system could potentially be used, like collision warning and braking systems, as an advanced driver assistance system (ADAS) or it could be coupled with full autonomous driving (AD) software system as a fail-safe protection. As opposed to developing methods for specific situations, this proposed system can broadly be applied to any autonomous driving scenario, while augmenting the existing planning framework, to provide fail-safe protection.

As part of this work, we propose a risk-aware crash mitigation system (RCMS), to aug-

ment any existing motion planner (MP), that enables an autonomous vehicle to perform evasive maneuvers in high-risk situations and minimize the severity of collision if a crash is inevitable. In order to facilitate a smooth transition between RCMS and MP, we develop a novel activation mechanism that combines instantaneous as well as predictive collision risk evaluation strategies in a unified hysteresis-band approach. For trajectory planning, we deploy a modular receding horizon optimization-based approach that minimizes a smooth situational risk profile, while adhering to the physical road limits as well as vehicular actuator limits. We demonstrate the performance of our approach in a realistic simulation environment.

This chapter is structured as follows: Section 4.1 provides an overview of the problem; Section 4.2 summarizes the existing methods available in the literature to solve the problem; Section 4.3 formalizes the problem; Section 4.4 details the various components of the proposed approach, and highlights its implementation and analytical details; Section 4.5 analyzes the performance of the proposed approach by running a series of tests and providing a discussion on the results; Section 4.6 summarizes the approach while providing concluding remarks and future outlook; and, Section 4.7 presents a detailed discussion on the collision avoidance strategies that were considered as part of this work.

4.1 Overview

To motivate our proposed framework, consider the scenario presented in Figure 4.1, where an ego vehicle has vehicles traveling on its side as well as behind. If one of the vehicles on the adjacent lanes swerves into the ego vehicle's lane (possibly due to a blind spot), the ego vehicle has

to perform a drastic maneuver to avoid a collision, or minimize the severity of it, if it is unavoidable. It cannot simply use emergency braking [83] due to the trailing vehicle, and it cannot swerve into the other lane due to the presence of the other vehicle. Therefore, it has to finesse its way around the vehicles, depending on the space availability and actuation limits, to take the least risky action. To handle such intricate scenarios, we develop RCMS.

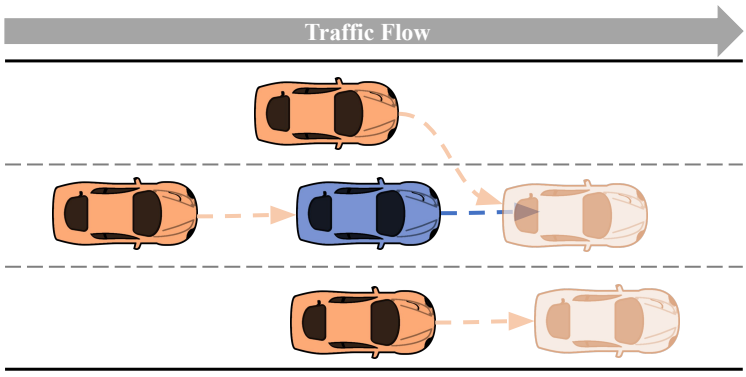

Figure 4.1: **Motivational Example**. The ego vehicle (in blue) is cruising on a highway, with a vehicle following behind, and two vehicles traveling on either side in adjacent lanes. If one of the vehicles in the adjacent lanes suddenly swerves towards it, the ego vehicle is put in a high-risk situation where a collision may be unavoidable. The ego vehicle will have to make an evasive maneuver that ideally avoids a crash but if it's inevitable, then choose an action that minimizes the severity of collision.

The design of RMCS involves two components: (i) a novel activation mechanism and, (ii) a trajectory generation method. The activation mechanism considers both the instantaneous as well as predictive collision risk evaluation strategies under a hysteresis band to trigger the trajectory generation method. The trajectory generation method performs situational risk analysis through smooth functional evaluation and optimizes it in a receding horizon optimization-based framework. Due to its modular nature, RCMS has the capability to augment any existing motion planning framework while incorporating modern prediction algorithms.

4.2 Literature Review

While crash mitigation in an AD setting depends also on timely detection [86, 87] and prediction [88], we focus on the planning and decision-making aspect of the problem. In that regard, most existing literature focuses on risk-based techniques. Lee and Kum [89] evaluate situational risk through a predictive occupancy map (POM) and use a sampling-based technique for trajectory generation. They employ a simplistic threshold-based activation mechanism and a fixed time-based deactivation. The sampling-based approach restricts the solution search space to the set of predetermined samples which may be very limiting in high-risk scenarios, where the difference between a trajectory that avoids a collision to the one that does not may be minimal. Moreover, the fixed time-based deactivation runs the risk of deactivating the system before getting the vehicle to a safe state or leaving it in an even worse situation during the transition.

Wang et al. [90] present a real-time Model Predictive Control (MPC) algorithm that uses a potential crash severity index (CSI) to select the least dangerous action. However, in an effort to improve the computational complexity, they linearize the dynamical model and subsequently convexify the optimization problem, potentially adversely affecting the feasibility of the control actions, which may prove detrimental in a collision avoidance situation. Moreover, their work does not consider when to activate the system, limiting its usefulness for ADAS applications.

Shang et al. [91] combine artificial potential fields with MPC and verify that against a Hamilton Jacobi reachability (HJ) based approach. They use a *non-smooth* energy-based cost function which may adversely affect the computational complexity, as no timing statistics for the algorithm are provided, while the HJ-based approach notably has high computational complexity so a simpler

unicycle dynamical model is used. Moreover, their relatively simple rule-based activation mechanism is prone to running into issues pertaining to ineffective triggering, as discussed in Section 4.4.1.

Qin et al. [92] integrate a high-fidelity model with tire slip forces while considering MAIS (Maximum Abbreviated Injury Severity) 3+ probability as CSI while using time-to-collision (TTC) to switch between three modes of operation: path following, crash avoidance, and crash mitigation. However, such an approach raises concerns regarding its real-time applicability since no remarks regarding its computational complexity were made in the paper. Furthermore, as mentioned previously, using a simplistic condition-based triggering mechanism is not adequate for effective switching between the crash mitigation system and the regular motion planner.

In this work, we address the aforementioned drawbacks of existing methods while benefiting from the efficiency of optimization-based techniques by designing RCMS, composed of an activation mechanism and a modular trajectory generation component. The activation mechanism combines instantaneous as well as predictive collision risk evaluation in a hysteresis band to facilitate a smooth transition between RCMS and MP, which is important since the goals of the two systems are fundamentally different, as discussed in Section 4.4.1. The trajectory generation component minimizes the situational risk, evaluated through a smooth function while considering actuation, dynamical, and road limits, as detailed in Section 4.4.2. We verify the performance of our approach while providing timing statistics to ascertain its real-time applicability, in the simulation of high-risk collision-prone scenarios in Section 4.5.2.

4.3 Formulation

In this section, we first recap the algorithmic pipeline, that has been discussed in previous chapters, before elaborating upon the road, observation, and vehicle models that are used within the RCMS framework.

4.3.1 Algorithmic Pipeline

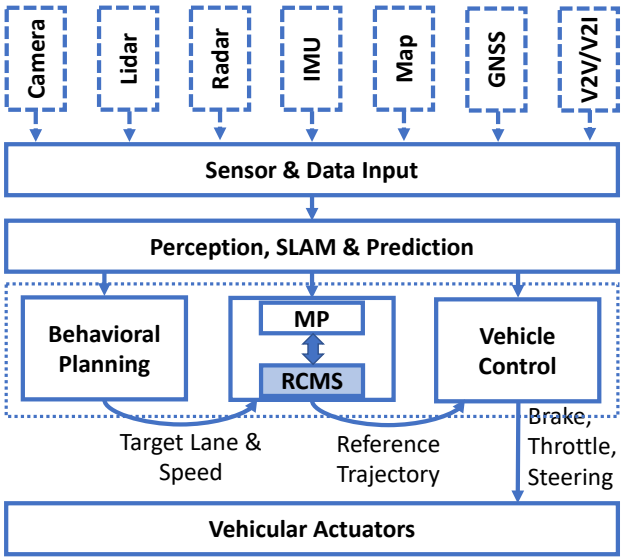

Figure 4.2: **Algorithmic Pipeline**. The Perception and Simultaneous Localization and Mapping (SLAM) modules process the raw sensory input to locate the AV within the environment. The navigation stack (outlined by the dotted rectangle), composed of behavioral planning, motion planning as well as our crash mitigation system, and vehicle control modules, uses this information to control the AV through actuation commands (brake, throttle, and steering).

Figure 4.2 provides a visual representation of the data flow among various algorithmic modules onboard an autonomous vehicle (AV). The navigation stack, enclosed by the dotted rectangle, consists of several modules whose nomenclature follows the conventions established in [8]. This modular architecture facilitates the seamless integration of external modules while maintaining the

overall system integrity. Additional insights into this framework can be found in Section 1.1. The primary objective of this research is to augment any existing motion planning module with a crash mitigation system, as depicted in Figure 4.2.

Notation

We reserve the variable $k \in \mathbb{R}$ to represent the current time instant such that any variable defined as $\zeta(k)$ represents the evaluation of ζ at the current instant.

4.3.2 Road Model

Even though the proposed methodology can be applied to any general setting, we limit the scope of this work to a multi-lane highway setting, as depicted in Figure 4.3. In this setting, we define spatial coordinates (x, y) with respect to a global Cartesian coordinate system. We assume that the road limits $\mathbb{B}(k) \subset \mathbb{R}^2$ in this coordinate system as well as the speed limit $V(k)$ are known to us. The road limits can typically be estimated while the speed limit can be obtained through either the map or perception information.

4.3.3 Observation Model

We limit the ego vehicle's visibility range to its field of view, denoted by $\mathbb{F}(k)$, as shown in Figure 4.3. With the set of vehicles in the environment represented by $\mathbb{V}(k) \subset \mathbb{Z}_{>0}$, the set of

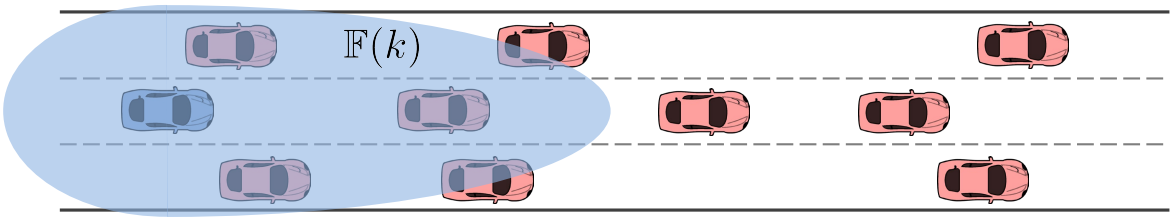

Figure 4.3: **Observation Model**. The observation space of the ego vehicle (in blue) is limited to its field of view, denoted by the set $\mathbb{F}(k)$ (highlighted in blue). Therefore, it is only able to observe the neighboring vehicles (in orange) that are present within this set.

vehicles visible to the ego vehicle is given by:

$$\mathbb{O}(k) = \left\{ i \in \mathbb{V}(k) \;\middle|\; p_i(k) \in \mathbb{F}(k) \right\} \tag{4.1}$$

where $p_i(k) = \begin{bmatrix} x_i(k) & y_i(k) \end{bmatrix}^{\mathrm{T}}$ denotes the position of vehicle i's center of mass in the global Cartesian coordinate frame, with subscript $_0$ reserved for the ego vehicle.

Remark 24. *Active perception through the receding horizon approach (Section 4.4.2) enables the ego vehicle to handle challenges pertaining to partial information, resulting from occlusion, sensory limitations, etc., as demonstrated in Section 2.3.1.3. Moreover, various streams of input data, e.g. V2X communication methods, can provide additional information to overcome such challenges, as shown in Section 2.4.*

4.3.4 Vehicle Model

We utilize the nonlinear kinematic bicycle model which provides a good balance between efficiency and accuracy [93]. Thus, the states, $X(k)$, and control inputs, $U(k)$, of the ego vehicle

at time instant k are defined as:

$$X(k) = \begin{bmatrix} x_0(k) & y_0(k) & \theta_0(k) & v_0(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{X}(k), \tag{4.2}$$

$$U(k) = \begin{bmatrix} a(k) & \delta(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{U}(k), \tag{4.3}$$

where $x_0(k)$, $y_0(k)$, $\theta_0(k)$, and $v_0(k)$ respectively denote the x-coordinate (m), y-coordinate (m), yaw angle with respect to the x-axis (rad), and speed (m/s), whereas $a(k)$ and $\delta(k)$ respectively denote acceleration (m/s^2), and steering angle (rad). The sets $\mathbb{X}(k) = \mathbb{B}(k) \times \mathbb{R}_{[0,2\pi)} \times \mathbb{R}_{[0,2V(k)]}$ and $\mathbb{U}(k) = \mathbb{R}^2_{[U_{\min}, U_{\max}]}$ respectively denote the feasible states and actuation limits. Then, the system dynamics read:

$$\begin{aligned} x_0(k+1) &= x_0(k) + T_s \cdot (v_0(k)\cos(\theta_0(k)), \\ y_0(k+1) &= y_0(k) + T_s \cdot (v_0(k)\sin(\theta_0(k)), \\ \theta_0(k+1) &= \theta_0(k) + T_s \cdot \left(\frac{v_0(k)}{L}\tan(\delta(k))\right), \\ v_0(k+1) &= v_0(k) + T_s \cdot a(k), \end{aligned} \tag{4.4}$$

where T_s corresponds to the sampling time. Further details regarding this model can be found in [93].

Remark 25. *The set of feasible speeds is expanded to twice the speed limit as it may be necessary to exceed the speed limit in order to avoid a crash and maintain safety.*

Remark 26. *In our dynamical model, we do not incorporate the tire model or friction forces [94]. Although these factors may play a significant role in extreme situations, we have to be mindful*

of the efficiency-accuracy tradeoff. Since our focus here is on the motion planning layer, rather than the control layer (refer to Figure 4.2), where challenges arise from the inclusion of collision avoidance constraints (Section 4.4.2) and a longer planning horizon, incorporating higher fidelity models comes at a cost of degraded computational efficiency.

4.4 Approach

This section describes the activation mechanism responsible for triggering the RCMS and the receding horizon optimization-based trajectory generation module, together with its various components, that outputs a reference trajectory to the low-level modules.

4.4.1 Activation Mechanism

An essential component of RCMS is its activation mechanism which decides when and how to activate it. This is because the objective of RCMS is fundamentally different from that of MP which operates under normal (low-risk) operating conditions. Unlike MP, RCMS places minimal emphasis on auxiliary metrics such as waypoint following, travel time minimization, passenger comfort maximization, etc., and focuses solely on the fundamental need to ensure the safety of the ego vehicle. Specifically, it considers only the local observations in the vicinity of the ego vehicle to decide on actions that will place the ego vehicle in a relatively safe state as soon as possible. Ideally, the goal of MP is to have a high level of safety and efficiency so that RCMS is never called. However, exogenous factors such as the unexpected behavior of on-road vehicles, sensor failures, weather conditions, etc. may potentially put the ego vehicle in situations where MP is

rendered infeasible, in which case, RCMS has to be called upon. Due to the fundamentally different objectives of RCMS and MP, the output of the two can at times be contradictory. Therefore, an unnecessary activation of RCMS can negatively impact passenger comfort, or in the worst case, even jeopardize the safety of the ego vehicle.

Being cognizant of the underlying differences in the objectives of the two modules and the potential undesirable consequences of improper switching, we propose a novel activation mechanism that facilitates a smooth transition between the two systems. In the existing literature, we notice the use of either instantaneous or predictive risk evaluation strategies to determine the necessity to activate the crash mitigation system. In terms of the instantaneous and predictive risk evaluation methods respectively, Gaussian overlap [95] and time-to-collision (TTC) [85,92,96,97] appear to be the preferred choices of the research community due to their simplicity and efficiency. However, these methods are usually deployed standalone in a 'bang-bang' fashion which runs the risk of constant switching between the two systems, adversely affecting passenger comfort as a result. Moreover, having them deployed independently also runs the risk of underestimating the risk in certain situations due to their underlying formulations. To elaborate, consider the following two scenarios. In the first case, the ego vehicle is traveling behind a human-driven vehicle (HDV) with negligible headway and 0 relative velocity, leading to ∞-TTC. If the leading vehicle decides to brake suddenly, it will lead to a crash so this is a high-risk situation that is not captured by the TTC metric but is captured by the ellipsoid overlap metric due to the close proximity of the two vehicles. On the other hand, consider the previous scenario but with considerable headway between the two vehicles. In this case, if the HDV decides to brake, the ellipsoid overlap will

not have a high enough value until the two vehicles get into close proximity of each other, but by that time, it may be too late to take any evasive actions, so TTC, with its predictive nature, has a better chance of anticipating the developing high-risk situation. To counter these drawbacks and ensure smooth operation, we combine instantaneous and predictive risk evaluation methodologies in a unified hysteresis-based activation mechanism.

As for instantaneous collision risk evaluation, motivated by [95], we model the risk associated with vehicle i as a bivariate Gaussian distribution $\tilde{p}_i(k) \sim \mathcal{N}(p_i(k), \Sigma_i(k))$. Here, the mean corresponds to the position of the vehicle while the variance matrix is determined by the length, width, and orientation of the vehicle as follows:

$$\Sigma_i(k) = R_{\theta_i(k)} \begin{bmatrix} \beta_l L_i & 0 \\ 0 & \beta_w W_i \end{bmatrix} R^{\mathrm{T}}_{\theta_i(k)}, \tag{4.5}$$

where L_i and W_i define the length and width of vehicle i with β_l and β_w corresponding to the respective scaling factors while $R_{\theta_i(k)}$ represents the 2D rotation matrix with the rotation angle $\theta_i(k)$.

We then evaluate the ego vehicle's overall collision risk $\kappa(k)$ as the maximum of its pairwise collision risk, $\kappa_i(k)$, with vehicle i where $\kappa_i(k)$ is given as the *product sum* of their distributions [95]. This is analogous to likelihood since when two distributions are completely overlapped, the likelihood is highest and when the distributions are not overlapped, the likelihood is very low. We

evaluate $\kappa(k)$ analytically as follows:

$$\kappa(k) = \max_{i \in \mathbb{O}(k)} \kappa_i(k), \tag{4.6}$$

$$\kappa_i(k) = \eta_i(k) \cdot e^{\frac{1}{2}\left(\omega_i^\top \Omega_i^{-1} \omega_i - p_0^\top \Sigma_0^{-1} p_0 - p_i^\top \Sigma_i^{-1} p_i\right)}, \tag{4.7}$$

where $\Omega_i = \left(\Sigma_0^{-1} + \Sigma_i^{-1}\right)$, $\omega_i = \Omega_i \left(\Sigma_0^{-1} p_0 + \Sigma_i^{-1} p_i\right)$, and $\eta_i(k)$ is a normalizing/scaling factor.

Remark 27. *For the sake of brevity, we do not show the dependence of variables in (4.7) on time instant k.*

Remark 28. *To keep the formulation generalized, we let $\eta_i(k)$ be vehicle i and time k dependent which gives us the flexibility to modify the risk based on a vehicle's class, e.g. truck, emergency, bicycle, etc., and its behavior over time. This gives us the ability to incorporate the vehicular behavioral risk, that was expounded upon in Section 3.3.3.3.*

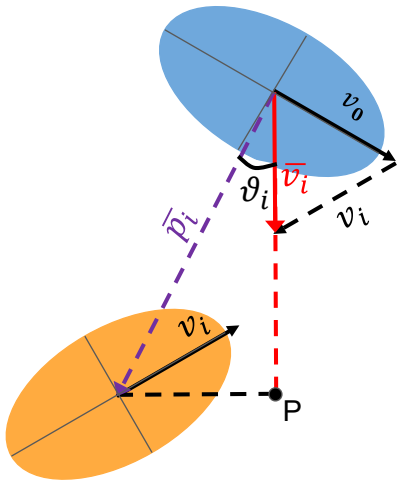

Figure 4.4: **Time-to-Closest-Encounter (TTCE) Calculation**. With the ego vehicle shown in blue and the vehicle i shown in orange, the time to collision is calculated based on the relative velocity ($\bar{v}_i$) as well as the relative displacement ($\bar{p}_i$) between the two vehicles.

Regarding the predictive collision risk metric, TTC is typically evaluated based only on longitudinal displacement between vehicles [85]. However, for the crash mitigation system in a highway setting, lateral motion plays as much of a role, if not more, as longitudinal motion. Due to the direction of motion in the longitudinal direction, the motion planner already tends to maintain a desired safety margin in that direction but the lane width typically restricts the safety margin in the lateral direction. For instance, it is easier to anticipate and react to the behavior of a vehicle traveling further ahead, even if it is acting erratically, but it's harder to react to a vehicle traveling in the adjacent lane that suddenly decides to swerve in the direction of the ego vehicle. Therefore, we formulate Time-to-Closest-Encounter (TTCE), τ_i, as our predictive risk evaluation metric based on the relative velocity between the ego vehicle and vehicle i as follows:

$$\tau(k) = \min_{i \in \mathbb{O}(k)} \tau_i(k) \tag{4.8}$$

$$\tau_i(k) = \frac{\|\bar{p}_i(k)\| \cos(\vartheta_i(k))}{\|\bar{v}_i(k)\|} = \frac{\bar{p}_i(k) \cdot \bar{v}_i(k)}{\|\bar{v}_i(k)\|^2} \tag{4.9}$$

where $\bar{p}_i(k) = p_i(k) - p_0(k)$ and $\bar{v}_i(k)$ correspond respectively to the relative displacement and velocity between the ego vehicle and vehicle i, as shown in Figure 4.4, while the second equality in (4.9) follows from the fact that:

$$\bar{p}_i(k) \cdot \bar{v}_i(k) = \|\bar{p}_i(k)\| \|\bar{v}_i(k)\| \cos(\vartheta_i(k)).$$

We combine the instantaneous (4.6) and predictive collision risk evaluation metrics with a hysteresis band, depicted in Figure 4.5, to come up with an overall activation mechanism, outlined

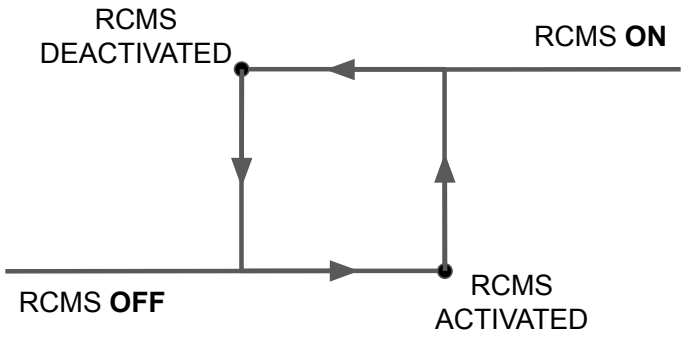

Figure 4.5: **Hysteresis Band for RCMS Activation**. Different activation and deactivation thresholds enable a much smoother transition between MP and RCMS, as compared to a 'bang-bang' approach.

in Algorithm 1. With the hysteresis band, higher activation thresholds (κ_a and τ_a) prevent unnecessary activation of the crash mitigation system while lower deactivation thresholds (κ_d and τ_d) allow for the ego vehicle to get to a much safer state before handing the control back over to the motion planner. Moreover, the two conditions in line 6 of Algorithm 1 ensure that the predictive collision risk is evaluated only when necessary. Specifically, $\bar{v}_i(k) \cdot \bar{p}_i(k) < 0$ ensures that the ego vehicle and vehicle i are moving towards each other while $\|\bar{p}_i(k)\| \sin(\vartheta_i(k)) < L_0 + L_i + \epsilon$ ensures that at the point of closest encounter ('P' in Figure 4.4), the vehicles are in close proximity to each other where ϵ governs how close of a proximity to consider. In the 2D Cartesian coordinate system, we can evaluate $\|\bar{p}_i(k)\| \sin(\vartheta_i(k))$ using the following relation:

$$\bar{p}_i(k) \times \bar{v}_i(k) = \|\bar{p}_i(k)\| \|\bar{v}_i(k)\| \sin(\vartheta_i(k))$$

$$\|\bar{p}_i(k)\| \sin(\vartheta_i(k)) = \frac{\bar{p}_i(k) \times \bar{v}_i(k)}{\|\bar{v}_i(k)\|}.$$

Algorithm 1 RCMS Activation Mechanism

$\kappa(k) \leftarrow 0$
$\tau(k) \leftarrow \infty$
while system running **do**
  for $i \in \mathbb{O}(k)$ **do**
    $\kappa_i(k) \leftarrow (4.7)$
    if $\bar{v}_i(k) \cdot \bar{p}_i(k) < 0$ **and** $\|\bar{p}_i(k)\| \sin(\vartheta_i(k)) < L_0 + L_i + \epsilon$ **then**
      $\tau_i(k) \leftarrow (4.8)$
    else
      $\tau_i(k) \leftarrow \infty$
    end if
  end for
  $\kappa(k) \leftarrow (4.6)$
  $\tau(k) \leftarrow (4.8)$
  if $\kappa(k) > \kappa_a$ **or** $\tau(k) < \tau_a$ **then**
    Activate RCMS
  else if $\kappa(k) < \kappa_d$ **and** $\tau(k) > \tau_d$ **then**
    Deactivate RCMS
  end if
end while

4.4.2 Trajectory Generation

Once the crash mitigation system has been activated, its job, just like that of the motion planner, is to provide a reference trajectory comprised of future waypoints (position as well as speed profile), for the low-level modules (see Figure 4.2) to track and follow. For trajectory generation, we utilize a receding horizon optimization [13] based formalism that allows us to minimize the situational risk profile while accounting for vehicular dynamics, actuator limits, and road boundaries, ensuring the feasibility of output trajectory.

Situational Risk Model

To quantify the spatially distributed situational risk perceived by the ego vehicle at time instant k, we formulate the instantaneous risk function $\rho(k)$ as an aggregate of individual agent-specific risk functions $\rho_i(k)$, for each $i \in \mathbb{O}(k)$, as well as road boundary risk $\rho_r(k)$. We choose parametric continuous smooth functions to model $\rho(k)$. In particular, taking inspiration from a skewed Gaussian distribution [98], we formulate a skewed hyperbolic quadratic function to represent $\rho_i(k)$, and we model $\rho_r(k)$ with a univariate Gaussian function. Before expanding on the functional form for $\rho_i(k)$, we first shed some light on the density function $\phi_s(p_s)$ of a skewed Gaussian distribution, outlined below:

$$\phi_s(p_s) = 2\phi(p_s; \mu_s, \Sigma_s)\Phi(q_s^{\mathrm{T}} p_s), \tag{4.10}$$

where $p_s \in \mathbb{R}^2$ and $q_s \in \mathbb{R}^2$ correspond respectively to position and direction-oriented skew parameter in the Cartesian coordinate frame while ϕ and Φ represent the density and distribution functions of a bivariate Gaussian distribution parametrized by mean μ_s and covariance Σ_s.

To obtain a situational risk profile with its spatial density similar to that of a Gaussian mixture while ensuring that each agent has a barrier around that prevents the ego vehicle from 'going through' it, we opt to use a reciprocal quadratic function ψ_i instead of ϕ. Moreover, to orient and scale the symmetric risk distribution in the direction of an agent's motion, we use a simpler Sigmoid function σ_i instead of Φ which requires the evaluation of the error function. Then, having

$\bar{p}(k) = \{\bar{p}_i(k) \mid i \in \mathbb{O}(k)\}$, $\rho(k)$ is given by:

$$\rho(k; \bar{p}(k)) = \sum_{i \in \mathbb{O}(k)} \rho_i(k; \bar{p}_i(k)) + \rho_r(k) \tag{4.11}$$

$$\rho_i(k; \bar{p}_i(k)) = \psi_i(k; \bar{p}_i(k)) \sigma_i(k; \bar{p}_i(k)) \tag{4.12}$$

$$\psi_i(k; \bar{p}_i(k)) = \frac{\tilde{\eta}_i(k)}{\alpha_g + \bar{p}_i^{\mathrm{T}}(k) \widetilde{\Sigma}_i^{-1}(k) \bar{p}_i(k)} \tag{4.13}$$

$$\sigma_i(k; \bar{p}_i(k)) = \frac{1}{1 + \exp(-\alpha_s \bar{p}_i^{\mathrm{T}}(k) v_i(k))} \tag{4.14}$$

$$\rho_r(k) = \gamma_r \exp\left(-\alpha_r \bar{p}_r^{\mathrm{T}}(k) \Gamma(k) \bar{p}_r(k)\right) \tag{4.15}$$

where α_g controls the relative gradient of the agent distribution; α_s controls skewness of the agent distribution; $\tilde{\eta}_i(k)$ serves a similar normalization purpose to $\eta_i(k)$ in (4.7); $\widetilde{\Sigma}_i(k)$ is defined analogously to $\Sigma_i(k)$ in (4.5) but the distinction is made since the scaling and rotation parameters need not be the same; γ_r controls the scaling for road distribution; α_r controls the gradient for road distribution; $\bar{p}_r(k) = p_r(k) - p_0(k)$, with $p_r(k) \in \mathbb{B}(k)$, denotes the relative coordinates of road boundary; and $\Gamma(k)$, with the help of Frenet coordinates system, ensures that the road risk is effective in only the lateral direction.

Prediction Model

Given the extensive literature on motion prediction methods for autonomous vehicles [16], there exists a range of approaches that can be integrated into the RCMS framework. However, the lack of data for high-risk near-miss or collision scenarios raises doubts about the applicability of many state-of-the-art methods. To avoid making assumptions about the nature of neighboring

agents as it may lead to degraded overall behavior, we opt to use a constant acceleration prediction model for RCMS. Making assumptions about the behavior of agents, such as attributing aggressive (adversarial) or defensive (cooperative) intent to them, increases the risk of misjudging the true nature of the agent's behavior in such stressful scenarios. This can lead to undesirable outcomes, such as focusing on minimizing the severity of a crash but ultimately causing the crash when it was actually avoidable or opting for a collision avoidance strategy but eventually increasing crash severity when it was indeed unavoidable. In practice, most high-risk situations result from human drivers' negligence [99, 100] which justifies the use of our simple behavior-agnostic prediction model formalized below:

$$x_i^k(j+1) = x_i^k(j) + T_s v_i^k(j)\cos(\theta_i(k)) \tag{4.16}$$

$$y_i^k(j+1) = y_i^k(j) + T_s v_i^k(j)\sin(\theta_i(k)) \tag{4.17}$$

$$v_i^k(j+1) = v_i^k(j) + T_s a_i(k) \tag{4.18}$$

where $\theta_i(k)$ and $a_i(k)$ are vehicle i's estimated heading and acceleration values at time instant k while $x_i^k(j)$, $y_i^k(j)$ and $v_i^k(j)$ are vehicle i's predicted x-coordinate, y-coordinate and speed values at a future time step j w.r.t. time instant k, with $x_i^k(0) = x_i(k)$, $y_i^k(0) = y_i(k)$ and $v_i^k(0) = v_i(k)$. Then, the relative predicted position is denoted by $\bar{p}_i^k(j) = \begin{bmatrix} x_i^k(j) & y_i^k(j) \end{bmatrix}^{\mathrm{T}} - p_0(k)$ and $\bar{p}^k(j) = \{\bar{p}_i^k(j) \mid i \in \mathbb{O}(k)\}$.

Remark 29. *The modular nature of RCMS allows for the incorporation of modern prediction algorithms in case better models become available for the task at hand.*

Objective Function

The objective function is formulated to minimize the accumulative predictive situational risk, defined in (4.11), over the planning horizon H while placing relatively low emphasis on control actions regulation as the priority is to ensure safety.

$$J(k) = \sum_{j=1}^{H} \rho^k(j; \bar{p}^k(j)) + {U^k}^{\mathrm{T}}(j) R(k) U^k(j), \tag{4.19}$$

where $R(k)$, chosen such that $\max {U^k}^{\mathrm{T}}(j) R(k) U^k(j) << \max \rho^k(j)$, places a time-varying penalty on control actions which in turn ensures passenger comfort.

Remark 30. *$R(k)$ is allowed to be time-varying so that its value is set inversely proportional to the sum of normalized instantaneous and predictive collision avoidance risks i.e.*

$$\frac{1}{R(k)} \propto \frac{2\kappa}{\kappa_a + \kappa_d} + \frac{\tau_a + \tau_d}{2\tau},$$

which ensures that the emphasis on control action minimization (or passenger comfort) is further decreased with an increased risk of collision with other agents.

Complete Optimization Problem

The receding horizon optimization problem is posed as a nonlinear program with its formulation provided below:

$$\min_{\mathcal{X}^k,\, \mathcal{U}^k} \; J(k) \tag{4.20}$$

subject to:

$$X^k(0) = X(k) \tag{4.21}$$

$$X^k(j+1) = f(X^k(j), U^k(j)) \quad \forall j \in \mathbb{Z}_{[0,H-1]}, \tag{4.22}$$

$$X^k(j) \in \mathbb{X}(k) \quad \forall j \in \mathbb{Z}_{[0,H-1]}, \tag{4.23}$$

$$U^k(j) \in \mathbb{U}(k) \quad \forall j \in \mathbb{Z}_{[0,H-1]}, \tag{4.24}$$

where $\mathcal{X}^k = \{X^k(j) \mid j \in \mathbb{Z}_{[0,H]}\}$ and $\mathcal{U}^k = \{U^k(j) \mid j \in \mathbb{Z}_{[0,H-1]}\}$, respectively representing the future states and controls, denote the optimization variables.

Remark 31. *A crucial requirement for timely collision avoidance and crash severity mitigation is a high algorithmic computational efficiency which is achieved by the smoothness and continuity of the objective function [101], as verified in Section 4.5.2.*

Remark 32. *With minimal improvement in the computational efficiency of the non-linear program by linearizing the dynamics in our experimentation, we opt to use the non-linear dynamical model as its higher fidelity ensures the feasibility of outputs even in high-risk situations.*

4.5 Evaluation

This section details the experimental setup and demonstrates the performance of RCMS in our test case scenarios in Python-based as well as CARLA simulations.

4.5.1 Experimental Setup

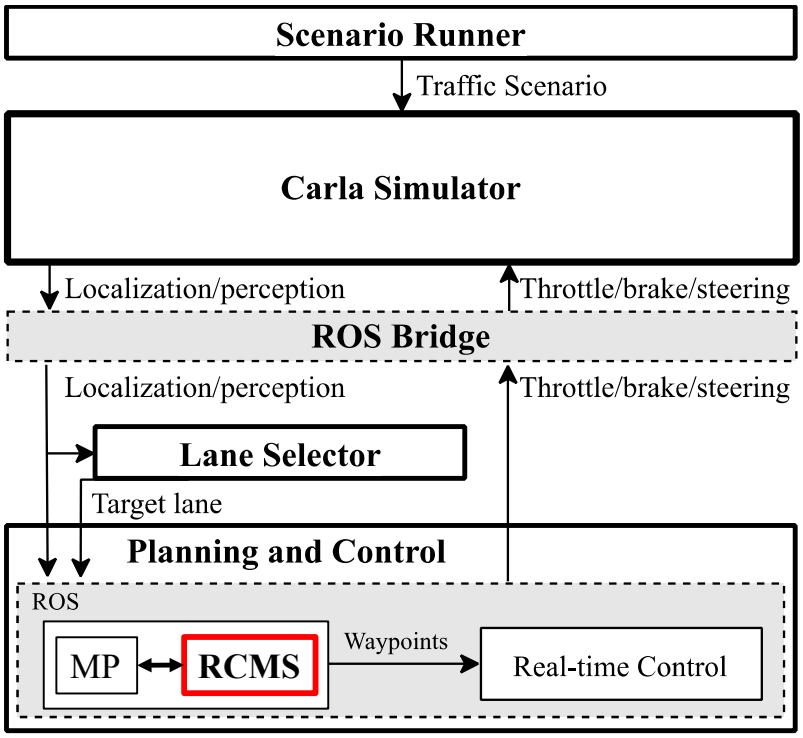

Figure 4.6: **CARLA Simulation Setup**. Scenario Runner configures the scenario for the CARLA Simulator, which communicates with the Lane Selector as well as the Planning and Control ROS nodes via the ROS bridge node. The Planning and Control node incorporates both the MP and the RCMS modules, with the activation mechanism determining which of the two is operational.

The experimentation is conducted on a computer equipped with an AMD Ryzen 7 5800h × 16 processor and NVIDIA GeForce RTX 3080 graphics card, running Ubuntu 20.04 LTS. To solve the non-linear program for RCMS, we employ the interior point optimizer *IPOPT* [102]

using *MA27* linear solver, from the HSL library [103], within a CasADi [104] environment. The optimization process requires $0.06s$ on average and $0.08s$ in the worst case, with $H = 30$ and $T_s = 0.1$, showcasing the potential for real-time applicability.

For the CARLA-based simulation in Section 4.5.2.2, the experimental setup is depicted in Figure 4.6. It is composed of the CARLA Simulator and Scenario Runner (Versions 0.9.11) [77], Lane Selector module [57], RCMS module (Section 4.4), and MP and Controller modules [70].

4.5.2 Results

We first demonstrate our approach on scenarios developed in a Python-based simulation environment in Section 4.5.2.1 before moving on to the demonstration in a CARLA simulation environment in Section 4.5.2.2.

4.5.2.1 Python Simulations

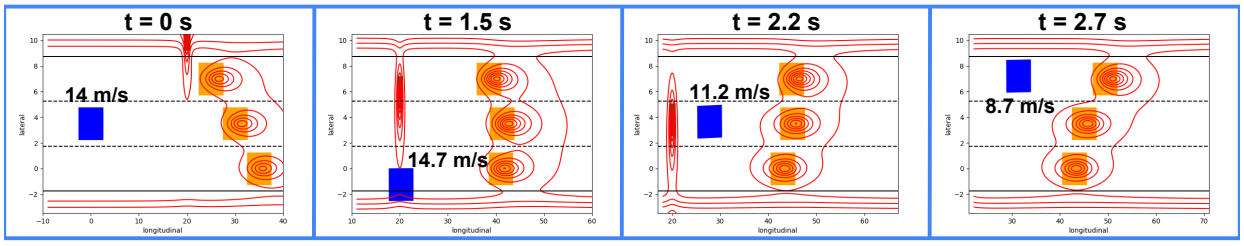

Figure 4.7: **Scenario I.** With the on-road agents depicted in orange, and the situational risk contour plot depicted in red, the ego vehicle (in blue) notices a moving object (animal, pedestrian, etc.) approaching laterally from the end of the road, leading to a high collision risk, which activates the RCMS at $t = 0s$. The ego vehicle then swerves right to avoid a collision with the object when suddenly two of the vehicles traveling ahead in the right and center lanes successively stop abruptly in the middle of the highway at $t = 1.5s$, maintaining the high collision risk. Then, the ego vehicle swerves smoothly to the left-most lane to place the ego vehicle in a relatively safe state before handing the control over to the MP at $t = 2.7s$.

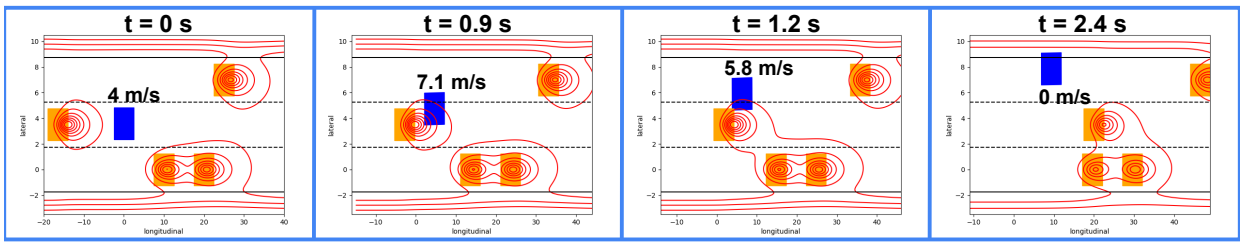

Figure 4.8: **Scenario II.** With the on-road agents depicted in orange, and the situational risk contour plot depicted in red, the ego vehicle (in blue) traveling slowly in the center lane notices a fast-moving vehicle approaching rapidly from behind, leading to a high collision risk, which activates the RCMS at $t = 0s$. With two other vehicles traveling close up ahead towards its right, the ego vehicle accelerates and swerves left simultaneously to minimize the severity of the collision, leading to a near-miss situation at $t = 1.2s$. Due to the drastic maneuver, the ego vehicle has to make use of the shoulder to stabilize before handing the control back to the MP at $t = 2.4s$.

To evaluate the performance of RCMS, we consider two scenarios: (i) Scenario I: Object approaching laterally (Figure 4.7); (ii) Scenario II: Fast-approaching rear-end vehicle (Figure 4.8). These scenarios are designed to evaluate different properties of RCMS. Scenario I tests whether the proposed framework has the ability to perform multiple evasive actions successively while Scenario II evaluates the ability of RCMS to operate at the actuation limits to minimize the severity of a potential collision. The corresponding speed, acceleration, and steering profiles are illustrated in Figure 4.9.

In Scenario I, with the object approaching laterally from the end of the road, the ego vehicle first assesses the situation to see if it can brake and swerve left to go behind the object. However, with the imposed actuation limits making that impossible, it then decides to accelerate and swerve right, while accessing the shoulder, to escape the object from its front at $t = 1.5s$. Upon noticing the successively stopping vehicles thereafter, it decides to brake and swerve left to avoid those vehicles before ending up in a safe zone at $t = 2.7s$ and handing the control back to MP. In Scenario II, the ego has to first accelerate while swerving left to avoid the fast-approaching rear-

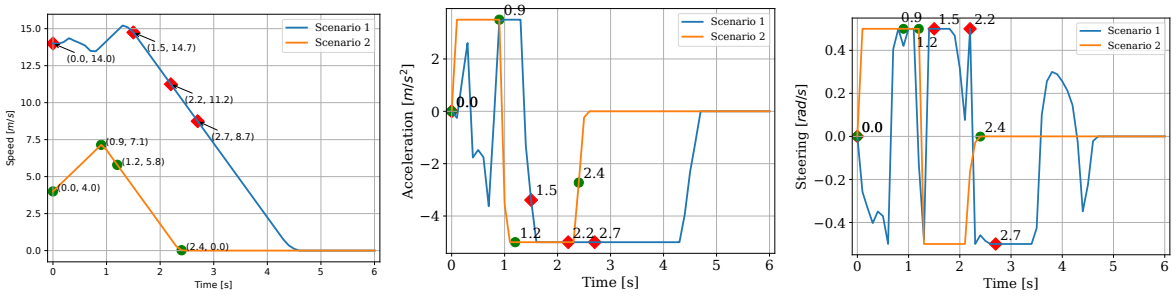

Figure 4.9: **Control Plots for Scenarios 1 and 2.** The actuation limits are given by $U_{\text{min}} = \begin{bmatrix} -5 & -0.5 \end{bmatrix}^{\text{T}}$ and $U_{\text{max}} = \begin{bmatrix} 3.5 & 0.5 \end{bmatrix}^{\text{T}}$, with negative steering (δ) values representing steering to the right. In scenario 1, the ego vehicle initially switches between acceleration and deceleration while assessing the situation before deciding to slam on the brakes and steer aggressively to avoid the laterally moving object as well as the stopping vehicles. In scenario 2, the ego vehicle operates at the steering and acceleration limits to barely escape the rear-end speeding vehicle. The numbers indicated on the plots highlight the values at different time instances recorded in Figure 4.7 and Figure 4.8.

end vehicle. Once it barely escapes at around $t = 1s$, it slams on the brake while steering back right to avoid ramming into the shoulder, before reaching a safe state at $t = 2.4s$ and handing the control back to MP.

4.5.2.2 CARLA Simulation

To effectively evaluate the utility of RCMS in a realistic setting, we implement the motivational scenario outlined in Section 4.1 in a CARLA simulation environment. The scenario, demonstrated in Figure 4.10, consists of a four-lane highway segment, with lanes labeled as $0 - 3$ from left to right, relative to the direction of traffic. The ego vehicle, as depicted in Frame 1, is initialized to travel on lane 1 with no leading vehicle, a tailgating vehicle, and two relatively slow vehicles traveling ahead on adjacent lanes 0 and 2. As the ego vehicle is approaching to overtake the vehicles in adjacent lanes, the vehicle in lane 0 suddenly swerves in front of the ego vehicle,

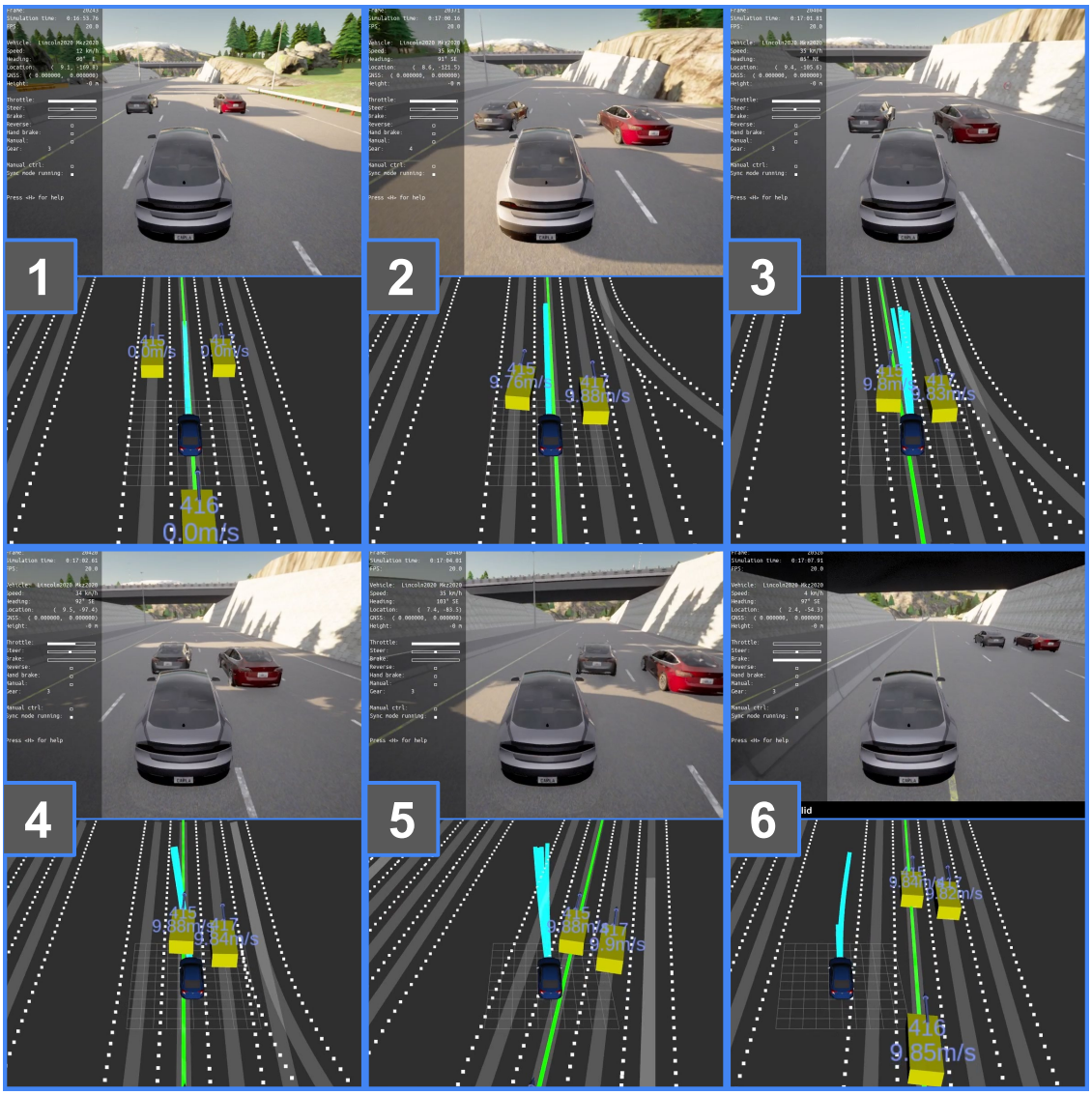

Figure 4.10: **CARLA Simulation Scenario**. On a stretch of a multi-lane highway, lanes 0 (left), 1 (center), 2 (right), and 3 (right-most) are all available for traveling while shoulder access is also available for emergency situations. The ego vehicle's motion over the course of the simulation is depicted through the numbered frames. The turquoise-colored lines represent the ego vehicle's planned trajectory.

as shown in Frame 2, rendering the MP infeasible. In the absence of the RCMS module, the two backup strategies for MP, i.e. emergency braking and continuing on the current trajectory, both lead to a collision. In the case of emergency braking, there is a collision with the trailing vehi-

cle while in the case of continuing on the current trajectory, there is a collision with the vehicle swerving in from lane 0.

With the integration of RCMS, the ego vehicle carefully navigates around the surrounding vehicles, considering space availability and actuation limits, to safely avoid a collision, as depicted through Frames $3-6$. First, the ego vehicle cautiously maneuvers to the right, to see if it can exploit the available space to pass between the vehicles, as observed in Frame 3. However, with the actuation limits preventing that, it decides to slow down slightly to accommodate the swerving vehicle, as seen in Frame 4. Wary of the negligible distance to the now leading vehicle, RCMS carefully steers the ego vehicle towards the left, as seen in Frame 5, to place it in a safe state while accessing the shoulder, as seen in Frame 6, before handing the control back to MP.

4.6 Conclusion

In conclusion, we have developed a novel risk-aware crash mitigation system that comprises an activation mechanism and a modular trajectory generation component to perform evasive maneuvers in high-risk collision-prone situations. The activation mechanism effectively combines instantaneous and predictive collision risk evaluation within a hysteresis band to facilitate a smooth transition between RCMS and MP, with their distinct objectives, to maintain passenger comfort. Meanwhile, the trajectory generation component minimizes situational risk through a smooth function while considering actuation, dynamical, and road limits. We have validated the real-time applicability and performance of our approach by conducting simulations of two high-risk scenarios which assessed the ability to perform successive evasive maneuvers at the vehicle's actuation

limits.

Our future work entails the implementation of RCMS on a physical small-scale setup to evaluate its robustness to various real-world uncertainties before deploying it on actual vehicles.

4.7 Appendix

This section provides an in-depth discussion of the existing collision avoidance strategies that were considered over the course of this work. By doing so, we aim to reemphasize the relevance and importance of our proposed methodology.

4.7.1 Collision Avoidance

An important aspect of any motion planning algorithm, or its crash mitigation counterpart, is the collision avoidance strategy employed in the algorithm. Since collision avoidance by its nature is a non-convex problem, this aspect of the planning task proves to be a bottleneck in terms of computational complexity. The sampling-based approaches [47, 55, 89], which do have low computational complexity, try to evaluate a cost function on a predefined set of trajectories and select the trajectory with minimum overall cost. However, due to the evaluation of a limited number of trajectories, these methods are unable to find the optimal solution over the entire state and action space. Such a rigorous search may be necessary, especially in crash mitigation scenarios, to find a trajectory that potentially avoids a collision over the one that does not. Therefore, the improved complexity of the sampling-based methods comes at a cost of degraded performance. The performance can be improved by increasing the size of the sample space but that again comes at a

cost of increased complexity in accordance with the efficiency-accuracy tradeoff. Moreover, the complete state and action space can only be exploited in the limiting case of the number of samples approaching infinity so these methods will always have a hard time matching the performance of their continuous space optimization-based counterparts.

In terms of the continuous space optimization-based collision avoidance strategies, several prevalent techniques in the literature [105–107] define safety requirements as hard constraints within the optimization problem. These methods offer the advantage of "guaranteed collision avoidance" in normal low-risk operating conditions when the real-world situation aligns well with the predictions of the model, as discussed in Section 2.3.2.3. However, in high-risk collision-prone scenarios, these methods encounter feasibility issues unless accompanied by slack variables (soft constraints) [107, 108] or when opting for a chance-constrained formulation [109, 110]. Moreover, a fundamental problem observed with such methods is their high computational complexity arising from the non-convex constraints. Therefore, for our crash mitigation system, we adopt a risk-minimization-based approach that provides satisfactory collision avoidance performance with significantly improved computational complexity while losing out on safety guarantees, which are inherently unavailable for scenarios where a collision is potentially unavoidable.

When it comes to risk evaluation-based strategies [90, 96, 111] or potential-function approaches [26, 112–114], it is crucial to strike a balance between optimality and runtime complexity. If the optimal solution is obtained within the allocated time, these methods can successfully generate a path to avoid obstacles. However, given the nonlinearity of the objective function, it is highly probable that the optimal solution cannot always be obtained within the designated timeframe. In

such cases, two possible approaches can be adopted to ensure real-time operation: (i) Use the solution from the previous time step, or (ii) Use the best feasible solution found within the allocated time. In the case of the first approach, if obtaining the optimal solution with the global minima for the formulated optimization problem proves to be challenging, the method will continuously rely on past solutions based on outdated information. This reliance on outdated information undermines the advantages of the receding horizon framework. As a result, after a few time steps where the optimal solution is not obtained within the allocated time, the planner will base its decisions on predictions from a long time ago, which may differ significantly from the current situation. This situation poses a risk to the safety of all agents involved. In the worst-case scenario, the planner may exhaust all past solution steps to exploit if no optimal solution is obtained within the time steps corresponding to the control horizon, leading to the same infeasibility problem again.

The second approach relies on using the best available solution obtained within the given time limit at the current instant. With the typical gradient-based optimization methods [101], the optimizer starts off with an initial guess and recursively improves upon it through gradient-based updates while maintaining the feasibility of the solution until an optimal solution is found or the time limit is reached. Thus, the preference is ensuring feasibility over achieving optimality. When applying this approach to a risk or potential minimization-based strategy, special consideration needs to be placed on the proper design of objective function. Typically, many existing approaches [90, 114–116] assign a finite constant risk to the obstacles observed in the environment. In dense traffic scenarios, such a risk evaluation function tends to yield feasible solutions that intersect with obstacles, thereby increasing the potential severity of collisions. To mitigate this issue, we propose

our own risk evaluation function, detailed in Section 4.4.2, that assigns a significantly high risk to the obstacles, especially at their centers. Consequently, the optimizer is compelled to quickly find solutions that do not intersect with obstacles head-on, effectively minimizing the potential severity of collisions if they were to occur.

For the development of RCMS, we thoroughly explored and evaluated various collision avoidance strategies prior to designing our situational risk minimization framework within a receding horizon optimization-based approach. The smooth functional risk profile provides continuous and reliable situational risk assessment which when optimized over generates paths that avoid collisions and minimize their severity, if they are inevitable, by minimizing the collision area overlap. Even with the imposition of an optimizer time-limit to facilitate real-time applicability, the situational risk profile encourages the optimizer to quickly move away from solutions that 'directly go through' obstacles, ensuring the validity of the sub-optimal solutions. For the sake of completeness, we elaborate upon the various considered methodologies, which help further highlight the contributions of our research and provide the rationale behind our design choices.

4.7.1.1 Farkas' Lemma

In the Cartesian coordinate system, any polygon can be represented by the set $\mathbb{P} = \{z \mid Az \leq b\}$ where $z = \begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$. Therefore, if each vehicle $i \in \mathbb{O}(k)$ is modeled as a rectangle, its spatial representation is given by $\mathbb{P}_i(k) = \{z \mid A_i(k)z \leq b_i(k)\}$ where $A_i(k)$ contains its orientation information while $b_i(k)$ contains its positional and dimensional (length and width) information. More details regarding the definitions of $A_i(k)$ and $b_i(k)$ can be found in [105]

and [69]. Now, in order to enforce the collision avoidance condition at a given time instant, which states that the two polygons *should not* intersect, we pose the following condition:

$$\mathbb{P}_0(k) \cap \mathbb{P}_i(k) = \{z \mid \bar{A}_i(k)z = \begin{bmatrix} A_0^{\mathrm{T}}(k) & A_i^{\mathrm{T}}(k) \end{bmatrix}^{\mathrm{T}} z \leq \begin{bmatrix} b_0^{\mathrm{T}}(k) & b_i^{\mathrm{T}}(k) \end{bmatrix}^{\mathrm{T}} = \bar{b}_i(k)\} = \emptyset \quad (4.25)$$

where, as per usual, subscript $_0$ is reserved for the ego vehicle.

This condition, in its current form, is not applicable to a standard optimization problem since it requires the *non-existence* of any z that satisfies $\bar{A}_i(k)z \leq \bar{b}_i(k)$. Therefore, to transform this condition into a set of inequalities that can be posed as constraints to an optimization problem, we benefit from the following lemma [105]:

Lemma 2 (Farkas' Lemma)**.** *Either the system $Rp \leq q$, with $p \in \mathbb{R}^n$, has a solution or the system $R^{\mathrm{T}}\alpha = 0$ has a solution with $q^{\mathrm{T}}\alpha < 0$ and $\alpha \geq 0$.*

Lemma 2, when applied to our problem, states that *either* there exists some $z \in \mathbb{R}^2$ that solves the primal problem given by $\bar{A}_i(k)z \leq \bar{b}_i(k)$ (hence, a collision) *or* there exists some $\alpha_i(k) \geq 0$ with $\bar{b}_i^{\mathrm{T}}(k)\alpha_i(k) < 0$ that solves the dual problem given by $\bar{A}_i^{\mathrm{T}}(k)\alpha_i(k) = 0$. Therefore, to enforce the collision avoidance condition, it is sufficient to ensure the feasibility of the dual problem, which in turn ensures the infeasibility of the primal problem.

In order to formulate the collision avoidance constraints, we introduce auxiliary α_i variables

for each $i \in \mathbb{O}(k)$. Then, the constraints read:

$$\alpha_i(k) \geq 0 \tag{4.26}$$

$$\bar{A}_i^{\mathrm{T}}(k)\alpha_i(k) = \begin{bmatrix} A_0^{\mathrm{T}}(k) & A_i^{\mathrm{T}}(k) \end{bmatrix} \alpha_i(k) = 0 \tag{4.27}$$

$$\bar{b}_i^{\mathrm{T}}(k)\alpha_i(k) = \begin{bmatrix} b_0^{\mathrm{T}}(k) & b_i^{\mathrm{T}}(k) \end{bmatrix} \alpha_i(k) \leq -\epsilon \tag{4.28}$$

The constraints above are only for the current time instant k so in order to extend them to the receding horizon optimization framework, we have to introduce variables for every future time step $j \in \mathbb{Z}_{[0,H]}$ i.e. replace $\alpha_i(k)$, $\bar{A}_i(k)$ and $\bar{b}_i(k)$ with $\alpha_i^k(j)$, $\bar{A}_i^k(j)$ and $\bar{b}_i^k(j)$ respectively in (4.26), (4.27), and (4.28). At first glance, these constraints may appear linear but upon closer look, it becomes evident that they are indeed non-linear and non-convex. Recall from Section 4.4.2 that the optimization variables for the receding horizon optimization problem comprise the future states ($\mathcal{X}^k$) and control variables ($\mathcal{U}^k$) for the ego vehicle. Then, we have the product of the trigonometric functions of $\theta_0^k(j)$ - from $A_0^k(j)$ - with $\alpha_i^k(j)$ in (4.27) and the product of positional variables ($x_0^k(j)$ and $y_0^k(j)$) - from $b_0^k(j)$ - with $\alpha_i^k(j)$ in (4.28) which leads to non-linearity and non-convexity. This, in turn, leads to a higher computational complexity for the optimization problem.

Moreover, with the introduction of $\alpha_i^k(j) \in \mathbb{R}^{8\times H\times N_v(k)}$, $\bar{A}_i^k(j) \in \mathbb{R}^{8\times 2\times H\times N_v(k)}$ and $\bar{b}_i^k(j) \in \mathbb{R}^{8\times H\times N_v(k)}$, where 8 comes from the combined sides of the two rectangles, 2 comes the spatial dimension of the Cartesian coordinate system and $N_v(k) = |\mathbb{O}(k)|$ represents the number of vehicles in the observation set of the ego vehicle, we significantly increase the scale of the optimization problem. As a consequence, the runtime complexity increases further, resulting in a loss of real-

time applicability of the approach.

Remark 33. *Lemma 2 requires that the inner product of q and α ($\bar{b}_i^k(j)$ and $\alpha_i^k(j)$ in our case) be strictly negative i.e $q^{\mathrm{T}}\alpha < 0$. However, the optimization problem formulation in the standard form [101] restricts the constraints to non-strict inequalities ($\leq$, $\geq$) because in a continuous space setting, a strict inequality defines an open set where a point arbitrarily close to the boundary can be chosen and no optimal solution may be found, as a result. Moreover, under practical considerations, i.e. considering floating point errors, there essentially is no difference between the ($<$) and ($\leq$) conditions. Therefore, to make the condition from Lemma 2 conform to the formulation of a standard optimization framework, we require that a tolerance factor of $\epsilon > 0$ be introduced s.t. the condition transforms to the form $q^{\mathrm{T}}\alpha \leq -\epsilon$. The conditions for an appropriate choice for ϵ still need to be investigated for our application. If such a modification is not made, then the trivial solution of $\alpha = 0$ satisfies the two conditions from the lemma, regardless of the values of elements in matrix A and vector b, rendering our collision avoidance condition invalid.*

4.7.1.2 Disjunctive Constraints

As elaborated in Section 4.7.1.1, with the vehicles represented as rectangles in the Cartesian coordinate system, the collision avoidance condition is given by (4.25). Considering the rectangular vehicular geometries, we can pose a set of disjunctive constraints to model the collision

avoidance condition for the receding horizon optimization problem as follows [107]:

$$\begin{aligned}\{x_0^j(k) - x_i^j(k) \geq L_b^k(j)\} \vee \{x_i^j(k) - x_0^j(k) \geq L_f^k(j)\} \vee \\ \{y_0^j(k) - y_i^j(k) \geq L_r^k(j)\} \vee \{y_i^j(k) - y_0^j(k) \geq L_l^k(j)\}\end{aligned} \tag{4.29}$$

where $L_b^k(j)$, $L_f^k(j)$, $L_r^k(j)$, and $L_l^k(j)$ represent the vehicle-dependent backward, forward, right, and left safety margins respectively at the future time step $j \in \mathbb{Z}_{[0,H]}$. For the sake of accuracy, these distances are allowed to be time step j dependent in order to incorporate the predicted orientation (yaw) of vehicle $i \in \mathbb{O}(k)$ into the safety margins. Essentially, the condition in (4.29) states that the ego vehicle should be present either behind vehicle i, or ahead of it, or towards its right, or towards its left, while maintaining appropriate safety margins.

In order to convert the disjunctive constraints into a set of conjunctive constraints, as appropriate for a standard optimization problem, we make use of the big-M method, coupled with the introduction of auxiliary binary variables, as follows:

$$x_0^j(k) - x_i^j(k) \geq L_b^k(j) - Mb_{i,1}^k(j) \tag{4.30}$$

$$x_i^j(k) - x_0^j(k) \geq L_f^k(j) - Mb_{i,2}^k(j) \tag{4.31}$$

$$y_0^j(k) - y_i^j(k) \geq L_r^k(j) - Mb_{i,3}^k(j) \tag{4.32}$$

$$y_i^j(k) - y_0^j(k) \geq L_l^k(j) - Mb_{i,4}^k(j) \tag{4.33}$$

$$\sum_{n=1}^{4} b_{i,n}^k(j) \leq 3 \tag{4.34}$$

where $M \gg 0$ (big-M) and $b^k_{i,n}(j) \in \{0,1\}, n \in \mathbb{Z}_{[1,4]}$ is responsible for making a choice between the set of 4 constraints. Having $b^k_{i,n}(j) = 1$ ensures the automatic satisfaction of the constraint corresponding to $n \in \mathbb{Z}_{[1,4]}$ due to $M \gg 0$ while (4.34) ensures that, at most, 3 of the 4 constraints can be automatically satisfied, or in other words, at least one of them has to be strictly enforced.

However, with the introduction of $b^k_{i,n}(j) \in \{0,1\}^{4\times H\times N_v(k)}$, where 4 comes from the number of constraints to choose from, H corresponds to the planning horizon, and $N_v(k) = |\mathbb{O}(k)|$ represents the number of vehicles in the observation set of the ego vehicle, we increase the scale of the optimization problem while converting it into a mixed-integer program. As a consequence, the runtime complexity of the optimization problem increases, affecting the real-time applicability of the approach.

4.7.1.3 Elliptical Constraint

In the Cartesian coordinate system, an ellipse centered at $q \in \mathbb{R}^2$ can be represented by the set $\mathbb{E} = \{z \mid (z-q)^{\mathrm{T}} Q^{-1}(z-q) \leq 1\}$ where $z = \begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$, and Q is a positive definite symmetric matrix containing the orientation (rotation) and dimensional information (length and width) of the ellipse. Therefore, if each vehicle $i \in \mathbb{O}(k)$ is modeled as an ellipse, its spatial representation is given by $\mathbb{E}_i(k) = \{z \mid (z-p_i(k))^{\mathrm{T}} Q_i^{-1}(z-p_i(k)) \leq 1\}$ where $Q_i(k)$ contains the vehicle i's orientation (yaw) and dimensional (length and width) information. More details regarding the definition of $Q_i(k)$ can be found in [117]. Now, the collision avoidance condition states that two ellipses *should not* intersect i.e. $\mathbb{E}_0(k) \cap \mathbb{E}_i(k) = \emptyset$ where, as per usual, subscript $_0$ is reserved for the ego vehicle. In order to enforce this condition, we pose the following constraint

for the optimization problem: [106]:

$$\bar{p}_i^{\mathrm{T}}(k)\bar{Q}_i(k)p_i(k) \geq 1 \tag{4.35}$$

where $\bar{Q}_i(k)$ is an appropriately defined augmented shape matrix, considering the relative orientation and motion between $\mathbb{E}_0(k)$ and $\mathbb{E}_i(k)$.

The constraint above is only for the current time instant k so in order to extend it to the receding horizon optimization framework, we have to introduce variables for every future time step $j \in \mathbb{Z}_{[0,H]}$ i.e. replace $\bar{p}_i(k)$ with $\bar{p}_i^k(j)$ and $\bar{Q}_i(k)$ with $\bar{Q}_i^k(j)$. For an accurate representation of the augmented safe set into the future, the future orientation information of the ego vehicle ($\theta_0^k(j)$) needs to be incorporated into $\bar{Q}_i^k(j)$. Then, the product between the trigonometric functions of $\theta_0^k(j)$ - from $\bar{Q}_i^k(j)$ - with the positional variables ($x_0^k(j)$ and $y_0^k(j)$) - from $p_0^k(j)$ - leads to non-linearity and non-convexity, increasing the complexity of the problem. Moreover, even with the simplifying assumption, stating that the orientation of the ego vehicle stays the same for the duration of the planning horizon, which may prove detrimental in a crash mitigation scenario, the problem already has high complexity due to the inherent non-convexity resulting from the presence of $\geq$ sign in the constraints. The constraints essentially require the ego vehicle (represented by a point mass at $p_0^k(j)$) to be present outside the augmented elliptical (convex) safe set, defined by $\bar{Q}_i^k(j)$. This non-convexity in constraints leads to the high computational complexity of the optimization problem.

Chapter 5: Conclusion

This dissertation addresses some critical research problems that lie on the path to bringing the idea of widespread public presence of fully autonomous vehicles to fruition. We focus on scenarios where autonomous vehicles need to operate alongside human-driven vehicles. This brings forth numerous challenges associated with the inherent unpredictability of human behavior, that must be effectively addressed before autonomous vehicles can gain the general public's approval.

In this dissertation, we address various navigation tasks where the core requirements encompass efficient real-time decision-making, reliable motion prediction and behavior consideration of on-road agents, and safety prioritization for all the agents involved while ensuring passenger comfort. The specific problems addressed in our research include bidirectional highway overtaking, highway maneuvering in dense traffic, and crash mitigation on highways. To address these problems, we develop a modular multi-timescale navigation architecture with the goal of performing safe, efficient, and comfortable driving. The navigation task comprises the decision-making, planning, and control subtasks but we limit our scope, in this dissertation, to the decision-making and planning aspects due to the prevalence of a variety of control paradigms in the existing literature. The notable characteristics of the proposed approaches include real-time operation capability, consideration of behavior variability of on-road agents, modularity in design, and optimality across

various metrics. As for our verification and validation procedures, we perform extensive comparative analyses over singular case studies as well as randomized Monte Carlo scenarios in realistic simulation environments.

To conclude, we briefly recap our main contributions to the problems addressed in this dissertation while providing a synopsis of the future outlook.

5.1 Summary of Contributions

We develop algorithms that enable autonomous vehicles to operate seamlessly within the existing traffic ecosystem while accounting for the diverse driving patterns exhibited by human drivers. To that end, we address a variety of commonly encountered driving scenarios.

Firstly, we discuss the overtaking scenario on a bidirectional two-lane undivided highway where the core emphasis is on maintaining safety as opposed to enhancing performance, by minimizing travel time, since any miscalculation can lead to an at-fault potentially fatal head-on collision. We develop non-cooperative as well as cooperative optimization-based approaches while employing realistic sensing models that account for sensor occlusion. This gains our approach the ability to operate in a realistic sensing range while accounting for occlusion and the ability to retract an initiated overtake maneuver, paving the way for active perception.

Then, we move on to the maneuvering problem in traffic on a multi-lane highway, where an overly conservative or defensive approach may be detrimental to the ego vehicle as well as the surrounding vehicles so a balanced approach, compensating between safety and performance, is required. This problem focuses on the importance of foresight as well as explicit behavior consid-

eration of on-road agents through risk evaluation in decision-making and planning. We propose interaction and risk agnostic, and interaction and risk aware optimization-based approaches that achieve high computational efficiency through various reformulations and algorithmic modifications.

Finally, we address the crash mitigation problem where an autonomous vehicle finds itself in an extremely precarious situation with a collision being imminent, so it has to take evasive actions. Since the safety guarantees in this situation are non-existent, the goal is to execute potentially drastic maneuvers to ideally avoid a collision or minimize its severity if it is indeed unavoidable. We present a novel risk-aware framework composed of an activation mechanism responsible for triggering the crash mitigation system and an optimization-based trajectory generation module that computes an evasive strategy based on situational risk minimization.

5.2 Future Work

The proposed algorithms' rigorous formulations coupled with their verification and validation procedures carried out in high-fidelity simulations serve as compelling proof-of-concepts and demonstrate their strengths relative to the various existing methods. However, for their widespread adoption and further development towards practical implementation on actual autonomous vehicles, it is essential to first evaluate their performance on a small-scale physical setup that closely emulates real-world conditions. By implementing the approach on realistic small-scale vehicles that accurately capture low-level vehicular dynamics, we can evaluate our navigation framework's robustness in the face of real-world uncertainties, disturbances, and time delays. Enhancing real-

world performance may involve incorporating various noise and time delay models, which, in theory, can seamlessly be handled by the modular structure of the proposed method. Therefore, the near-future outlook for the proposed navigation framework and the associated algorithms include implementing, fine-tuning, and experimenting in controlled small-scale real-world environments in the hopes of having some of these algorithms deployed on publicly available autonomous vehicles.

Bibliography

[1] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation*, pages 987–993. IEEE, 2010.

[2] Ryosuke Okuda, Yuki Kajiwara, and Kazuaki Terashima. A survey of technical trend of adas and autonomous driving. In *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, pages 1–4. IEEE, 2014.

[3] Istvan Barabás, Adrian Todoruţ, N Cordoş, and Andreia Molea. Current challenges in autonomous driving. In *IOP conference series: materials science and engineering*, volume 252, page 012096. IOP Publishing, 2017.

[4] Adam Ziebinski, Rafal Cupek, Hueseyin Erdogan, and Sonja Waechter. A survey of adas technologies for the future perspective of sensor fusion. In *International Conference on Computational Collective Intelligence*, pages 135–146. Springer, 2016.

[5] P. Koopman and M. Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.

[6] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[7] Yorie Nakahira, Quanying Liu, Terrence J Sejnowski, and John C Doyle. Diversity-enabled sweet spots in layered architectures and speed–accuracy trade-offs in sensorimotor control. *Proceedings of the National Academy of Sciences*, 118(22), 2021.

[8] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.

[9] Yonghwan Jeong and Seongjin Yim. Model predictive control-based integrated path tracking and velocity control for autonomous vehicle with four-wheel independent steering and driving. *Electronics*, 10(22):2812, 2021.

[10] Jingjing Jiang and Alessandro Astolfi. Lateral control of an autonomous vehicle. *IEEE Transactions on Intelligent Vehicles*, 3(2):228–237, 2018.

[11] Chuan Hu, Rongrong Wang, Fengjun Yan, and Nan Chen. Output constraint control on path following of four-wheel independently actuated autonomous ground vehicles. *IEEE Transactions on Vehicular Technology*, 65(6):4033–4043, 2015.

[12] Armin Norouzi, Ali Barari, and Hadi Adibi-Asl. Stability control of an autonomous vehicle in overtaking manoeuvre using wheel slip control. *International Journal of Intelligent Transportation Systems Research*, 18(2):320–330, 2020.

[13] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, 2007.

[14] Yixiao Liang, Yinong Li, Amir Khajepour, and Ling Zheng. Holistic adaptive multi-model predictive control for the path following of 4wid autonomous vehicles. *IEEE Transactions on Vehicular Technology*, 70(1):69–81, 2021.

[15] Eugenio Alcalá, Vicenç Puig, and Joseba Quevedo. Lpv-mpc control for autonomous vehicles. *IFAC-PapersOnLine*, 52(28):106–113, 2019.

[16] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1–14, 2014.

[17] Faizan M. Tariq, Nilesh Suriyarachchi, Christos Mavridis, and John S. Baras. Autonomous vehicle overtaking in a bidirectional mixed-traffic setting. In *2022 American Control Conference (ACC)*, 2022.

[18] Faizan M Tariq, Nilesh Suriyarachchi, Christos Mavridis, and John S Baras. Cooperative bidirectional mixed-traffic overtaking. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2494–2501. IEEE, 2022.

[19] Faizan M. Tariq, David Isele, John S. Baras, and Sangjae Bae. Slas: Speed and lane advisory system for highway navigation. In *2022 61st IEEE Conference on Decision and Control (CDC)*, 2022.

[20] Benoit Vanholme, Dominique Gruyer, Benoit Lusetti, Sebastien Glaser, and Said Mammar. Highly automated driving on highways based on legal safety. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):333–347, 2012.

[21] Christina Miller, Christian Pek, and Matthias Althoff. Efficient mixed-integer programming for longitudinal and lateral motion planning of autonomous vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1954–1961. IEEE, 2018.

[22] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P How. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on control systems technology*, 17(5):1105–1118, 2009.

[23] Weria Khaksar, Khairul Salleh Mohamed Sahari, and Tang Sai Hong. Application of sampling-based motion planning algorithms in autonomous vehicle navigation. *Autonomous Vehicle*, 735, 2016.

[24] Carl-Johan Hoel, Krister Wolff, and Leo Laine. Automated speed and lane change decision making using deep reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2148–2155. IEEE, 2018.

[25] M. Kaushik, V. Prasad, K. M. Krishna, and B. Ravindran. Overtaking maneuvers in simulated highway driving using deep reinforcement learning. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1885–1890, 2018.

[26] Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, and Lydie Nouvelière. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):589–606, 2010.

[27] Giuseppe Franze and Walter Lucia. A receding horizon control strategy for autonomous vehicles in dynamic environments. *IEEE Transactions on Control Systems Technology*, 24(2):695–702, 2015.

[28] Jongsang Suh, Heungseok Chae, and Kyongsu Yi. Stochastic model-predictive control for lane change decision of automated driving vehicles. *IEEE Transactions on Vehicular Technology*, 67(6):4771–4782, 2018.

[29] Shilp Dixit, Umberto Montanaro, Mehrdad Dianati, David Oxtoby, Tom Mizutani, Alexandros Mouzakitis, and Saber Fallah. Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2310–2323, 2019.

[30] Fisnik Sulejmani, Florian Reiterer, Amin Assadi, and Luigi del Re. Autonomous overtaking assistant for country road scenarios. In *2020 American Control Conference (ACC)*, pages 1217–1222. IEEE, 2020.

[31] Aneesh Raghavan, Jieqiang Wei, John S Baras, and Karl Henrik Johansson. Stochastic control formulation of the car overtake problem. *IFAC-PapersOnLine*, 51(9):124–129, 2018.

[32] Yulong Gao, Frank J Jiang, Karl H Johansson, and Lihua Xie. Stochastic modeling and optimal control for automated overtaking. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1273–1278. IEEE, 2019.

[33] Nilesh Suriyarachchi, Faizan M. Tariq, Christos Mavridis, and John S. Baras. Real-time priority-based cooperative highway merging for heterogeneous autonomous traffic. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2019–2026, 2021.

[34] Nilesh Suriyarachchi and John S. Baras. Shock wave mitigation in multi-lane highways using vehicle-to-vehicle communication. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 1–7, 2021.

[35] Junlan Chen, Ke Wang, Huanhuan Bao, and Tao Chen. A design of cooperative overtaking based on complex lane detection and collision risk estimation. *IEEE Access*, 7:87951–87959, 2019.

[36] Ruoqi Deng, Boya Di, and Lingyang Song. Cooperative collision avoidance for overtaking maneuvers in cellular v2x-based autonomous driving. *IEEE Transactions on Vehicular Technology*, 68(5):4434–4446, 2019.

[37] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius Brito Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago Meireles Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, page 113816, 2020.

[38] Shih-Lin Lin and Bing-Han Wu. Application of kalman filter to improve 3d lidar signals of autonomous vehicles in adverse weather. *Applied Sciences*, 11(7), 2021.

[39] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[40] Rachid Attia, Rodolfo Orjuela, and Michel Basset. Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, 52(2):261–279, 2014.

[41] Georg Schildbach and Francesco Borrelli. Scenario model predictive control for lane change assistance on highways. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 611–616. IEEE, 2015.

[42] Rafael Lazimy. Mixed-integer quadratic programming. *Mathematical Programming*, 22(1):332–349, 1982.

[43] I. Griva, S.G. Nash, and A. Sofer. *Linear and Nonlinear Optimization: Second Edition*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), 2009.

[44] Shilp Dixit, Saber Fallah, Umberto Montanaro, Mehrdad Dianati, Alan Stevens, Francis Mccullough, and Alexandros Mouzakitis. Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Annual Reviews in Control*, 45:76–86, 2018.

[45] Zhen Wang, Xiaowei Shi, and Xiaopeng Li. Review of lane-changing maneuvers of connected and automated vehicles: models, algorithms and traffic impact analyses. *Journal of the Indian Institute of Science*, 99(4):589–599, 2019.

[46] Bustanul Arifin, Bhakti Yudo Suprapto, Sri Arttini Dwi Prasetyowati, and Zainuddin Nawawi. The lateral control of autonomous vehicles: A review. In *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 277–282, 2019.

[47] Alyssa Pierson, Wilko Schwarting, Sertac Karaman, and Daniela Rus. Navigating congested environments with risk level sets. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5712–5719. IEEE, 2018.

[48] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, pages 2575–2582. IEEE, November 2018.

[49] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

[50] Daniel V McGehee, Elizabeth N Mazzae, and GH Scott Baldwin. Driver reaction time in crash avoidance research: Validation of a driving simulator study on a test track. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 44, pages 3–320. Sage Publications Sage CA: Los Angeles, CA, 2000.

[51] S. Krauss, P. Wagner, and C. Gawron. Metastable states in a microscopic model of traffic flow. *Phys. Rev. E*, 55:5597–5602, May 1997.

[52] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, 62:1805–1824, Aug 2000.

[53] Jakob Erdmann. Sumo's lane-changing model. In Michael Behrisch and Melanie Weber, editors, *2nd SUMO User Conference*, volume 13 of *Lecture Notes in Control and Information Sciences*, pages 105–123. Springer Verlag, 2015.

[54] Guodong Liu, Siyu Chen, Ziqian Zeng, Huijie Cui, Yanfei Fang, Dongqing Gu, Zhiyong Yin, and Zhengguo Wang. Risk factors for extremely serious road accidents: Results from national road accident statistical annual report of china. *PLoS one*, 13(8):e0201587, 2018.

[55] Sangjae Bae, David Isele, Kikuo Fujimura, and Scott J Moura. Risk-aware lane selection on highway with dynamic obstacles. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 652–659. IEEE, 2021.

[56] Patrick Olsen. Cadillac tops tesla in consumer reports' first ranking of automated driving systems. *Consum. Rep*, 2018.

[57] Arne Kesting, Martin Treiber, and Dirk Helbing. General lane-changing model mobil for car-following models. *Transportation Research Record*, 1999(1):86–94, 2007.

[58] Anirudha Majumdar and Marco Pavone. How should a robot assess risk? towards an axiomatic theory of risk in robotics. In *Robotics Research*, pages 75–84. Springer, 2020.

[59] Yimin Chen, Chuan Hu, and Junmin Wang. Motion planning with velocity prediction and composite nonlinear feedback tracking control for lane-change strategy of autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 5(1):63–74, 2019.

[60] Wei Liu and Marcelo H. Ang. Incremental sampling-based algorithm for risk-aware planning under motion uncertainty. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058, 2014.

[61] Mustafa Mukadam, Akansel Cosgun, Alireza Nakhaei, and Kikuo Fujimura. Tactical decision making for lane changing with deep reinforcement learning. 2017.

[62] Jiachen Yang, Alireza Nakhaei, David Isele, Kikuo Fujimura, and Hongyuan Zha. Cm3: Cooperative multi-goal multi-stage multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2020.

[63] Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. Driving in dense traffic with model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5385–5392. IEEE, 2020.

[64] Piyush Gupta, Demetris Coleman, and Joshua E. Siegel. Towards physically adversarial intelligent networks (pains) for safer self-driving. *IEEE Control Systems Letters*, 7:1063–1068, 2023.

[65] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and systems*, volume 2, pages 1–9. Ann Arbor, MI, USA, 2016.

[66] Jaime F Fisac, Eli Bronstein, Elis Stefansson, Dorsa Sadigh, S Shankar Sastry, and Anca D Dragan. Hierarchical game-theoretic planning for autonomous vehicles. In *2019 International conference on robotics and automation (ICRA)*, pages 9590–9596. IEEE, 2019.

[67] David Isele. Interactive decision making for autonomous vehicles in dense traffic. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3981–3986. IEEE, 2019.

[68] Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50):24972–24978, 2019.

[69] Chang Liu, Seungho Lee, Scott Varnhagen, and H Eric Tseng. Path planning for autonomous vehicles using model predictive control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 174–179. IEEE, 2017.

[70] Sangjae Bae, Dhruv Saxena, Alireza Nakhaei, Chiho Choi, Kikuo Fujimura, and Scott Moura. Cooperation-aware lane change maneuver in dense traffic based on model predictive control with recurrent neural network. In *2020 American Control Conference (ACC)*, pages 1209–1216. IEEE, 2020.

[71] Kazuhide Okamoto, Karl Berntorp, and Stefano Di Cairano. Similarity-based vehicle-motion prediction. In *2017 American Control Conference (ACC)*, pages 303–308. IEEE, 2017.

[72] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.

[73] Christian Pek, Peter Zahn, and Matthias Althoff. Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules. In *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017.

[74] A. Richards and J. How. Mixed-integer programming for control. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2676–2683 vol. 4, 2005.

[75] Hugues Marchand, Alexander Martin, Robert Weismantel, and Laurence Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123(1-3):397–446, 2002.

[76] Bernardo Cuteri, Carmine Dodaro, Francesco Ricca, and Peter Schüller. Constraints, lazy constraints, or propagators in asp solving: An empirical analysis. *Theory and Practice of Logic Programming*, 17(5-6):780–799, 2017.

[77] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[78] Hans Föllmer and Alexander Schied. Stochastic finance. In *Stochastic Finance*. de Gruyter, 2016.

[79] Shaun S Wang. A class of distortion operators for pricing financial and insurance risks. *Journal of risk and insurance*, pages 15–36, 2000.

[80] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[81] R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[82] Ravi Kumar Kolla, LA Prashanth, Sanjay P Bhat, and Krishna Jagannathan. Concentration bounds for empirical conditional value-at-risk: The unbounded case. *Operations Research Letters*, 47(1):16–20, 2019.

[83] I Isaksson-Hellman and M Lindman. Evaluation of rear-end collision avoidance technologies based on real world crash data. *Proceedings of the Future Active Safety Technology Towards zero traffic accidents (FASTzero), Gothenburg, Sweden*, pages 9–11, 2015.

[84] Wesley Hulshof, Iain Knight, Alix Edwards, Matthew Avery, and Colin Grover. Autonomous emergency braking test results. In *Proceedings of the 23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, pages 1–13. National Highway Traffic Safety Administration Washington, DC, 2013.

[85] Vicente Milanés, Joshué Pérez, Jorge Godoy, and Enrique Onieva. A fuzzy aid rear-end collision warning/avoidance system. *Expert Systems with Applications*, 39(10):9097–9107, 2012.

[86] Marcel P Huijser, Patrick T McGowen, Whisper Camel, et al. Animal vehicle crash mitigation using advanced technology phase i: Review, design, and implementation. 2006.

[87] Marcel P Huijser, Tiffany D Holland, Angela V Kociolek, Aaron M Barkdoll, Jessica D Schwalm, et al. Animal-vehicle crash mitigation using advanced technology: phase ii, system effectiveness and system acceptance. Technical report, Oregon. Dept. of Transportation. Research Unit, 2009.

[88] Marcus Müller, Michael Botsch, Dennis Böhmländer, and Wolfgang Utschick. Machine learning based prediction of crash severity distributions for mitigation strategies. *Journal of Advances in Information Technology*, 9(1):15–24, 2018.

[89] Kibeom Lee and Dongsuk Kum. Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map. *IEEE Access*, 7:52846–52857, 2019.

[90] Hong Wang, Yanjun Huang, Amir Khajepour, Yubiao Zhang, Yadollah Rasekhipour, and Dongpu Cao. Crash mitigation in motion planning for autonomous vehicles. *IEEE transactions on intelligent transportation systems*, 20(9):3313–3323, 2019.

[91] Xu Shang and Azim Eskandarian. Emergency collision avoidance and mitigation using model predictive control and artificial potential function. *IEEE Transactions on Intelligent Vehicles*, 2023.

[92] Yechen Qin, Ehsan Hashemi, and Amir Khajepour. Integrated crash avoidance and mitigation algorithm for autonomous vehicles. *IEEE Transactions on Industrial Informatics*, 17(11):7246–7255, 2021.

[93] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE intelligent vehicles symposium (IV)*, pages 1094–1099. IEEE, 2015.

[94] Yiqi Gao, Theresa Lin, Francesco Borrelli, Eric Tseng, and Davor Hrovat. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In *Dynamic systems and control conference*, volume 44175, pages 265–272, 2010.

[95] Ahmadreza Moradipari, Sangjae Bae, Mahnoosh Alizadeh, Ehsan Moradi Pari, and David Isele. Predicting parameters for modeling traffic participants. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 703–708. IEEE, 2022.

[96] Guofa Li, Yifan Yang, Tingru Zhang, Xingda Qu, Dongpu Cao, Bo Cheng, and Keqiang Li. Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios. *Transportation research part C: emerging technologies*, 122:102820, 2021.

[97] Yizhen Zhang, Erik K Antonsson, and Karl Grote. A new threat assessment measure for collision avoidance systems. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 968–975. IEEE, 2006.

[98] Adelchi Azzalini and A Dalla Valle. The multivariate skew-normal distribution. *Biometrika*, 83(4):715–726, 1996.

[99] Dahlia Sam, Cyrilraj Velanganni, and T Esther Evangelin. A vehicle control system using a time synchronized hybrid vanet to reduce road accidents caused by human error. *Vehicular communications*, 6:17–28, 2016.

[100] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015.

[101] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[102] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

[103] Iain S Duff. Sparse system solution and the hsl library. *Some topics in industrial and applied mathematics*, 8:78–94, 2006.

[104] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[105] Matthias Gerdts, René Henrion, Dietmar Hömberg, and Chantal Landry. Path planning and collision avoidance for robots. *Numerical Algebra, Control and Optimization*, 2(3):437–463, 2012.

[106] Yiqi Gao, Andrew Gray, Ashwin Carvalho, H Eric Tseng, and Francesco Borrelli. Robust nonlinear predictive control for semiautonomous ground vehicles. In *2014 American control conference*, pages 4913–4918. IEEE, 2014.

[107] Tom Schouwenaars, Bart De Moor, Eric Feron, and Jonathan How. Mixed integer programming for multi-vehicle path planning. In *2001 European control conference (ECC)*, pages 2603–2608. IEEE, 2001.

[108] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA, 1997.

[109] Lars Blackmore, Masahiro Ono, and Brian C Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011.

[110] Alexander T Schwarm and Michael Nikolaou. Chance-constrained model predictive control. *AIChE Journal*, 45(8):1743–1752, 1999.

[111] Haitian Tan, Guangquan Lu, and Miaomiao Liu. Risk field model of driving and its application in modeling car-following behavior. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11605–11620, 2021.

[112] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[113] Jerome Barraquand, Bruno Langlois, and J-C Latombe. Numerical potential field techniques for robot path planning. *IEEE transactions on systems, man, and cybernetics*, 22(2):224–241, 1992.

[114] Pengwei Wang, Song Gao, Liang Li, Binbin Sun, and Shuo Cheng. Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm. *Energies*, 12(12):2342, 2019.

[115] Zhiqiang Zuo, Xu Yang, Zhicheng Zhang, and Yijing Wang. Lane-associated mpc path planning for autonomous vehicles. In *2019 Chinese Control Conference (CCC)*, pages 6627–6632. IEEE, 2019.

[116] Yadollah Rasekhipour, Amir Khajepour, Shih-Ken Chen, and Bakhtiar Litkouhi. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1255–1267, 2016.

[117] Koju Nishimoto, Riku Funada, Tatsuya Ibuki, and Mitsuji Sampei. Collision avoidance for elliptical agents with control barrier function utilizing supporting lines. In *2022 American Control Conference (ACC)*, pages 5147–5153. IEEE, 2022.