

ABSTRACT

Title of Dissertation: **COOPERATIVE MULTI-AGENT
SENSING, PLANNING, AND CONTROL
FOR CONNECTED AUTONOMOUS VEHICLES**

Chethana Nilesh Suriyarachchi
Doctor of Philosophy, 2023

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

Modern transportation networks are witnessing ever-increasing levels of vehicle automation as subsystems originally controlled by the human driver are taken over by automated control systems. Connected autonomous vehicles (CAVs) represent the latest technological breakthrough in this domain, which promises increased safety, improved efficiency, and better accessibility. Recently, more and more academic and industrial research interest has been focused on guaranteeing the safe operation of a single CAV by harnessing its advanced sensing and computation capability. However, CAVs also have another very important capability whose potential remains largely untapped: communication. Communication among CAVs (V2V - Vehicle to Vehicle) and with infrastructure (V2I - Vehicle to Infrastructure) provide CAVs the ability to share progressively gathered information, policies, and rules, and pursue controls related to both local and global objectives (in space and time). Thus, CAVs can collaboratively achieve levels of safety and efficiency

that cannot be achieved by a single vehicle working alone.

With this new focus on cooperative CAV control, we propose methods and associated algorithmic implementations, through which CAVs can contribute to solving some of the persistent problems which have been plaguing road and highway infrastructure for decades. We provide problems such as highway merge junction bottlenecks, highway traffic shock waves, and signalized or unsignalized intersection management with vastly superior solutions by harnessing the communication capabilities of CAVs. In both the domains of fully autonomous traffic and mixed traffic (CAVs coexisting with human driven vehicles), communication-based CAV control allows for safer operation with improved global throughput of traffic flow. In a world facing increasing shortages of traditional fuel sources, the ability of cooperative CAV control to reduce overall fuel consumption levels is also quite notable.

The algorithms proposed in this dissertation, which use heuristic, optimization, and learning-based control, can be broadly categorized as cooperative multi-agent control in which CAV sensing and/or actuation data are communicated. Sharing of this data allows CAVs to gather information about downstream traffic conditions, and overcome problems such as hidden obstacles due to sensor occlusion and dangerous conditions ahead or out of a single vehicle's "line of sight". This increased collection of information available to the control center (either ego vehicle or infrastructure) can then be used to compute better controls leading to safer, more efficient operation, and the ability to prevent the occurrence of certain unwanted traffic conditions. This potential capability of CAVs is explored using both centralized (often V2I-based) and decentralized (often V2V-based) control strategies. Moreover, as ensuring the safe operation of the overall system is a primary con-

cern, in addition to achieving the desired performance, safety is a fundamental constraint built into all these methods.

Our overall goal is to provide practically implementable algorithms that achieve high levels of performance and are capable of real-time operation while demonstrating robustness to variations in the environment. As such, we pay special attention to realistic driving environments including the impact of features such as multiple lanes, mixed traffic, heterogeneous traffic (cars, buses, trucks, emergency vehicles, etc.), and limitations in both communication and computation resources. We investigate the properties of communication networks used, as well as the ability of the control systems to adapt to imperfections in actuation, sensing, and communication.

COOPERATIVE MULTI-AGENT SENSING, PLANNING, AND CONTROL FOR CONNECTED AUTONOMOUS VEHICLES

by

Chethana Nilesh Suriyarachchi

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

© Copyright by
Chethana Nilesch Suriyarachchi
2023

Dedication

To my partner Leena, and my parents, Chandima and Nayani.

Acknowledgments

First and foremost, I would like to express my gratitude to my advisor, Professor John S. Baras, for giving me the opportunity to work in one of the most interesting, challenging, and impactful research fields of our era. It has been an honor to work under Prof. Baras whose tireless passion, focus on solving real-world problems and insightful guidance helped shape me into the researcher I am today. His eagerness to learn new things and his motivation to meet any challenge were infectious characteristics that I will always admire and attempt to emulate. My dissertation research has also been supported by the A. James Clark School of Engineering at the University of Maryland (UMD), with the Dean's Fellowship in 2018-2019 and the Maryland Robotics Center GRA Program Fellowship in 2022-2023, and the Office of Naval Research (ONR), with the grant N00014-17-1-2622.

I would like to thank all 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 endeavors. My interaction with Prof. Manocha also taught me valuable experiences in collaboration across teams and departments on joint research. I must acknowledge the valuable experience I gained from my research internships at STEER Tech, MD, in 2019, the Nokia Bell Labs, NJ, in 2020, and the Mitsubishi Electric Research Laboratories (MERL), MA, in 2022. I would like to thank my industrial supervisors, Dr. Anuja Sonalker the CEO of STEER Tech, Mr.

Jay Gaonkar and Mr. Sander van Valkenburg of Nokia Bell Labs, and Dr. Rien Quirynen and Dr. Stefano Di Cairano of Mitsubishi Electric Research Laboratories respectively. Both Dr. Anuja Sonalker and Dr. Rien Quirynen especially have been very close advisors who were always willing to listen and help in both research and external life challenges/decisions.

My dissertation work would not have been the same without the people that supported me during the process. First, I would like to thank my parents, Chandima and Nayani, and my sister, Shenali for their unwavering support and advice throughout all the trials and tribulations of life. Then, there are the members of Prof. Baras' research group, who I have been working with for the past years, and have shaped me through discussions, presentations, and collaborations. I would like to thank the entire group, and especially my friends, Christos, Faizan, Fatemeh, Panos, Asim, Charles, Praveen and Sandeep. I would also like to thank Mrs. Kimberly Edwards for her ever-present support with any problem that may have occurred within the university. Of course, my lifelong friends Hasun, Kavinda, Nandula, Yasas, Medha, and Adrianna, should also always be acknowledged for their unconditional support and heartfelt friendship.

My furball golden retriever Loki cannot go without mention as the constant source of stress relief and joy in my life. Last but definitely not least, I would like to thank my long-term partner and now fiancé, Leena, who has actively been supporting me at all times, withstanding the pressure of being in a relationship with a Ph.D. candidate while being one herself. She was the first person I met at UMD, and has remained by my side throughout this journey that would not have been possible without her.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	v
List of Tables	ix
List of Figures	x
List of Abbreviations	xiii
Chapter 1: Introduction	1
1.1 Challenges in coordinated CAV control	4
1.1.1 Choosing centralized vs decentralized control	5
1.1.2 Impact of heterogeneous vehicles	6
1.1.3 CAV control in mixed traffic	6
1.1.4 Importance of real-time control	7
1.1.5 Optimization vs Heuristic vs Reinforcement learning control	8
1.1.6 V2X Communication and knowledge representation	10
1.2 Contributions in Coordinated CAV control	11
1.2.1 Highway on-ramp merging control	11
1.2.2 Complex unsignalized intersections	14
1.2.3 Large-scale signalized multi-intersection control	17
1.2.4 Highway shock wave dissipation	19
1.2.5 Vehicular Communication (V2X) modeling and information sharing	22
Chapter 2: Cooperative Highway Merging for Heterogeneous Mixed Traffic	25
2.1 Evolution of Methodologies	25
2.2 Problem Description and Structure	28
2.2.1 Modeling the physical system	28
2.2.2 Optimal merging control formulation	31
2.3 Handling Heterogeneous Vehicles	33
2.4 Priority-based Real-time capable method for cooperative merging of Heterogeneous Traffic	34

2.4.1	Outline and System structure	34
2.4.2	Optimal sequence generation	36
2.4.3	Optimization subsystem	38
2.5	System performance and impact	43
2.5.1	Performance metrics and comparisons	45
2.5.2	Main performance metric results	47
2.5.3	Delay metric results	49
2.5.4	Fuel consumption metric results	50
2.5.5	Real-time operation	51
2.6	Further extensions to centralized highway merging control	51
2.6.1	Full capacity utilization lane changing controller	52
2.6.2	Changes to allow for mixed traffic	53
2.6.3	Improvements to merging sequence generation	57
2.6.4	Results from improved highway merging control system	59
2.7	Decentralized V2V Communication-based Method for Real-time Cooperative Highway Merging of Heterogeneous Autonomous Traffic	64
2.7.1	Dividing the problem into inter-related sub-problems	65
2.7.2	Optimization problem for a single vehicle group	71
2.7.3	Importance of V2V Communication	71
2.8	Future work	73
2.8.1	Comparison against full-scale optimization formulation	73
2.8.2	Macroscopic effects of the proposed merging method	74
2.8.3	Reinforcement learning extension	75
Chapter 3:	Optimal Real-time Multi-Agent Planning and Control for Dynamic Intersections	76
3.1	Introduction	76
3.1.1	Related Work	78
3.1.2	Main Contributions:	80
3.2	Problem Formulation	81
3.2.1	Modeling the Physical Intersection	82
3.2.2	Vehicle Dynamics and Control Variables	83
3.2.3	Dynamic Traffic Intersection Planning & Control Problem	85
3.2.4	Auction Theory	87
3.3	GAMEOPT: Methodology	87
3.3.1	GAME: Selecting the Priority Order	88
3.3.2	Multi-lane Intersection Planning and Control	93
3.3.3	OPT: Computing Optimal Velocities	94
3.4	Experiments and Results	98
3.4.1	Experiment Setup	98
3.4.2	Evaluation Metrics and Baselines	98
3.4.3	Efficiency Analysis	99
3.4.4	Effect of Overbidding and Underbidding	101

3.5	Extensions to GAMEOPT intersection control	101
3.5.1	Extension to heterogeneous vehicles	101
3.5.2	Extension of multi-bidding strategy for multi-sequence generation	103
3.5.3	Extended results with the updated version of GAMEOPT	107
Chapter 4: Coordination and Control of Traffic Lights and Mixed Traffic in Multi-Intersection Environments		
		117
4.1	Introduction	117
4.2	Problem Description	120
4.2.1	Modeling of Physical Road Network	121
4.2.2	Modeling of Controlled Vehicles (CAVs)	122
4.2.3	Modeling of Human Driven Vehicles (HDVs)	122
4.2.4	Traffic Light Intersection Model	123
4.3	MILP Formulation for Centralized Traffic Coordination	124
4.3.1	CAV Control Trajectories	125
4.3.2	Collision Avoidance	127
4.3.3	Intersection Membership	129
4.3.4	HDV Modeling and Prediction	130
4.3.5	Traffic Light Timing Constraints	134
4.3.6	MILP Objective Function	135
4.4	Decentralized Traffic Coordination	136
4.4.1	Decentralized Architecture	137
4.4.2	Vehicle Assignment to MILP-ITCs	137
4.5	Numerical validation of results with simulations	138
4.5.1	Traffic Simulation	139
4.5.2	Comparison Metrics and Baselines	140
4.5.3	Performance analysis	141
4.6	Hierarchical Traffic Coordination extension	146
4.6.1	Heuristic central traffic light command (H-CTLIC)	147
4.6.2	Optimization central traffic light command (O-CTLIC)	148
4.6.3	ITC cost function	153
4.6.4	Results and numerical validation of Hierarchical Traffic Coordination	154
4.7	Conclusions	158
Chapter 5: Shock Wave Mitigation in Multi-lane Highways using Vehicle-to-Vehicle Communication		
		159
5.1	Evolution of Methodologies	159
5.2	Problem overview	160
5.3	Modeling the network and vehicles	161
5.3.1	Modeling the physical highway structure	161
5.3.2	Vehicle Model	162
5.4	Microscopic traffic models	164
5.4.1	Krauss Car-following Model	165

5.4.2	Lane change model	166
5.5	V2V Communication-based Shock wave dissipation method	167
5.5.1	Traffic state estimation	167
5.5.2	Shock wave detection	169
5.5.3	Control for shock wave dissipation	171
5.6	Experimental Setup and Results	173
5.6.1	Proposed method Performance	174
5.6.2	Comparison between Methods	176
5.6.3	Impact of CAV penetration levels	178
5.7	Improvements to Shock-Wave Dissipation Control	179
5.7.1	Proactive Shock-Wave Dissipation Controller	179
5.7.2	CAV's Lane-Changing Controller	183
5.7.3	Improved proactive control results	184
5.8	Deep Reinforcement Learning Shock-Wave Dissipation Control	189
5.8.1	Deep Reinforcement Learning Formulation	191
5.8.2	DRL Experimental Setup and Results	198
5.9	Future Work	202
5.9.1	Accurate modeling of V2V communication	203
5.9.2	Improvements to Reinforcement learning based shock wave control	204
Chapter 6:	V2X communication modeling and applications	205
6.1	Introduction	205
6.2	Communication technology and simulation overview	207
6.2.1	V2I communication capabilities	207
6.2.2	V2V communication capabilities	208
6.2.3	Communication modeling software	209
6.3	Simplified communication modeling for control applications	210
6.3.1	Algorithm features and structure	210
6.3.2	Incorporating communication limitations in algorithms	211
6.3.3	Trade-off in communication bandwidth vs system performance	212
6.4	Cooperative bidirectional overtaking application	212
6.4.1	Overview	213
6.4.2	Modeling sensor occlusion	215
6.4.3	Information sharing using V2V communication	216
6.4.4	Experiments and results	217
6.5	Graph Networks for Highway Traffic State Estimation in Mixed Traffic Settings	220
6.5.1	Dynamic multi-graphs for representing traffic state	221
6.5.2	Global State Graphs by combining local CAV state graphs	222
6.5.3	GNN-based behavior and traffic prediction on graphs	224
Chapter 7:	Conclusion and Future Work	226
	Bibliography	231

List of Tables

2.1	System parameters used in testing	45
3.1	Comparison with prior work: We highlight approaches for navigating unsignal- ized intersections based on the type of intersection, optimality guarantees, and real- time capability.	79
4.1	Mapping from 9 traffic light states to the value 0 (red) or 1 (green) for each of the 12 traffic light directions.	124
4.2	Centralized computation cost using GUROBI, with varying traffic inflow rates and $N = 10, r_{\text{cav}} = 67\%$	136

List of Figures

1.1	On-ramp merging into a multi-lane highway.	12
1.2	Unsignalized Intersection with multiple input lanes.	14
1.3	Traffic road network with 5 signalized intersections.	17
1.4	Circular multi-lane highway loop for shock wave simulation.	20
1.5	Bidirectional traffic overtaking scenario.	23
2.1	On-ramp Merging Regions and Infrastructure Model	29
2.2	Optimal highway merging algorithm architecture	35
2.3	Experimental highway merge setup.	44
2.4	Comparison of performance for varying loop density	48
2.5	Comparison of delay faced with varying loop density	49
2.6	Comparison of fuel consumption with loop density	50
2.7	On-ramp merging into a multi-lane highway with CAVs and HDVs.	60
2.8	Performance in terms of throughput and density in comparison to baseline zipper merge, with increasing vehicle input flow rate. (CAV penetration level: 100%)	61
2.9	Performance in terms of delays faced by vehicles and fuel consumption in comparison to baseline zipper merge, with increasing vehicle input flow rate. (CAV penetration level: 100%)	62
2.10	Performance in terms of emergency vehicle time-to-goal and truck fuel consumption in comparison to baselines, with increasing vehicle input flow rate. (CAV penetration level: 100%)	62
2.11	Performance in terms of throughput and density in comparison to baseline zipper merge, with increasing vehicle input flow rate. (100%, 80% and 60% CAV penetration levels.)	63
2.12	Vehicle group formation for centralized merging.	66
3.1	GAMEOPT vs. Traffic Lights: We present a new approach for optimal real-time planning and control in dynamic, multi-agent unsignalized intersections. In this figure, the light blue regions denote the control zone. We show that at identical input traffic flow levels, our approach outperforms even traffic light infrastructure, resulting in less queues and enabling smooth traffic flow.	77
3.2	Regions of interest: We highlight the control zone in which vehicles communicate with the control tower and the conflict zone in which vehicles cross over to their desired target road.	83

3.3	GAMEOPT overview: Our approach begins by reading the positions and velocities of all agents in the control zone (blue). Our approach is hybrid; in the planning phase, GAMEOPT collects the bids from every agent and generates an optimal priority sequence (Section 3.3.1). In the controls phase, we use an optimization-based trajectory planner to compute the optimal velocity for each agent that satisfies the priority order while simultaneously guaranteeing safety and real-time performance (Section 2.4.3).	84
3.4	Conflict handling at multi-lane intersections. We propose a novel strategy for conflict handling using vehicle grouping based on non-colliding trajectories.	113
3.5	Comparison with baselines: GAMEOPT increases throughput by approximately 25% and 174% compared to TRAFFIC-LIGHT and AUCTION, respectively. Furthermore, our method reduces the time-to-goal by 75% and 81%, respectively, and improves fuel efficiency by 33% and 66%, respectively, compared with TRAFFIC-LIGHT and AUCTION.	114
3.6	Impact of increasing the speed limit and unbalanced inflow rates on the average time-to-goal. GAMEOPT exhibits improvements with a higher speed limit and invariance to unbalanced flow.	114
3.7	Performance in terms of throughput and time-to-goal in comparison to baselines, with increasing vehicle input flow rate.	114
3.8	Performance in terms of fuel consumption and CO2 emissions in comparison to baselines, with increasing vehicle input flow rate.	115
3.9	Performance in terms of emergency vehicle time-to-goal and truck fuel consumption in comparison to baselines, with increasing vehicle input flow rate.	115
3.10	Performance in terms of throughput and time-to-goal in comparison to baselines, with unbalanced vehicle input flow rate. Total flow rate fixed at 5200 <i>veh/hr</i> .	115
3.11	Performance in terms of fuel consumption and CO2 emissions in comparison to baselines, with unbalanced vehicle input flow rate. Total flow rate fixed at 5200 <i>veh/hr</i> .	116
3.12	Performance in terms of emergency vehicle time-to-goal and truck fuel consumption in comparison to baselines, with unbalanced vehicle input flow rate. Total flow rate fixed at 5200 <i>veh/hr</i> .	116
4.1	Schematic of the proposed central traffic controller (CTC) directly controlling connected automated vehicles (CAVs) and indirectly affecting the motion of human driven vehicles (HDVs) by commanding traffic lights.	120
4.2	Traffic road network with 5 intersections, where each road segment consists of 3 lanes in both directions and 4 road segments where vehicles can flow in and out of the road network.	121
4.3	Each 4-way traffic intersection, with 3 lanes in each road segment, includes traffic lights to control the flow in each of 12 possible directions.	123
4.4	Traffic road network with 5 signalized intersections in SUMO. Blue vehicles are HDVs and red vehicles are CAVs.	139
4.8	Directed graph representation for optimization-based CTLC (O-CTLC) of the transportation network.	153

4.9	SUMO results over 2000s for increasing inflow rates $r_{in} = 0.42, 0.54, 0.66$ and 0.78 veh/s.	155
4.10	Computation time comparison for increasing inflow rates $r_{in} = 0.42, 0.54, 0.66$ and 0.78 veh/s.	157
5.1	Modeling a highway section including vehicles	162
5.2	Modeling sensing capability of CAVs	164
5.3	Circular multi-lane highway loop simulation	174
5.4	Variation in trajectories of vehicles	175
5.5	Variation in velocities of vehicles	176
5.6	Velocity Standard Deviation for different methods	177
5.7	Average velocity comparison for different methods	177
5.8	Velocity std. dev. for varying CAV penetration levels	178
5.9	Closed-loop velocity-position profiles for different parameter values $\delta_i \in [0, 5, 10, 15, 20, 30]$ ($a_i = 1, d_c = 300$). Here v_i, s_i , are the initial velocity and position of a CAV that detects a shock wave at a distance s_{det}^i from the worst case minimum average velocity $v_{det}^i(t)$	180
5.10	Variation in trajectories of vehicles.	184
5.11	Variation in velocities of vehicles.	185
5.12	Velocity Standard Deviation for different methods.	186
5.13	Average velocity comparison for different methods.	187
5.14	Flow diagram for Deep RL shock wave control architecture.	196
5.15	Circular multi-lane highway simulation	199
5.16	Variation in trajectories of vehicles.	200
5.17	Deep RL training performance	201
6.1	Bidirectional traffic overtaking scenario.	213
6.2	Sensor Occlusion Model.	215
6.3	Overtaking success over varying flow rate.	218
6.4	Overtakes attempted over varying flow rate.	218
6.5	Overtaking success over CAV penetration level.	219
6.6	Overtakes attempted over CAV penetration level.	220
6.7	Co-evolving Multi-graph Model	223

List of Abbreviations

CAV	Connected Autonomous Vehicle
CTC	Central Traffic Controller
CTLC	Central Traffic Light Command
DRL	Deep Reinforcement Learning
EV	Emergency Vehicle
GNN	Graph Neural Networks
H-CTLC	Heuristic Central Traffic Light Command
HDV	Human Driven Vehicle
ITC	Intersection Traffic Controller
KL	Kullback-Libeler
LC	Lane Changing
LP	Linear Programming
LSTM	Long Short Term Memory Networks
MERL	Mitsubishi Electric Research Laboratories
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Programming
MIQP	Mixed Integer Quadratic Programming
MPC	Model Predictive Control
O-CTLC	Optimization Central Traffic Light Command
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
RNN	Recurrent Neural Networks

SSA	Sponsored Search Auction
SUMO	Simulation of Urban Mobility Software
UMD	University of Maryland
V2I	Vehicle to Infrastructure Communication
V2V	Vehicle to Vehicle Communication
V2X	Vehicle to Everything Communication
VANET	Vehicular Ad hoc Network
WAVE	Wireless Access in Vehicular Environments
WLAN	Wireless Local Area Network

Chapter 1: Introduction

Over the past decade, we witnessed an overwhelming interest in research related to connected vehicles, electrified vehicles, and self-driving vehicles. The introduction of these vehicles, which we collectively call Connected Autonomous Vehicles (CAVs), is rapidly changing from what was once a futurist dream to a present-day reality. These CAVs have the potential to improve safety, efficiency, and accessibility which are the three major performance criteria in road infrastructure-based transportation systems.

Both safety and efficiency are becoming an increasing concern as vehicular demand continues to increase much faster than improvements in road infrastructure can keep up. These higher levels of vehicle densities along with increases in dangerous driving behavior such as distracted driving have led to some alarming statistics. In the United States, 42915 lives were lost in 2021 [1, 2] due to road traffic accidents which is the highest number recorded since 2007. There was an estimated \$340 Billion cost to American society in 2019 due to traffic crashes [3]. Furthermore, on the efficiency side it is shown that in 2022, an average driver in the United States annually incurs a time cost of \$869, a fuel cost of \$546, and a time wastage of 51 hours due to congestion [4]. The unfortunate truth is that this economic impact of congestion is only expected to further increase with the increasing cost of fossil fuels as it has also been shown that fuel prices have a small effect

on the amount people traveled. Therefore, we find ourselves in a situation in which the abilities of CAVs to potentially improve safety and efficiency need to be harnessed to the extent possible.

It is also important to note that the introduction of CAVs is not a sudden transition but instead a gradual process in which vehicles are changing from systems that completely rely on human drivers for decision-making and control, to automated systems which perform a large range of activities without the need for human intervention. Technologies such as adaptive cruise control, lane-keeping assist, and automated parking have already become commonplace in modern vehicles. Furthermore, onboard sensing and communication technologies have also seen drastic improvements. Modern vehicles often come equipped with radar sensors, vision-based sensors, and even lidar ranging sensors. This allows vehicles to observe more of their surroundings and control algorithms can leverage this information in order to make better decisions. The other major improvement appearing in vehicles is in communication technologies. The advent of 5G networking has enabled vehicles to share larger amounts of data at low latency, from vehicle to vehicle (V2V) and from vehicle to infrastructure (V2I) (Collectively called V2X). These changes are some of the major forces contributing to the development and acceptance of CAVs today. With these scientific breakthroughs pushing the technology of CAVs forward, a lot of attention is now being put on governing policies for these vehicles, infrastructure improvements to facilitate these vehicles, and even government incentive programs for the adoption of these vehicles. This is primarily due to the multitude of benefits presented by CAVs, which include safety improvements, efficiency improvements, quality of life improvements, and even the positive impact CAVs can have on the environment.

When we analyze the most common CAV research areas, it is evident that a large portion of research focuses on the micro-level behavior planning and control of CAVs. Focus is placed on ensuring safety, achieving robustness with respect to handling a diverse range of driving scenarios, and enabling wide-range applicability of CAVs in various domains. This research provides a great step forward in terms of individual vehicle safety and the practical applicability of CAVs. These improvements play a major role in the increasing use of CAVs in multiple domains ranging from personal transportation vehicles to autonomous trucks for goods delivery to autonomous taxi systems. It is important to note here that this research is carried out from the point of view of an individual CAV, by focusing on the use of enhanced onboard sensing and computation to achieve safer and more efficient driving. However, it is often true that applying a control that is optimal from one vehicle's perspective does not necessarily result in the best global strategy. The use of multi-agent control strategies that aim to reach a globally optimal result often vastly outperforms single-agent strategies in terms of overall safety, efficiency, and performance achieved. This is especially true in the case of CAV control during traffic bottleneck conditions such as highway merge junctions, highway shock waves, and complex intersections. In these conditions optimally coordinating groups of CAVs yield far better results than each CAV trying to optimize its own individual control.

From a global perspective, it becomes apparent that the true power of CAVs can only be harnessed by leveraging both the individual controllability aspect of CAVs along with their communication capabilities which unlock their potential for cooperation and collaboration. To achieve this the word "Connected" in the term CAV plays an equally important role as the word "Au-

tonomous”. When compared to single-agent control, when algorithms attempt to coordinate multiple agents in order to achieve a globally optimal state, the communication capabilities of CAVs play a very important role. This communication can take the form of Vehicle to Infrastructure (V2I) communication often used in many centralized control strategies or even Vehicle to Vehicle (V2V) communication often used in decentralized control strategies. In addition to enhanced communication methods, the control algorithms also need to adapt and improve in order to provide real-time computation capability even in the presence of multiple agents.

The primary goal of this dissertation is to explore the numerous possibilities brought about by communication-based coordinated CAV control. Here, the focus is placed on the challenges, applications, and various other aspects of the cooperative CAV control domain.

1.1 Challenges in coordinated CAV control

It is quite clear that coordinated CAV control can provide many advantages in the form of improved safety and efficiency. When vehicles and infrastructure communicate with each other, they can share critical information such as intent, detected road agents (vehicles and pedestrians), road hazards, and even predicted driver behavior. This information can then be used for multiple tasks such as overcoming blind spots caused by sensor occlusion, decision-making with regards to which vehicle receives the right of way, and reacting in advance to hazardous environmental conditions. While these advantages are numerous, there are also significant challenges that need to be addressed for the successful implementation of cooperative control strategies.

1.1.1 Choosing centralized vs decentralized control

Centralized control methods focus on collecting all information at one point, choosing the optimal decision for all agents, and communicating a recommended control strategy back to each of the agents. In contrast, decentralized methods do not have one fixed agent with access to all relevant data and responsible for decision-making. Instead, they either have a process of vehicle grouping, appointing a leader on the fly and computing a decision for the group, or, each agent computing its own decision based on the information provided to it by other agents within communication range. Centralized methods are often associated with V2I communication and have geographically fixed operating locations while decentralized methods are associated with V2V communication and can be implemented at any location.

Both centralized and decentralized control methods offer different advantages and the choice of which one should be used often depends on the application domain. Centralized methods relying on infrastructure facilities often have a lot more computation capability thus allowing them to compute controls for a large number of vehicles in real-time and often reach higher levels of overall performance. Computation units on infrastructure would have larger antennas with longer communication ranges. They would also not have energy usage constraints as they don't rely on battery power. On the other hand, the primary advantage of decentralized methods is their ease of implementation, wide applicability, and the lack of reliance on infrastructure investments. It is also important to point out that it is much easier to implement governing policies and cyber security methods in centralized methods than in decentralized methods. This is due to the fact that the control node remains in a fixed location and the introduction and maintenance of these nodes

would often fall under the responsibility of a governing agency. The choice between these two strategies will often boil down to the computation needs, cost of implementation, safety concerns, and required performance level related to the application in question.

1.1.2 Impact of heterogeneous vehicles

The term “Heterogeneous vehicles” refers to the different types of vehicles on modern road transportation networks. These types include passenger cars, cargo vans, trucks, emergency vehicles, motorcycles, and many more. These vehicle types often have diverse needs and requirements. For example, emergency vehicles prioritize speed above all else while trucks would want to reduce their acceleration and deceleration events due to high fuel costs. Furthermore, the dynamics of these vehicle classes can be substantially different. This involves characteristics such as acceleration and braking capabilities as well as other factors like steering rate and top speed. Therefore, when a coordinated control algorithm is being implemented it is important to take these diverse requirements into consideration. Algorithms that consider all vehicles as one homogeneous class often yield highly sub-optimal results in real-world applications.

1.1.3 CAV control in mixed traffic

The term mixed traffic relates to the conditions in which human driven vehicles (HDVs) and connected autonomous vehicles (CAVs) coexist in the same environment. This situation introduces many complexities that don’t surface in systems involving only autonomous agents. Human driving unpredictability and latency in reaction time can create many complex situations that au-

onomous need to account for and adjust accordingly. In the case of guaranteeing safety, human driving behavior needs to be accurately classified in order to set the correct levels of safety margins and control thresholds. For example, as a safety consideration, a CAV needs to give a HDV classified as aggressive more space due to the large variance in its possible future actions. As the safety of the system may depend on the accuracy in the prediction of a human driver's future actions, this area of research has received a lot of recent research attention. Additionally, many problems faced by modern transportation networks can often be linked back to the issues related to human driving. Problems such as highway shock waves, deadlock at intersections, collisions due to rule-breaking, and general inefficiencies at bottlenecks highlight the impact of human-related issues such as delayed reaction times, lack of communication, non-cooperative decision making, and unpredictability of future actions. As an increasingly greater percentage of vehicles become CAVs, research shows that many of these problems can be alleviated. This is mainly due to the fact that the CAVs dispersed among the HDVs can act as Lagrangian control points which can influence the actions of all vehicles around them. Applying a suitable control strategy to these CAVs can have a drastic effect on the overall flow of traffic.

1.1.4 Importance of real-time control

Most of the applications related to CAV control occur in highly dynamic environments with agents moving at high speeds. In order for a CAV to operate safely in these environments it needs high control update frequencies. Control algorithms need to adapt to changes in the environment and adjust the control strategy within a very short time. This means that the control algorithms

developed need to be capable of very fast computation times. Practically in most situations, the controls need to be updated at a frequency of at least $1Hz$ and often above $10Hz$. In fact, higher expected vehicle velocities may even require faster control updates. If the control algorithm can compute and provide outputs at the desired control frequency we refer to these algorithms as real-time capable.

1.1.5 Optimization vs Heuristic vs Reinforcement learning control

Many coordinated CAV control problems involve the search for the optimal set of controls each agent should follow in order to achieve some global objectives such as safety and efficiency. As such, optimization-based methods such as those involving model predictive control are often quite suited to the problems considered. Many of these problems have elegant optimization-based solutions that achieve excellent performance while guaranteeing safe operation with a mix of hard and soft constraints. However, as the size of the problem, observation space, and action space grows larger, certain optimization formulations become increasingly unable to compute controls in real time. In some of these situations, we find that rule-based heuristic control can often provide a good enough solution. This means that in certain structured cases, the sub-optimality introduced by a heuristic method is within our desired tolerances and the computation speed increase outweighs the minimal drop in performance.

Additionally, in many control problems related to traffic networks, a large amount of data can usually be collected regarding the environment conditions via vehicle or infrastructure mounted sensors. This data is used to model the state of the environment with details involving vehicles,

pedestrians, other agents, and obstacles in the vicinity. In most heuristic and optimization formulations, certain data points are selected or combined in order to form the input state of the system. In most cases, a large amount of data is discarded and effectively unused in the computation process in these methods. This is mostly due to the fact that complex optimization problems which consider all this data are impossible to compute in real-time. This is a major issue since most automated vehicle-related control problems require online real-time computation capability due to the dynamic nature of these environments.

Deep reinforcement learning (RL) methods try to overcome these issues by finding a suitable direct relationship between the input state and output actions. This allows us to use a much larger portion of the available data and leave it up to the neural network to discover the relationships between the input data and output actions of the system. Deep RL algorithms are also capable of real-time operation on a learned network structure. After training is complete the learned policy can be used to select suitable actions resulting in a real-time capable system. However, Deep RL systems have significant disadvantages that need to be accounted for. The training process can be very unsafe and therefore initial training has to be done in simulation. Furthermore, when moving from simulation to real-world training, there may be difficulty guaranteeing safety if the sim to real gap is large. This is a major concern in control for automated vehicles as safety is the top priority. Many auxiliary systems need to be in place for the safe deployment of an RL-based system and even with all this support, it may be difficult to provide any fixed guarantees on the system's operational safety.

1.1.6 V2X Communication and knowledge representation

Modern CAVs often have a large number of sensors on board which generate enormous amounts of data continuously. This data is often too large to be stored on onboard storage and once utilized is discarded in most cases. Due to this fact, it is very important to extract all the useful information from this data before it is discarded. It may also be the case that even this extracted information may be too large to store indefinitely. Due to these problems, the task of knowledge representation becomes essential. This is further compounded by the need to share this data between vehicles and possibly infrastructure due to the additional constraints of network bandwidth and communication delays. The need for minimizing the amount of information that needs to be shared over the V2X communication network becomes more critical in real-time control applications. Sharing large amounts of sensor data is often infeasible and leads to sub-par performance. As such extracting the needed information features and representing them in appropriate data structures are essential steps in a realistic efficient communication-based control strategy for CAVs.

Knowledge representation is the process of storing the gathered information in a meaningful way so that it can be used for inferring relationships among the data, understanding what information is useful for each task, and selecting the minimum amount of information that needs to be shared in communication-based systems. Many different methods such as gridmaps, graph data structures, tree data structures, and organized databases have been suggested for the task of representing some of the more complex information collections. While each of these methods has their own advantages and disadvantages, the selection of the method to be used depends on the task specification, constraints on the system, and expected relationships within the captured data.

In the case in which the data captured by an autonomous vehicle has both spatial and temporal relationships, either temporal or spatial data aggregation methods or graph-based methods such as dynamically evolving multi-graphs show a lot of promise. Certain highly time-critical applications may even need further knowledge compaction to obtain the minimum information that needs to be shared. In such cases, it is possible to leverage the computation capability onboard CAVs to perform initial reasoning and feature extraction tasks so that the information that needs to be shared over the network is minimized.

1.2 Contributions in Coordinated CAV control

While the use of coordinated CAV control is widely applicable in almost all driving scenarios, there are certain traffic bottleneck conditions that really highlight the advantages and possibilities of this research domain. The conditions include highway merge junctions, highway shock wave conditions, and complex intersections. If coordinated CAVs can be shown to have significant advantages in these complex scenarios, then their applicability in other simpler scenarios is a given. In this dissertation we focus on some key coordinated CAV control application domains and in this section, we present our contributions to each of these domains. These application domains will later be explained in detail in the form of the chapters that make up the rest of this dissertation.

1.2.1 Highway on-ramp merging control

Highway on-ramp merge junctions remain a major bottleneck in transportation networks. This section describes a method of increasing highway merging efficiency by optimally coordinat-

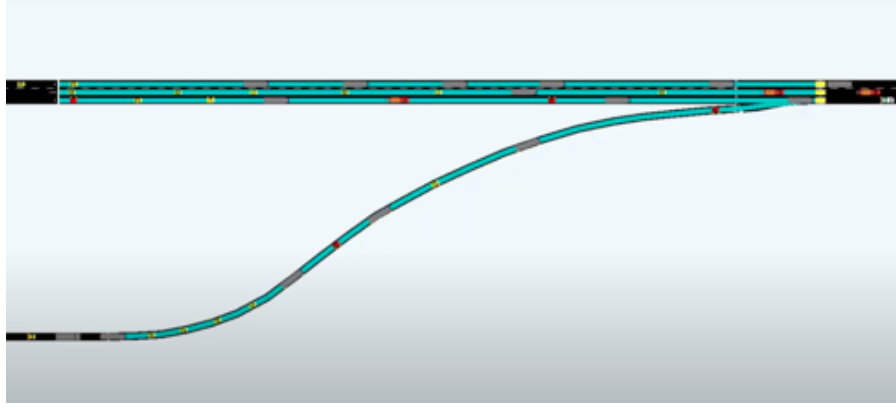


Figure 1.1: On-ramp merging into a multi-lane highway.

ing CAVs in order to maximize the flow of vehicles while satisfying all safety constraints. The general highway merging problem can be divided into two stages. The selection of the optimal merging sequence (order in which to merge) and controlling the actuation of all the vehicles in order to achieve the selected sequence. Sequence selection can become very difficult as the number of possibilities grows exponentially with the number of vehicles involved. Another consideration in the merging problem is the challenges introduced by the presence of heterogeneous traffic. Different types of vehicles may prioritize different aspects such as speed maximization or minimization of acceleration and deceleration tasks. Therefore, a practical controller for solving the merging problem should be capable of finding an optimized solution in real-time while considering the varying needs brought about by mixed traffic.

Our key contributions in the highway merging domain are as follows,

- Formulate a centralized hybrid rule and optimization-based strategy for computing controls for highway merging. Controls are computed by the central control infrastructure with information obtained via V2I communication.

- Achieve real-time computation capability with a parallel computation architecture achieved by breaking down the problem into rule-based merging sequence generation and then parallel optimization per sequence. The performance of our system can scale with available computation resources with solutions yielding higher performance when higher compute power is available.
- Account for the diverse needs of heterogeneous traffic by optimizing according to each vehicle's specific requirements.
- Guarantee safety with hard constraints and additional safety margins for discrepancies in sensing and actuation.
- Additional full highway capacity utilization with separate lane changing controller which maximizes the usage of all available highway lanes.
- Extension to allow for mixed traffic. HDVs coexist with CAVs with HDVs provided with larger safety margins due to lower predictability.
- Formulate a decentralized version of the algorithm with vehicle grouping and multiple distributed control computations based on information shared via V2V communication.
- Results are verified in a realistic traffic simulator under varying traffic conditions. Presented results showing significant improvements in key traffic performance metrics, such as higher throughput, lower density, reduced delay, and increased fuel efficiency.

1.2.2 Complex unsignalized intersections

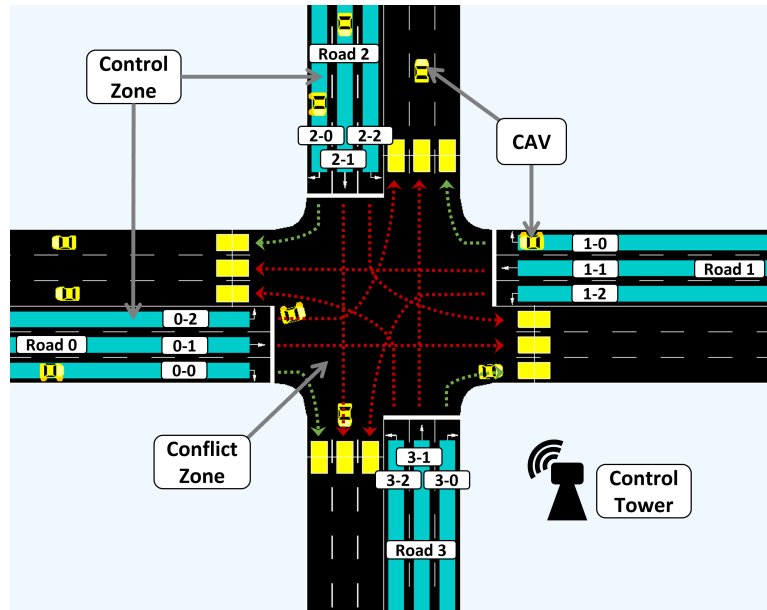


Figure 1.2: Unsignalized Intersection with multiple input lanes.

The urban driving domain is quite different from the previously discussed highway driving domain. It is often less structured, has many diverse obstacles and the environment is less predictable. However, the desired driving speeds in urban environments are lower than on highways to help counteract the higher levels of environmental uncertainty. This uncertainty mostly stems from the lack of knowledge and difficulty in behavior prediction of the multiple dynamic obstacles in the vicinity of a CAV. These obstacles can take many forms such as other vehicles, pedestrians, and cyclists. Additionally, as the environment is often quite cluttered, the ego vehicle (primary CAV) is unable to see many obstacles due to sensor occlusion. No matter how many sensors are fitted onto a vehicle, in a cluttered environment it will face numerous blind spots which could contain potential hazards. One method to help alleviate this issue is to harness the communication capabilities of CAVs and enable them to share information with each other or with roadside

infrastructure. Working together it is often possible for CAVs to drastically reduce the number of blind spots faced by each other. This would help make autonomous urban driving a lot safer and more efficient. This increased awareness and environmental knowledge would also allow vehicles to move faster while maintaining all safety criteria. Furthermore, the capabilities of this type of communication-based method would continue to improve as more and more vehicles become connected.

Within the domain of urban driving, we pay special attention to the CAVs behavior at unsignalized intersections as they exhibit a statistically high rate of occurrence of collisions. Both signalized and non-signalized intersections offer different types of challenges which need to be handled appropriately. These intersections often have many blind spots due to sensor occlusion, and vehicles may have to adapt control strategies very rapidly in order to prevent collisions. This process can be made safer and more efficient if vehicles are able to communicate and cooperate with each other. To achieve this We propose GAMEOPT: a novel hybrid approach to cooperative intersection control for dynamic, multi-lane, unsignalized intersections. Safely navigating these complex and accident-prone intersections requires simultaneous trajectory planning and negotiation among drivers. GAMEOPT is a hybrid formulation that first uses an auction mechanism to generate a priority entrance sequence for every agent, followed by an optimization-based trajectory planner that computes velocity controls that satisfy the priority sequence. This coupling operates at real-time capable computation speeds of less than 10 milliseconds per control cycle in high-density traffic of more than 10,000 vehicles/hr, $100\times$ faster than other comparable fully optimization-based methods, while providing guarantees in terms of fairness, safety, and efficiency.

Our key contributions in the unsignalized intersection control domain are as follows,

- System achieves game-theoretic optimality with the priority order sequence computed being fair, efficient, and tractable.
- System also achieves optimality in terms of the controls computed to achieve a given sequence being reachable and yielding the best performance. It ensures that the optimal trajectories satisfy the priority order with safety guarantees.
- Real-time computation achieved by the hybrid auction mechanism and optimization-based formulation which allows GAMEOPT to operate on average at control frequencies greater than $10Hz$ with more than 10000 vehicles per hour input traffic flow rates.
- System capable of handling dynamic input flow rates where the traffic flow in different arms of the intersection can vary with time. The proposed system can easily adapt to situations in which the flows in the different arms are significantly different from each other resulting in an unbalanced flow.
- Multi-lane capability is achieved by creating a formulation that can generalize to any number of lanes on each arm of the intersection. Furthermore, an increase in performance is obtained by allowing many vehicles to enter the intersection simultaneously with a vehicle grouping system which allows non-conflicting vehicles from different lanes to enter the intersection at the same time.
- Extending capabilities to allow the system to handle heterogeneous traffic with diverse needs in both the auction and optimization sub-sections.

- Extending the auction-based sequence generator to a multi-bidding multi-sequence strategy in which it can generate a collection of sequences by changing its focus on various criteria such as vehicle position, speed, waiting time, vehicle type, and individual vehicle preference.
- Our proposed system is tested on the SUMO simulator, and our results show that we improve throughput by at least 25%, time taken to reach the goal by 75%, and fuel consumption by 33% compared to auction-based approaches and signaled approaches using traffic-lights and stop signs.

1.2.3 Large-scale signalized multi-intersection control

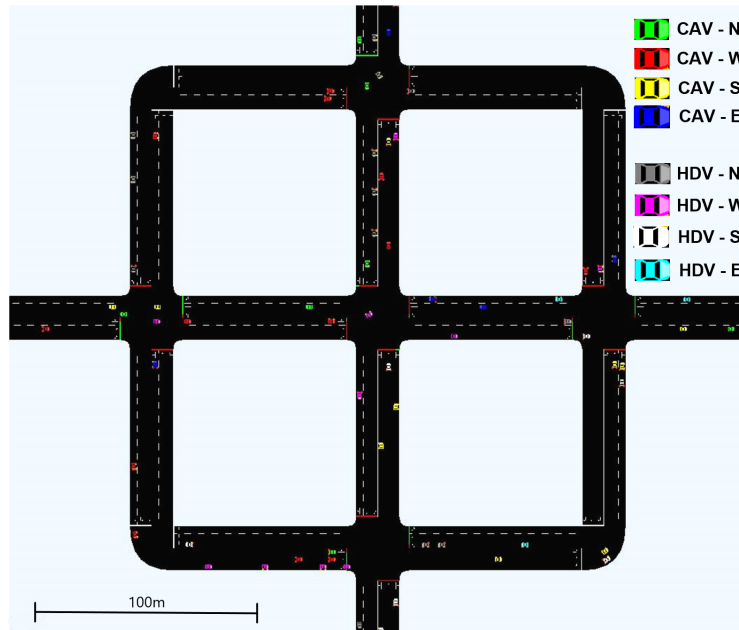


Figure 1.3: Traffic road network with 5 signalized intersections.

Coordinating the flow of traffic through urban areas with multiple intersections is a complex problem whose solution has the potential to improve safety, increase throughput, and optimize energy efficiency. Many existing macro-level approaches attempt to find the best strategies for

switching traffic lights at multiple connected intersections to obtain the best traffic flow. In addition to controlling traffic lights, the introduction of connected and automated vehicles (CAVs) offers opportunities in terms of additional sensing and actuation points within the traffic network. By jointly controlling CAVs and traffic lights we can create a system that significantly outperforms existing methods in terms of traffic throughput, time spent, and fuel consumption costs. This section proposes a centralized implementation for the joint coordination and control of both traffic signals and mixed traffic, including CAVs and human driven vehicles (HDVs), in a network of multiple connected traffic intersections. Mixed-integer linear programming (MILP) is used to compute safe control trajectories for both CAVs and traffic light signals, which minimize overall congestion and fuel consumption. Computational tractability with a 1 Hz control frequency, even at high traffic densities, is then achieved with a decentralized method extension.’ A further extension introduces a hierarchical control strategy capable of performing at speeds close to the decentralized strategy while achieving performance closer to the centralized strategy.

Our key contributions in the signalized multi-intersection CAV and traffic light control domain are as follows,

- We describe a MILP formulation for vehicle and intersection coordination, with support for multiple intersections, mixed traffic, lane changes, HDV predictions, and a more advanced traffic light switching model.
- Mixed traffic operation capability is obtained by a switched prediction model for HDVs along with increased safety tolerances to account for the lower predictability of HDV behavior.

- A decentralized implementation, which works by decoupling the MILP, is proposed to allow multiple intersection traffic controllers (ITCs), to achieve real-time computations. This computation gain is obtained at the cost of overall system performance and optimality.
- An additional hierarchical implementation is then proposed to bridge the optimality gap between centralized and decentralized implementations. This method creates a central traffic light command which uses a macro-level traffic model to provide guidance to the individual intersection ITCs. Together this system operates at real-time capable speeds with a minimal reduction in overall performance.
- All components of our system are built with a modular software architecture, allowing it to adapt to any traffic scenario.
- We perform extensive validations of the proposed approach based on SUMO, using realistic simulations of vehicle demand, behavior, and interactions.
- Our approaches are shown to provide improvements of around 32-60%, 90-96% and 40-60% in travel time, waiting time, and fuel consumption, respectively, when compared to gap-based adaptive and timed traffic lights.

1.2.4 Highway shock wave dissipation

Highway shock waves are persistent naturally occurring phenomena, which have been plaguing road and highway infrastructure for decades. They are caused by the delays in reaction times of Human Driven Vehicles (HDVs) resulting in unnecessary congestion in highway networks. Shock

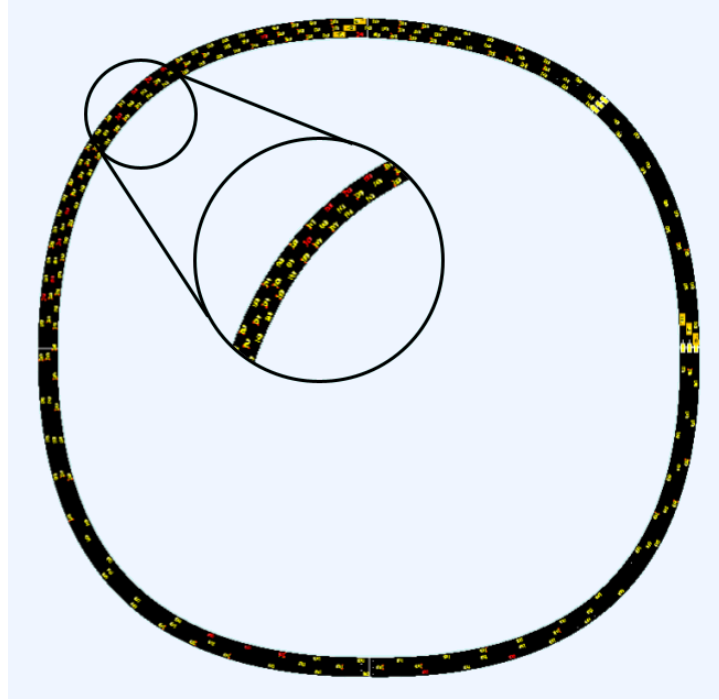


Figure 1.4: Circular multi-lane highway loop for shock wave simulation.

waves, also known as stop-and-go waves, are characterized by the traffic condition in which vehicles accelerate and decelerate in a periodic manner. This is a common cause of overall network efficiency degradation. Shock waves are also a major contributor to increased fuel consumption on highways, due to the large variance in velocity experienced by vehicles facing these conditions. In high traffic density highway conditions, these shock waves can easily be generated by even the simplest of triggers such as a few cars slowing down due to a distraction on the side of the road. Other common triggers for shock waves include busy highway merge junctions, disabled vehicles on the side of the road, vehicle collisions, and road construction activities. Additionally, once a shock wave forms, if left unchecked, it usually resolves only when the demand (vehicles entering the highway) reduces. As such, in high density periods, shock waves can last for hours on end. The main contributing factor to shock wave generation is the latency introduced by human decision-

making. Here, the focus is placed on using the communication and sensing capabilities of modern CAVs to solve the problem of stop-and-go waves in highway networks.

The major issue in dealing with shock waves lies in the difficulties in sensing highway conditions and subsequently actuating vehicles in a suitable manner to dissipate the shock wave. Some prior solutions to this involve infrastructure-based sensing and variable speed limits. However, with the advent of CAVs, many more options for sensing, actuation, and control have appeared. Even at low percentages of CAVs to HDVs (hereafter referred to as CAV penetration levels), the CAVs can act as Lagrangian sensors, collecting local information about the state of highway traffic. This local information can thereafter be combined to obtain a global state of traffic. Furthermore, CAVs can also act as Lagrangian actuators, allowing the control of themselves and their surrounding vehicles via inter-vehicle interactions. Thus, even the slowing down and speeding up of HDVs can be achieved by suitable actuation of the CAVs dispersed among the HDVs. The process of shock wave dissipation involves the use of CAVs as moving bottlenecks which reduce the flow into a congested high density area. This gives enough time for the congestion to clear up and over time if applied correctly can result in complete shock wave dissipation.

Our work on shock wave mitigation focuses on introducing a communication-based cooperative control method for CAVs in multi-lane highways in a mixed-traffic setting. The method allows for proactive control application and exhibits good shock wave dissipation performance even with low CAV penetration levels. Our contributions in shock wave mitigation are as follows,

- Introduce a V2V communication-based shock wave mitigation strategy. CAVs collect information on neighboring vehicles. Share compacted information with other CAVs with

V2V communication. Each CAV chooses the best control strategy based on the collected information.

- We design a class of proactive shock wave mitigating CAV controllers which can be tuned to either prioritize shock wave dissipation performance or passenger comfort. Thus, the choice of controller depends on design parameters such as the aggressiveness of the driving behavior allowed.
- Implement a lane-changing controller which focuses on placing CAVs strategically across all lanes. This positioning of autonomous agents plays a major role in shock wave dissipation in multi-lane scenarios.
- We show that even at very low CAV penetration levels our controller exhibits good shock wave dissipation capability with full dissipation often achieved in less than 10 minutes.
- Introduce a multi-agent shared policy deep reinforcement learning approach for shock wave mitigation. We use the shared-parameter Proximal Policy Optimization (PPO) reinforcement learning strategy for obtaining the controls for each CAV in the simulation. We also built a custom SUMO-Gym wrapper for the multi-lane highway simulation with custom-designed observation space, action space, and rewards for each agent.

1.2.5 Vehicular Communication (V2X) modeling and information sharing

Generally, most coordinated CAV control algorithms are developed and tested in high-fidelity simulation environments. Due to this, it is important that these simulations accurately model the

state of the communication networks used. In a safety-critical application like self-driving cars, the importance of network characteristics such as latency, bandwidth, and packet loss should not be underestimated. Control algorithms need to be designed to take into account the shortcomings of communication systems and be able to adapt accordingly to guarantee the safety and comfort of passengers and pedestrians. Some methods to solve this issue could involve having adequate tolerances, backup redundancy methods, and stochastic approximations for received information. We also find that it is important to study the trade-off between the amount of information communicated and the relative system performance achieved. Here, the focus is placed on identifying the relationships between network capability, the minimum amount of information to be shared, and overall system performance achieved. The first stage of this process involves the creation of a simplified communication modeling system for control applications.

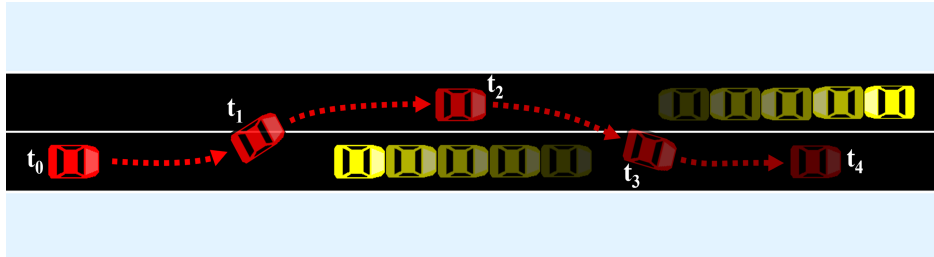


Figure 1.5: Bidirectional traffic overtaking scenario.

In order to showcase the importance of knowledge compaction and information sharing we implement a control algorithm for the cooperative bidirectional overtaking application. This application domain is significantly affected by blind spots in each CAVs observation space resulting from sensor occlusion. To overcome these difficulties, we propose 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 pro-

posed method in this section aims to combine the benefits of cooperation-driven communication-based CAV control systems with the safety guarantees of an MI-MPC optimization-based controller for the domain of bidirectional overtaking in mixed traffic. The main contribution of this work involves the implementation of a V2V 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 method was tested using the realistic SUMO traffic simulation system, and the results and comparisons are presented. We show that our 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 CAV penetration levels and lane density on the performance of our method.

Chapter 2: Cooperative Highway Merging for Heterogeneous Mixed Traffic

2.1 Evolution of Methodologies

Highway on-ramps are one of the biggest causes of congestion on modern highways. Due to the lack of coordination and limited visibility among vehicles, it is very common to see a traffic buildup around highway on-ramps [5]. The delays caused at on-ramps are some of the major contributors to overall system efficiency degradation [6]. In periods of high demand, the effect of slow-moving vehicles attempting to merge into a stream of fast-moving vehicles creates a bottleneck that often leads to traffic buildup in a large radius around the merge junction. Solving this problem requires vehicles to cooperate with each other and create gaps to enable smooth merging at high speeds [7]. This unfortunately is extremely difficult to achieve with human-driven vehicles. However, with the advent of CAVs, much more information has been made available for improving this overall process.

The problem of improving the throughput in highway merge junctions has been evolving with the advent of new technologies. Prior to the introduction of CAVs, most of the work was focused on ramp metering strategies [8], which control how many vehicles can merge [9], but don't consider the micro-simulation aspects of the merging process. However, as vehicles become

more advanced with improved sensing, decision-making, and communication capabilities, we now see a lot of effort in improving the merge process by taking advantage of the cooperativeness between connected vehicles. These efforts are often categorized either as rule-based methods or optimization-based methods.

Rule-based methods attempt to obtain near-optimal solutions with minimal computation cost. Some initial work on a zipper-based method where vehicles take turns in merging is presented by Sarvi and Kuwahara [10] and Scarinci *et al.* [11]. The work by Awal *et al.* [12] and more recently Ding *et al.* [13] showcase the improvements cooperation between vehicles can provide. While these algorithms are capable of real-time operation the result is usually non-optimal. There is no guarantee that the optimal sequence is found among the merging sequences generated, and it is difficult to model the interaction between vehicles.

The optimization-based methods aim to overcome this and obtain the optimal merge sequence and optimal control for all vehicles near the merge junction. However, this process generally has a very high computation cost. Different methods aim to circumvent this problem using many different kinds of assumptions. The work by Rios-Torres *et al.* [14, 15] demonstrates how the unconstrained optimization problem can be solved in real-time with Hamiltonian analysis. However, when the fully constrained problem incorporating collision constraints is considered the method presented in [14, 15] cannot be used. Optimization is restricted to one ramp vehicle and one mainline facilitating vehicle in the work by Zhou *et al.* [16]. Research by Chen *et al.* [17] attempts to solve the constrained problem using a two-level Mixed Integer Quadratic Programming (MIQP) optimization formulation. However, this is very computationally costly and cannot

be solved in real-time. In fact, they only compute sequences with one ramp vehicle for a very short time duration. This same problem is faced by other optimal control-based approaches such as that of Li *et al.* [18].

Another interesting approach to formulating this merging problem is the virtual slot-based approach [19]. Here a central controller creates virtual slots on the highway which each vehicle is then allocated to. This allows vehicles to be controlled centrally by controlling the slot parameters. This type of formulation performs poorly in the presence of mixed traffic and also does not provide an optimal solution to maximizing merging throughput. A similar virtual vehicle concept is also introduced by Uno *et al.* [20]. While all cooperative merging algorithms require some form of vehicle to vehicle or infrastructure communication, the work by Ito *et al.* [21] shows how the communication overhead can be reduced by using different broadcast strategies.

Contribution

Because of the nature of the highway merging problem, rule-based methods tend to yield sub-optimal solutions while methods based on end-to-end optimization are, by definition, computationally expensive and can hardly be used in real-time. Moreover, the existing methods invoke the assumption of homogeneous traffic, while the potential additional complications of mixed/heterogeneous traffic conditions have not been adequately studied.

The main contribution of this work to the merging problem is to ameliorate both these issues by creating a real-time operation capable hybrid rule and optimization-based merging algorithm with a parallel computation design. This hybrid approach leads to better performance without sac-

rificing real-time computation capabilities. This also allows the solution provided by the algorithm to be improved depending on the available computation resources. The varying requirements of different classes in heterogeneous traffic are handled by incorporating the concept of individual vehicle priorities into the optimization formulation.

Finally, the algorithm is tested on a realistic traffic simulator under varying traffic conditions, and the results and comparisons are presented. We show that our method yields improved traffic performance metrics, such as higher throughput, lower density, reduced delay, and improved fuel efficiency.

2.2 Problem Description and Structure

In this section, we formally define the fully automated highway merging problem as a large optimal control problem, and show that, by decomposing it into two simpler decoupled optimization problems, which employ a parallel computation architecture, a near-optimal solution can be computed in real-time.

2.2.1 Modeling the physical system

Consider the abstracted model of a highway on-ramp as shown in figure 2.1. We define a control zone, where all vehicles in this zone communicate with a central controller (V2I communication) and each other (V2V communication) to decide individual optimum velocities and paths to be followed. The control zone encompasses both the main road and the on-ramp. We denote the number of vehicles in the control zone by n , with m vehicles located in the mainline, and

$r = n - m$ located on the on-ramp. Vehicles are allowed to merge from the on-ramp onto the main road in the merge zone, which is located at the end of the control zone.

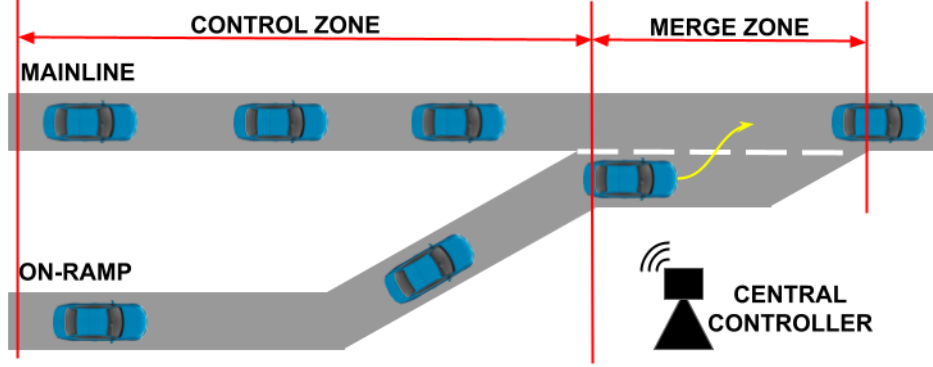


Figure 2.1: On-ramp Merging Regions and Infrastructure Model

In practice, the actual dynamics of the vehicles are unknown and highly non-linear, and, therefore, cannot be accurately modeled by the solver. In this regard, we assume that the low-level control of a vehicle is managed by a local controller w_i , which is also responsible for the control of lateral motion that keeps the vehicle in the lane. Therefore, the vehicle can be modeled as a point mass moving along the center of the lane according to a non-linear differential equation:

$$\dot{s}_i = f(t, s_i, w_i), \quad s_i(t_i^0) = s_i^0 \quad (2.1)$$

where t^0 is the time the vehicle enters the control zone. Throughout this manuscript, $s_i(t)$ will be measured as the distance of the vehicle i from the merge zone. Therefore, we can define the high-level vehicle dynamics by the following velocity control scheme:

$$\begin{aligned} \dot{s}_i &= v_i \\ v_i(t) &= u_i(t) \end{aligned} \quad (2.2)$$

where $s_i(t)$, and $v_i(t)$ denote the position and velocity of each vehicle i , respectively, along the direction of the lane, for $i \in \{1, \dots, n\}$.

Remark 1. *The control input $u_i(t)$ is the command velocity for vehicle i . It is necessary that the target set by this control is reachable by the low-level vehicle controller w_i and system (2.1). It is understood that since u_i will be the solution of the optimal merging problem, it should be designed to guarantee safe operation under small delays induced by the dynamics in (2.1).*

In addition to $s_i(t)$ and $v_i(t)$, we assign a priority value $p^i \in \mathbb{R}_+^2$ to each vehicle i , which affects the merging decision, and an indicator variable $b^i \in \{0, 1\}$ which signifies whether the vehicle is on the on-ramp or the mainline. The priority value has two scalar components, speed prioritization (p_s^i) based on the focus placed on vehicle velocity and speed variation prioritization (p_v^i) based on the tolerance to changes in velocity. Lastly, each vehicle is assumed to have a length l^i , and bounded acceleration capabilities characterized by its maximum acceleration a_{max} and maximum braking a_{min} capabilities, which carry information about lower-level variables such as the mass of the vehicle. Therefore, all cars are connected autonomous vehicles (CAV) completely defined by the state vectors:

$$x_i(t) = [s_i(t), v_i(t), p^i, b^i, l^i, a_{max}^i, a_{min}^i]^T \quad (2.3)$$

for $i \in \{1, \dots, n\}$. Lastly, regarding the parameters of the main road and the on-ramp, we assume speed limits $\bar{v}_m$ and $\bar{v}_r$, respectively, and we denote the length of the control zones by L_m and L_r . Without loss of generality, we will assume in this work that $\bar{v}_m = \bar{v}_r = \bar{v}$, a constant value that can be decided by the designer. The constant parameter tuple $M_s = \{M_{sr}, M_{sl}\}$, which denotes a safety margin between two vehicles, is also defined, and can be chosen by the designer. Here, M_{sr}

and M_{sl} represent the safety margins needed to prevent rear-end collisions and lateral collisions respectively.

2.2.2 Optimal merging control formulation

The highway merging problem can be formulated as an optimal control problem, where the goal is to estimate the optimal command velocities u_i for all $i \in \{1, \dots, n\}$, according to an appropriately defined objective function. Essentially, we want to minimize the time it takes for all vehicles to merge, which can be written as a minimization of the merging time of the last vehicle to be merged:

$$\begin{aligned} \min_{\{u_i\}} \quad & \max_i t_f^i \\ \text{s.t.} \quad & C(\{x_i\}, \bar{v}, L_m, L_r, M_s) \end{aligned} \tag{2.4}$$

where $t_f^i \in \arg \min_{\tau} \{s_i(\tau) = 0\}$ is the merging time of vehicle i . The set C is a set of constraints, to be defined later, that depend on the state vectors $\{x_i\}$ of the vehicles, equation (2.2), and the parameters introduced in Section 2.2.1. Based on the CAV assumption, near an optimal configuration, we expect that all the velocities $v_i(t)$ will be close to the limit $\bar{v}$. Thus, we can expect to achieve a near-optimal solution to the minimum-time optimization problem (2.4) by solving:

$$\begin{aligned} \min_{\{u_i\}} \quad & \sum_{i=1}^n p_s^i \lambda (u_i - \bar{v})^2 + p_v^i (1 - \lambda) (u_i - v_i)^2 \\ \text{s.t.} \quad & C(\{x_i\}, \bar{v}, L_m, L_r, M_s) \end{aligned} \tag{2.5}$$

The first term of the new objective function tries to keep the velocity command as close to the speed limit as possible, since higher velocities result in shorter merging times. The second

term refers to Remark 1, and acts as a regularization term, penalizing a fast change in the velocity command u_i with respect to the current velocity of the car. Lastly, p_s^i and p_v^i correspond to the priority values explained below and define a heterogeneous traffic model. We note that this new objective function consists of a weighted sum that yields a Pareto optimal point with respect to the parameter λ which can be chosen by the designer.

However, the optimization problem (2.5) is still practically intractable, since the set C contains constraints with respect to the order of the vehicles in their respective queues in the two lanes. Observe that the solution to the optimal control problem (2.5), also yields the optimal sequence q with respect to which the vehicles should merge. This makes it a mixed-integer optimization problem which is typically solved by solving (2.5) over all possible merging sequences q after an exhaustive search. This essentially decouples the two problems of generating a sequence and finding the optimal velocity commands u_i . However, the total number of possible sequences q grows exponentially with the number of vehicles in the control zone. As iterating through such a large number is impossible in real-time applications, in the next section, we propose a heuristic approach to search inside the sequence space. This approach can reduce the computational complexity by orders of magnitude, while the complexity-performance trade-off can be controlled by the designer with respect to the computational power of the system. We also transform the remaining optimization problem into a set of quadratic programming problems, one for each generated sequence, that can be solved in parallel and in real-time.

2.3 Handling Heterogeneous Vehicles

Most algorithms found in the literature treat all vehicles as equal in the merging process and use methods such as first in first out (FIFO) to decide the merging sequence. However, this approach is sub-optimal and performs very poorly in the presence of heterogeneous traffic. This is further complicated, since, different vehicles should be assigned different priorities based on multiple factors such as:

- Vehicle type, size, and mass
- Emergency vehicle
- On-ramp or mainline vehicle
- Current speed
- Vehicle's future intent

Some examples of this are as follows. A loaded heavy 16-wheeler truck on the highway should not be forced to slow down drastically to enable an on-ramp vehicle to merge. Vehicles should give way to emergency vehicles. A vehicle intending on taking the next off-ramp can be slowed to enable other vehicles to merge in front.

To handle these requirements, prior to the optimization stage which selects the optimal merging sequence, a priority assignment node updates all the priority values of the vehicles in the control zone. The algorithm considers two types of priority values for every vehicle:

1. Speed Prioritization (p_s^i)

2. Speed Variation Prioritization (p_v^i)

Speed priority is concerned with how much a vehicle desires to maximize its own speed. In mixed traffic, this is especially important for emergency vehicles. On the other hand, Speed Variation priority is based on the vehicle's deterrence to change its current velocity. While emergency vehicles would not mind sacrificing speed variation to gain higher speed the opposite is true for trucks since a high-speed variation would involve incurring a high fuel cost. The steps used in updating these priorities are as follows. Each vehicle class has a baseline speed priority and variation priority value. This value is then adjusted depending on the vehicle's current speed, whether it is on the mainline or not, and whether it has been waiting in the queue for a long time. This modification in priority depending on vehicle delay acts as a fairness factor to ensure that even low-priority vehicles will get a chance to merge eventually. Finally, an adjustment for the flow over effect caused by high-priority vehicles is made. This effect involves increasing the priority of all vehicles ahead of a high-priority vehicle in a single-line queue. This ensures that the high-priority vehicle will not be slowed down too much by low-priority vehicles ahead of it in the queue.

2.4 Priority-based Real-time capable method for cooperative merging of Heterogeneous Traffic

2.4.1 Outline and System structure

The optimal highway merging algorithm is formulated with many sub-components responsible for handling different aspects of the overall control process. Figure 2.2 describes the intercon-

nections and information flow among the algorithm sub-components. The algorithm is designed

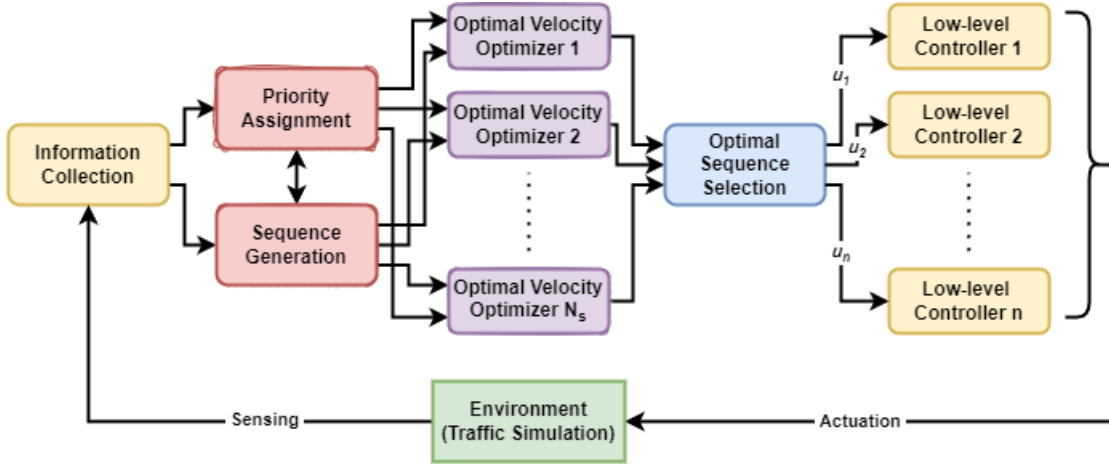


Figure 2.2: Optimal highway merging algorithm architecture

to operate in a cyclic manner with all tasks repeating at a specified control frequency. For a single time instance, the process starts by collecting state information from vehicles in the control zone. This information is then used for generating vehicle priorities and generating merging sequences. Each merging sequence $(q \in \{1, \dots, N_s\})$ generated is tested for feasibility and optimality and then the best sequence q^* is selected. The optimal command velocities corresponding to the best sequence are then provided to the low-level controllers onboard each CAV.

The two major tasks in the implementation of a fully automated highway merge system are: (a) the selection of the optimal merging sequence in which vehicles should merge, and (b) the implementation of this merging sequence by controlling each individual vehicle.

When a vehicle enters the control zone the centralized controller assigns to it a unique ID, that takes values between 1 and N , where N is the maximum number of vehicles that can fit inside the control zone.

The vehicles are then added to either the mainline queue or the on-ramp queue.

2.4.2 Optimal sequence generation

When deciding the best sequence in which to merge highway traffic the total number of possible sequences grows exponentially with the number of vehicles in the control zone. For example, in the practical case with $n = 30$ vehicles ($m = 15$ mainline and $r = 15$ ramp) in the control zone, the number of possible sequences is $\frac{(m+r)!}{m!r!} = 155117520$. As iterating through such a large number is impossible in a real-time application, this approach starts at an initial seed sequence based on time to merge and computes perturbations to this sequence depending on available computing power. Once we have a good selection of sequences to check, then our problem reduces to choosing which of these possible sequences provides the best performance. Therefore, our algorithm will analyze all possible vehicle inputs in order to make the optimal selection. In order to do this we need to calculate the optimal vehicle control input (acceleration/deceleration) needed by each vehicle to achieve each desired merge sequence. This calculation of inputs for all of the vehicles in the control zone for a specific merge sequence is formulated as a quadratic programming optimization problem.

In the process of generating the possible sequences which the optimizer needs to compare, we find that the number of possible highway merge sequences is limited due to two factors. The length of the control zone decides the maximum number of vehicles in the sequence and the requirement for order in individual lanes which means mainline vehicles can't overtake other mainline vehicles and the same is true for ramp vehicles. Using this fact, in this algorithm, a sequence is formed by injecting the on-ramp vehicles into the string of mainline vehicles. There are two extreme scenarios that may be considered when injecting ramp vehicles into the main line vehicle

queue. The first criterion is time to merge (TTM) (t_M) which takes into account the vehicle's speed and position. In this criterion, slow-moving vehicles may be allocated to the end of the sequence even though they are close to the merging point. The other criterion is distance to merge (DTM) (d_M). Here, vehicles moving fast will need to slow down in order to allow a slow-moving vehicle closer to the merging point to merge. We find that the optimum sequence lies in between these two extremes. Therefore, this algorithm follows a scalarization approach and performs a trade-off analysis between DTM and TTM controlled by the parameter α :

$$(1 - \alpha) * t_M^i + \alpha * d_M^i, \quad \forall i \in \{1, \dots, n\} \quad (2.6)$$

to compute the merge criteria from which the test sequences are generated. In order to generate a range of sequences, we vary α from 0 to 1 and the resolution is selected according to the desired quality of the sequences, available compute power, and traffic levels in the control zone. When the number of vehicles in the control zone increases the resolution of α can be made fine at the cost of computation.

A key feature of this algorithm is that the optimization process for each sequence is designed to run in parallel. This means that we can increase the α resolution depending on the available compute cores without sacrificing real-time performance. Every sequence thus generated is then fed into the optimizer in order to rank and select the best sequence and control variable values.

2.4.3 Optimization subsystem

The optimal control problems, one for each specific sequence under consideration q , seek to find the optimal command velocities of each vehicle u_i , which would facilitate the merging of vehicles according to the sequence q in an optimal way.

2.4.3.1 Objective function

The objective function to obtain u_i for a given sequence q , is formulated as:

$$J(u_i|q) = \sum_{i=1}^n p_s^i \lambda (u_i - \bar{v})^2 + p_v^i (1 - \lambda) (u_i - v_i)^2 \quad (2.7)$$

In addition to the interpretation given in Section 2.2.2, the objective function $J(u_i|q)$ can be viewed as a trade-off between minimizing the control effort (fuel consumption) and maximizing the throughput of vehicles through the merge zone. Throughput maximization is achieved through the minimization of the difference between the vehicle's command velocity (u_i) and the speed limit ($\bar{v}$). This stems from the fact that the throughput through the merging bottleneck is maximized when the speeds of all the participating vehicles are maximized. This formulation focused on mean velocity maximization is also based on the inherent nature of CAVs to maximize velocity in order to reach target destinations in the shortest possible time.

2.4.3.2 Constraints

The constraints (set C in (2.5)) imposed on this optimization problem ensure safety of operation during merging (Remark 1), by preventing collisions between vehicles, while providing controls that lie within the capabilities of each vehicle.

First, the command velocity $u_i(t)$, for all $i \in \{1, \dots, n\}$ and $0 \leq t \leq t_f^i$, should be bounded based on the speed limits:

$$0 \leq u_i(t) \leq \bar{v} \quad (2.8)$$

as well as based on the acceleration capabilities of each vehicle:

$$a_{min}^i \Delta t \leq u_i(t) - v_i(t) \leq a_{max}^i \Delta t \quad (2.9)$$

where Δt is set as the time resolution in our computations.

The next constraint refers to the dynamics of the system and considers the new expected position of vehicle i after time Δt :

$$s_i(t + 1) = s_i(t) - \Delta t \frac{v_i(t) + u_i(t)}{2} \quad (2.10)$$

The assumption that u_i is chosen such that the vehicle can reach it in time Δt is justified by (2.9) and the second term of the objective function (2.7).

The expected position parameter computed by (2.10) is used to ensure rear-end collisions do

not occur on the mainline and on-ramp respectively:

$$|s_j(t+1) - s_{j'}(t+1)| \geq l^j + M_{sr} \quad (2.11)$$

$$|s_k(t+1) - s_{k'}(t+1)| \geq l^k + M_{sr} \quad (2.12)$$

for all $j, j' \in \{1, \dots, m\}$ mainline vehicles with $j \neq j'$ and for all $k, k' \in \{1, \dots, r\}$ ramp vehicles with $k \neq k'$.

Here, M_{sr} represents the safety margin used to prevent rear-end collisions.

The equations (2.11) and (2.12) are then simplified by substituting the new expected position $s_i(t+1)$ values from equation (2.10). The norm is removed using the ordered precedence assumption which prevents vehicles from overtaking each other. We then obtain the form given in equations (2.13) and (2.14) for the prevention of all rear-end collisions on both the mainline and on-ramp:

$$u_j - u_{j+1} \geq (v_{j+1} - v_j) + \frac{2}{\Delta t}(s_j - s_{j+1} + l^j + M_{sr}) \quad (2.13)$$

$$u_k - u_{k+1} \geq (v_{k+1} - v_k) + \frac{2}{\Delta t}(s_k - s_{k+1} + l^k + M_{sr}) \quad (2.14)$$

for all $j, j+1$ vehicles belonging in the mainline and $k, k+1$ vehicles located on the ramp.

We then compute time to merge t_m^i for each vehicle i by:

$$t_m^i = \frac{s_i(t+1)}{u_i} \quad (2.15)$$

Then the constraint,

$$t_m^i \leq t_m^{i+1} - \frac{l^i}{u_i} - \frac{l^{i+1}}{u_{i+1}} - M_{sl} \quad (2.16)$$

for all consecutive vehicles $i, i + 1$ in sequence q , is responsible for ensuring that individual commands u_i are chosen such that vehicles merge in the order of the sequence under consideration q . Here, M_{sl} represents the safety margin used to prevent lateral collisions during merging.

By substituting from equations (2.10) and (2.15), we simplify equation (2.16) to (2.17). We also eliminate the l^{i+1} term since we restrict distance measurements to be always taken from a vehicle's front bumper. Therefore the following equation:

$$u_{i+1}(s_i - \frac{\Delta t}{2} \cdot v_i + l^i + M_{sl}) \leq u_i(s_{i+1} - \frac{\Delta t}{2} \cdot v_{i+1} - M_{sl}) \quad (2.17)$$

for all consecutive vehicles $i, i + 1$ in sequence q , plays the role of both preventing lateral collisions during merging as well as enforcing the order of merging to follow the tested sequence q .

The constraints given by equations (2.8),(2.9),(2.13),(2.14) and (2.17) along with the objective function in (2.7), form the optimization problem for optimal vehicle command velocity computation, given a test sequence q .

As a last comment, we note that the presence of the variable b^i in the set of constraints C in (2.5) causes the initial formulation to be a mixed-integer quadratic programming (MIQP) optimization problem, which would be difficult to solve in real-time for this large-scale problem. By incorporating the effect of b^i into the computed priority p^i and by setting up separate constraints for the mainline and ramp, as described above, we have reformulated the problem as a quadratic

programming (QP) optimization problem. This is a very important result that has enabled fast computation of the solution of each optimization problem in real time.

2.4.3.3 Optimal Sequence Selection

If a feasible solution is provided by the optimization above, the final cost value for the tested sequence q and the output u_i for each vehicle are stored. Once the optimization is carried out for each possible sequence q , the feasible sequences are sorted with respect to the minimum objective value achieved. Then the best sequence is compared to the sequence assigned in the previous computation iteration. Generally, the computed best sequence along with the suggested u_i command velocities are broadcast to all the vehicles. However, if the previous sequence was also recalculated in the current cycle and if the cost reduction in selecting the new sequence when compared to the previous sequence is less than a predetermined switching threshold then the previous sequence is transmitted to all the vehicles along with the new suggested u_i command velocities corresponding to the previous sequence. This ensures that the algorithm will not keep oscillating between two closely related sequences with comparable costs.

It is also important to note that this method allows the merging sequence to be changed at every update cycle, which is an essential feature for heterogeneous traffic. For example, when an emergency vehicle (EV) enters the control zone, the merging sequence should be modified to give precedence to the EV. This ability to switch rapidly and adapt to a new merge sequence is a core component of our approach.

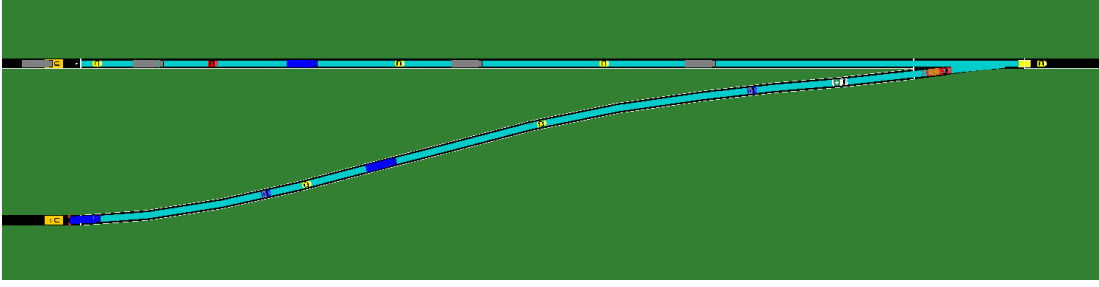
2.4.3.4 Low-level vehicle controller

The optimal sequence and suggested u_i command velocities are received by each of the vehicles in the control zone. In this formulation, we do not consider the delay of transmission and assume the data is available almost immediately via technologies such as 5G connectivity. The low-level vehicle controller (VC) in each vehicle will calculate the optimal control parameters (acceleration and deceleration) needed in order to achieve the desired u_i command velocity within the control time Δt duration.

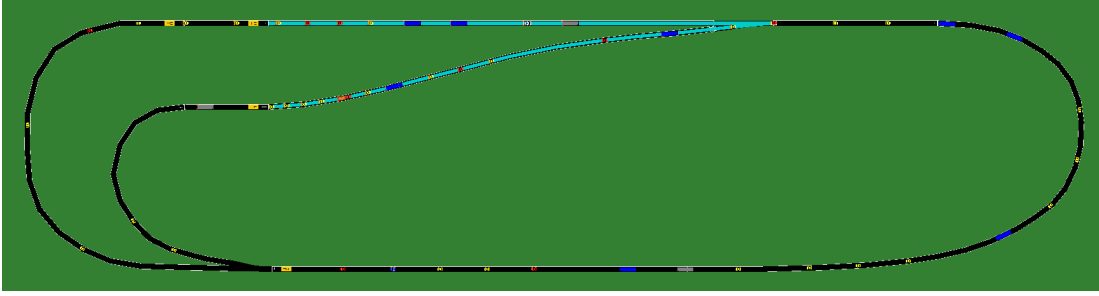
This low-level controller should be made robust enough to handle perturbations in both sensing and actuation. In practical scenarios, one cannot assume perfect information and actuation. There will always be certain failures in sensing such as receiving incorrect information broadcast from other vehicles or malfunctions in onboard sensing systems. Therefore, the VC will attempt to reach the target velocity as long as it is safe to do so. The multitude of sensors on board CAVs can be leveraged to check whether it is safe for the vehicle to reach its target velocity.

2.5 System performance and impact

We evaluate the performance of the proposed approach in a highway section simulation using the SUMO simulator [22]. The simulation setup for the merge junction is shown in figure 2.3a. The quadratic programming optimization problems (Section 2.4.3) are solved numerically using the Gurobi software (version 9.1.1) [23]. The controller communicates with the simulator using the TraCI traffic controller interface. All simulations and optimization algorithms run on a personal computer with an Intel i7-8750H CPU and 32GB of RAM. To simulate a continuous stream of



(a) Merge junction simulation.



(b) Highway loop simulation.

Figure 2.3: Experimental highway merge setup.

vehicles and ensure that on average the ratios of different vehicle classes remain constant, we implemented a highway loop network as shown in figure 2.3b. Probabilistic rerouting is used to decide what percentage of the overall traffic is sent through the ramp. As we consider only one mainline lane the rerouting parameter is set to $r_p = 0.5$ for all tests. Multiple tests are conducted for different loop densities l_d (number of vehicles in the cyclic highway simulation). The loop density takes values in $l_d \in [20, 80]$, while ensuring that all vehicle classes were suitably represented. All tests were performed for a 2-hour (7200s) time period, which reduces the variance of the values of the performance scores of each algorithm tested. All the parameters used in the tests conducted are given in Table 2.1.

Table 2.1: System parameters used in testing

Parameter	Value
Control zone length	300 m
Simulation duration	2 hrs
Time step (Δt)	100 ms
Maximum number of sequences	12
Speed limit ($\bar{v}$)	27 m/s
Mainline-Ramp rerouting ratio (r_p)	0.5
Objective trade-off (λ)	0.7
M_{sr}	2.0
M_{sl}	10.0

2.5.1 Performance metrics and comparisons

The key performance metrics used to evaluate merging algorithms are (a) *throughput* m_T (number of vehicles that can merge onto the highway per hour), (b) *density* m_D (number of vehicles on the link), and (c) *delay* m_L (Average delay experienced by vehicles compared to the ideal travel time). As density and throughput are related, we also consider their combined effect using (d) *mean velocity* m_V , which is obtained by $m_V = m_T/m_D$. Since we focus on mixed traffic special attention is paid to the delay faced by emergency vehicles. The total energy expended, (e) *fuel consumption* m_F , is also taken into consideration which shows the importance in the minimization of large fluctuations of the control commands. In this case, we also specifically compare fuel consumption in heavy vehicles, such as trucks, as they consume larger quantities of fuel during acceleration tasks.

In order to showcase the properties and appropriately assess the performance of our methodology, we compare it with three different approaches. First, we make a comparison with the baseline case of on-ramp merging without cooperation among vehicles. In the baseline case, ramp

vehicles always give way to mainline vehicles. Furthermore, we also compare against two commonly used collaborative merging strategies, which better showcases the strengths and potential weaknesses of our approach. The first of the two is a FIFO-based approach, in which each vehicle is assigned merging velocities in a greedy manner based on the order it enters the control zone. The second method utilizes a more advanced heuristic strategy known as zipper merging, in which the merge sequence is roughly chosen based on distance to merge with alternating selection between ramp and mainline. It is important to note that all these methods do not consider the heterogeneous nature of the traffic and all vehicles are treated equally. To our knowledge, as this is the first work to introduce a real-time end-to-end optimization-based merging algorithm for heterogeneous traffic, no direct comparison with other optimization-based algorithms found in the literature would be fair or informative. This is also compounded by the fact that other constrained optimization-based methods cannot compute a real-time solution for the large-scale simulation used in this comparison.

The main objectives in the highway merge problem involve maximizing the mean velocity (m_V) and throughput (m_T), while minimizing the density (m_D) on each lane. It is also necessary to consider the minimization of the overall fuel consumption (m_F) during merging and the delay (m_L) experienced by individual vehicles.

2.5.2 Main performance metric results

From the fundamental equation of traffic flow, we know that density m_D , the throughput m_T , and the mean velocity m_V , are linked according to equation (2.18).

$$m_T = m_V * m_D \quad (2.18)$$

Therefore, these three factors are considered in unison. In practice, the main goal of a highway merge control algorithm is to keep the traffic flowing as fast as possible with minimum density buildup on the lanes.

The values of these performance metrics for each of the methods compared as a function of the loop density of the vehicles are shown in figure 2.4. We observe that the methods that utilize cooperation between vehicles perform drastically better than the baseline approach, in which vehicles do not cooperate with each other. In figure 2.4b we observe that our method is capable of maintaining low levels of lane density when compared to the other methods. This property becomes more pronounced as the loop density on the highway section increases. While in figure 2.4a, zipper merge is found to have similar throughput levels to our method, the benefit of our method becomes evident when we consider the combined effect of throughput and density which is shown in figure 2.4c. We observe that the effective mean velocity achieved in our method is considerably higher than all the other methods, especially in high loop density scenarios. Our method achieves a 282% improvement in comparison to the baseline case for mean velocity of flow. When compared to zipper merge, an improvement of 81% is achieved at high loop densities. However, we see that

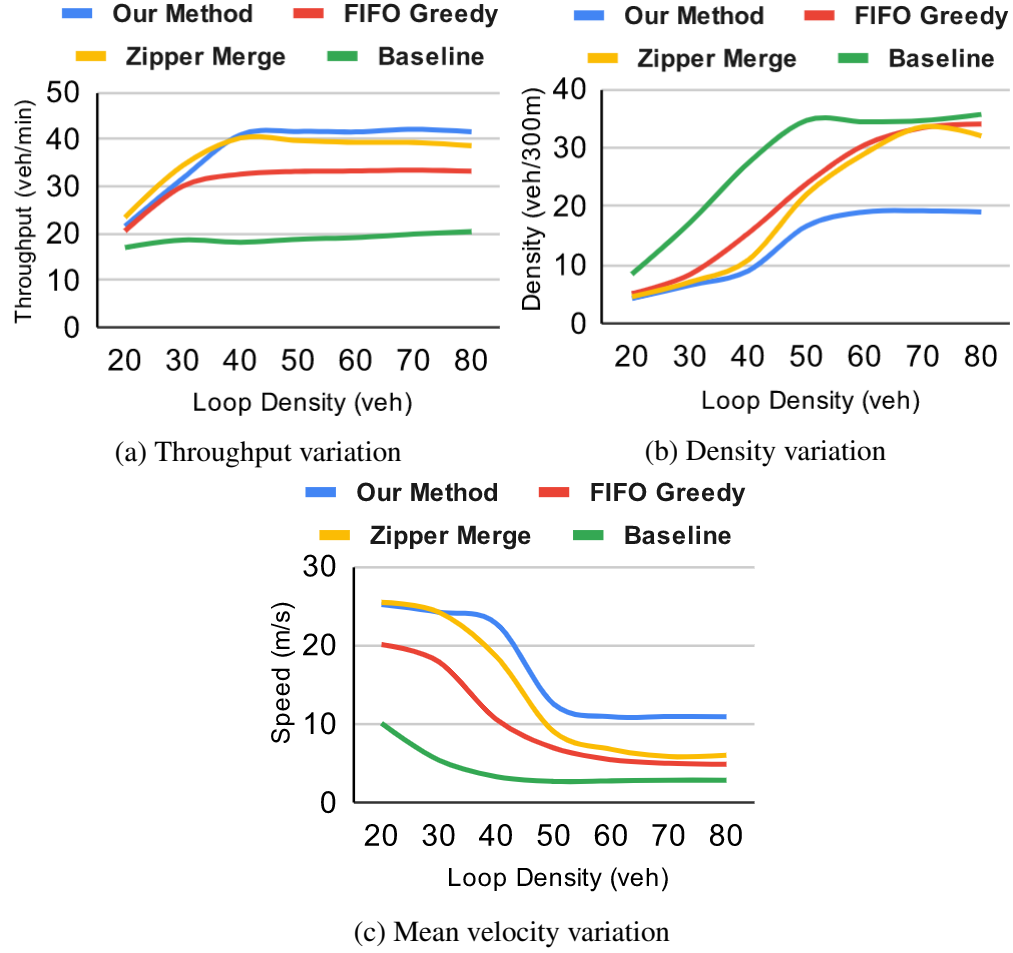


Figure 2.4: Comparison of performance for varying loop density

at low loop densities (few vehicles on the road) the zipper merge algorithm performs comparably to our method. This follows our intuition that when there are very few vehicles on the road we don't require a complex optimization-based algorithm and a simple heuristic-based algorithm performs adequately. However, as the traffic on the road increases, which leads to a bottleneck at merge junctions, the benefit of our optimization-based method becomes apparent.

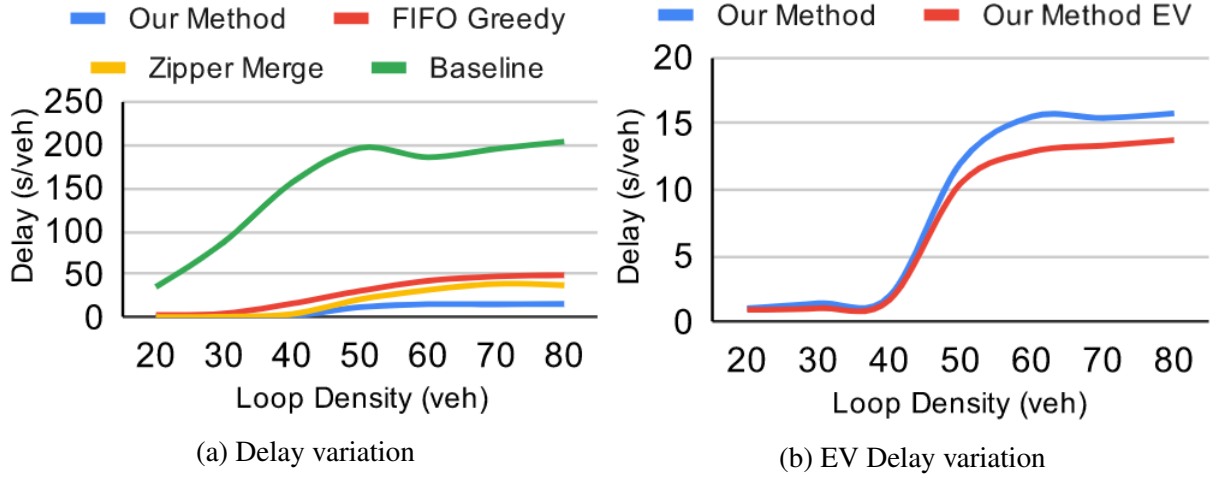


Figure 2.5: Comparison of delay faced with varying loop density

2.5.3 Delay metric results

The overall delay faced by the individual vehicles on the highway is also a key factor in merging control. The values for the delay metric are compared in figure 2.5, in which our method shows considerably better performance. In figure 2.5a which compares the average delay in seconds experienced by vehicles, we observe that our method achieves a 92% and 58% improvement when compared to the baseline case and zipper merge respectively.

Once again this improvement is more pronounced at high loop densities. When considering delay in heterogeneous traffic it is important to consider the impact of delay on different vehicle classes. This is especially true in the case of emergency vehicles (EVs). As the other algorithms tested consider all classes as equal, the delay faced by EVs is the same as the delay faced by other vehicle classes. In contrast, EVs in our approach face 17% less delays than other vehicle classes. For example, at a loop density of 80, the average delay faced by EVs is 13.7 *sec* in our method and 203.4 *sec* in the baseline case. Figure 2.5b depicts how EVs experience less delays than the other

vehicle classes in our method. It is important to note that the proposed method offers adjustable parameters which can be tuned to further reduce EV delay at the cost of reduced mean velocity for overall traffic flow.

2.5.4 Fuel consumption metric results

The effect of these algorithms on overall fuel efficiency is depicted in figure 2.6. Here, figures 2.6a and 2.6b show the fuel consumption per vehicle in all classes and the truck class respectively. As expected, all cooperation-based algorithms have much lower fuel consumption than the baseline case. Additionally, even among the cooperative methods, we see that at higher loop densities our proposed method yields the best fuel efficiency. While under the current parameters, the fuel savings percentages for trucks are similar to other classes, our method allows tuning the parameters to further reduce the truck class fuel usage, at the cost of reduced overall throughput.

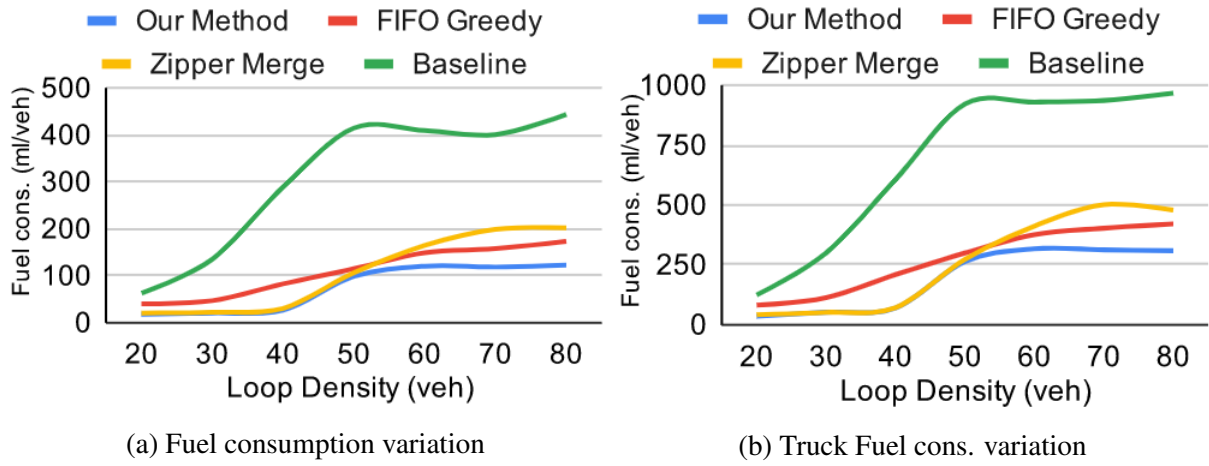


Figure 2.6: Comparison of fuel consumption with loop density

2.5.5 Real-time operation

The algorithm in our optimization-based approach operates at a frequency of $f_s = 10Hz$ (updates all control commands every 100ms). The time required for the optimizer to compute control outputs for a single sequence was around 25ms. Based on our computation power, in the tests carried out, up to 12 different sequences would be checked in each time step which, if computed sequentially would require roughly 300ms. However, as this algorithm was designed to run each sequence optimization in parallel, the optimization task for all sequences takes less than 50ms on a 12 core CPU. This showcases that the proposed approach is more than capable to work in real-time, as this is considerably less than the desired recalculation rate of 100ms. It is important to note that the optimality of the generated output control commands can be improved by searching over a larger sample of sequences. However, in order to achieve real-time operation this would require more computation resources. This means that if needed, by utilizing a processing unit with more computational resources (e.g. higher core count), the performance of this method can be further enhanced, while continuing to operate in real-time.

2.6 Further extensions to centralized highway merging control

We propose some key extensions to the original centralized merging control formulation in our previous work [24]. We focus on improving the overall throughput of the on-ramp merging process in multi-lane highways by utilizing the capacity of all the highway lanes. We extend the system to be capable of operating in mixed traffic conditions with the appropriate prediction of HDV behavior. We also update the merging sequence generation algorithm to be capable of

handling multiple criteria which have an effect on the output sequence.

2.6.1 Full capacity utilization lane changing controller

Our previous work in this area [24] focused on controlling only the outermost lane of the mainline highway. This is often quite inefficient since vehicles in the outermost lane will be forced to stay in the same lane until the merge junction is passed even if there is space in the inner lanes. If there is available capacity and empty space in the inner lanes of the highway mainline, then a separate controller can be introduced to get vehicles to switch lanes from outer lanes to inner lanes thus creating more space in the outermost lane and giving more opportunities for more ramp vehicles to merge successfully at higher speeds. This will then lead to increased overall throughput, increased average velocities, and reduced delays. To achieve this we introduce a mainline lane-change (LC) controller capable of detecting the density conditions on each lane, selecting vehicles that can perform safe maneuvers to inner lanes, and then providing commands to execute these lane-change maneuvers.

To perform full highway capacity utilization we extend the control zone to include all the other same-direction lanes of the highway. Therefore, the central controller receives information from all the vehicles in all the lanes of the mainline highway. Using this information the LC controller computes the density of vehicles on each of the highway lanes and decides if there is space in the inner lanes to accommodate for vehicles switching lanes from the outer lanes. If such a space is found, the LC controller then identifies vehicles in the outer lanes that can switch into the inner lanes. It then checks if these vehicles could perform a safe lane change maneuver into the

inner lanes. This safety check involves computing the leading and following vehicles in the target lane relative to the considered CAV. The parameters needed for this safety computation involve the time to change lane (T_{lc}), considered CAV's velocity (v_0), position (s_0), length (l^0), lead vehicles' velocity (v_l), position (s_l), length (l^l), and following vehicles' velocity (v_f) and position (s_f). We also define a displacement safety margin with respect to the lead vehicle (M_s^l) and the following vehicle (M_s^f). Note that all distances are computed from the front bumper of each vehicle. Given this information, we define the safety checks,

$$s_l - l^l - M_s^l + T_{lc}(v_l - v_0) \geq 0 \quad (2.19)$$

$$s_f - l^0 - M_s^f + T_{lc}(v_0 - v_f) \geq 0 \quad (2.20)$$

Here, equation (2.19) refers to the condition needed to be satisfied to safely change lanes without being affected by the lead vehicle in the target lane, and equation (2.20) refers to the condition needed to be satisfied to safely change lane without being affected by the following vehicle in the target lane. If both these safety checks are passed then the lane change controller will provide a lane change command to the considered CAV to change into the target inner lane. This lane change command will then be executed by the low-level lane changing controller onboard the CAV.

2.6.2 Changes to allow for mixed traffic

The process of including human driven vehicles (HDVs) in the traffic flow involves modeling HDVs, predicting HDV behavior, and adapting the CAV control algorithm to account for this

higher prediction variance uncontrollable agents (HDVs). Let the set of HDVs in the control zone be denoted by $\mathcal{V}^h$.

2.6.2.1 Modeling Human-Driven Vehicles

The modeling of human-driven vehicle behavior on highways requires a longitudinal car-following model and a lateral lane-changing model. The car-following model considers the interaction between the HDV and its lead vehicle. For this research we choose the Krauss model [25] for its accuracy and its ability to easily modify driving behavior parameters such as human reaction times and driving imperfections.

The model computes the safe following speed $v_i^s(t)$ by considering the impact of speed limits $\bar{v}$, vehicle acceleration capabilities a_{max}^i , the vehicle deceleration profile $b(v_i(t))$, distance gap $\Delta s_i(t)$ and speed $v_l(t)$ of the lead vehicle, time step Δt and driver reaction time τ_r as shown in equation (2.21).

$$v_i^s(t) = \min(\bar{v}, v_i(t) + a_{max}^i \Delta t, v_l(t) + \frac{\Delta s_i(t) - v_l(t) \tau_r}{\frac{v(t)}{b(v_i(t))} + \tau_r}) \quad (2.21)$$

The lane-changing model is responsible for appropriate lane selection and determining safe instances for performing lane changes. For this research, we use the lane-changing model developed by Erdmann [26].

2.6.2.2 HDV behavior prediction

When we consider an environment with mixed traffic, a key preceding step to computing the controls for the CAVs is the accurate prediction of HDV behavior. Since the behavior of HDVs cannot directly be controlled by the algorithm, the prediction model should be able to decide how each HDV would react to changes in the environment. Similar to our work in [27], we model the prediction of HDV behavior using a switched dynamical system to represent the reactions of HDVs to their surroundings. We model each HDV $i \in \mathcal{V}^h$ using the following state-dependent switched dynamics:

- **if** leading vehicle is present within a particular distance in front of HDV, the HDV follows the leading vehicle and maintains a safe following distance,
- **else** HDV travels at a fixed desired target speed which is a function of both the speed limit and the HDVs current velocity.

The choice of which constraint becomes active is governed by the current conditions faced by each HDV. Since the behavior of HDVs is governed by the lead vehicle ahead of each HDV, by controlling the CAVs which happen to be the lead vehicles of HDVs, we find that our system can indirectly control HDV behavior to a certain extent.

2.6.2.3 Constraints related to HDVs

The velocity constraints for HDVs are different from CAVs and are formulated in accordance with the switched dynamical system presented in 2.6.2.2. We compute the maximum possible

target velocity $u_i^m(t)$ in one time step which is related to the acceleration capabilities of the HDV $i \in \mathcal{V}^h$ as follows,

$$u_i^m(t) = v_i(t) + a_{max}^i \Delta t, \quad \forall i \in \mathcal{V}^h \quad (2.22)$$

If there exists a leading vehicle within range of HDV i , let this leading vehicle be referred to as i^l . The predicted target velocity $u_i(t)$ for all HDVs $i \in \mathcal{V}^h$ is set by either the speed limit $\bar{v}$, the target speed $u_{i^l}^l(t)$ of the leading vehicle or the maximum possible target velocity $u_i^m(t)$ as follows,

$$u_i(t) = \min\{\bar{v}, u_{i^l}^l(t), u_i^m(t)\}, \quad \forall i \in \mathcal{V}^h \quad (2.23)$$

Furthermore, the two sets of collision constraints introduced in section 2.4.3.2, for rear-end and lateral collisions are introduced for any pairing of CAVs with CAVs and HDVs with CAVs. However, constraints are not placed on pairings of HDVs with other HDVs since the system cannot control the outcome of this pairing. Note that in order to account for the increased variability in possible behavior of HDVs, for every constraint involving a CAV and HDV pairing the safety margins are increased. These increased margins add tolerances to the system so that the system can continue to be feasible and generate valid control velocities for CAVs even when the actual behavior of HDVs deviate from the predicted behavior. In simpler terms, CAVs give HDVs more space in order to account for the difficulty in accurately predicting a HDV's future behavior.

2.6.3 Improvements to merging sequence generation

The merging sequence generation node in our previous work [24], was only capable of using simple criteria for determining the vehicle sequence. We expand the capabilities of this system by expanding on both the base criteria used as well as the sorting algorithm which uses these criteria. For this purpose, we choose the base criteria as follows,

1. Time-to-merge (t_M^i)
2. Distance-to-merge (d_M^i)
3. Waiting time (w_M^i)
4. Vehicle priority (p_M^i)

Distance-to-merge considers only the position of the vehicles and therefore vehicles close to the merge zone will be allowed to merge first. Time-to-merge considers both the position and velocity of vehicles and allows fast-moving vehicles to merge before slow-moving vehicles which may be closer to the merge zone. Waiting time considers how long each vehicle has spent inside the control zone and allows vehicles that have been waiting longer to merge first. Finally, vehicle priority computed as discussed in 2.3, encapsulates the vehicle type requirements present in heterogeneous traffic. Here, note that time-to-merge and distance-to-merge are criteria that incorporate dynamics of the environment such as vehicle locations and speeds. The waiting time and vehicle priority criteria do not contain such dynamic information. As such sequences generated by considering only waiting time and vehicle priority are more likely to be infeasible when vehicle dynamics

are considered. As such we build a sequence generation algorithm that takes this into consideration. Therefore, our algorithm follows a scalarization approach and performs a trade-off analysis between t_M^i , d_M^i , w_M^i and p_M^i controlled by the parameters α, β and γ as follows:

$$\begin{aligned} & (1 - \beta) * [(1 - \alpha) * t_M^i + \alpha * d_M^i] \\ & + \beta * [(1 - \gamma) * p_M^i + \gamma * w_M^i], \quad \forall i \in \mathcal{V} \end{aligned} \tag{2.24}$$

Here, $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 0.5$ and $\gamma \in \{0, 1\}$. The convex combination of waiting time and vehicle priority does not add value and therefore γ takes only values of 0 or 1 in order to activate one criterion at a time. Furthermore, β is restricted to the range $[0, 0.5]$ to ensure that at least 50% of the sorting criteria is governed by time-to-merge and distance-to-merge which incorporate system dynamic constraints.

The combined criteria obtained from (2.24) is then used to sort the vehicles in the ramp and merging queues in order to generate the final merging sequence. In order to generate a range of sequences, we vary α from 0 to 1 and β from 0 to 0.5 and the resolution is selected according to the desired quantity of sequences, which is in turn governed by the available compute power and traffic levels in the control zone. When the number of vehicles in the control zone increases the resolution of α and β can be made fine at the cost of computation. Every sequence thus generated is then fed into the optimizer in order to rank and select the best-performing sequence and corresponding control variable values.

A key feature of our algorithm is that the optimization process for each generated sequence is designed to run in parallel. This means that we can increase the α and β resolution depending on the available compute cores without affecting our ability to operate with real-time computation

capability.

2.6.4 Results from improved highway merging control system

2.6.4.1 Simulation environment changes

To test the multiple new features added to the highway merging control algorithm such as full capacity utilization, mixed traffic capabilities, and improved merging sequence generation, a new simulation environment needed to be designed. We choose to move away from the loop design in [24], and create a single stretch of highway into which vehicles are injected with varying inflow rates. Vehicles are injected into the mainline and ramp lanes according to a Poisson arrival process with chosen mean. Each class of vehicle has its own mean arrival rate. The ratio of CAVs to HDVs can therefore be modified by changing the mean arrival rate of HDVs in contrast to CAVs. Changing the target mean value of this distribution also allows us to modify the amount of traffic injected into each lane. The exact simulation used involving a 3-lane highway and a single-lane ramp road is shown in figure 2.7.

2.6.4.2 Performance comparison

The updated highway merging cooperative control algorithm is tested against the key benchmark of zipper merging used in our previous work [24]. Note that the zipper merge algorithm used here in our simulation does not account for real-world conditions such as low visibility during merging and in fact, uses a long look-ahead distance to obtain the best performance. As such it represents an upper bound on what a realistic single agent highway on-ramp merging control algo-

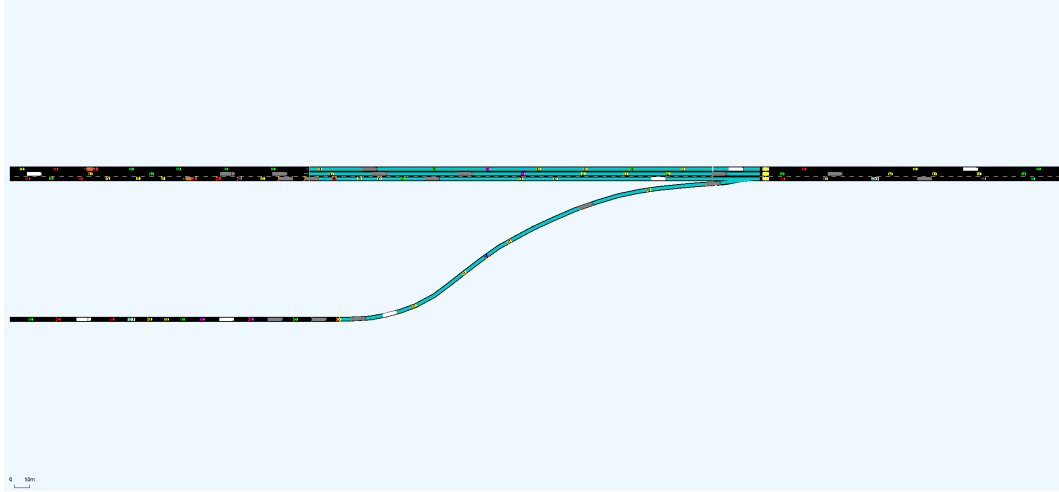


Figure 2.7: On-ramp merging into a multi-lane highway with CAVs and HDVs.

rithm could achieve. We also present our results with and without the highway capacity utilization lane changing controller active so as to observe the effect of adding this additional controller to the overall system. The performance metrics discussed in this section include,

- Throughput: Average number of vehicles passing through both the mainline and ramp per minute.
- Density: Average number of vehicles accumulating within the control zone.
- Delay: Average amount of delay faced by each vehicle in contrast to the ideal travel time.
- Fuel consumption: Average amount of fuel consumed by each vehicle in passing through the control zone.
- EV Delay: Average amount of delay faced by emergency vehicles in contrast to the ideal travel time.
- Truck Fuel consumption: Average amount of fuel consumed by trucks in passing through

the control zone.

Here, special attention is paid to emergency vehicle delays and truck fuel consumption as they represent the important edge cases that appear in optimization for heterogeneous traffic. A good heterogeneous traffic handling control system should be able to cater to the contradicting needs of emergency vehicles desiring minimum delays and trucks desiring minimum control effort.

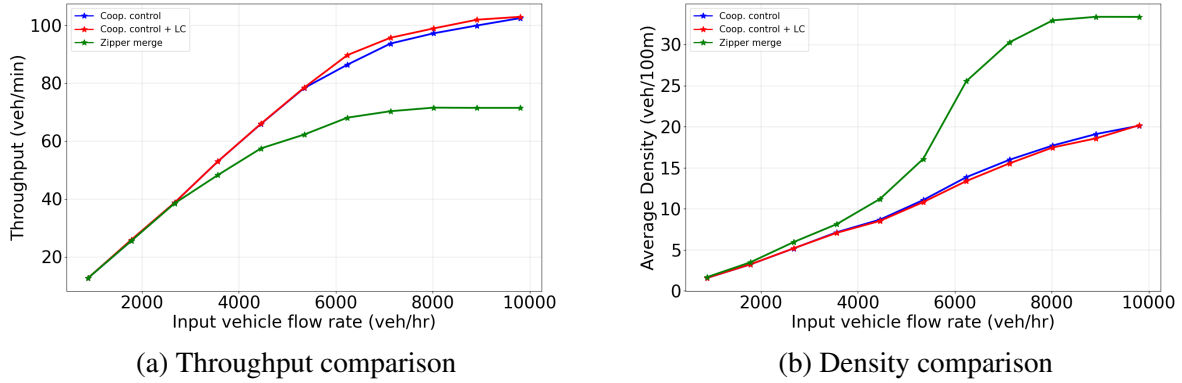


Figure 2.8: Performance in terms of throughput and density in comparison to baseline zipper merge, with increasing vehicle input flow rate. (CAV penetration level: 100%)

As shown in figure 2.8, we see that both our algorithm versions significantly outperform the zipper merge strategy in terms of throughput and density. The percentage improvement over the zipper merge strategy grows significantly higher with increasing input flow rate. This follows our intuition that at low traffic flow rates, even simple algorithms can perform well due to multiple opportunities available for safe merging. However, at high congestion conditions, the ability of multi-agent algorithms to coordinate each other's actions to cooperatively achieve efficient merging is very apparent. At high vehicle flow rates, we observe improvements to throughput of around 45% in comparison to zipper merge (figure 2.8a) and improvements to density of around 50% in comparison to zipper merge (figure 2.8b). A key feature to note is that the inclusion of the lane-

changing controller further improves the merging control algorithm's throughput performance by an additional 6%.

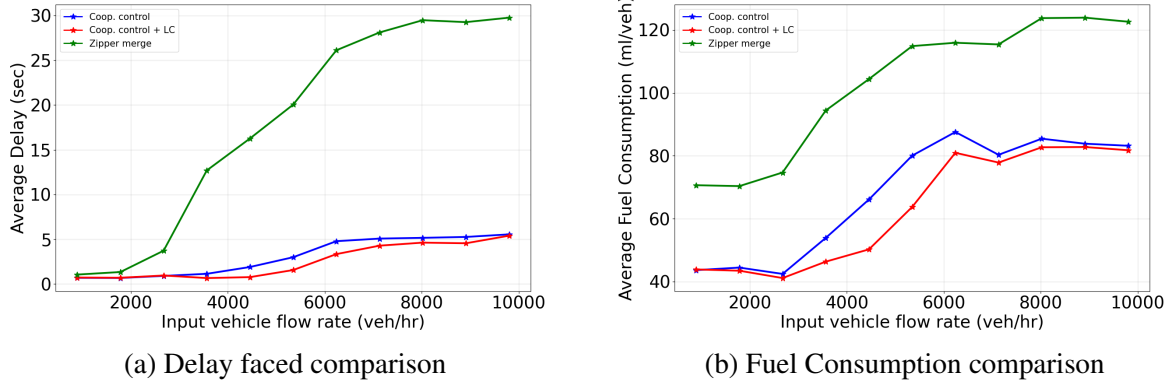


Figure 2.9: Performance in terms of delays faced by vehicles and fuel consumption in comparison to baseline zipper merge, with increasing vehicle input flow rate. (CAV penetration level: 100%)

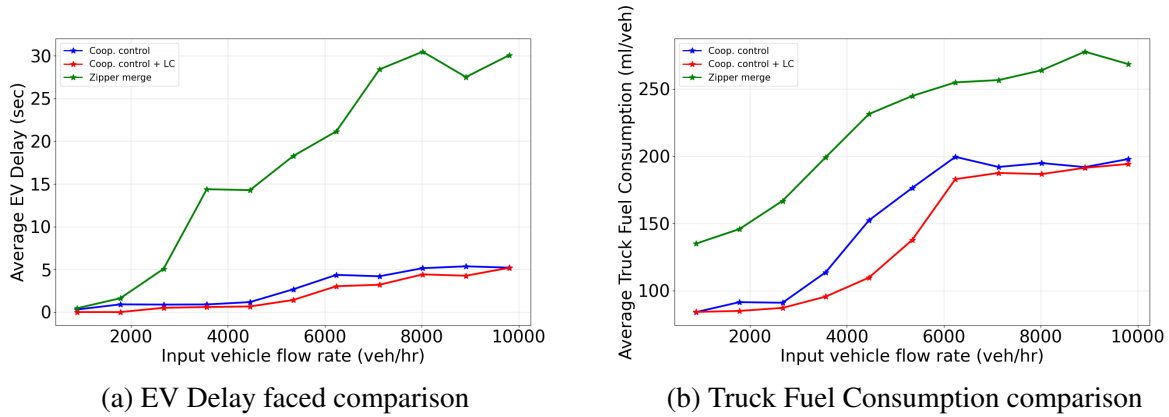


Figure 2.10: Performance in terms of emergency vehicle time-to-goal and truck fuel consumption in comparison to baselines, with increasing vehicle input flow rate. (CAV penetration level: 100%)

In figures 2.9 and 2.10 we compare our method's performance for the delay and fuel consumption metrics. Delay faced also showcases the trend of increased relative performance of our method at higher inflow rates. We observe improvements to delay faced of around 83% in comparison to zipper merge (figure 2.9a) and improvements to fuel consumption of around 33% in comparison to zipper merge (figure 2.9b). These improvements have a significant impact on overall

traffic efficiency by drastically reducing time wasted due to congestion and overall fuel consumption. Similar improvements are also seen in the special cases of EV delay (figure 2.10a) with an 85% reduction and in truck fuel consumption (figure 2.10b) with a 35% reduction when compared to the zipper merge method. We also observe that in both the delay faced and fuel consumption cases, the inclusion of the lane-changing controller for full highway utilization has a noticeable positive impact on performance.

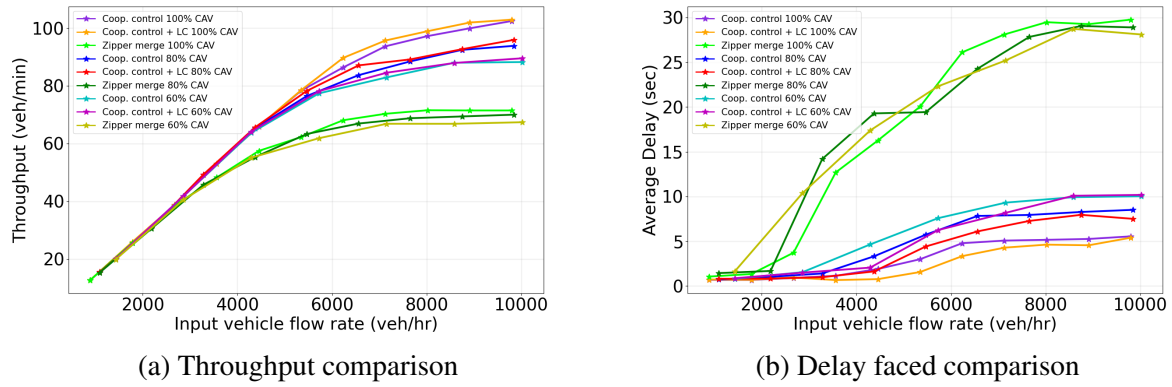


Figure 2.11: Performance in terms of throughput and density in comparison to baseline zipper merge, with increasing vehicle input flow rate. (100%, 80% and 60% CAV penetration levels.)

We also want to consider the influence of the CAV penetration level (percentage of CAVs in traffic flow) on overall system performance. As such we conduct experiments at CAV penetration levels of 100%, 80%, and 60%. These results are shown in figure 2.11 for both throughput and delay faced. It is clear that higher levels of CAV penetration lead to increased overall performance with an improvement of around 10% in throughput and 25% in delay faced when increasing CAV penetration from 60% to 80%. When comparing the performance at CAV penetration levels of 60% to 100%, we observe improvements of around 19% in throughput and 50% in delay faced. This can be attributed to the increased controllability of the traffic flow when the proportion of CAVs

to HDVs is higher. When there are more HDVs relative to CAVs (lower CAV penetration level), a lower proportion of the traffic flow can be controlled by the centralized merging control algorithm. Also since HDVs are given larger safety margins to account for their behavior unpredictability, the system cannot operate with the tighter tolerances it uses when most of the vehicles are CAVs. The combined effect of these two factors accounts for the performance drop we observe when the CAV penetration level is reduced. As the HDVs in the experiment use the zipper merge strategy, we expect the results to converge to the zipper merge outcome when the CAV penetration level is reduced even further.

Note that the new merging control optimization algorithm operates within the same computation constraints as the original algorithm and remains more than capable of real-time operation with a computation time below $100ms$ resulting in an attainable control frequency of $10Hz$.

2.7 Decentralized V2V Communication-based Method for Real-time Cooperative Highway Merging of Heterogeneous Autonomous Traffic

So far in this chapter, we saw that centralized methods for cooperative highway on-ramp merging offer drastic performance improvements when compared to methods without cooperation. However, these centralized methods require infrastructure upgrades and longer range communication capabilities with a central control tower. This is generally not possible for every single highway on-ramp. Having individual centralized controllers for each on-ramp or having reliable long-range communication to a distant centralized controller are often either financially or technically intractable. However, with the evolution of V2V communication technologies, we can

now make the assumption that vehicles within a relatively close distance of one another can communicate with near-zero latency. Thereby a V2V-based decentralized algorithm for cooperative merging could theoretically function in real-time while maintaining performance levels close to the centralized approach.

2.7.1 Dividing the problem into inter-related sub-problems

A primary concern in decentralized control of on-ramp merging with V2V communication is the decision regarding the number of vehicles that need to be considered in the merging strategy. There is a trade-off between the performance gain from considering more vehicles and the delay introduced by trying to communicate between vehicles located far apart. The advantages of considering a larger number of vehicles in a single group optimization problem are the improvements in performance, merging efficiency, and global optimality that can be achieved. However, the disadvantages of considering many vehicles include the communication concerns such as increased latency as well as the fact that solving the optimal merging control problem with a large number of vehicles in real-time is very computationally expensive. The computational complexity is found to increase exponentially with the number of vehicles considered and the onboard compute power available on the CAVs may become a limiting factor that governs the maximum number of vehicles that can be considered together. Therefore in solving the decentralized problem creating appropriate vehicle groups (figure [2.12](#)) play a major role.

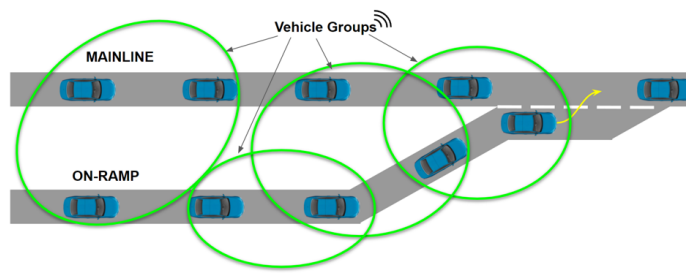


Figure 2.12: Vehicle group formation for centralized merging.

2.7.1.1 Grouping of vehicles

Multiple factors influence how vehicles should be grouped together for the V2V communication-based decentralized on-ramp merging algorithm. The first major limiting factor is communication latency and communication range. Since real-time operation is essential for this safety-critical application, we only consider single-hop communication between vehicles. While multi-hop communication is definitely possible, the latency it introduces would impact the real-time performance of this method. The communication range of individual vehicles may depend on the individual hardware present in each vehicle. Based on the IEEE 802.11p vehicular communication protocol we will define a standard communication range and assume that all vehicles are capable of communicating within this range. More details regarding this V2V communication model will be discussed later.

The other factor in vehicle groupings is the performance and computational complexity trade-off. If a larger number of vehicles are considered the resulting merging process will yield greater throughput and overall better merging performance. However, as the number of vehicles considered increases, the computational load on the hardware increases exponentially. Unlike in the centralized approach where computation was carried out in a grounded control tower, in the

decentralized approach this computation needs to take place onboard the CAVs. CAVs may not have access to very high-level computational hardware and also have limitations such as the limited power availability for running powerful computation systems. Due to these reasons, the decentralized algorithm needs to be considerably less computationally complex than the centralized version. The direct impact of this on the grouping of vehicles is that the maximum number of vehicles in a group will be limited by the computational cost of the algorithm.

Given these two criteria, the process of group formation would involve the following steps. Selecting a vehicle as the primary ego vehicle, in which the computation will take place. All other CAVs in which no computation is carried out are defined as secondary ego vehicles. Then search for vehicles in an increasing region centered around the primary ego vehicle. Vehicles are added to the group including the primary ego vehicle as long as either the maximum single-hop communication range or the maximum vehicles based on algorithm complexity have not been reached. Note that each group has only one primary ego vehicle. The number of primary ego vehicles (groups) that are needed to solve the problem as well as the management of cases in which vehicles may belong to multiple groups will also be discussed.

Process of group formation The outline of the steps involved in forming a vehicle group is as follows,

- Select a vehicle in the region prior to the merging zone as a primary ego vehicle. The criteria for the selection of this ego vehicle can include available compute power, distance from the merge zone, and separation from other possible primary ego vehicles.
- Search for vehicles in an expanding radius around the selected primary vehicle.

- Vehicles located are added to the group with the primary ego vehicle if they are within single hop communication range
- Stop searching for vehicles if the maximum number of permissible vehicles based on algorithm complexity is reached.
- Secondary vehicles in the group all transmit their location, velocity, and other metadata to the primary ego vehicle of the group.

2.7.1.2 Locally optimal strategies for vehicle groups

Once a vehicle group has been formed the primary ego vehicle obtains information related to all the other secondary vehicles in the group. The information collected from the secondary vehicles is as follows. Both the position $s_i(t)$ and velocity $v_i(t)$ at time t for secondary vehicle i are transmitted to the primary ego vehicle. In addition to $s_i(t)$ and $v_i(t)$, we assign a priority value $p^i \in \mathbb{R}_+^2$ to each vehicle i , which affects the merging decision, and an indicator variable $b^i \in \{0, 1\}$ which signifies whether the vehicle is on the on-ramp or the mainline. The priority value has two scalar components, speed prioritization (p_s^i) based on the focus placed on vehicle velocity and speed variation prioritization (p_v^i) based on the tolerance to changes in velocity. Furthermore, information is also collected on each vehicle's length l^i , and bounded acceleration capabilities characterized by its maximum acceleration a_{max} and maximum braking a_{min} capabilities, which carry information about lower-level variables such as the mass of the vehicle. Therefore, all

vehicles in the group are completely defined by the state vectors:

$$x_i(t) = [s_i(t), v_i(t), p^i, b^i, l^i, a_{max}^i, a_{min}^i]^T \quad (2.25)$$

for $i \in \{1, \dots, n\}$.

Once the state information of all the vehicles in the group is collected, the primary ego vehicle solves a constrained optimization problem which provides the merging sequence and the command velocities each member in the group should follow for optimal merging. The optimization formulation will be discussed in the following section. Once the command velocities are generated, they are transmitted using V2V communication to all the vehicles in the group. The local low-level controller on-board each vehicle and then actuates the vehicle in order for it to achieve this target command velocity.

2.7.1.3 Handling the cases involving overlapping subgroups

A major concern in this decentralized approach is the possibility of secondary ego vehicles being assigned to multiple groups as shown in figure 2.12. This means that the primary ego vehicles which compute the merging sequence and command velocities for each group may send conflicting information to the secondary vehicles which are a member of both groups. There are multiple possible methods for solving this issue and they all need to be explored further. One possible method involves dynamic member reassignment so that each vehicle only belongs to one group. So an assignment phase allocating vehicles to groups will be carried out prior to the optimization phase. The assignments need to be dynamically recomputed as vehicle leave or enter the merging

region. This process would involve the sharing of group membership information among vehicles. Another approach would involve vehicles forming an overall consensus on the merging sequence based on the proposed merging sequence provided by each group's primary ego vehicle. Conflict resolution and in certain cases re-computation will be needed to be addressed in this process. The delay introduced by this process will also need to be accounted for. This would also involve individual vehicles sharing their assigned merging sequences with other vehicles.

There is also the possibility of vehicles not belonging to any group. Since there are many ways to solve this issue the evaluation and selection of a suitable method needs to be carried out. One such method may involve a more involved process of primary ego vehicle selection which would ensure that the resulting groups formed encapsulate all the vehicles in the control region. However, this process would require higher communication overhead when compared to a simple arbitrary primary ego vehicle allocation process. This trade-off needs to be analyzed in detail. Another solution may be to continue the arbitrary leader allocation process until every vehicle belongs to at least one group. A secondary elimination round may also be carried out after this to find and eliminate redundant vehicle groups.

These issues and their possible solutions also highlight the need for deciding the number of groups needed to obtain the best solution to this problem. Selecting the minimum number of groups needed would lead to reduced computation overhead and increased algorithm speed. However, it is important to ensure that a single group does not grow too large, as this may result in a computation bottleneck forming in the primary ego vehicle of the large group.

2.7.2 Optimization problem for a single vehicle group

In this section we describe the optimization subsystem tasked with computing the merging sequence and control velocities for the vehicles in one subgroup. This computation is carried out in the primary ego vehicle of the group. Unlike in the centralized case, the number of vehicles involved in the optimization problem is far less. This means that either the merging sequence can be incorporated into the optimization problem or every possible sequence can be checked by brute force. An exploration of both these methods needs to be carried out and the method yielding the better performance-to-computation ratio will be selected. The effective method used per vehicle group will be identical to the formulation in the centralized method described in section [2.4.3](#).

2.7.3 Importance of V2V Communication

2.7.3.1 Impact of physical highway structure on communication

In the case of decentralized V2V communication-based highway merging it is essential to model the impact of the physical highway structure since the Cartesian distance is used in computing the communication range. Depending on physical parameters such as highway length, shape, and features such as obstacles the effective communication range for V2V communication is variable. For this work, we focus on the impact of the road structure or shape on communication. The performance of this algorithm is severely affected by how the ramp and mainline are positioned relative to each other. This is due to the fact that vehicles in the control zone at similar distances from the merge zone may in fact be either close to each other or far apart depending on the Euclidean

distance between them. For example, if the on-ramp and mainline run parallel to each other prior to merging then the Euclidean distance between the on-ramp and mainline vehicles will be small. However, if the on-ramp approaches the mainline in a perpendicular fashion then these distances will be large. The impact of this is that in the close to perpendicular case, by the time ramp vehicles enter the communication range of the mainline vehicles, there is limited time and distance for the vehicles to adjust their velocities to merge in an optimal way. The options available for merge sequence allocation and optimizations to handle heterogeneous vehicles drastically reduce in this case.

2.7.3.2 Achieving an inter-group consensus

V2V communication also plays an important role in the vehicle group and merging sequence coordination task. In these phases multiple back-and-forth communication steps may be required between the primary ego vehicle and secondary ego vehicles. The main tasks that require continuous communication are as follows,

- Appointing group leaders as primary ego vehicles
- Adding secondary vehicles into groups
- Removing redundant groups
- Handling conflicting information for members in multiple groups
- Reassigning groups as members leave or join the control region

The effective implementation of these subsystems of the algorithm depends heavily on a robust V2V communication system. These subsystems play a major role in managing conflicts and ensuring that smooth collision-free merging performance is obtained.

2.8 Future work

The work described in this chapter, and partially published in [24], has many possible extensions and improvements. This section focuses on the future directions of highway merging control systems.

2.8.1 Comparison against full-scale optimization formulation

Since the method presented in this chapter relates to a hybrid rule-based and optimization-based control for highway merging, in order to truly understand the impact of this method we need to implement a fully optimization-based method. This optimization-only method will be able to converge to the actual local optimal control for each of the vehicles in the control zone. However, as expected this method cannot match the original method in terms of speed and computation efficiency. In a domain in which real-time computation is critical, this algorithm thus becomes unusable. In this algorithm's defense, as the computational power of control systems improves there may come a day in the future when this large-scale optimization problem can be solved in real-time. Therefore, it is important to calculate and show the computation gap between the hybrid method and the purely optimization-based method. In this section, we will explore the most efficient formulation of the large-scale optimization problem, implement this in simulation and

compare the computational complexity of the two methods.

2.8.2 Macroscopic effects of the proposed merging method

Another interesting application of this centralized merging control method is its ability to act as a ramp metering method. This proposed method allows dynamic changes to the ratio of mainline vehicles to ramp vehicles that are allowed to merge. In the priority allocation system, we can adjust the priority assigned to both mainline and ramp vehicles thus modifying the ratios in which they merge. An interesting future research is the incorporation of this merging control algorithm as the ramp metering control algorithm in a large-scale traffic network's macro-level flow optimization system.

The goal of these large-scale network-level optimization systems is to find how to optimally conduct traffic in the network so that from a global perspective overall network throughput is maximized while minimizing the overall delay faced by vehicles. In order to achieve this the controls applied focus on restricting the rate of flow of vehicles on either the mainline or on-ramps. They compute the best merging rate to be applied at any time instance to each merge junction in order to satisfy the global optimization objectives. This merging rate is then implemented using various strategies such as ramp metering and variable speed limits.

Most modern ramp metering strategies found in the literature such as ALINEA [28] require traffic light-based on-off control at highway on-ramps. As mainline traffic lights are often not practical, some other methods introduce variable speed limits [29] for controlling vehicles on the mainline. Overall, these methods severely reduce the applicability of these methods due to the

infrastructure improvements needed and issues with the practical implementation of traffic lights at highway on-ramp junctions.

With the advent of CAVs, algorithms such as our proposed method are able to directly influence the merging ratio between ramp vehicles and mainline vehicles, without the need for traffic light or variable speed limit infrastructure. The next step in this research would involve combining our automated merging method with a large-scale network optimization system. Here, our method will be used at every on-ramp merge junction in the network. We would then compare our result with the existing methods such as ramp metering and variable speed limits. As our method directly controls vehicle velocities and also does not require the stop-go nature of traffic lights, intuitively we expect a significant performance improvement over existing methods.

2.8.3 Reinforcement learning extension

In the previous section, the optimal control velocities and merging sequence for a V2V communications-based group of vehicles were computed by an optimization formulation. This section aims to formulate the computation of the vehicle velocities and merge sequence as a deep reinforcement learning problem. The motivation behind this approach is the ability of RL algorithms to capture information and relationships among vehicles that are not considered in the optimization formulation.

Chapter 3: Optimal Real-time Multi-Agent Planning and Control for Dynamic Intersections

3.1 Introduction

Effectively navigating unsignalized intersections often requires careful planning due to low visibility and complex maneuvers such as unprotected left turns. 40% of all crashes, 50% of serious collisions, and 20% of fatalities occur at unsignalized intersections [30]. The notion of introducing vehicle-to-infrastructure (V2I) communication capable connected autonomous vehicles (CAVs), to provide additional sensing and actuation points in the traffic flow, allows for the development of new algorithms to handle these scenarios. This has been applied successfully in complex traffic bottlenecks such as highway merging and traffic shock waves [14, 24, 31]. Recent research into leveraging cooperation among CAVs to achieve safe, fair, and efficient intersection control has provided promising results with solutions from many fields, including game theory, auctions, optimization, and deep learning. While each domain has its own advantages, no single solution has been shown to be able to achieve all four key considerations of safety, efficiency, fairness, and real-time operation. Here, safety involves the prevention of collisions, efficiency involves maximizing the capacity of the intersection, and fairness involves treating all agents equitably. Real-time com-

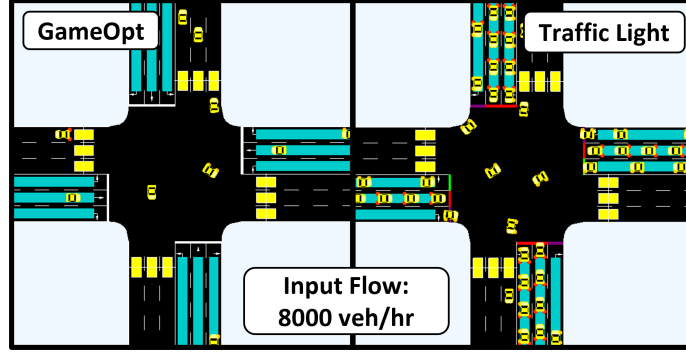


Figure 3.1: **GAMEOPT vs. Traffic Lights:** We present a new approach for optimal real-time planning and control in dynamic, multi-agent unsignalized intersections. In this figure, the light blue regions denote the control zone. We show that at identical input traffic flow levels, our approach outperforms even traffic light infrastructure, resulting in less queues and enabling smooth traffic flow.

putation is also essential in this scenario, as the system must react to dynamic changes in this fast-paced environment.

The general multi-agent intersection control problem can be broadly divided into two phases. The planning phase involves the selection of the optimal entrance sequence (order in which vehicles enter the intersection) and the control phase involves generating safe trajectories for all the vehicles in order to achieve the selected sequence. The system then transmits the output target trajectories to each vehicle’s local controller for execution. Here, sequence selection can become very difficult as the number of possibilities grows exponentially with the number of vehicles. This is further complicated in multi-lane scenarios, which should allow multiple non-conflicting vehicles to enter the intersection at the same time to achieve optimal throughput. Additionally, practical intersection scenarios are *dynamic*, which means that the number of vehicles that request to cross the intersection changes with time.

3.1.1 Related Work

In Table 3.1, we compare our approach with the current state-of-the-art in navigating unsignalized intersection scenarios on the basis of optimality guarantees, dynamic intersection handling, and real-world applicability through real-time computation capabilities.

3.1.1.1 Auctions

Auction mechanisms have been used to navigate unsignalized intersections in real-time with fairness and feasibility guarantees [32, 33]. But they work only in *static* scenarios, where the number of participating agents is fixed, yielding poor efficiency and throughput in scenarios where the number of vehicles is variable. Incentive-compatible auctions such as Sayin et al. [34] propose a mechanism in which agents are assigned turns based on their distance from the intersection and the number of passengers in the vehicle. Carlino et al. [33] and Rey et al. [35] propose a similar mechanism but use a monetary-based bidding strategy. Buckman et al. [36] integrate a driver behavior model [37] within the first-in first-out framework to incorporate human social preference. More recently, Chandra et al. [32] proposed an auction based on the driving behavior of the agents.

3.1.1.2 Game theory

Game-theoretic approaches have also been used for autonomous vehicle navigation [38, 39]. Li et al. [38] implement a Stackelberg game in which one agent is the leader (modeled by first-in first-out (FIFO) principle) and the other agent is a follower. Tian et al. [39], on the other hand, use a recursive k -level approach in which strategies at current levels are derived from previous levels.

Approach	Intersection	Optimality			Realtime
		Safety	Efficiency	Fairness	
Auctions	Static	✗	✗	✓	✓
Game Theory	Dynamic	✗	✗	✓	✗
Optimization	Dynamic	✓	✓	✗	✗
Deep Learning	Static	✗	✓	✗	✓
This work	Dynamic	✓	✓	✓	✓

Table 3.1: **Comparison with prior work:** We highlight approaches for navigating unsignalized intersections based on the type of intersection, optimality guarantees, and real-time capability.

However, all agents except the ego agent at the first level are assumed as static.

3.1.1.3 Optimization-based methods

These methods compute optimal trajectories and provide safety guarantees, however, they are unable to perform in real time due to high computational costs. Existing methods circumvent this issue by using unrealistic assumptions. Rios-Torres *et al.* [14, 40] solve the optimization problem in real-time using Hamiltonian analysis, which does not extend to the fully constrained problem. Computation is also restricted to fewer vehicles in single-lane roads, and no turning at intersections, to allow for close to real-time computing [41–43]. In Suriyarachchi *et al.* [24], we applied a hybrid rule and optimization-based control to the highway merging scenario to achieve real-time performance. However, the selection of sequences based on rules only provides a sub-optimal solution. Most of these methods do not scale well into multi-lane scenarios.

3.1.1.4 Deep learning

Deep learning methods involving reinforcement learning [44] and recurrent neural networks [45] learn a planning and control policy, to be used by agents approaching an intersection. In practice, these policies do not generalize well to different environments and often do not provide guarantees in terms of safety and fairness. Furthermore, most learning methods do not easily extend to dynamic multi-agent planning and control scenarios.

3.1.2 Main Contributions:

In the field of cooperative unsignalized multi-lane intersection control, existing work is unable to combine optimality guarantees in safety, efficiency, and fairness, with real-time performance. In order to bridge this gap, we develop a novel hybrid planning and control algorithm for navigating unsignalized dynamic intersections by combining auctions with optimal control. This is the first such framework, with implications that extend beyond intersection control to more general multi-agent systems, due to the challenging nature of dynamic mechanism design. Our main contributions are:

- *Game-theoretic optimality:* The priority order sequence is fair, efficient, and tractable (Section 3.3.1).
- *Safety guarantees:* The optimal trajectories satisfy the priority order with safety guarantees (Section 2.4.3).
- *Dynamic flow:* GAMEOPT handles varying traffic flow in different arms of the intersection

(Figure 3.6).

- *Real-time computation:* GAMEOPT operates on average at 1.16ms with more than 10000 vehicles per hour.
- *Multi-lane capability:* Increased performance by allowing many vehicles to enter the intersection simultaneously.
- *Efficiency:* We outperform state-of-the-art by improving throughput, time-to-goal, and fuel consumption on a realistic traffic simulator (Figure 3.5).

3.2 Problem Formulation

In this section, we formally define the fully automated unsignalized intersection planning and control problem as a large optimal control problem consisting of a hierarchy of two simpler decoupled optimization problems: mechanism design followed by vehicle trajectory planning. In a dynamic scenario consisting of n vehicles, our goal is to compute velocities v_i for each vehicle such that they autonomously navigate an unsignalized multi-lane intersection safely, fairly, and efficiently. We observe the following goals/constraints:

1. *Fairness:* The computed velocities, v_i , must maximize utility across all agents and be incentive compatible.
2. *Safety:* The trajectories generated by the optimizer must not result in collisions.
3. *Efficiency:* The overall algorithm must optimize throughput, time-to-goal, and fuel efficiency.

4. The overall planning and control must not assume access to the objective or utility functions of other agents.
5. The overall planning and control must not assume a dynamics model; rather, it should compute the dynamics for each agent assuming a simple forward motion model.

The first step in this process involves defining the physical characteristics of an unsignalized intersection as well as the dynamics and control of the CAVs involved.

3.2.1 Modeling the Physical Intersection

We define an unsignalized intersection as a single-lane or multi-lane four-way crossing that is not regulated either by traffic signals or right-of-way rules. In Fig. 3.2, we show an example of a single-lane intersection used in our work. We define a control zone of length L_c along each arm of the intersection; vehicles in the control zone can share state information and receive actuation commands using V2I communication protocols. In this research we assume zero transmission delay and that this V2I communication channel has perfect conditions. The center of the intersection is referred to as the conflict zone, where collisions are likely to occur. We denote the number of vehicles in the control zone in roads 0, 1, 2, 3 as n_1, n_2, n_3, n_4 , respectively, with the total number of vehicles in the control zone being $n = n_1 + n_2 + n_3 + n_4$.

Vehicles are autonomous and can freely choose between turning left, turning right, and going straight. The input flow rate of vehicles in each control zone of the intersection can be adjusted along with the ratio of vehicles that turn left, turn right, or go straight. We vary the input flow rate between 2,000 vehicles per hour to 10,000 vehicles per hour, where vehicles are generated

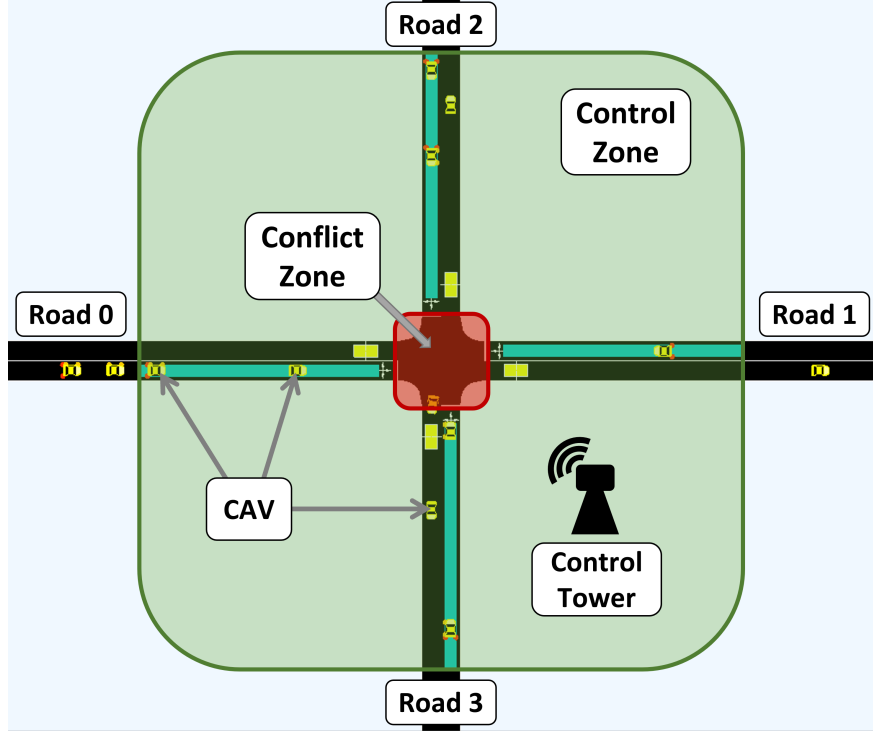


Figure 3.2: **Regions of interest:** We highlight the control zone in which vehicles communicate with the control tower and the conflict zone in which vehicles cross over to their desired target road.

according to a Poisson distribution.

3.2.2 Vehicle Dynamics and Control Variables

In real-world scenarios, vehicle dynamics are hard to model accurately due to their non-linearity. Standard practice in control theory suggests the use of a local controller, z_i , that produces low-level controls for the longitudinal and lateral motion of a vehicle according to the following equation,

$$\frac{\delta s_i}{\delta t} = \mathcal{F}(t, s_i, z_i). \quad (3.1)$$

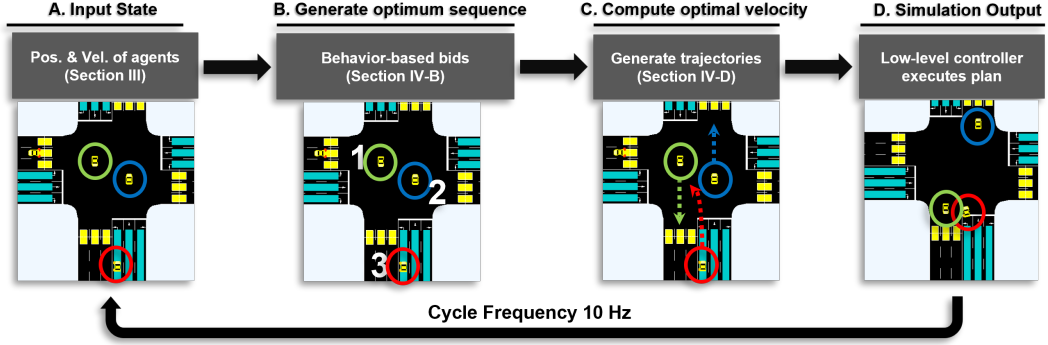


Figure 3.3: **GAMEOPT overview:** Our approach begins by reading the positions and velocities of all agents in the control zone (blue). Our approach is hybrid; in the planning phase, GAMEOPT collects the bids from every agent and generates an optimal priority sequence (Section 3.3.1). In the controls phase, we use an optimization-based trajectory planner to compute the optimal velocity for each agent that satisfies the priority order while simultaneously guaranteeing safety and real-time performance (Section 2.4.3).

Here, we represent $\mathcal{F}$ as the unknown non-linear dynamic function. s_i denotes the distance of the agent i from the intersection measured in the Frenet coordinate system.

Dynamics: Since $\mathcal{F}$ is generally unknown, we define the high-level dynamics by the following velocity control scheme:

$$\begin{aligned} \frac{\delta s_i}{\delta t} &= v_i \\ v_i(t) &= \mathbf{u}_i(t) \end{aligned} \tag{3.2}$$

for $i \in \{1, \dots, n\}$.

Control variables: The control input, $\mathbf{u}_i(t)$, represents the command velocity for vehicle i . We compute the optimal value for $\mathbf{u}_i(t)$ with respect to safety, efficiency, and reachability guarantees described in the following sections.

State vector: In addition to $s_i(t)$ and $v_i(t)$, we assign an indicator variable $l_c^i(t) \in \{0, 1, \dots, L_n - 1\}$, which signifies which lane the vehicle is currently in, with L_n representing the number of lanes

on each road. We also assign a bidding value $b_i(t)$ to each vehicle according to 3.3.1, which allows for the computation of the sequence order in which vehicles would cross the intersection. We also obtain the length l_i , and maximum acceleration a_i^{max} and deceleration a_i^{min} capabilities of each CAV i . The CAVs are thereby completely defined by the state vectors:

$$\Phi_i(t) = [s_i(t), v_i(t), l_c^i(t), b_i(t), l_i, a_i^{max}, a_i^{min}]^\top \quad (3.3)$$

for $i \in \{1, \dots, n\}$. We also define the speed limit, $\bar{v}$, and a parameter tuple $M_s = \{M_{sr}, M_{sl}\}$, which represents the safety margins needed to prevent rear-end and lateral collisions.

3.2.3 Dynamic Traffic Intersection Planning & Control Problem

The dynamic traffic intersection problem involves computing suitable, collision-free, continuous trajectories for all vehicles approaching an intersection to achieve efficient and fair traffic flow. This problem can be formulated as an optimal control problem, with the objective of computing the optimal command velocities u_i for each CAV $i \in \{1, \dots, n\}$, resulting in the minimization of the maximum time taken by each CAV to cross the intersection:

$$\begin{aligned} \min_{\{u_i\}} \quad & \max_i t_f^i \\ \text{s.t.} \quad & \mathcal{C}(\{\Phi_i\}, \bar{v}, \mathbb{S}(M_s)) \end{aligned} \quad (3.4)$$

where t_f^i is the intersection crossing time of vehicle i . $\mathcal{C}$ is a set of constraints, discussed in Section 2.4.3, that depend on the state vectors $\{\Phi_i\}$ of the CAVs, and safety requirements $\mathbb{S}(M_s)$ which prevent collisions between vehicles.

Equation 3.4 is, however, intractable to optimize directly because it implicitly assumes that agents are navigating the intersection according to the optimal priority order, q^* . Equation 3.4 can be equivalently solved by optimizing jointly over sequences, q , and the control velocities, $\mathbf{u}_i$, needed to implement the sequence.

$$\begin{aligned} \min_{\{\mathbf{u}_i, q\}} \quad & \sum_{i=1}^n \lambda(\mathbf{u}_i - \bar{v})^2 + (1 - \lambda)(\mathbf{u}_i - v_i)^2 \\ \text{s.t.} \quad & \mathcal{C}(\{\Phi_i\}, \bar{v}, \mathbb{S}(M_s), q) \end{aligned} \tag{3.5}$$

We solve this optimization problem in Section 2.4.3.

Intractability of Equation (3.5): The optimal control velocities, $\mathbf{u}_i$, depend on the sequence order q of the vehicles entering the conflict zone. The naive solution is to solve a mixed-integer optimization problem for every possible entrance sequence q by performing an exhaustive search. This process involves two sub-problems—generating a sequence q followed by computing the optimal velocity commands $\mathbf{u}_i$ for that sequence. The total number of possible sequences q grows exponentially with the number of vehicles in the control zone which would take the form $\mathcal{O}\left(\frac{(n_1+n_2+n_3+n_4)!}{n_1!n_2!n_3!n_4!}\right)$, where $n=n_1+n_2+n_3+n_4$ is the number of vehicles in the control zone. Therefore, current methods to solve Equation 3.5 are intractable.

We propose the use of auctions to obtain an optimal priority sequence order q^* in $\mathcal{O}(n \log n)$ time, significantly reducing the combinatorial runtime complexity of the mixed-integer formulation. We run the optimization over $\mathbf{u}_i$ for the selected optimal sequence q^* which can be solved in real-time.

3.2.4 Auction Theory

Auctions are a central theme in mechanism design. An auction allocates K items among n agents. Each agent i has a private valuation ζ_i and submits a bid $\mathbf{b}_i$ to receive at most one item of value α_i . In any auction, we have $\mathbf{b}_1 > \mathbf{b}_2 > \dots > \mathbf{b}_K$ and $\alpha_1 > \alpha_2 > \dots > \alpha_K$. Each auction has an allocation rule (who gets what), a payment rule (who pays what), and a utility function (how much value an agent gets based on their bid). In a sponsored search auction, which is the model we are using in this work, the allocation rule is that the agent with the j^{th} highest bid is allocated the j^{th} most valuable item, α_j . The quasi-linear utility θ_i [46] incurred by i is as follows,

$$\theta_i(\mathbf{b}_i) = \zeta_i \alpha_i - \sum_{j=i}^k \mathbf{b}_{j+1} (\alpha_j - \alpha_{j+1}). \quad (3.6)$$

The term $\zeta_i \alpha_i$ denotes the time reward gained by i by moving on the i^{th} turn. The payment term, $\sum_{j=i}^k \mathbf{b}_{j+1} (\alpha_j - \alpha_{j+1})$, represents the risk [47] associated with moving on that turn. It follows that an allocation of a conservative agent to a later turn (smaller α) also presents the lowest risk and vice-versa.

3.3 GAMEOPT: Methodology

We present our overall approach in Figure 3.3. Computation begins by receiving the positions, velocities, and initial bids from all agents in the control zone via V2I communication (Section 3.2). In the following planning phase, this information along with the processed bids is used to generate an optimal priority order or sequence, which determines the order in which agents

will enter the intersection (Section 3.3.1). The control phase of our approach then involves using an optimization-based trajectory planner to compute the optimal velocity (Section 2.4.3) for each agent, which satisfies the priority order while simultaneously guaranteeing safety and real-time performance.

3.3.1 GAME: Selecting the Priority Order

Choosing an optimal ordering for agents to navigate unsignalized traffic scenarios is equivalent to allocating each agent a turn in which they would cross the intersection. Such an allocation depends on the incentives of the agents which, in many cases, are not known apriori. Auctions model the incentives of agents in unsignalized traffic scenarios using an optimal combination of bidding, allocation, and payment strategies. In the rest of this section, we present the auction framework for generating an optimal priority order, followed by an analysis of its fairness.

The SSA (Section 3.2.4) is an ideal mechanism to generate a priority order for the agents. *The agent with the highest priority bid is allowed to navigate the scenario first, followed by the agent with the next highest priority, and so on.* This algorithm runs in polynomial time since the main computation at this stage is dominated by sorting the agents' bids [48]. The priority value of an agent is its true value, ζ_i , and our approach accommodates many different metrics that can be used for computing the priority value for an agent. For instance, our approach can work with both driver behavior-based [32], distance-based [36], and monetary-based [33] bidding strategies. We compute the priority value based on its velocity and its distance from the intersection. More specifically, agents that are closer to the intersection and have higher velocities have higher priority

to cross the intersection.

More formally, recall that $s_i(t)$ measures the distance of agent i from the intersection, and we use τ_i to denote the time i will reach the intersection, based on its current velocity and $s_i(t)$. Then, the priority value for agent i is,

$$\zeta_i = (c_1 - s_i(t)) \times (c_2 - \tau_i), \quad (3.7)$$

where c_1 and c_2 are constants representing maximum bounds on control zone length and time to intersection respectively. To avoid a single lane dominating the intersection (due to a constant flow of fast-moving vehicles) and to prevent congestion, we reward agents that have been waiting in queue for a long time. Thus, $\zeta_i \leftarrow w_i \zeta_i$, where w_i represents the waiting time reward value. Such a strategy of mitigating congestion has been commonly used in auction-based approaches [33]. Finally, to deal with multi-agent dynamic traffic, we implement an overflow strategy where, if an agent with a higher priority is behind an agent with a lower priority, then the former transfers some amount of their bid to the latter. This phenomenon is commonly observed in agent-based motion models such as the Social Forces model.

In [32], the authors show that the sponsored search auction is game-theoretically optimal for static intersections consisting of 1 vehicle on each arm of the intersection. In this work, we extend the proof for dynamic traffic intersections where we must take into account three factors—*(i)* multiple vehicles on each arm of the intersection and *(ii)* lower priority vehicles blocking higher priority vehicles, and *(iii)* providing waiting time rewards to agents.

Fairness Analysis of q^*

From a game-theoretic perspective, fairness implies that no agent is incentivized to “cheat” or, in other words, this occurs when the dominant strategy for each agent is to bid their true valuation ζ_i . Incentivizing traffic agents to bid their true value ($b_i = \zeta_i$) as a dominant strategy is known as incentive compatibility [46]. Chandra et al. [32] showed that SSAs implemented at static intersections with a single agent on each arm are incentive-compatible. We extend the theoretical framework of static SSAs to dynamic intersections with multiple vehicles, taking into account *overflow*, which is when a lower-priority vehicle blocks a higher-priority one.

Theorem 1. *Incentive compatibility for dynamic intersections:* *For each agent i for $i = 1, 2, \dots, n$ and $n = n_1 + n_2 + n_3 + n_4$ where n_j represents the number of vehicles on the j^{th} arm, bidding $b_i = \zeta_i$ is the dominant strategy.*

Proof. 1. Use SSA to compute an ordering over all agents. But this can generate false bids overbids and underbids.

2. For example, an agent may be assigned to go second on account of a higher priority vehicle.

3. But the latter vehicle may not even conflict with the first vehicle. Conversely, agent i might be asked to go first, risking a collision.

4. Therefore, it is important to identify conflicting and non-conflicting lane groups. Run the SSA in conflicting groups only.

5. At this point, we can say that since we have optimized conflicting groups and non-conflicting

groups do not collide anyways, the resulting auction is incentive-compatible

□

The next desired property in a fair auction is welfare maximization [46] which maximizes the total time reward (Section 3.2.4) earned by every agent. We show that SSAs maximize social welfare for dynamic intersections as well.

Theorem 2. Welfare maximization for dynamic intersections: *Social welfare of an auction is defined as $\sum_i \zeta_i \alpha_i$ for each agent i for $i = 1, 2, \dots, n$ and $n = n_1 + n_2 + n_3 + n_4$ where n_j represents the number of vehicles on the j^{th} arm. Bidding $\mathbf{b}_i = \zeta_i$ maximizes social welfare for every agent.*

Proof. We can show that welfare is maximum for each set of conflicting lane group. We need to show that social welfare of the system is maximized when the welfare of each individual conflicting lane group is maximized. Our proof is based on induction. Base case ($n = 1$): for $n = 1$, this reduces to the static auction case [32]. Hypothesis: Suppose the current system consists of $n > 1$ conflicting groups and that the social welfare of this system is maximized. Then we want to show that the addition of $|n + 1|$ agents (belonging to the $(n + 1)^{\text{th}}$ conflicting group also maximizes social welfare of the system. Now, welfare of system with n conflicting groups is

$$\sum_{i=1}^n \sum_{j=1}^{|i|} \zeta_j^i \alpha_j^i$$

From the hypothesis, we know that the above is maximum. Now if the $(n + 1)^{\text{th}}$ conflicting group is added, then we can maximize that using the static SSA auction [32]. The resulting sum is a sum of two optimized terms, so the resulting welfare of the system remains maximum. □

Finally, we propose a novel strategy to address overflow by transferring a portion of the higher priority agent's bid to the lower priority agent i . We denote such a transfer by $(\hat{a}, \hat{b} = a \xrightarrow{c} b)$, where $\hat{a} = a - c, \hat{b} = b + c$. More formally,

Theorem 3. *Overflow prevention:* *In a current SSA, suppose there exists $(i, j) \in [n_k] \times [n_k], k = 1, 2, 3, 4$, such that $p_i > p_j$ and $s_i[t] > s_j[t]$. Let $\hat{\zeta}_i, \hat{\zeta}_j = (\zeta_i \xrightarrow{q} \zeta_j)$. If*

$$q < \zeta_i \left(1 - \frac{\alpha_{i+m}}{\alpha_i}\right) - \sum_{s=i}^{i+m-1} \zeta_{s+1} \left(\frac{\alpha_s - \alpha_{s+1}}{\alpha_i}\right),$$

the new SSA with $\hat{\zeta}_i, \hat{\zeta}_j$ as the new priority values for i, j is incentive compatible.

Proof. The difference in utility for agent i by transferring some of her valuation to agent j is given by the following equation. Here, we assume that by executing this transfer, agent i gained a jump of m :

$$\begin{aligned} \Delta u = (v_k - q)\alpha_k - [v_{k+1}(\alpha_k - \alpha_{k+1} + v_{k+2}(\alpha_{k+1} - \alpha_{k+2} + \dots \\ + v_{k+m}(\alpha_{k+m-1} - \alpha_{k+m}))] - v_k\alpha_{k+m} \end{aligned} \quad (3.8)$$

Rearranging the terms, $\Delta u > 0$ if,

$$\begin{aligned} v_k(\alpha_k - \alpha_{k+m}) - \sum_{j=k}^{k+m-1} v_{j+1}(\alpha_j - \alpha_{j+1}) > q\alpha_k \\ \implies q < v_k \left(1 - \frac{\alpha_{k+m}}{\alpha_k}\right) - \frac{1}{\alpha_k} \sum_{j=k}^{k+m-1} v_{j+1}(\alpha_j - \alpha_{j+1}) \end{aligned} \quad (3.9)$$

□

As a final remark, Carlino et al. [33] show that multiplying a bid by a waiting time reward does not change the incentive compatibility of the auction. We defer the proofs of Theorems 1, 2, and 3 to [49].

3.3.2 Multi-lane Intersection Planning and Control

A key capability of our method is its ability to handle traffic effectively in multi-lane intersections. The extension from single-lane to multi-lane is not straightforward. Many assumptions that can be made in single lanes (such as one vehicle at a time in the conflict zone) do not hold up in a multi-lane scenario. Therefore, we introduce a system to classify all vehicles in the control zone into groups based on their desired trajectory (origin road and intention) in order to handle conflicts among vehicles accurately. Group x - y : $x=road_num$, $y=intention$ (Here intention values are: $0=right_turn$, $1=go_straight$, $2=left_turn$). For example, a vehicle from road 1 taking a left turn would be in group 1-2. Fig. 3.4a shows the trajectories that vehicles in each lane can take, along with the labeling used to allocate groups based on these trajectories, marked on each of the lanes.

Here, we notice that vehicles taking right turns at the intersection (groups 0-0, 1-0, 2-0, 3-0) do not conflict with any other vehicle groups. Therefore, they are allowed to enter into the intersection freely, and the constraints in the optimization in 2.4.3, reflect this. For each of the remaining lane groups, we need to identify which other lane groups can be allowed to enter the intersection at the same time. We denote this collection as the non-conflicting group set for each

corresponding main lane group, as shown in Table 3.4b. For example, consider the main lane group 0-1. The groups with trajectories that do not conflict with this main group are 0-2, 1-1, and 2-2 along with all the right turn groups (0-0, 1-0, 2-0, 3-0). What this effectively means is that any vehicles belonging to the non-conflicting group set of a main lane group will be allowed to enter the intersection at the same time as a vehicle of the main lane group. Note that vehicles in the right turn at intersection groups (0-0, 1-0, 2-0, 3-0) are directly marked as non-conflicting and are included in the non-conflicting groups list for all vehicle groups by default. The non-conflicting groups are based on the assumption that the vehicles approaching the conflict zone have completed all strategic lane changes.

3.3.3 OPT: Computing Optimal Velocities

The Optimization Problem:

Based on the optimal priority sequence q^* from Section 3.3.1, the optimal command velocities, $\mathbf{u}_i$, for each vehicle are computed by optimizing Equation 2.5 for a fixed sequence. The objective function restated:

$$\mathcal{J}(\mathbf{u}_i|q^*) = \min_{\{\mathbf{u}_i\}} \sum_{i=1}^n \lambda(\mathbf{u}_i - \bar{v})^2 + (1 - \lambda)(\mathbf{u}_i - v_i)^2 \quad (3.10)$$

The first quadratic term of Equation 3.10 prevents $\mathbf{u}_i$ from deviating from the speed limit $\bar{v}$ (results in improved throughput). The second quadratic term minimizes the control effort applied to optimize fuel efficiency. Below we discuss constraints on safety and the influence of q^* on Equation (3.10).

Constraints on Safety and Compliance with q^* :

We consider both longitudinal (rear-end) collisions in the control zone and collisions in the conflict zone. Rear-end collisions along each of the lanes in the control zone are prevented by ensuring all vehicles in the same lane maintain a safe distance between each other. Formally,

$$\left| s_j^k(t+1) - s_{j'}^k(t+1) \right| \geq l_j + M_{sr} \quad (3.11)$$

for all k lanes in the control zone and for all agents j, j' with $j \neq j'$ in the k^{th} lane. l_j represents the length of the agent j and M_{sr} is a safety margin added to maintain a safe gap between vehicles and to account for imperfections in sensing and actuation. We compute the future position $s_i(t+1)$ using,

$$s_i(t+1) = s_i(t) - \Delta t \left(\frac{v_i(t) + \mathbf{u}_i(t)}{2} \right) \quad (3.12)$$

Here, Δt is defined as the planning time step of the controller.

Next, we also need to ensure that no collisions occur inside the conflict zone. This task reduces to ensuring no vehicles belonging to conflicting lane groups (3.3.2) enter the intersection simultaneously. Therefore, for each vehicle i in the sequence q^* , paired with a vehicle j in a conflicting group where i has higher priority over j , we add the constraint,

$$t_c^i + \frac{(l_i + M_{sl})}{\mathbf{u}_i} \leq t_c^j \quad (3.13)$$

where $t_c^i = \frac{s_i(t+1)}{u_i}$ represents the time at which i would arrive at the intersection. The above constraint ensures that command velocities u_i are chosen such that vehicles proceed through the conflict zone in the order of the sequence under consideration q^* . Note that this constraint is applied to every conflicting pair of vehicles; non-conflicting vehicles are allowed to enter the intersection at the same time, increasing the efficiency of the system. The safety margin M_{sl} corresponds to the width of the intersection and prevents lateral collisions from occurring when conflicting vehicles cross the intersection.

The safety constraints with respect to ordering q^* in Equations (3.11) and (3.13) result in an intractable mixed-integer quadratic programming (MIQP) optimization problem. This problem is further compounded due to lane indexing, which introduces a new variable indicating the agent's lane. We remove the dependency over the lane index by separating the constraints for each lane and using the multi-lane conflict resolution scheme introduced in 3.3.2. This results in the simplification of Equations (3.11) and (3.13), reformulating the original MIQP as a quadratic programming (QP) optimization problem; this QP can be solved in real-time. More formally, Equation (3.11) can be simplified as,

$$\begin{aligned} u_j^k - u_{j+1}^k &\geq (v_{j+1}^k - v_j^k) \\ &+ \frac{2}{\Delta t} (s_j^k - s_{j+1}^k + l_j + M_{sr}) \end{aligned} \tag{3.14}$$

for all k lanes in the control zone and for all $j, j+1$ consecutive vehicles belonging to lane k . And Equation (3.13) can be simplified as,

$$\mathbf{u}_j \left(s_i - \frac{\Delta t}{2} \cdot v_i + l_i + M_{sl} \right) \leq \mathbf{u}_i \left(s_j - \frac{\Delta t}{2} \cdot v_j \right) \quad (3.15)$$

for all vehicles i in sequence q^* and for all vehicles j in a conflicting group to vehicle i .

Finally, we enforce reachability by bounding the command velocities by the speed limit, $\bar{v}$, as well as the acceleration and braking capabilities of each individual vehicle.

$$0 \leq \mathbf{u}_i(t) \leq \bar{v} \quad (3.16)$$

$$a_i^{\min} \Delta t \leq \mathbf{u}_i(t) - v_i(t) \leq a_i^{\max} \Delta t \quad (3.17)$$

Solving OPT: The constraints (3.14), (3.15), (3.16), and (3.17) are provided to OPT with objective function given by Equation (3.10), which is solved numerically using Gurobi (version 9.1.2) [23].

The computed target $\mathbf{u}_i$ command velocities are transmitted to each of the CAVs in the control zone. Note that we do not consider delays in communication and assume data is transmitted instantaneously via V2I communication. The low-level controller on-board each CAV then computes the acceleration or deceleration needed, to achieve the desired $\mathbf{u}_i$ command velocity within the control time (Δt) duration.

Finally, a key feature of our approach is that it is robust to drift in the executed control. Our approach operates at a cycle frequency of 100 ms after which new command velocities are obtained, updating the previous controls. Therefore, any deviation between the optimal control and the executed control is erased during every 100 ms update cycle.

3.4 Experiments and Results

In this section, we evaluate the capabilities of our approach, study the impact of varying environmental parameters, and perform comparisons against other real-time capable methods.

3.4.1 Experiment Setup

We evaluate the performance of GAMEOPT in a multi-lane four-arm intersection simulation using the SUMO platform [50] with the following parameters; *Control zone length* (L_c) = 150m, *Speed limit* ($\bar{v}$) = 20m/s *Objective trade-off* (λ) = 0.7, *Safety margins*: $M_{sr} = 2.0$ and $M_{sl} = 25.0$. The high-level controller uses the TraCI traffic controller interface to communicate with the simulation. All simulations and optimization algorithms run on a personal computer with an Intel i7-8750H CPU and 32GB of RAM. We provide results in additional scenarios with varying lanes and traffic density in [49].

3.4.2 Evaluation Metrics and Baselines

The key performance metrics used to evaluate merging algorithms are *throughput* (number of vehicles that can travel through the intersection per min), *time-to-goal* (average time vehicles spend in the control zone), and *fuel consumption*. To showcase the properties and appropriately assess the performance of GAMEOPT, we compare it with two other real-time capable baseline approaches. The first baseline we choose to compare with is an auction framework where the bidding strategy is based on the order in which agents arrive at the intersection, similar to the FIFO principle. To test the limits of our approach, we also compare against well-tuned traffic lights (light

timing set to maximize performance), a well-established signaled intersection traffic management system. We refer to these baselines as TRAFFIC-LIGHT and AUCTION in the following sections.

3.4.3 Efficiency Analysis

From the blue curves in Figure 3.5, we observe that GAMEOPT exhibits linearly increasing throughput with increasing input flow while the other two methods become capped after a point. Similarly, we see that GAMEOPT is capable of maintaining low temporal delays as well as low fuel consumption across all values of input flow. This highlights the consistent performance capabilities of the GAMEOPT method, even at very high input flow rates such as 10,000 vehicles per hour. Note that the input flow rate is defined as the total number of vehicles that enter the intersection simulation per hour on all four roads.

Results and Comparison with Baselines: In Figure 3.5, we show results of comparing GAMEOPT with TRAFFIC-LIGHT and AUCTION based on throughput, time-to-goal, and fuel efficiency. Perhaps the most significant result is that not only does GAMEOPT outperform the AUCTION approach, it also outperforms TRAFFIC-LIGHT, a well-established method optimized for signalized intersections. In Figure 3.5a, we observe that GAMEOPT linearly increases in throughput (equivalent to maximizing flow through the intersection) with increasing flow rate. At high flow rates, GAMEOPT performs approximately 25% and 174% better when compared to TRAFFIC-LIGHT and AUCTION, respectively. We also note that these relative performance improvements become more apparent with the increase in flow rate.

This follows our intuition that with fewer vehicles, we do not require a complex optimization-

based method; a simpler signal-based method suffices. However, as the traffic on the road increases, which often leads to increased congestion and delays, the benefit of GAMEOPT becomes apparent. Furthermore, from Figures 3.5b and 3.5c, we observe that at high flow rates, our method reduces the time-to-goal by 75% and 81%, respectively, and reduces fuel consumption by 33% and 66%, respectively, compared with TRAFFIC-LIGHT and AUCTION. This shows that GAMEOPT results in less waiting in queues along with less acceleration and braking tasks, resulting in improved fuel efficiency.

We also analyze the impact of increasing road speed limits from 20m/s to 25m/s, as this is a possibility in fully autonomous systems. In Figure 3.6a, we observe that while TRAFFIC-LIGHT and AUCTION both remain unchanged, GAMEOPT exhibits reduced time-to-goal. This shows that the overall capabilities of GAMEOPT can be increased further with improved road infrastructure. We also test the impact of creating unbalanced vehicle demand on different arms of the intersection by varying the inflow rate on each arm. Assuming we keep the total number of vehicles constant, some arms experience unusually high traffic density. As expected, from Figure 3.6b we observe that AUCTION's and TRAFFIC-LIGHT's performance worsens whereas GAMEOPT remains unchanged, demonstrating robustness to this configuration.

Real-time Operation: The algorithm in our optimization-based approach operates at a frequency of $f_s = 10\text{Hz}$ (updates all control commands every 100ms). The time required for the optimizer to compute control outputs for a single sequence has a mean of 1.16ms, standard deviation of 1.73ms, and a worst-case maximum of 17.25ms. This is significantly lower than the controller update rate of 100ms, thus enabling real-time operation. The fastest optimization-based method in

literature takes up to 2.4 seconds for 50 vehicles [43, 51] in a simple single-lane environment. In comparison, GAMEOPT takes less than 18 milliseconds for 50+ vehicles in a multi-lane setting. This is a speed-up of around $130\times$ in considerably more complex environments.

3.4.4 Effect of Overbidding and Underbidding

According to Theorem 1 in Section 3.3.1, overbidding or underbidding yields a lower utility [32]. In simulation, when the auction is combined with a trajectory planner, we observe even more drastic consequences. In the case of overbidding, the agent submits a higher than expected bid to the trajectory planner. The planner accommodates this higher bid by reducing a conflicting agent's speed, causing the latter to slow down and create a dangerous situation with possible rear-end collisions. In the case of underbidding, the trajectory planner reduces the speed of the agent that is underbidding, resulting in possible rear-end collisions.

3.5 Extensions to GAMEOPT intersection control

3.5.1 Extension to heterogeneous vehicles

When dealing with heterogeneous multi-class traffic we find that different classes of vehicles can have significantly different goals. Some classes may prefer to prioritize reaching their destination and speed while others would prefer to maintain fixed speeds and not perform too many acceleration tasks in order to save fuel. For example, an ambulance cares more about speed maximization, while a large semi-trailer truck would rather minimize its overall acceleration tasks due to the relatively large fuel costs associated with these actions. In addition to vehicle class, we

find that other factors such as total waiting time, current speed, and individual vehicle intent or aggressiveness play a major role in what features each car prioritizes. In order to achieve a more optimal control solution it is necessary that each vehicle should be treated differently based on its specific needs. As such we assign a vehicle type c_i which depends on the class of vehicle and a driver preference (aggressiveness) value d_i , which takes a value from 0 to 1 indicating a vehicles bias towards either driving as fast as possible (towards 1) or reducing fuel usage (towards 0). We also include the waiting time $w_i(t)$ indicating how long it has been since the vehicle entered the control zone. Thus the extended state vectors of each CAV are given by:

$$\Phi_i(t) = [s_i(t), v_i(t), l_i^c(t), w_i(t), c_i, d_i, G_i, l_i, a_i^{max}, a_i^{min}]^T \quad (3.18)$$

3.5.1.1 Changes to objective function

When we are dealing with heterogeneous traffic it becomes increasingly important to treat individual vehicles differently depending on their needs and priorities. There are two main types of priorities we consider in this research. The first type is the Speed Priority (P_i^s), which represents each vehicle's desire to maximize its individual velocity. The second type is the Speed Variation Priority (P_i^v), which represents each vehicle's desire to minimize variations to its current speed, thereby minimizing acceleration tasks and overall fuel consumption. Based on the vehicle class c_i and vehicle preference d_i , a priority assignment node computes these two types of priorities (P_i^s, P_i^v) for each vehicle i in the control zone. Each vehicle depending on its class c_i has upper and lower bounds on its priority values. The exact value assigned within these class bounds takes into

account each vehicle's individual preference d_i . While the λ parameter in Equation (3.10) provides a way to trade-off between speed maximization and fuel savings it only does so in a global sense. In order for each individual vehicle to adjust its preference towards speed maximization or fuel savings, we introduce the two priority terms (P_i^s, P_i^v) into the objective function resulting in the following updated objective function:

$$\mathcal{J}(\mathbf{u}_i|q^*) = \min_{\{\mathbf{u}_i\}} \sum_{i=1}^n \lambda P_i^s(\mathbf{u}_i - \bar{v})^2 + (1 - \lambda) P_i^v(\mathbf{u}_i - v_i)^2 \quad (3.19)$$

Here, each individual vehicle i , can affect the bias towards either throughput or fuel savings depending on its own individual preferences captured in its two priority values. This allows for a more optimal solution that can cater to the individual needs of each vehicle.

3.5.2 Extension of multi-bidding strategy for multi-sequence generation

The Sponsored Search Auction (SSA), as detailed in Section 3.2.4, presents an optimal mechanism for generating a priority order for agents. This algorithm allows the agent with the highest priority bid to navigate the scenario first, followed by the agent with the next highest priority, and so on. With a polynomial runtime, dominated by sorting the agents' bids, the SSA is an efficient solution [48]. In GameOpt [52], agents' bids were based on a single dimension - the current kinematics of each agent. However, in this work, we propose a more advanced bidding language where agents can submit multi-dimensional bids, expressing their preferences using various features or currencies such as driver behavior-based [32], distance-based [36], and time-based [33].

Formally, we represent the set of bidding currencies as $\mathcal{F}^m$, where $f \in \mathcal{F}^m$ denotes an

m -dimensional feature vector comprising bidding currency values. We further define $\Omega^m \subseteq \mathbb{R}^m$ as the space of coefficient vectors $\omega \in \Omega^m$. A master bid, $\mathbf{b}_i : \mathcal{F}^m \times \Omega^m \rightarrow \mathbb{R}$, is then generated and passed on to the SSA. To address multi-agent dynamic traffic, we implement an overflow strategy where, if an agent with a higher priority is behind an agent with a lower priority, the former transfers a portion of their bid to the latter, similar to the Social Forces model.

Typically, SSAs employ a single bidding currency in their frameworks, reducing the space of bidding strategies to $\mathcal{F}^1$. Although some recent studies [53, 54] propose extensions that allow bidding on multiple slots or keywords, these auctions still utilize a single currency (bidding over keywords). However, utilizing a single bidding currency poses several challenges, primarily that if an agent has a specific budget B for multiple keywords and spends 90% of B towards a high-profile keyword, they may not have sufficient funds left to bid on other keywords.

In contrast, multi-currency bidding strategies theoretically allow agents to realize multiple budgets and thus, expand their power to express their preferences. However, multi-currency bidding strategies are challenging to implement in practice, as many currencies lack constraints, and the dimension of the space $\mathcal{F}^m$ grows exponentially with m .

In this work, we propose a method for handling multi-currency bidding strategies. Specifically, we consider the features *time to intersection* (T_I^i), *distance to intersection* (D_I^i), *waiting time* (W_I^i), and *driver preference or assertiveness* (A_I^i). Thus the full feature (bidding currency) set is defined as $f = [T_I^i, D_I^i, W_I^i, A_I^i]$, each of which constitutes a continuous feature, with $m = 4$. The master bid $\mathbf{b}_i$ is then computed as follows:

$$\mathbf{b}_i = \omega^\top \mathbf{f} \quad (3.20)$$

Which using the coefficients ω_k where $k \in \{1, \dots, m\}$ can also be written as,

$$\mathbf{b}_i = \omega_1 * T_I^i + \omega_2 * D_I^i + \omega_3 * W_I^i + \omega_4 * A_I^i \quad (3.21)$$

Next, we describe the equations for computing each bidding feature. Recall that $s_i(t)$ measures the distance of agent i from the intersection, and we use $\tau_i(t) = s_i(t)/v_i(t)$ to denote the time in which i shall reach the intersection based on its current velocity and $s_i(t)$. The bidding capacity of vehicles should be higher when they are closer to the intersection in terms of time and distance. Therefore, the features T_I^i and D_I^i for agent i are computed as follows:

$$T_I^i = (c_1 - \tau_i(t)) \quad (3.22)$$

$$D_I^i = (c_2 - s_i(t)) \quad (3.23)$$

where c_1 and c_2 are constants representing maximum bounds on control zone length and time to intersection respectively. Next, agents specify a preference for waiting in queue for a long time, by the cumulative time it has been waiting in the control zone $w_i(t)$ (available in vehicle state vector in equation (3.18)). Feature W_I^i is then set as $W_I^i = w_i(t)$. Finally, agents also express their behavior through a driver preference assertiveness feature A_I^i . For these features, vehicles with higher waiting time and higher preference have a higher bidding capacity. The computation

of the preference feature A_I^i additionally incorporates the effects of heterogeneous vehicle types. Therefore, different classes of vehicles will have varying bounds on their preference bidding capacity A_I^i . For example, on a scale of 1 to 10, emergency vehicles would have the highest bidding range of $\{7 \text{ to } 10\}$ and passenger cars would have a lower range such as $\{1 \text{ to } 5\}$. Therefore the maximum bidding capacity within the driver preference assertiveness feature A_I^i is dependent on the vehicle class. This allows vehicles with higher priority needs to be given preference over other vehicle classes during the sequence order generation phase. Here, the actual vehicle class c_i and individual driver preference d_i are available in the vehicle state vector in equation (3.18).

By using the combined master bid obtained for each CAV i as shown in equation (3.21), a sponsored search auction is conducted to generate the final feasible merging sequence q . Here, vehicles with higher combined bids would get the opportunity to pass through the intersection before vehicles with lower bids. As the bidding system is directly linked to the dynamics of the system, the sequence generated by the auction ensures that vehicles higher up in the sequence have the dynamic capability to pass the intersection before vehicles lower down in the sequence. Certain other features such as the overflow strategy discussed also play a role in ensuring the overall feasibility of the generated sequence.

Generation of multiple feasible sequences

Thus far we described the method of generating a single feasible sequence q , using the set of bidding currencies as $\mathcal{F}^m$, and the space of coefficient vectors $\omega \in \Omega^m$. The choice of coefficients ω_k where $k \in \{1, \dots, m\}$, is a tuning parameter available to the central controller which

can be changed in order to generate multiple feasible sequences. It is difficult to tell which sequence would be truly optimal in terms of maximizing vehicle throughput while minimizing fuel consumption at this stage. Therefore it is the responsibility of the GAME subsystem (section 3.3.1) of the control system, to generate a suitable collection of candidate feasible sequences to be tested by the optimization OPT subsystem (section 2.4.3) for optimality. Thus, each of the candidate sequences q , generated in GAME, will be fed into the optimization OPT subsystem in parallel to find the best sequence q^* , whose corresponding target command velocities would be sent to each CAV in the control zone.

3.5.3 Extended results with the updated version of GAMEOPT

3.5.3.1 Changes to performance metrics and baselines

The key performance metrics used to evaluate control algorithms are *throughput* (number of vehicles that can travel through the intersection per min), *time-to-goal* (average time vehicles spend in the control zone), and *fuel consumption*. We also add the metric of *CO2 consumption* as carbon emissions are an important criterion in overall vehicle emissions. Furthermore, to showcase the performance in the presence of heterogeneous vehicles, we include separate results on *time-to-goal of EVs* (Emergency Vehicles) and *fuel consumption of trucks*.

We introduce an additional baseline to the original two real-time capable baselines used in GAMEOPT [52], which include the Auction method which, is essentially an advanced multi-lane stop sign and timed traffic lights tuned to fit the input flow rate. The new baseline is a gap-based adaptive traffic light control strategy [55]. This adaptive strategy allows traffic lights to dynam-

ically adapt to current traffic conditions in the incoming lanes of an intersection by prolonging traffic phases whenever a continuous stream of traffic is detected. We refer to these baselines as TRAFFIC-LIGHT, ACTUATED-TRAFFIC-LIGHT, and STOP-SIGN. We refer to our method as COOP-CONTROL.

3.5.3.2 Performance analysis of improved GAMEOPT

The results of these experiments for the two main traffic performance metrics of throughput and time spent are shown in figure 3.7. It is observed that our method (COOP-CONTROL) significantly improves performance relative to the traffic light-based methods with the relative improvement in performance increasing with increasing input flow rate. This follows our intuition that advanced multi-agent control methods provide a significant advantage when congestion is high. At low congestion levels, even simpler methods are often sufficient. The STOP-SIGN method's performance however is very poor even at moderately high input flow rates. In fact beyond a certain vehicle flow rate, due to the low throughput in the STOP-SIGN method, the input lanes become fully packed and it is impossible to increase the input flow rate any further. Our proposed COOP-CONTROL method is observed to provide improvements to *throughput* of around 25% over the TRAFFIC-LIGHT and ACTUATED-TRAFFIC-LIGHT methods and around 137% improvement over the STOP-SIGN method as shown in figure 3.7a. Additionally, as shown in figure 3.7b, our COOP-CONTROL method improves the average time spent by vehicles (*time-to-goal*) by around 70% over the TRAFFIC-LIGHT and ACTUATED-TRAFFIC-LIGHT methods and around 83% over the STOP-SIGN method.

Here, we find that both the timed TRAFFIC-LIGHT and ACTUATED-TRAFFIC-LIGHT methods offer similar performance which shows how well the timing on the TRAFFIC-LIGHT method has been tuned to match the rate of vehicle inflow on each of the intersections input arms.

The key indicators we consider for efficiency in terms of environmental factors are total fuel consumption and CO2 emissions. The performance of our COOP-CONTROL method for these metrics is shown in figure 3.8, where we observe significant improvements over baseline methods. Once again we observe that the level of improvement increases at higher input vehicle flow rates. In figure 3.8a, we observe a 50% and 80% reduction in fuel consumption relative to the TRAFFIC-LIGHT and STOP-SIGN methods respectively. The same trend is observed for CO2 emissions in figure 3.8b, where we see a reduction of 50% and 80% relative to the TRAFFIC-LIGHT and STOP-SIGN methods respectively. This ability of our method to reduce fuel usage and CO2 emissions by such significant margins could have a major impact on overall environmental sustainability. In our COOP-CONTROL method fewer vehicles are forced to unnecessarily slow down and speed up which equates to less fuel usage to cover the same distance. Additionally, when acceleration is necessary the control algorithm attempts to minimize the amount of acceleration required.

To showcase our method's ability to handle heterogeneous traffic we also show a comparison of the performance impact for the two major edge cases of emergency vehicles minimizing their *time-to-goal of EVs* and the minimizing of *fuel consumption of trucks*. In a system where each vehicle's actions can affect the other neighboring vehicles, these two goals of minimizing time and fuel consumption are generally opposing each other. However as shown in figure 3.9, our method can successfully minimize both metrics in order to satisfy the individual needs of each

vehicle class. Over the standard benchmark of the TRAFFIC-LIGHT method, our COOP-CONTROL algorithm is shown to provide reductions of around 75% and 54% for *time-to-goal of EVs* and *fuel consumption of trucks* respectively. When compared to figures 3.7b and 3.8a, we see that EVs face fewer relative delays than the rest of the traffic and trucks relatively consume less fuel than the rest of the traffic.

3.5.3.3 Impact of unbalanced traffic inflow on improved GAMEOPT version

A major benefit of our method is its ability to adapt to dynamic changes to the input vehicle flow rate on each of the input arms of the intersection. When all the arms have equal input flow the system is symmetrical and we consider the input flow to be balanced. However, when the input flow rate on one or more of the arms is noticeably higher than the others, we consider the input flow to be unbalanced. To evaluate our method's robustness to unbalanced traffic inflow we conduct experiments where the amount of traffic originating in the horizontal lanes (from West to East or vice versa) is gradually increased relative to the traffic originating in the vertical lanes (from North to South or vice versa). We then plot the same performance metrics against the increasing ratio of horizontal lane to vertical lane traffic. Note that in these experiments, to allow for accurate comparisons, the flow rates on each arm are adjusted so that the overall total input flow rate remains fixed at 5200 *veh/hr*.

The effect of unbalanced traffic inflow on throughput and time spent is presented in figure 3.10. Our proposed COOP-CONTROL method is unperturbed by changes in the traffic inflow balance. Both the throughput and time spent remain steady under increasingly unbalanced input flow

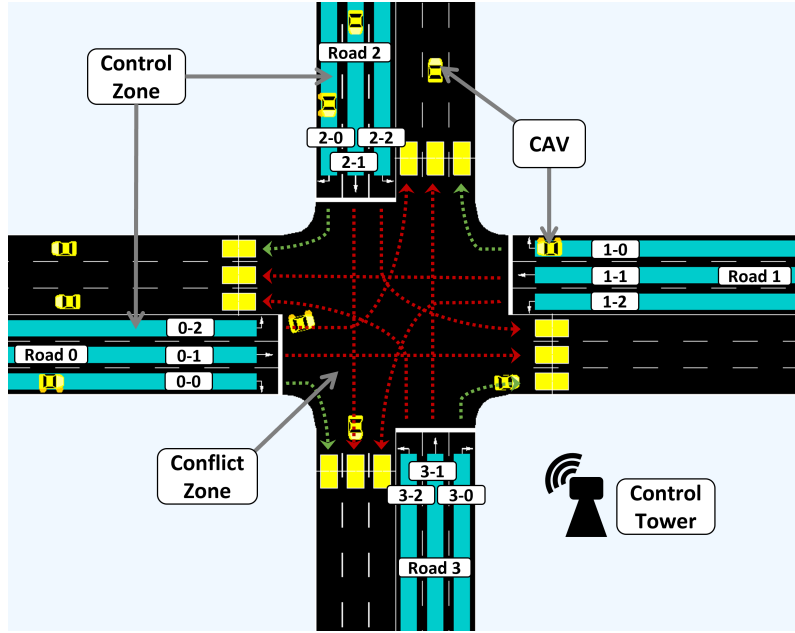
when our proposed COOP-CONTROL method is used. The timed TRAFFIC-LIGHT approach is the most affected method, with performance dropping by about 20% and 45% in throughput and time spent when the balance between horizontal and vertical lanes changes from 1 : 1 to 4 : 1. This is expected since the TRAFFIC-LIGHT method timing is tuned to handle a symmetric load of traffic. As such, in symmetric inflow conditions (1 : 1) the TRAFFIC-LIGHT performs equivalent to the ACTUATED-TRAFFIC-LIGHT method which is designed to adapt to possible changes in traffic symmetry. Therefore we see a lesser performance drop of around 5% in throughput in the ACTUATED-TRAFFIC-LIGHT method when the traffic becomes more unbalanced. While the STOP-SIGN method generally performs much worse than its competitors, we observe an interesting phenomenon of improved performance when the traffic flow becomes increasingly unbalanced. The STOP-SIGN method is inherently designed to handle unbalanced traffic as and when required. When the traffic is more unbalanced, then the traffic moving through the intersection is less diverse in terms of possible directions it takes. As such the vehicles moving in the majority-use directions can pass the intersection with less waiting especially when they belong to non-conflicting groups. This is the primary reason the stop sign method performs better when traffic is less balanced with around a 10% improvement in throughput in more unbalanced traffic.

The same observations hold in terms of fuel consumption and CO₂ emissions as shown in figure 3.11. Our COOP-CONTROL method which offers the best performance also maintains this performance irrespective of the balance of traffic flow. The TRAFFIC-LIGHT shows the highest performance degradation while the ACTUATED-TRAFFIC-LIGHT method lies somewhat in between. As in the earlier case the STOP-SIGN method performance improves in unbalanced traffic and is

actually capable of achieving fuel consumption related performance equivalent to the TRAFFIC-LIGHT method in the extreme unbalanced traffic condition.

Finally, as shown in figure 3.12, we find that these same observed trends hold in the specific heterogeneous vehicle cases of emergency vehicle time to goal and truck fuel consumption.

Overall we identify that our COOP-CONTROL method demonstrates high levels of robustness to changes in the balance of input traffic flow. It not only exhibits the highest performance among all other competing methods but is also able to maintain this level of performance irrespective of the level of symmetry in the input traffic flow.



(a) Multi-lane intersection structure with lane-based groups

Lane group	No-conflict group set
0-1	0-2, 1-1, 2-2
0-2	0-1, 1-2, 3-1
1-1	0-1, 1-2, 3-2
1-2	0-2, 1-1, 2-1
2-1	1-2, 2-2, 3-1
2-2	0-1, 2-1, 3-2
3-1	0-2, 2-1, 3-2
3-2	1-1, 2-2, 3-1

(b) Non-conflicting trajectory groups at intersection

Figure 3.4: **Conflict handling at multi-lane intersections.** We propose a novel strategy for conflict handling using vehicle grouping based on non-colliding trajectories.

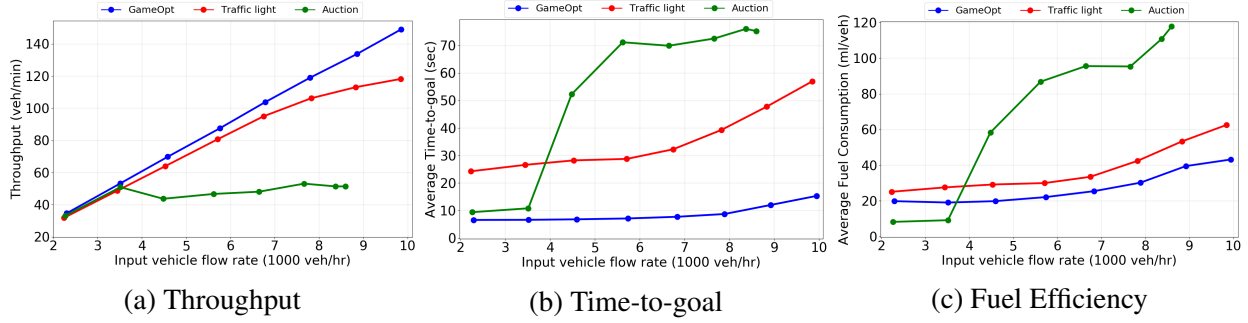


Figure 3.5: **Comparison with baselines:** GAMEOPT increases throughput by approximately 25% and 174% compared to TRAFFIC-LIGHT and AUCTION, respectively. Furthermore, our method reduces the time-to-goal by 75% and 81%, respectively, and improves fuel efficiency by 33% and 66%, respectively, compared with TRAFFIC-LIGHT and AUCTION.

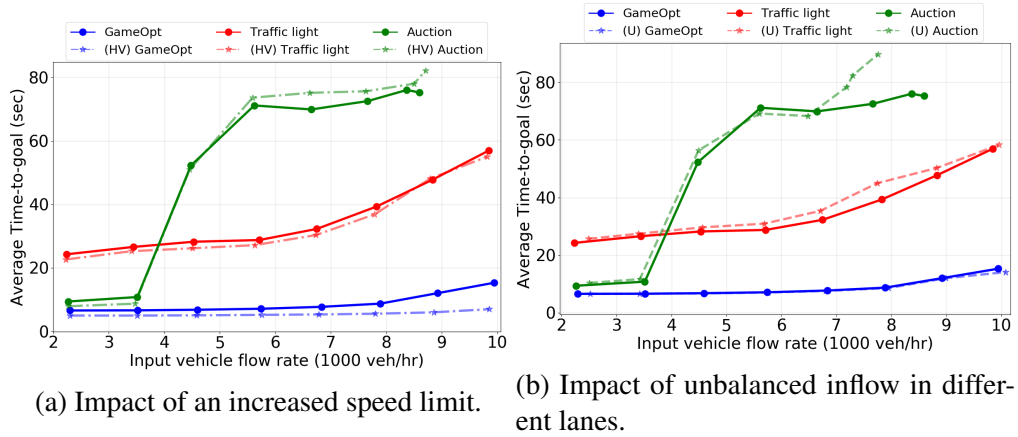


Figure 3.6: Impact of increasing the speed limit and unbalanced inflow rates on the average time-to-goal. GAMEOPT exhibits improvements with a higher speed limit and invariance to unbalanced flow.

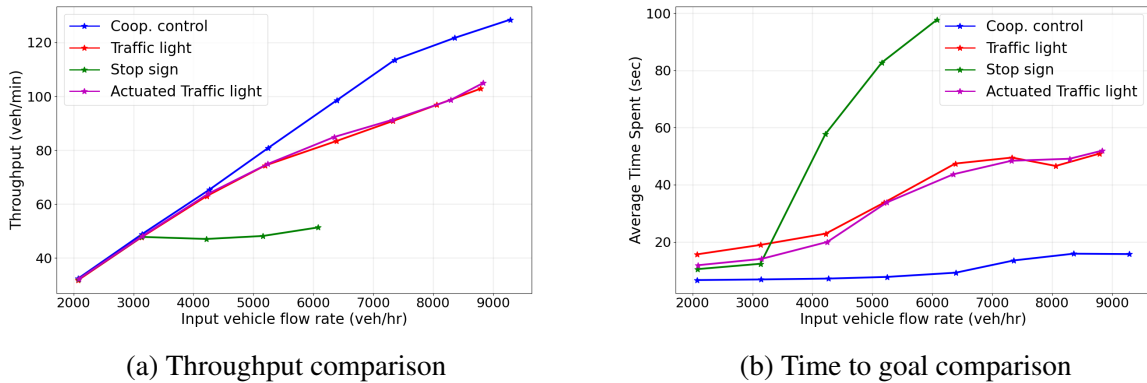
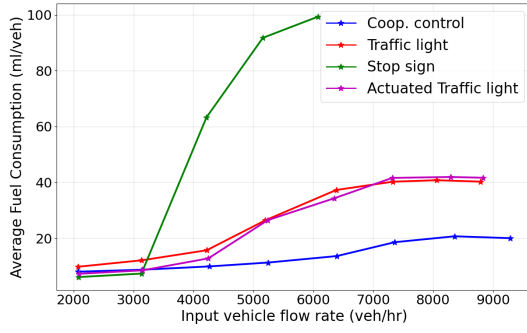
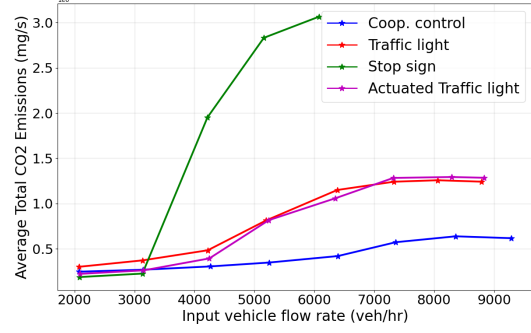


Figure 3.7: Performance in terms of throughput and time-to-goal in comparison to baselines, with increasing vehicle input flow rate.

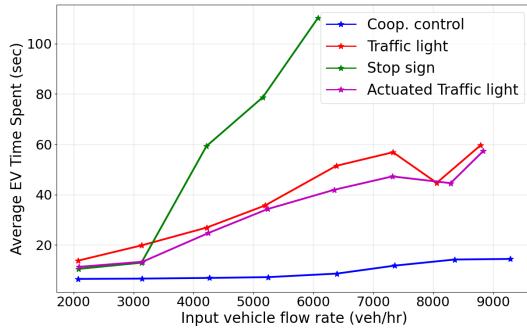


(a) Fuel Consumption comparison

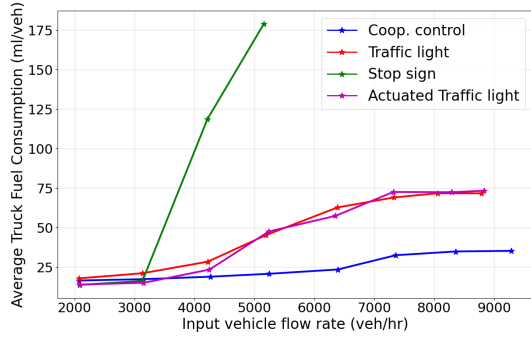


(b) CO2 emissions comparison

Figure 3.8: Performance in terms of fuel consumption and CO2 emissions in comparison to baselines, with increasing vehicle input flow rate.

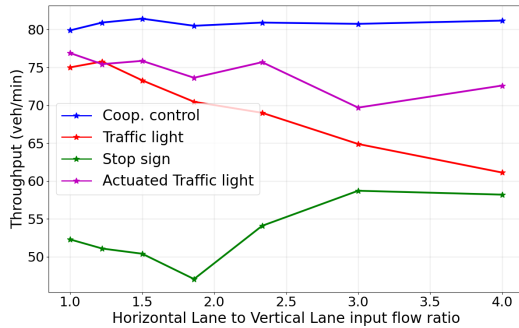


(a) EV Time to goal comparison

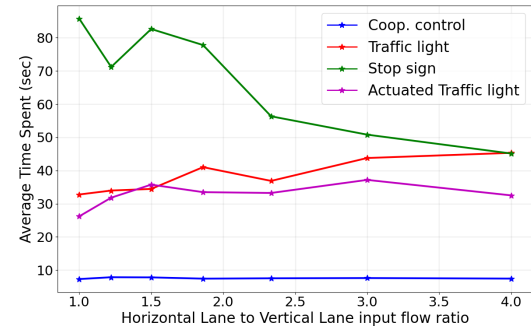


(b) Truck Fuel Consumption comparison

Figure 3.9: Performance in terms of emergency vehicle time-to-goal and truck fuel consumption in comparison to baselines, with increasing vehicle input flow rate.

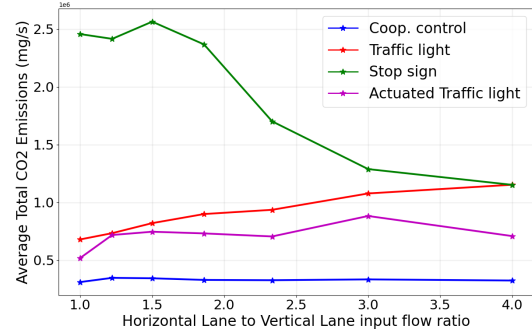
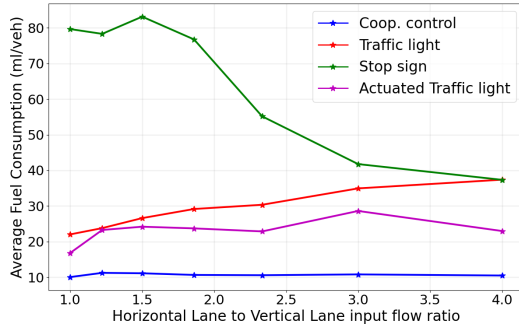


(a) Throughput comparison with unbalanced inflow



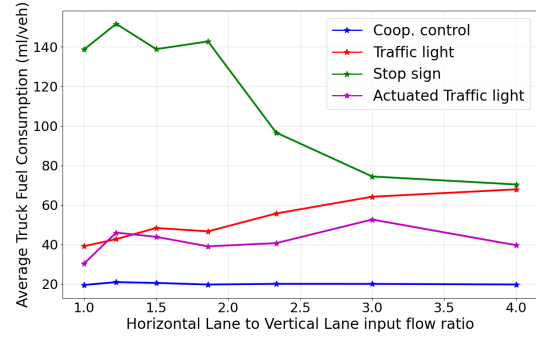
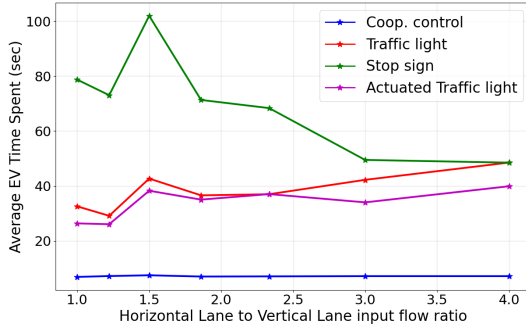
(b) Time to goal comparison with unbalanced inflow

Figure 3.10: Performance in terms of throughput and time-to-goal in comparison to baselines, with unbalanced vehicle input flow rate. Total flow rate fixed at 5200 *veh/hr*.



(a) Fuel Consumption comparison with unbalanced inflow (b) CO2 emissions comparison with unbalanced inflow

Figure 3.11: Performance in terms of fuel consumption and CO2 emissions in comparison to baselines, with unbalanced vehicle input flow rate. Total flow rate fixed at 5200 veh/hr .



(a) EV Time to goal comparison with unbalanced inflow (b) Truck Fuel Consumption comparison with unbalanced inflow

Figure 3.12: Performance in terms of emergency vehicle time-to-goal and truck fuel consumption in comparison to baselines, with unbalanced vehicle input flow rate. Total flow rate fixed at 5200 veh/hr .

Chapter 4: Coordination and Control of Traffic Lights and Mixed Traffic in Multi-Intersection Environments

4.1 Introduction

Coordination and control of vehicles traveling through urban areas is an open challenge in transportation networks. Most traditional control systems are based on timed traffic light signals or density-adaptive traffic light signals [56]. However, these systems may change with the introduction of connected automated vehicles (CAVs) that would allow far more sensing and actuation points within the traffic flow [57]. In addition to collecting traffic information via onboard sensors on CAVs, traffic control systems can suggest CAV actions that optimize the traffic flowing through an urban network [58]. Furthermore, while human driven vehicles (HDVs) cannot be directly controlled by these control systems, the presence of neighboring CAVs and traffic lights significantly improves the degree of controllability of the overall traffic flow in the network.

The main objective of an urban traffic control system is to allow vehicles to travel through the network in the shortest possible time while minimizing overall fuel consumption. To achieve this goal, a central traffic controller (CTC) is responsible for generating both CAV and traffic light control trajectories while ensuring safety, fairness, and optimal traffic throughput. Achieving

these goals in the presence of mixed traffic adds an extra layer of complexity due to the need for sufficiently accurate predictions of HDV behavior.

Literature review

Intelligent traffic light control has been studied for decades, with methods ranging from time-based switching to density-based switching using heuristics [59], optimization [60] and deep learning [61, 62]. For multi-intersections, an adaptive dynamic programming approach was presented in [63]. These methods typically allow for reactive control which can be sub-optimal and lead to vehicles having to slow down to a stop at intersections. Additionally, most of the existing research focuses on macro-level traffic modeling which often abstracts details and results in a loss of efficiency at the micro-level. Some existing algorithms aim to use traffic light timing information to reduce the fuel consumption impact [64, 65]. Research interest is also placed in CAV control for navigating through unsignalized intersections [40, 52, 66] and merging points [24, 58, 67]. However, these algorithms typically require all vehicles to be CAVs and deal with only local control for a single traffic intersection.

In the context of centralized control of multi-intersection environments with mixed traffic including both CAVs and HDVs, receding horizon methods based on mixed-integer linear programming (MILP) using either a road segment-based [68] or time-based formulation [69] have recently been proposed. Many remaining challenges exist in extending this work to real-time capable systems that can operate under realistic conditions and high-density traffic.

Contributions

Targeting multi-intersection mixed traffic coordination, we propose both centralized and decentralized MILP-based strategies for the joint computation of CAV trajectories and traffic light switching commands, and we validate their operational capability in realistic traffic simulations. Compared to the prior work in [68, 69], our novel contributions are,

- We describe a MILP formulation for vehicle and intersection coordination, with support for multiple intersections, mixed traffic, lane changes, HDV predictions, and a more advanced traffic light switching model.
- We propose a decentralized implementation by decoupling the MILP, resulting in multiple intersection traffic controllers (ITCs), to achieve real-time computations.
- We built the system with a modular software architecture, allowing it to adapt to any traffic scenario.
- We perform extensive validations of the proposed approach based on SUMO [22], using realistic simulations of vehicle demand, behavior, and interactions.

In Section 4.2, we describe the problem statement, modeling, and objectives, and Section 4.3 describes the MILP for centralized coordination. The decentralized approach is presented in Section 4.4, and Section 4.5 presents the SUMO simulation results. Finally, Section 4.7 concludes the chapter.

Notation: $\mathbb{R}, \mathbb{R}_+, \mathbb{R}_{0+}$ ($\mathbb{Z}, \mathbb{Z}_+, \mathbb{Z}_{0+}$) are the set of real, positive real, and non-negative real (integer) numbers, and $\mathbb{B} = \{0, 1\}$. The logical operators *and*, *or*, *xor*, *not* are $\wedge, \vee, \underline{\vee}, \neg$, and the

logical operators *implies* and *equivalent (if and only if)* are $\implies$, $\iff$. Inequalities between vectors are intended component-wise.

4.2 Problem Description

In the multi-intersection traffic coordination problem, the central traffic controller (CTC) computes the most suitable CAV trajectories and traffic light state trajectories to allow for smooth traffic flow, as described in Figure 4.1. The presence of mixed traffic creates the need for accurate HDV behavior prediction in order to achieve the best performance. The overarching goals of the system involve the maximization of traffic throughput while minimizing a combination of waiting time and fuel consumption. Here, we aim at executing the control center at a sampling frequency of 1 Hz, i.e., using a sampling time period $T_s = 1$ sec, similar to [68, 69].

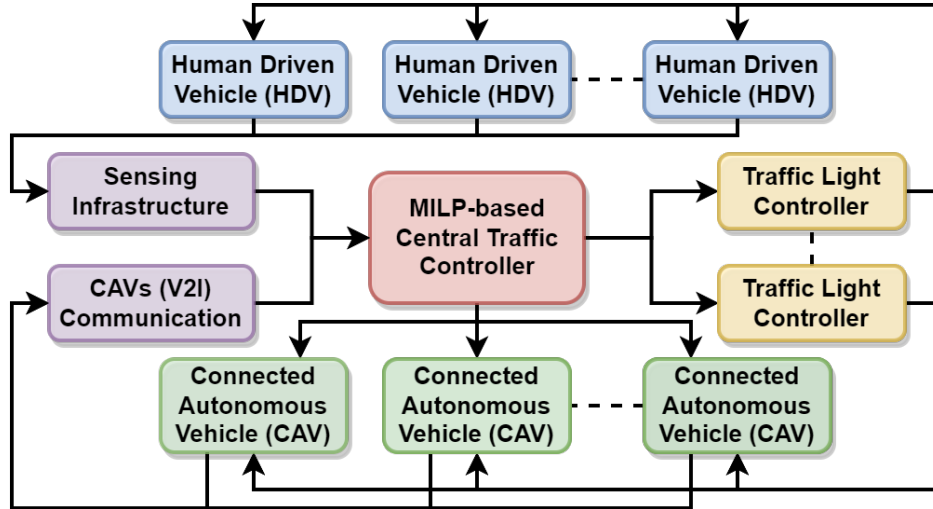


Figure 4.1: Schematic of the proposed central traffic controller (CTC) directly controlling connected automated vehicles (CAVs) and indirectly affecting the motion of human driven vehicles (HDVs) by commanding traffic lights.

4.2.1 Modeling of Physical Road Network

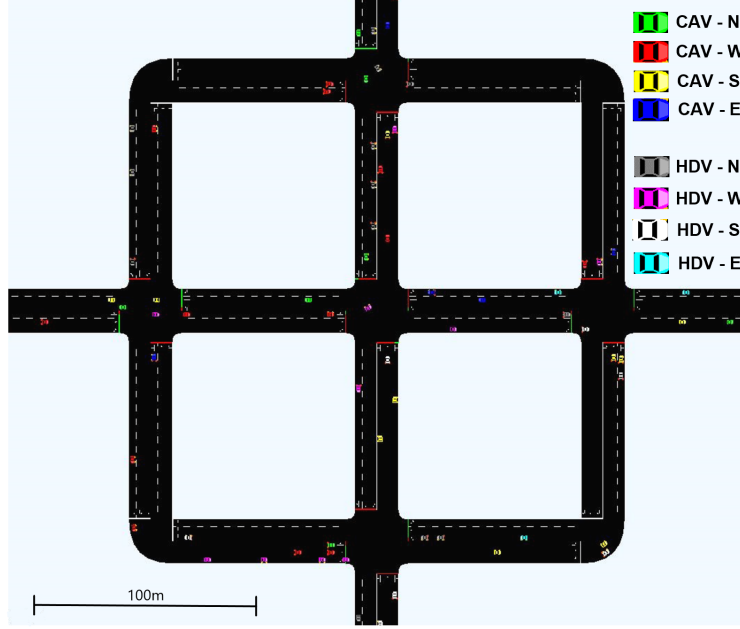


Figure 4.2: Traffic road network with 5 intersections, where each road segment consists of 3 lanes in both directions and 4 road segments where vehicles can flow in and out of the road network.

The key design parameters involved in modeling the physical road network are the number of intersections, the number of road segments, and the number of lanes in each road segment. In this work, we consider a case study based on a traffic network with 5 intersections, with connecting road segments as shown in Figure 4.2, each with 3 lanes in both directions. The distance between the centers of adjacent intersections is set to $100m$. We further use the index set $\mathcal{J} = \{1, \dots, 5\}$ to denote each of the traffic intersections. The speed limit $\bar{v}$ of the road network is set to $15m/s$. The vehicles can be either CAVs or HDVs resulting in mixed traffic, and the CAV penetration level r_{cav} (ratio of CAVs to the total number of vehicles) in the simulation can be adjusted. We use the index sets $\mathcal{V}^c$ and $\mathcal{V}^h$ to denote each of the CAVs and HDVs in the road network, where $\mathcal{V} = \mathcal{V}^c \cup \mathcal{V}^h$ and $\mathcal{V}^c \cap \mathcal{V}^h = \emptyset$.

4.2.2 Modeling of Controlled Vehicles (CAVs)

Each CAV $i \in \mathcal{V}^c$ has an onboard control architecture, capable of computing actuation commands (throttle and braking) in order to follow a sequence of velocity targets set by the CTC. This onboard controller also computes the required steering angles to keep the vehicle in the lane and to execute lane change maneuvers as instructed by the CTC. From the CTC's perspective, CAVs can therefore be modeled as agents with known dimensions moving along the center of the lane. CAVs are also assumed to be able to relay information to and from intersection infrastructure using vehicle-to-infrastructure (V2I) communication. In this work, we assume perfect communication and do not model V2I network conditions such as delays and packet loss.

4.2.3 Modeling of Human Driven Vehicles (HDVs)

Similar to the work in [69], we model the predictions of HDV behavior using a switched dynamical system to represent the reactions of HDVs to the, possibly changing, traffic rules. In the present work, we model each HDV $i \in \mathcal{V}^h$ using the following state-dependent switched dynamics:

- HDV stops at traffic light, **if** traffic light for its particular traffic direction is red and HDV is within a predetermined distance from stopping zone, **otherwise**
- HDV follows leading vehicle and maintains safe following distance, **if** leading vehicle is within a particular distance in front of HDV, **otherwise**
- HDV travels at a fixed desired target speed.

In Section 4.3, we describe how these switched dynamics can be modeled using mixed-integer

constraints.

4.2.4 Traffic Light Intersection Model

Traffic lights at intersections are modeled as a collection of finite states where a state contains the status of each of the individual light signals of a particular intersection. The traffic lights of a junction allow vehicles to pass through the intersection $j \in \mathcal{J}$ in a specific direction $d \in \mathcal{D}$ if the traffic light phase status $\psi_j^d = 1$ (green), and no vehicles can pass through the intersection j in the direction d if $\psi_j^d = 0$ (red). For a 4-way intersection, where each road segment contains 3 lanes, there are 12 possible directions as shown in Figure 4.3, i.e., we define the index set $\mathcal{D} = \{1, \dots, 12\}$.

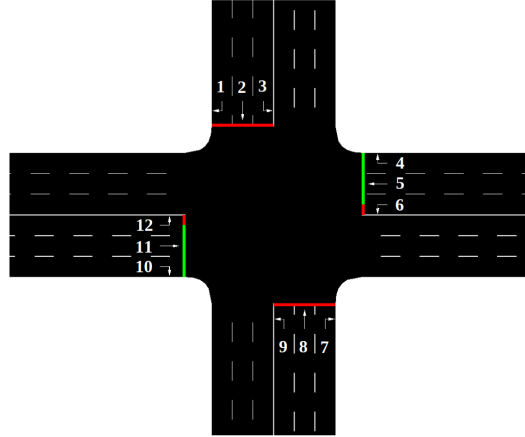


Figure 4.3: Each 4-way traffic intersection, with 3 lanes in each road segment, includes traffic lights to control the flow in each of 12 possible directions.

A state Φ_j^s of intersection $j \in \mathcal{J}$ is defined as a collection of non-conflicting traffic directions that are allowed to enter the intersection j simultaneously. For the 4-way intersection, with 3 lanes in each road segment, 9 possible states can be identified, i.e., we define the index set $\mathcal{S} = \{1, \dots, 9\}$. A mapping from state Φ_j^s to direction ψ_j^d is introduced in Table 4.1 for each intersection

$j \in \mathcal{J}$, state $s \in \mathcal{S}$, and direction $d \in \mathcal{D}$. Here, $\psi_j^d = 0$ or $\psi_j^d = 1$ indicates that a specific direction's traffic light is red or green, respectively.

State (Φ)	Phase Direction (ψ)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	1	0	0	0	0	1	1	0
3	1	0	0	0	0	1	1	0	0	0	0	1
4	1	1	0	0	0	0	1	1	0	0	0	0
5	0	0	1	1	0	0	0	0	1	1	0	0
6	1	0	0	0	0	0	0	0	0	1	1	1
7	0	0	0	1	1	1	1	0	0	0	0	0
8	0	0	0	0	0	0	1	1	1	1	0	0
9	1	1	1	1	0	0	0	0	0	0	0	0

Table 4.1: Mapping from 9 traffic light states to the value 0 (red) or 1 (green) for each of the 12 traffic light directions.

4.3 MILP Formulation for Centralized Traffic Coordination

The centralized traffic controller (CTC) is responsible for computing control trajectories for CAVs, including target velocities and lane changes, and traffic light commands while enforcing traffic rules and predicting the behavior of HDVs. Compared to the prior work in [69], the MILP-based CTC formulation in the present research includes lane-change behavior for both CAVs and HDVs, as well as more advanced modeling for traffic lights and HDV predictions.

4.3.1 CAV Control Trajectories

The vehicle dynamics involve the evolution of its position state p_i along the direction of motion in the traffic network (see Fig. 4.2), according to the target velocity v_i , i.e.,

$$p_i(t+1) = p_i(t) + \frac{v_i(t) + v_i(t+1)}{2} \Delta t, \quad (4.1)$$

for each CAV $i \in \mathcal{V}^c$, and where Δt is the discretization time. The initial target velocity $v_i(0)$ is set to the current vehicle's velocity, given by sensors and/or through V2I communication. To ensure realistic target velocities, the velocity assigned by the CTC to CAVs needs to be within the speed limit, $v_i(t) \leq \bar{v}_i$, and the maximum acceleration $\bar{a}_i^{\max}$ and deceleration $\bar{a}_i^{\min}$ limit of each vehicle

$$v_i(t) - \bar{a}_i^{\min} \Delta t \leq v_i(t+1) \leq v_i(t) + \bar{a}_i^{\max} \Delta t. \quad (4.2)$$

In order to model the lateral dynamics of vehicles, we introduce binary variables, $\sigma_i^h = 1$ if vehicle $i \in \mathcal{V}$ is in lane h , where $h \in \{1, \dots, m_1\} = \mathbb{Z}_1^{m_1}$ and m_1 is the maximum number of lanes on a road segment. These variables model lane changes, ensure vehicles enter intersections in the correct lane, and perform collision avoidance in multi-lane scenarios. At every time step, each vehicle is always in exactly one of the traffic lanes, i.e., $\sum_h \sigma_i^h = 1$.

Lane change behavior

Lateral CAV dynamics are modeled as discrete switching between lanes. We introduce two binary variables $l_{i,u}^c(t), l_{i,d}^c(t) \in \{0, 1\}$ to represent a lane change up and down respectively as follows,

$$\begin{aligned}\sigma_i^h(t) \wedge l_{i,u}^c(t) &\implies \sigma_i^{h+1}(t+1), \\ \sigma_i^h(t) \wedge l_{i,d}^c(t) &\implies \sigma_i^{h-1}(t+1).\end{aligned}\tag{4.3}$$

The implications in (4.3) can be enforced using the following inequality constraints for $i \in \mathcal{V}$ and $t \in \mathbb{Z}_0^{N-1}$

$$\begin{aligned}1 - (1 - \sigma_i^h(t)) - (1 - l_{i,u}^c(t)) &\leq \sigma_i^{h+1}(t+1) \leq 1, \\ 1 - (1 - \sigma_i^h(t)) - (1 - l_{i,d}^c(t)) &\leq \sigma_i^{h-1}(t+1) \leq 1.\end{aligned}\tag{4.4}$$

Similarly, we implement the implication for no lane change

$$\sigma_i^h(t) \wedge \neg l_{i,u}^c(t) \wedge \neg l_{i,d}^c(t) \implies \sigma_i^h(t+1).\tag{4.5}$$

Finally, we prevent lane changes both up and down at the same time, along with lane change down from the lowest lane, and lane change up from the highest lane as

$$\begin{aligned}l_{i,u}^c(t) + l_{i,d}^c(t) &\leq 1, \\ l_{i,d}^c(t) + \sigma_i^1(t) &\leq 1, \quad l_{i,u}^c(t) + \sigma_i^{m_l}(t) \leq 1.\end{aligned}\tag{4.6}$$

4.3.2 Collision Avoidance

Collision avoidance constraints are placed between vehicle pairs whose trajectories may lead to potential collisions. Specifically, vehicles within the same road segments and adjoining road segments are paired up for collision checking. For these vehicle pairs, we enforce that the lead vehicle should remain ahead of the following vehicle, while both vehicles are in the same lane,

$$\begin{aligned}
& \{p_{i_1}(t) \leq p_{i_2}(t)\} \wedge \{\sigma_{i_1}^h(t+1) = \sigma_{i_2}^h(t+1)\} \\
& \implies p_{i_1}(t+1) \leq p_{i_2}(t+1), \\
& \{p_{i_1}(t+1) \leq p_{i_2}(t+1)\} \wedge \{\sigma_{i_1}^h(t+1) = \sigma_{i_2}^h(t+1)\} \\
& \implies p_{i_1}(t) \leq p_{i_2}(t),
\end{aligned} \tag{4.7}$$

for $h \in \mathbb{Z}_1^{m_1}$, two vehicles $i_1, i_2 \in \mathcal{V}$, and $t \in \mathbb{Z}_0^{N-1}$.

Taking into account spatial dimensions of vehicles and safety margins, we modify the above equations to include vehicle lengths L_i and safety distances, using a base value D^s and a velocity-

dependent component with coefficient k^v . We propose the mixed-integer inequality constraints

$$\begin{aligned}
& p_{i_1}(t) + R_{i_1, i_2} \geq p_{i_2}(t) - M\gamma_{i_1 i_2}^c(t) \\
& \quad - M(1 - \gamma_{i_1 i_2}^l(t)) + L_{i_1} + D^s + k^v v_{i_2}(t) - s_{i_2}^c, \\
& p_{i_1}(t+1) + R_{i_1, i_2} \geq p_{i_2}(t+1) - M\gamma_{i_1 i_2}^c(t) \\
& \quad - M(1 - \gamma_{i_1 i_2}^l(t)) + L_{i_1} + D^s + k^v v_{i_2}(t) - s_{i_2}^c, \\
& p_{i_1}(t) + R_{i_1, i_2} \leq p_{i_2}(t) + M(1 - \gamma_{i_1 i_2}^c(t)) \\
& \quad + M(1 - \gamma_{i_1 i_2}^l(t)) - L_{i_2} - D^s - k^v v_{i_1}(t) + s_{i_1}^c, \\
& p_{i_1}(t+1) + R_{i_1, i_2} \leq p_{i_2}(t+1) + M(1 - \gamma_{i_1 i_2}^c(t)) \\
& \quad + M(1 - \gamma_{i_1 i_2}^l(t)) - L_{i_2} - D^s - k^v v_{i_1}(t) + s_{i_1}^c, \\
& \gamma_{i_1 i_2}^l(t) \geq \sigma_{i_1}^h(t+1) + \sigma_{i_2}^h(t+1) - 1,
\end{aligned} \tag{4.8}$$

where R_{i_1, i_2} denotes the initial relative distance between vehicles i_1 and i_2 , and $s_i^c \geq 0$ represents the slack variable that is added to ensure feasibility in the presence of model mismatch and disturbances. The presence of the slack variables s_i^c , in combination with the safety margins $D^s + k^v v_i(t)$, account for the possible mismatch between predicted and actual vehicle behavior. In addition, the receding horizon implementation of the control center solves the MILP at each sampling time period $T_s = 1$ sec in order to correct possible prediction errors. The binary variable $\gamma_{i_1 i_2}^l \in \{0, 1\}$ is 1 if the vehicles i_1 and i_2 are in the same lane, and the binary variable $\gamma_{i_1 i_2}^c \in \{0, 1\}$ defines the order between vehicles i_1 and i_2 to enforce collision avoidance. Note that the constraints in (4.8) additionally prevent lane changes that are predicted to lead to collisions at future time steps.

4.3.3 Intersection Membership

We introduce the matrix F_{sd} to model the mapping from traffic light states to individual traffic light signals for each direction, as shown in Table 4.1. The relationship between traffic light states and individual traffic light signals for intersection $j \in \mathcal{J}$ is enforced by the constraints

$$\psi_j^d(t) \geq F_{sd} \Phi_j^s(t), \quad \psi_j^d(t) \leq 1 - (1 - F_{sd}) \Phi_j^s(t), \quad (4.9)$$

for all states $s \in \mathcal{S}$ and all directions $d \in \mathcal{D}$ in each intersection $j \in \mathcal{J}$. Additionally, the traffic lights should be in exactly one state at any time $\sum_{s \in \mathcal{S}} \Phi_j^s = 1$.

To represent the presence of vehicle $i \in \mathcal{V}$ in intersection $j \in \mathcal{J}$ at time $t \in \mathbb{Z}_0^N$, we introduce binary variables $\delta_{ij}^d(t) \in \{0, 1\}$. We also introduce a given binary value $a_{ij}^d \in \{0, 1\}$ that indicates if a vehicle $i \in \mathcal{V}$ intends to cross intersection $j \in \mathcal{J}$ in direction $d \in \mathcal{D}$, as part of its future route plan. For each vehicle, we know the relative position at which it enters $P_{i,j}^{\text{in}}$ and exits $P_{i,j}^{\text{out}}$ the intersection. Therefore, we set the variables δ_{ij}^d based on the following bi-conditional

$$p_i(t) \in [P_{i,j}^{\text{in}}, P_{i,j}^{\text{out}}] \iff \sum_d a_{ij}^d \delta_{ij}^d(t) = 1. \quad (4.10)$$

We can implement the above bi-conditional using a big-M formulation for the mixed-integer in-

equalities as

$$\begin{aligned}
p_i(t) &\leq P_{i,j}^{\text{in}} + s_i^\delta(t) + M\delta_{ij}^l(t), \\
p_i(t) &\geq P_{i,j}^{\text{out}} - s_i^\delta(t) - M\delta_{ij}^u(t), \\
\delta_{ij}^l(t) + \delta_{ij}^u(t) &\leq 1 + \sum_d a_{ij}^d \delta_{ij}^d(t), \\
p_i(t) + M\delta_{ij}^d(t) &\leq P_{i,j}^{\text{out}} - s_i^\delta(t) + M, \\
p_i(t) - M\delta_{ij}^d(t) &\geq P_{i,j}^{\text{in}} + s_i^\delta(t) - M,
\end{aligned} \tag{4.11}$$

where $M > 0$ is a large constant value, $\delta_{ij}^l, \delta_{ij}^u \in \{0, 1\}$ are auxiliary binary variables, and $s_i^\delta \geq 0$ denotes a slack variable to ensure feasibility in the presence of a model mismatch.

We can impose an upper bound on the number of vehicles that can be inside the intersection at the same time

$$\sum_i \delta_{ij}^d(t) \leq C_j^d \psi_j^d(t), \quad \forall j, d, \tag{4.12}$$

where C_j^d is the intersection capacity (upper bound based on physical intersection dimensions) for a specific direction $d \in \mathcal{D}$ and ψ_j^d is the traffic light signal for that direction, i.e., $\psi_j^d = 0$ (red) or $\psi_j^d = 1$ (green). Constraint (4.12) additionally ensures that vehicles can only enter the intersection if the corresponding traffic light signal is green.

4.3.4 HDV Modeling and Prediction

Using the HDV model in Section 4.2.3, and depending on which mode is currently active for each HDV $i \in \mathcal{V}^h$, we compute a safe reference velocity $v_i^s(t)$ that is used to predict the HDV trajectories. To model mode switching, we introduce the binary variables $b_i^f(t), b_i^x(t) \in \{0, 1\}$ to

indicate whether an HDV $i \in \mathcal{V}^h$ is in the lead vehicle following mode or stopping for red traffic light mode, respectively.

4.3.4.1 HDV lead vehicle following

For each HDV $i \in \mathcal{V}^h$, we introduce the index $i_l \in \mathcal{V}$ to denote its lead vehicle, if a vehicle exists ahead of the HDV within a predetermined distance from the HDV. This lead vehicle could be either a CAV or another HDV. The safe following distance is defined using a base value D^f and a velocity-dependent component with coefficient k^f . In order to define the binary variable $b_i^f(t)$, we implement the following bi-conditional

$$p_{i_l}(t) + R_{i_l,i} - p_i(t) - L_{i_l} \leq D^f + k^f v_i(t) \iff b_i^f(t) = 1, \quad (4.13)$$

where $R_{i_l,i}$ denotes the initial relative distance between vehicles $i_l \in \mathcal{V}$ and $i \in \mathcal{V}^h$, and given the vehicle length L_{i_l} . The bi-conditional can be enforced by the mixed-integer inequality constraints

$$\begin{aligned} p_{i_l}(t) + R_{i_l,i} - p_i(t) - L_{i_l} &\geq D^f + k^f v_i(t) - M b_i^f(t), \\ p_{i_l}(t) + R_{i_l,i} - p_i(t) - L_{i_l} &\leq D^f + k^f v_i(t) + M(1 - b_i^f(t)). \end{aligned} \quad (4.14)$$

4.3.4.2 HDV stopping at red traffic light

Setting the binary indicator variable $b_i^x(t)$ for stopping at a red light involves both the state of the associated traffic light signal and the set membership of the HDV in the braking region prior to the intersection. We introduce two binary variables b_i^l and b_i^u such that the HDV is in the braking

region when both these variables are 1. For each HDV $i \in \mathcal{V}^h$, we denote the initial distance to the next intersection as P_i^{tl} , and the safe stopping distance is defined using a base value D^{I} and a velocity-dependent component with coefficient k^{I} . We implement the following implications

$$\begin{aligned}
P_i^{\text{tl}} - p_i(t) \leq D^{\text{I}} + k^{\text{I}}v_i(t) &\implies b_i^{\text{l}}(t) = 1, \\
p_i(t) \leq P_i^{\text{tl}} &\implies b_i^{\text{u}}(t) = 1, \\
b_i^{\text{l}}(t) = 1 \wedge b_i^{\text{u}}(t) = 1 \wedge \psi_j^{\text{d}}(t) = 0 &\iff b_i^{\text{x}}(t) = 1,
\end{aligned} \tag{4.15}$$

where $j \in \mathcal{J}$ denotes the next intersection such that the HDV stops if and only if the HDV is inside the braking region and the corresponding traffic light is red. These implications can be enforced by inequality constraints as

$$\begin{aligned}
P_i^{\text{tl}} - p_i &\geq D^{\text{I}} + k^{\text{I}}v_i - Mb_i^{\text{l}}, \\
p_i + Mb_i^{\text{u}} &\geq P_i^{\text{tl}}, \\
b_i^{\text{l}} + b_i^{\text{u}} &\leq 1 + b_i^{\text{x}} + \psi_j^{\text{d}}, \quad \psi_j^{\text{d}} \leq (1 - b_i^{\text{x}}), \\
-M(1 - b_i^{\text{x}}) &\leq P_i^{\text{tl}} - p_i \leq D^{\text{I}} + k^{\text{I}}v_i + M(1 - b_i^{\text{x}}),
\end{aligned} \tag{4.16}$$

where we omit the time index t for compact notation.

4.3.4.3 HDV safe reference velocity

The safe reference velocity v_i^s for HDV $i \in \mathcal{V}^h$ is defined based on the indicators b_i^f and b_i^x , using the following implications

$$\begin{aligned} b_i^x(t) = 1 &\implies v_i^s(t) = 0, \\ b_i^x(t) = 0 \wedge b_i^f(t) = 1 &\implies v_i^s(t) = v_{i_1}(t), \\ b_i^x(t) = 0 \wedge b_i^f(t) = 0 &\implies v_i^s(t) = \bar{v}, \end{aligned} \tag{4.17}$$

where v_{i_1} denotes the velocity of the leading vehicle $i_1 \in \mathcal{V}$ for HDV $i \in \mathcal{V}^h$, and $\bar{v}$ denotes the desired target speed. These implications are enforced by the constraints

$$\begin{aligned} v_i^s &\leq M(1 - b_i^x), \\ -M(1 - b_i^f) - Mb_i^x &\leq v_i^s - v_{i_1} \leq M(1 - b_i^f) + Mb_i^x, \\ -Mb_i^f - Mb_i^x + \bar{v} &\leq v_i^s \leq \bar{v}, \end{aligned} \tag{4.18}$$

where we omit the time index t for compact notation.

4.3.4.4 HDV prediction model

The HDV prediction must respect the HDV's acceleration limits. We introduce the binary variables $b_i^{\max}$ and $b_i^{\min}$ to check whether these limits are reached. The HDV velocity prediction is

given by

$$v_i^s(t+1) - v_i(t) \geq \bar{a}_i^{\max} \Delta t \iff b_i^{\max}(t) = 1, \quad (4.19)$$

$$v_i^s(t+1) - v_i(t) \leq -\bar{a}_i^{\min} \Delta t \iff b_i^{\min}(t) = 1,$$

$$b_i^{\max}(t) = 1 \implies v_i(t+1) = v_i(t) + \bar{a}_i^{\max} \Delta t,$$

$$b_i^{\min}(t) = 1 \implies v_i(t+1) = v_i(t) - \bar{a}_i^{\min} \Delta t, \quad (4.20)$$

$$b_i^{\max}(t) = 0 \wedge b_i^{\min}(t) = 0 \implies v_i(t+1) = v_i^s(t+1).$$

4.3.5 Traffic Light Timing Constraints

We introduce a binary variable $l_{j,s}^{\text{tl}}(t) \in \{0, 1\}$ for each traffic light state $s \in \mathcal{S}$ of each intersection, where $l_{j,s}^{\text{tl}}(t) = 1$ if and only if the traffic light state changes, i.e.,

$$\Phi_j^s(t+1) - \Phi_j^s(t) \leq l_{j,s}^{\text{tl}}(t) \leq \Phi_j^s(t+1) + \Phi_j^s(t), \quad (4.21)$$

$$\Phi_j^s(t) - \Phi_j^s(t+1) \leq l_{j,s}^{\text{tl}}(t) \leq 2 - \Phi_j^s(t+1) - \Phi_j^s(t). \quad (4.22)$$

We introduce a variable $\hat{l}_j^{\text{tl}}(t) \in \{0, 1\}$ to denote if the overall state of the traffic intersection has changed, i.e.,

$$\sum_{s \in \mathcal{S}} l_{j,s}^{\text{tl}}(t) - 1 \leq \hat{l}_j^{\text{tl}}(t) \leq \sum_{s \in \mathcal{S}} l_{j,s}^{\text{tl}}(t). \quad (4.23)$$

Then, depending on the binary variable $\hat{l}_j^{\text{tl}}(t)$, a timer variable $t_j^{\text{tl}}(t)$ can be reset or updated as follows

$$\begin{aligned} -(1 - \hat{l}_j^{\text{tl}}(t))M &\leq t_j^{\text{tl}}(t+1) \leq (1 - \hat{l}_j^{\text{tl}}(t))M, \\ t_j^{\text{tl}}(t) + \Delta t - \hat{l}_j^{\text{tl}}(t)M &\leq t_j^{\text{tl}}(t+1) \leq t_j^{\text{tl}}(t) + \Delta t + \hat{l}_j^{\text{tl}}(t)M. \end{aligned} \quad (4.24)$$

We additionally impose a minimum time between two consecutive traffic light change commands as well as an upper bound for the counter variable as follows

$$t_{\min} - M(1 - \hat{l}_j^{\text{tl}}(t)) \leq t_j^{\text{tl}}(t) \leq t_{\max}. \quad (4.25)$$

4.3.6 MILP Objective Function

The objective function of the proposed MILP formulation for the CTC reads as the maximization of

$$J = \sum_{t=1}^N \sum_{i \in \mathcal{V}} (w_1 p_i(t) - w_2 |\sigma_i(t) - \sigma_i^{\text{ref}}| \quad (4.26a)$$

$$-w_3 l_{i,u}^c(t) - w_3 l_{i,d}^c(t) - w_4 s_i^\delta(t) - w_5 s_i^c(t)), \quad (4.26b)$$

with user-defined weight values $w_i \geq 0$ for $i = 1, \dots, 5$. The first term is to maximize the traveled distance $p_i(t)$ of all vehicles in (4.26a). We penalize vehicles for not being in their preferred lane by minimizing $|\sigma_i(t) - \sigma_i^{\text{ref}}|$, in which the absolute value can be handled by defining auxiliary variables in the MILP formulation. We introduced a penalty on unnecessary lane changes by minimizing $l_{i,u}^c(t), l_{i,d}^c(t)$ in (4.26b). In addition, we penalize the slack variables $s_i^\delta(t), s_i^c(t) \geq 0$

that ensure feasibility in the presence of model mismatch and disturbances. Large penalty weight values $w_4, w_5 \gg 0$ are used to ensure that a solution with $s_i^\delta(t) = 0$ and $s_i^c(t) = 0$ is computed, whenever feasible.

4.4 Decentralized Traffic Coordination

The main concern with the proposed centralized control center is the combinatorial complexity of the MILP solution, i.e., the computational cost generally increases exponentially with the number of vehicles and the prediction horizon length. As shown in Table 4.2, at high densities (an inflow rate r_{in} of 1.44 veh/s), the average computation time can easily exceed 60 sec. The inflow refers to the total number of vehicles that enter the road network in Fig. 4.2 from north, east, south, and west. However, since our desired update frequency is 1 Hz, the centralized system is not tractable under current hardware/processing limitations and with current state-of-the-art MILP solvers such as GUROBI [23].

Traffic Inflow Rate r_{in} [veh/s]	Mean CPU Time [sec]
Low (0.36)	0.21
Medium (0.72)	1.37
High (1.44)	64.12

Table 4.2: Centralized computation cost using GUROBI, with varying traffic inflow rates and $N = 10$, $r_{\text{cav}} = 67\%$.

In order to gain computational tractability, a natural way to decompose the centralized coordination problem is by solving a sub-problem for each traffic intersection individually with considerably fewer optimization variables. This decentralized implementation will be described next.

4.4.1 Decentralized Architecture

In the proposed decentralized implementation, each intersection is assumed to have an independently operating control center, which is further referred to as an intersection traffic controller (ITC). The ITC at each intersection operates fundamentally similar to the centralized control center, based on the same MILP formulation as described in Section 4.3, but with each of the variables restricted to only the relevant intersection, its traffic lights and immediately approaching vehicles within a predetermined *control zone* around the traffic intersection. Each ITC computes CAV velocities and lane change trajectories as well as the traffic light phase switching trajectories for its particular intersection, and the computations for each ITC may be executed in parallel on separate computing units. Adapting the MILP formulation from Section 4.3 to the decentralized implementation, for each ITC, we consider one intersection $j \in \mathcal{J}$ and a subset $\mathcal{V}_j \subseteq \mathcal{V}$ of the vehicles in the traffic network.

4.4.2 Vehicle Assignment to MILP-ITCs

In the decentralized implementation, the HDV modeling described in Section 4.3.4 can be used to model both HDVs and CAVs that are controlled by a different ITC. Therefore, each CAV could be assigned to a particular ITC “for control” or “for prediction”. This is used in the proposed decentralized implementation to increase safety, especially when vehicles transition from one ITC to the next. The assignment of vehicles to the subsets $\mathcal{V}_j$, for each ITC $j \in \mathcal{J}$, is performed according to the following rules:

1. Each HDV is assigned to its next intersection’s ITC for prediction if it is within a predeter-

mined distance.

2. Each CAV is assigned to its next intersection's ITC for control if it is within a predetermined distance.
3. Each CAV is assigned to its previous intersection's ITC for prediction, i.e., it will be treated as a HDV by that MILP-ITC if it is within a predetermined distance.

The latter set of rules ensures that the velocity and lane change commands for each CAV are computed by at most one ITC. If a CAV is not assigned to any ITC, given the predetermined threshold distance, then control is returned to the onboard control architecture. In the region immediately after a traffic intersection, both HDVs and CAVs are treated equally and their behavior is predicted to prevent possible collisions with any vehicles further upstream. Each ITC provides controls to CAVs only prior to the intersection and while the vehicle is physically within the intersection. Control is handed back to the vehicle or to the next junction's ITC (if junctions are very close) after the CAVs exit the intersection.

4.5 Numerical validation of results with simulations

In this Section, we present extensive simulation results to validate the performance of the proposed implementations for the traffic coordination and control system, and to investigate its computational tractability. The mixed traffic simulations are built on the SUMO [22] platform. The control systems are implemented in `Matlab` and the MILP problems are solved using `GUROBI` [23]. The communication between `Matlab` and SUMO is handled by the `traci4matlab` [70] software. The simulations are executed on a desktop with an Intel i7-10700k processor and 64GB of RAM.

4.5.1 Traffic Simulation

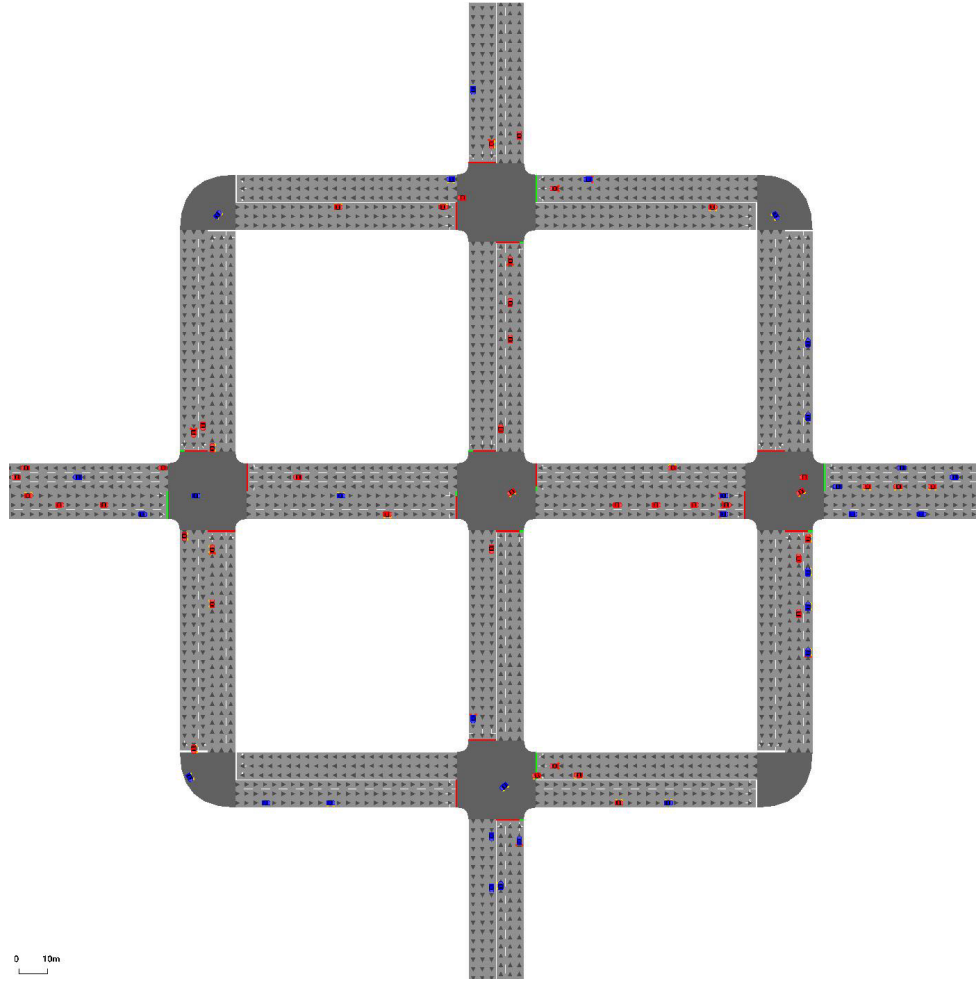


Figure 4.4: Traffic road network with 5 signalized intersections in SUMO. Blue vehicles are HDVs and red vehicles are CAVs.

In the conducted experiments, the traffic inflow from each of the four input directions (see Fig. 4.2) are kept equal to each other, resulting in a balanced flow of vehicles in the road network. The rate of traffic inflow for CAVs and HDVs is varied independently, in order to create multiple scenarios with varying traffic inflow rates r_{in} and CAV penetration levels r_{cav} (ratio of CAVs to the total number of vehicles). Vehicles enter the simulation according to a Poisson process with an

expected value of arrivals equal to the desired input flow rate for each inflow road. Each vehicle entering the network has an assigned destination road and the system chooses between possible routes from source to destination based on predefined probabilities to provide a balanced traffic flow throughout the network. In the simulated vehicle model of SUMO [22], parameters for the level of driver imperfection are set to low for CAVs and relatively high for HDVs. This results in lower predictability of HDV behavior due to higher variance in human actions. Our simulation setup with 5 active intersections is shown in figure 4.4.

4.5.2 Comparison Metrics and Baselines

The key metrics used in evaluating comparable traffic control systems are the average travel time of vehicles flowing through the network, the average time spent waiting at traffic lights per vehicle, and the average fuel consumption per vehicle. In addition to these metrics, we also present the average computation times (τ) for each of the proposed control architectures. The runtime (τ_{ITC}) of the decentralized architecture involves computing multiple MILPs in parallel, one for each intersection's ITC. We report both the average over time of the CPU runtime for the most expensive MILP at each time step (dark blue bar in Figures 4.5,4.6,4.7), as well as the average over time of cumulative CPU runtimes to solve all MILPs at each time step (light blue bar in Figures 4.5,4.6,4.7),

$$\tau_{\text{ITC}}^{\max} = \frac{1}{T} \sum_{t=1}^T \max_{j \in \mathcal{J}} \tau_{\text{ITC}}^j(t), \quad \tau_{\text{ITC}}^{\text{sum}} = \frac{1}{T} \sum_{t=1}^T \sum_{j \in \mathcal{J}} \tau_{\text{ITC}}^j(t). \quad (4.27)$$

As a reference, we use the standard baseline of a time-based traffic lights switching system

referred to as *Timed traffic lights*. The traffic light timings are tuned to suit the densities being tested. All four inflow roads in an intersection are treated equally in terms of traffic light timing since the overall inflow to the network is balanced. Furthermore, to provide a more challenging benchmark, we implemented a gap-based adaptive traffic light control strategy [55]. This adaptive strategy allows traffic lights to dynamically adapt to current traffic conditions in the incoming lanes of an intersection by prolonging traffic phases whenever a continuous stream of traffic is detected. This is a strategy that requires the traffic light controller to have a clear understanding of the input traffic flow into the intersection.

4.5.3 Performance analysis

The overall performance of the systems shows that both centralized and decentralized control strategies significantly outperform the timed traffic lights. We observed considerable improvements of around 60%, 96%, and 60% in travel time, waiting time, and fuel consumption, respectively, when using the centralized strategy over the timed traffic lights method. Similarly, when comparing the centralized strategy against the adaptive traffic lights method, we observed improvements of around 32%, 90%, and 40% in travel time, waiting time, and fuel consumption, respectively. These significant reductions in waiting times correspond to reduced idling of vehicles at traffic lights, which leads to more eco-driving strategies, which in turn results in fuel savings.

The centralized strategy also outperforms the decentralized strategy in most metrics by a smaller margin, at the cost of increased computation times. We observed a reduction in performance of around 2%, 3%, and 9% in travel time, waiting time, and fuel consumption, respectively,

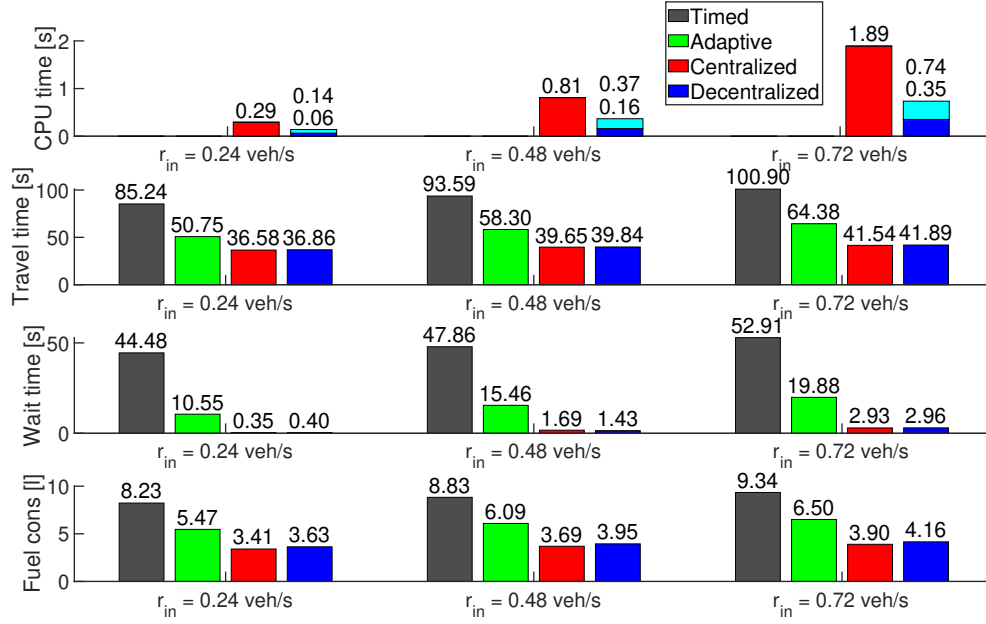


Figure 4.5: SUMO results over 2000 s to compare centralized MILP, decentralized MILPs versus timed and adaptive traffic lights for increasing inflow rates $r_{in} = 0.24, 0.48$ and 0.72 veh/s, with $N = 10, r_{cav} = 50\%$.

when using the decentralized strategy over the centralized strategy. However, the decentralized strategy CPU computation times are around 89% less than the centralized strategy.

4.5.3.1 Impact of vehicle inflow rate

The total number of vehicles present in the network is directly proportional to the inflow rate r_{in} of vehicles into the network. As the inflow rate increases, as observed in Figure 4.5, the travel time, waiting time, and fuel consumption all show worse performance, across all the control strategies. This is expected since intersections have a maximum capacity and therefore some vehicles will be forced to wait when the density is too high. However, even at high inflow rates that cause the timed traffic lights to have 100 sec average travel time and 53 sec average waiting time, the CTC reduces these values to 42 sec and 3 sec, respectively. Even compared to the

adaptive traffic lights, the CTC demonstrates a 36%, 85%, and 40% improvement in travel time, waiting time, and fuel consumption. The key issue with the CTC method is that at higher inflow rates, long computation times exceeding 2 sec are experienced, which does not allow us to reach the target control frequency of 1 Hz. In comparison, we observe that the decentralized ITCs allow for fast computations below 1 sec, while only resulting in around a 5% suboptimality in terms of other key performance metrics.

4.5.3.2 Impact of CAV penetration

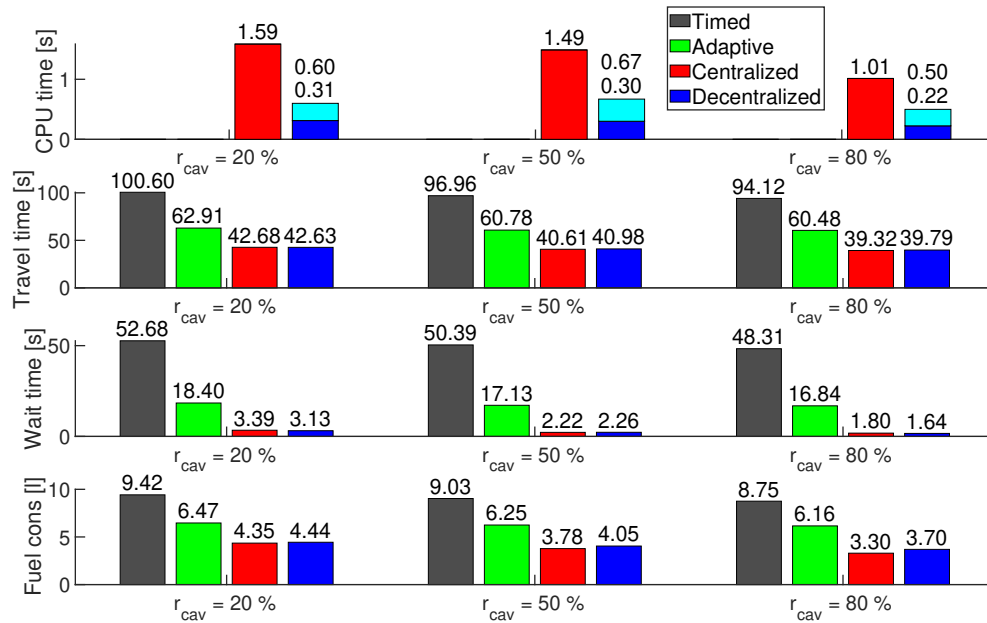


Figure 4.6: SUMO results over 2000 s to compare centralized MILP, decentralized MILPs versus timed and adaptive traffic lights for increasing CAV penetration levels $r_{cav} = 20, 50$ and 80% , with $N = 10, r_{in} = 0.60$ veh/s.

The percentage of CAVs plays a major role in the performance of the control system. A greater amount of CAVs offers multiple actuation points in the traffic flow which leads to better controllability of the overall traffic network. For example, as observed in Figure 4.6, using a

$N = 10$ planning horizon and 0.6 veh/sec inflow rate, in the CTC we observe performance gains of 8%, 46% and 24% in travel time, waiting time and fuel consumption, respectively, when switching from 20% to 80% CAV penetration. Furthermore, we also observe that the prediction of HDVs is a more computationally expensive process than the CAV trajectory computation, due to the increased number of integer variables involved. Thereby, the average computation times reduce by 36%, when switching from 20% to 80% CAV penetration.

4.5.3.3 Impact of planning horizon

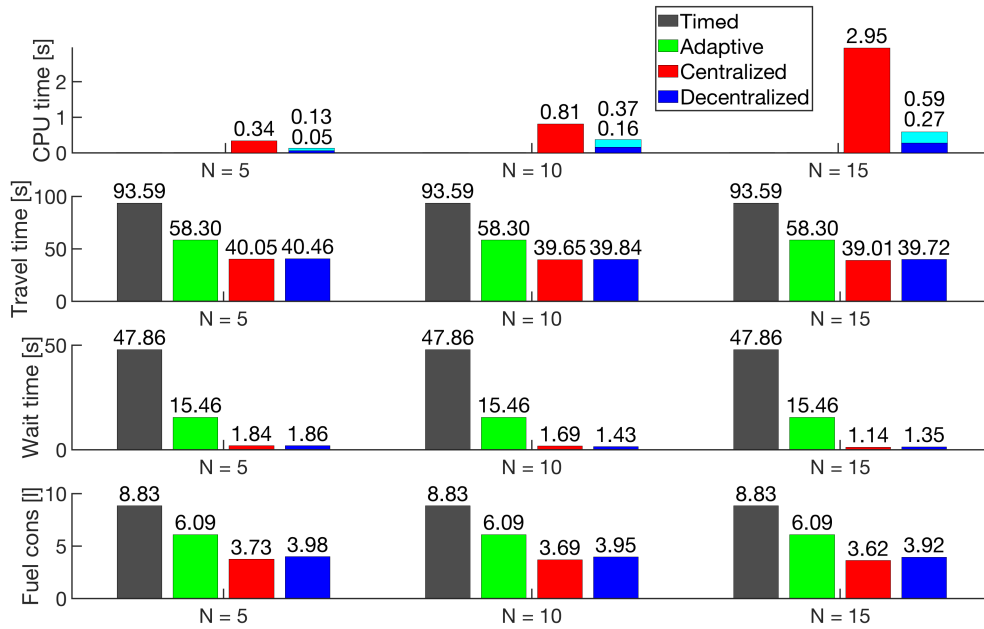


Figure 4.7: SUMO results over 2000 s to compare centralized MILP, decentralized MILP versus timed and adaptive traffic lights for increasing horizon lengths $N = 5, 10, 15$, with $r_{in} = 0.48$ veh/s, $r_{cav} = 50\%$.

The planning horizon length of the MILP problem is a major factor in the performance-computation trade-off in intersection traffic control. We observe that longer planning horizons lead to improved performance in terms of the key metrics but also cause the computation time

to increase rapidly. As shown in Figure 4.7, at $N = 5$ horizon length, the CTC is capable of computing on average in about 0.3 sec. When the horizon length is increased to $N = 15$, the average computation time increases significantly to around 3 sec. This is where the decentralized strategy with multiple ITCs becomes more suitable. ITCs offer similar performance improvements obtained by increasing horizon length while keeping total computation times under 0.5 sec, even at the $N = 15$ horizon length.

4.5.3.4 Real-time feasibility

While computations for the decentralized system are significantly faster than for the centralized system, a drop in overall system performance was observed. The combined output of each ITC in the decentralized system is sub-optimal when compared to the solution provided by the CTC. This performance drop can be primarily attributed to the lack of global knowledge in each ITC. To achieve real-time feasibility at a 1 Hz control frequency, the CTC/ITC needs to solve its MILP problem in less than 1 sec. The CTC is able to achieve this threshold only at low traffic inflow rates and shorter planning horizons. However, at higher inflow rates and planning horizons, the CTC's average computation time can easily exceed 10 sec. The average runtime of the ITCs remain below 0.8 sec, making the decentralized strategy a more practically implementable real-time system under current hardware limitations.

4.6 Hierarchical Traffic Coordination extension

In section 4.4 related to the work in [27], we introduced a decentralized version with multiple ITCs that could solve the distributed optimization problem within the desired 1 sec control period thus enabling real-time operation. The disadvantage of this decentralized architecture was the sub-optimality introduced in terms of overall system performance when compared to the central controller. We find that this sub-optimality is primarily caused by the limited visibility of the overall state space each ITC has when operating independently of each other. To bridge this optimality gap we introduce a novel hierarchical control center architecture in which an additional central agent provides global-level insights to the individual ITCs. This proposed hierarchical control center consists of two major components. ITCs at each intersection (described in section 4.4) and a Central Traffic Light Command (CTLTC) which has a macro-level understanding of global traffic conditions.

The overall goal of the proposed CTLTC is to provide suggestions for the importance of each traffic direction (probability of use) in each intersection. Each intersection has 12 traffic light phase directions. The CTLTC should provide the probability that each of these phase directions will be utilized within the control horizon. Therefore the CTLTC will compute the optimal usage probabilities P_j^{d*} for each direction in each intersection $j \in \mathcal{J}$ which will be sent to the corresponding ITC.

The ITC of an intersection will receive these probabilities and attempt to achieve these probabilities over its control horizon. This is implemented as an additional term in the objective cost function of the ITC MPC. Note that the ITC is able to stray from the CTLTC suggestions if it leads

to an overall better objective value.

Due to the simplicity of the CTLC formulation, it is often possible to compute CTLC probabilities over a longer horizon without a noticeable negative impact on computation cost. This allows for a multi-time scale approach in which the CTLC will accumulate and predict global traffic flow information for a longer horizon while the ITCs can leverage this information for better performance while only having to compute over a shorter control horizon.

4.6.1 Heuristic central traffic light command (H-CTLC)

The first iteration of the CTLC involves a simple heuristic-based method for identifying traffic flow demand through each direction of the intersection. This method uses the one or two-hop future route information of the vehicles in the network. The steps in the heuristic-CTLC (H-CTLC) are as follows:

1. Obtain future route information from each vehicle in the network. (This information contains the next one or two junctions and the expected direction through the junctions)
2. For each direction in each intersection, sum up the number of times the intersection-direction (j, d) pair would be used by each vehicle (U_j^d)
3. Obtain the total usage of each intersection by summing up the total usage of each direction in that intersection (U_j)
4. Compute the probability of use of each intersection-direction (j, d) pair as $P_j^d = U_j^d / U_j$
5. Provide usage probabilities P_j^d to each ITC

4.6.2 Optimization central traffic light command (O-CTLC)

In order to improve the performance of the CTLC we introduce an optimization-based version of the CTLC (O-CTLC). O-CTLC builds a macro-traffic level understanding of the network traffic flow and computes the optimal usage probabilities P_j^d for each intersection-direction (j, d) pair.

The main steps in the Optimization-CTLC (O-CTLC) are as follows:

1. Build a macro-level traffic network structure using the network parameters such as road segment length, number of lanes, road segment connections and intersection connectivity, etc.
2. For each road segment r , define the possible future maneuvers $w \in \{\text{go straight, turn right, turn left}\}$
3. Compute input probabilities of usage P_r^w of each road segment-maneuver (r, w) pair. This can be computed using either historical data and domain knowledge or using the usage probabilities P_j^d of each junction computed as shown in H-CTLC. Computing P_r^w using historical data involves collecting the usage of each road and maneuver over a large enough time to obtain a reasonable expectation for ground truth values
4. Collect input information (initial state values) for optimization which includes initial vehicle positions and velocities
5. Run the optimization to compute the optimal usage probabilities P_j^{d*} for each intersection-direction (j, d) pair

6. Provide optimal usage probabilities P_j^{d*} to each ITC

4.6.2.1 O-CTLC formulation

O-CTLC input:

- P_r^w (Probabilities of usage of each road segment-maneuver (r, w) pair)
- Flow buffer (Filled based on vehicle positions)
- z_r^w (Number of vehicles in each road segment-maneuver (r, w) pair)
- Additional tuning parameters

O-CTLC output:

- P_j^{d*} (Optimal usage probabilities for each intersection-direction (j, d) pair)

The traffic model in O-CTLC can be described as a directed graph, see Figure 4.8, where each node corresponds to a road segment and each edge corresponds to a connection between road segments, i.e., a direction to cross through a traffic intersection in the transportation network. Let us describe each of the optimization variables in O-CTLC:

- $z_{r,w}(t) \in \mathbb{R}$: number of vehicles (traffic density) for each road segment-maneuver (r, w) pair at each time step $t \in [0, N]$
- $f_{r,w}^{\text{in}}(t) \in \mathbb{R}$: number of vehicles **entering** (in-flow) for each road segment-maneuver (r, w) pair at each time step $t \in [0, N - 1]$

- $f_{r,w}^{\text{out}}(t) \in \mathbb{R}$: number of vehicles **exiting** (out-flow) for each road segment-maneuver (r, w) pair at each time step $t \in [0, N - 1]$
- $e_{r,s}(t) \in \mathbb{R}$: number of vehicles transitioning (along edge in directed graph) from each road segment r to each road segment s at each time step $t \in [0, N - 1]$

4.6.2.2 O-CTLC constraints

Given these variables, the state dynamics for the macro-level traffic model then read as

$$z_{r,w}(t + 1) = z_{r,w}(t) + f_{r,w}^{\text{in}}(t) - f_{r,w}^{\text{out}}(t), \quad \forall t \in [0, N - 1], \forall r, w, \quad (4.28)$$

given the initial state values $z_{r,w}(0) = \hat{z}_{r,w}$ and given the in- and out-flow variables. The in- and out-flow variables are then defined by the following additional inequality constraints

- Edge maximum flow inequality constraints:

$$e_{r,s}(t) \leq A(r, s) f^{\text{max}}, \quad (4.29)$$

where f^{max} is the maximum flow and A is the adjacency matrix for the directed graph, i.e., $A(r, s) = 1$ if road segments r and s are connected, see Fig. 4.8, otherwise $A(r, s) = 0$.

- Flow output for each road segment is upper bounded by the number of vehicles in the road segment, for the particular maneuver $w \in \{\text{go straight, turn right, turn left}\}$, minus the number of vehicles that entered the road segment within a particular time window. The latter ensures time delay constraints for each road segment corresponding to the expected duration

for each vehicle to travel through the road segment, given the physical length of the road segment and the expected velocity for the vehicles. This corresponds to the following inequality constraints:

$$f_{r,w}^{\text{out}}(t) \leq z_{r,w}(t) - \sum_{i=1}^{\tau(r)} f_{r,w}^{\text{in}}(t-i), \quad (4.30)$$

where $\tau(r)$ denotes the expected time delay for vehicles traveling through road segment r and the in-flow values $f_{r,w}^{\text{in}}(t)$ are given for $-\tau(r) \leq t \leq -1$.

- Equality constraints for edge variables, given out-flow variables:

$$f_{r,w}^{\text{out}}(t) = e_{r,s}(t), \quad (4.31)$$

where transition from r to s corresponds to maneuver w . **Note that the latter equality constraints can be used by the solver to completely eliminate the edge variables from the optimization problem.** However, we use these edge variables for ease of notation.

- Equality constraints for in-flow variables:

$$f_{r,w}^{\text{in}}(t) = P_r^w \left(f_r^{\text{ext}}(t) + \sum_s e_{s,r}(t) \right) \quad (4.32)$$

where $f_r^{\text{ext}}(t)$ denotes the predicted external in-flow values and P_r^w denote the estimated probability for vehicles in road segment r to perform maneuver $w \in \{\text{go straight, turn right, turn left}\}$.

- Traffic intersection capacity limit constraints:

$$\sum_{(r,s) \in \mathcal{E}(j)} e_{r,s}(t) \leq z_j^{\max}, \quad (4.33)$$

where $\mathcal{E}(j)$ denotes the edges that belong to traffic intersection $j \in \mathcal{J}$.

4.6.2.3 O-CTLC convex linear programming

The objective of the optimization problem in O-CTLC can include the maximization of the traffic flow and the minimization of congestion, leading to the convex linear programming (LP) problem

$$\min_{Z, E, F} \quad \sum_{t=0}^N w_1 c(t) - \sum_{t=0}^{N-1} \sum_{r \in \mathcal{S}} \sum_{w \in \mathcal{W}} w_2 f_{r,w}^{\text{out}}(t) \quad (4.34a)$$

$$\text{s.t.} \quad \text{Macro-level traffic model: Eq. (4.28),} \quad (4.34b)$$

$$\text{Edge inequality constraints: Eq. (4.29),} \quad (4.34c)$$

$$\text{Out-flow bounds and delay constraints: Eq. (4.30),} \quad (4.34d)$$

$$\text{Out-flow and edge equality constraints: Eq. (4.31),} \quad (4.34e)$$

$$\text{In-flow and edge equality constraints: Eq. (4.32),} \quad (4.34f)$$

$$\text{Traffic intersection capacity limits: Eq. (4.33).} \quad (4.34g)$$

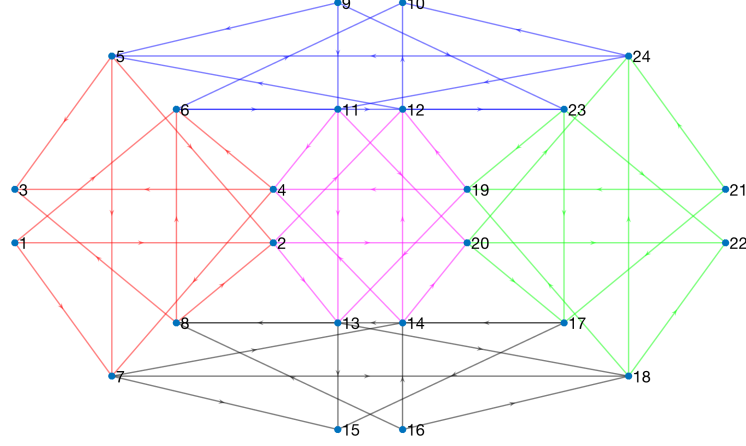


Figure 4.8: Directed graph representation for optimization-based CTLC (O-CTLC) of the transportation network.

given weight values $w_1 \geq 0$, $w_2 \geq 0$, and where the congestion variables $c(t)$ are defined by the following inequality constraints

$$c(t) \geq \frac{\sum_{w \in \mathcal{W}} z_{r,w}(t)}{\text{dist}(r)}, \quad \forall r \in \mathcal{S}, \quad (4.35)$$

where $\text{dist}(r)$ is the distance of road segment $r \in \mathcal{S}$.

4.6.3 ITC cost function

The ITC cost function is modified to penalize deviation from the CTLC probability of use per direction plan. The usage of each direction of the intersection is tracked using the δ intersection membership variable. (Note that since ITC only applies to one intersection $j = 1$). The following additional term which should be minimized needs to be added to the cost function;

$$J_{op} = \sum_{d \in \mathcal{D}} \left\| \sum_{t=1}^N \sum_{i \in \mathcal{A}} \left(\delta_i^d(t) - P^d \sum_{d \in \mathcal{D}} \delta_i^d(t) \right) \right\| \quad (4.36)$$

Here, P^d is the probability of usage per direction provided by the CTLC. To address the absolute value term in the objective function we introduce a new variable J_d^δ for each direction d . Accordingly, the objective function and corresponding additional constraints are added as follows:

$$J_{op} = \sum_{d \in \mathcal{D}} J_d^\delta \quad (4.37)$$

$$\sum_{t=1}^N \sum_{i \in \mathcal{A}} \delta_i^d(t) - P^d \sum_{t=1}^N \sum_{i \in \mathcal{A}} \sum_{d \in \mathcal{D}} \delta_i^d(t) \leq J_d^\delta \quad (4.38)$$

$$P^d \sum_{t=1}^N \sum_{i \in \mathcal{A}} \sum_{d \in \mathcal{D}} \delta_i^d(t) - \sum_{t=1}^N \sum_{i \in \mathcal{A}} \delta_i^d(t) \leq J_d^\delta \quad (4.39)$$

Therefore the final objective function given in equation 4.26 will be augmented as follows with a suitable value for the new weight coefficient (w_6).

$$J = w_1 \cdot J_p - w_2 \cdot J_l - w_3 \cdot J_c - w_4 \cdot J_{s,\delta} - w_5 \cdot J_{s,c} - w_6 \cdot J_{op} \quad (4.40)$$

4.6.4 Results and numerical validation of Hierarchical Traffic Coordination

The proposed hierarchical traffic coordination extension provides a way to bridge the gap in performance between centralized and decentralized coordination strategies while achieving significantly lower computation times than the centralized approach. Similar to the centralized and decentralized coordination strategies, we test the hierarchical traffic coordination strategy using the SUMO traffic simulator over 30 minutes of simulation time. We will explore both the perfor-

mance improvements as well as the changes in computation time brought about by the proposed hierarchical approach.

4.6.4.1 Traffic-related performance of hierarchical approach

As in our previous work [27], we compare our approaches using the key metrics of (a) Average travel time of vehicles, (b) Average waiting time of vehicles, and (c) Average fuel consumption of vehicles. We also compare against the baselines of timed traffic lights in which traffic signal switching occurs based on timings tuned to suit input traffic patterns, as well as an actuated traffic light system that uses a gap-based adaptive traffic light control strategy [55]. The combined results of our performance tests are shown in figure 4.9.

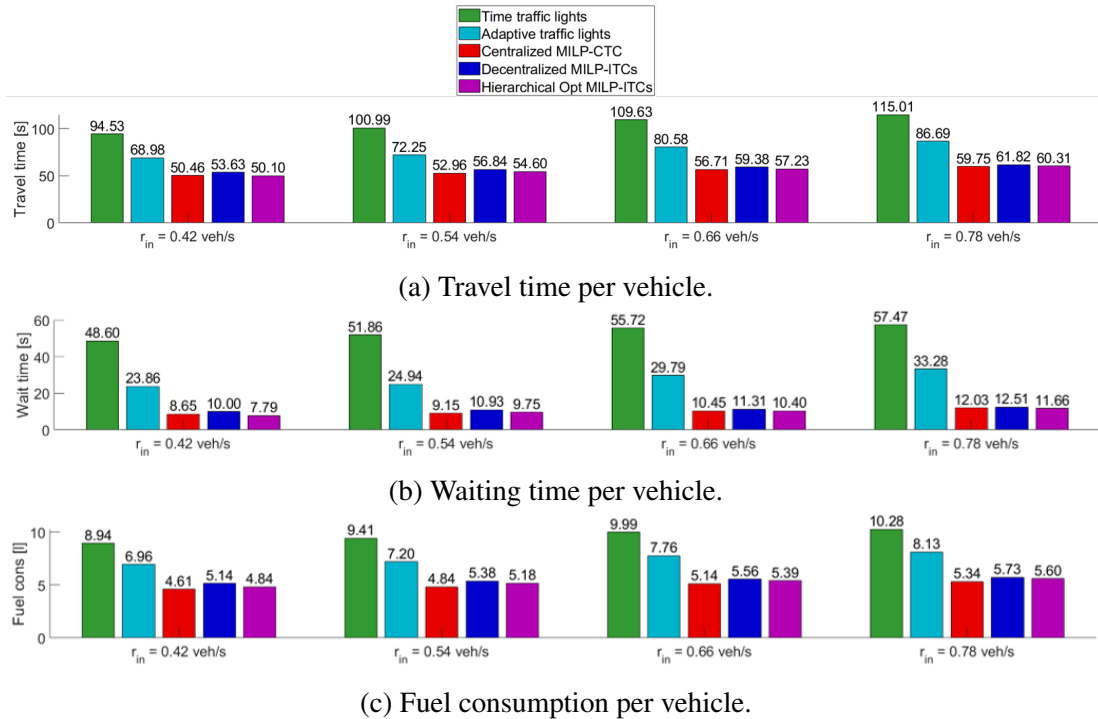


Figure 4.9: SUMO results over 2000s for increasing inflow rates $r_{in} = 0.42, 0.54, 0.66$ and 0.78 veh/s.

As shown in figure 4.9a, we observe around a 45-50% improvement over timed traffic lights and a 25-35% improvement over actuated traffic lights, with higher improvements observed at higher input flow rates. Note that the input flow rates are directly proportional to the amount of traffic in the network at a given time. The key observation to note is that the performance drop in travel times in comparison to the centralized method, is only about 1% for the hierarchical approach, while the decentralized approach suffers from a drop of around 5%. We observe the same phenomena in figure 4.9b, which shows the performance drop in waiting times in comparison to the centralized method, is less than 1% for the hierarchical approach, while the decentralized approach suffers from a drop of around 10-20%. In fact, in many cases, we observe that the hierarchical can outperform the centralized approach in terms of waiting times. This can be attributed to the fact that while the centralized controller, as well as the ITCs in the hierarchical method, are limited to a $N = 10$ horizon length in order to achieve realistic computation times, the CTLC component of the hierarchical method is allowed a $N = 20$ horizon length. This is possible since the CTLC which uses macro-level traffic data solves a convex optimization problem whose solution can be computed much faster than the MILP problems in the ITC and CTC. Thus the hierarchical method can plan further into the future in order to achieve a better result in terms of vehicle waiting times. Finally, as shown in figure 4.9c, we observe that the performance drop in fuel consumption efficiency in comparison to the centralized method, is around 5% for the hierarchical approach, while the decentralized approach suffers from a drop of around 10%. Overall we prove that the hierarchical approach is capable of bridging the optimality gap between the centralized and decentralized approaches.

4.6.4.2 Computation time of hierarchical approach

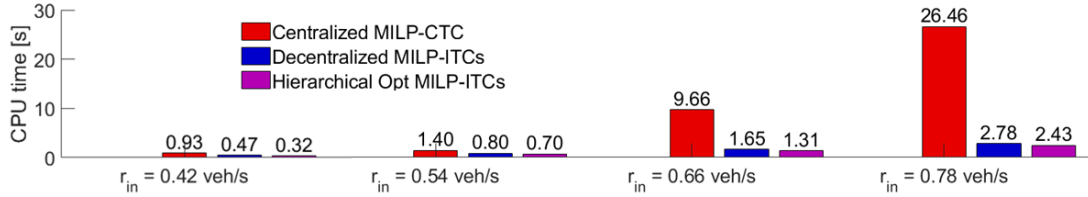


Figure 4.10: Computation time comparison for increasing inflow rates $r_{in} = 0.42, 0.54, 0.66$ and 0.78 veh/s.

The key requirement of the hierarchical approach is that it should be able to achieve computation times on par with the decentralized method while achieving traffic-related performance closer to the centralized approach. As shown in figure 4.10, we observe that the hierarchical approach achieves around 60-90% improvements in computation times in comparison to the centralized approach with greater improvements at higher input flow rates. The most interesting observation here is the fact that the hierarchical approach was actually able to surpass the computation times of even the decentralized approach by around 10-25%. We can attribute this improvement to the fact that the additional constraints introduced on the MILP problem at each ITC, by the CTLC in the hierarchical approach, lead to the individual MILP problems being solved faster. While not necessarily true in general, we expect that in this case, the added constraints help reduce the feasible region of the MILP problem thus allowing the solver to terminate faster due to the lower search effort needed. While at high input flow rates, we observe that even the hierarchical system slightly exceeds the target computation time of 1 second, it is important to note that these times were obtained on a medium-performance desktop computer. We believe that on a high-performance custom compute unit the system would realistically be implemented on, the computation time of the hierarchical

implementation should be comfortably below the 1 second requirement.

4.7 Conclusions

We presented a MILP approach for the computation of control trajectories for CAVs and traffic light signals in a network of multiple connected intersections with mixed traffic, including both CAVs and HDVs. Our proposed centralized traffic controller demonstrated a considerable increase in performance, with around 90-96%, 32-60%, and 40-60% improvements in wait times, travel times, and fuel consumption, respectively, over both adaptive and timed traffic light control methods. We also introduced a decentralized implementation, with multiple decoupled MILPs for each intersection ITC, which results in a relatively small suboptimality while remaining computationally tractable for higher traffic densities and longer planning horizons. To further reduce this suboptimality we introduced a hierarchical implementation that adds a central traffic light command (CTLIC) that uses macro-level traffic information to predict future network usage probabilities to feed to each intersection ITC. The resulting hierarchical implementation was shown to have traffic performance very similar to the centralized implementation while even outperforming the decentralized implementation in terms of computation times. The proposed algorithms were extensively tested in a multi-lane multi-intersection environment using SUMO traffic simulations, for varying traffic densities and CAV penetration levels. Future work may involve reducing the optimality gap in the decentralized method by providing global information on traffic conditions.

Chapter 5: Shock Wave Mitigation in Multi-lane Highways using Vehicle-to-Vehicle Communication

5.1 Evolution of Methodologies

The problem of traffic shock wave generation and mitigation was studied in field experiments conducted by Sugiyama *et al.* [71] and Stern *et al.* [72]. Here, vehicles were placed in a single file loop, and in [72] one ego vehicle was used to apply control for shock wave dissipation. The use of vehicle platoons to help reduce shock waves in single-lane roads was studied in [73]. An optimal control approach to this problem is explored in [74] which allows for multiple ego vehicles on a single lane ring-road, but requires the assumption that ego vehicles have access to the global traffic state. While these approaches yield effective results on single-lane roads, on multi-lane highways the control strategies need to be revised. The use of V2V communication for shock wave suppression is considered in [75], which explores the idea of changing driving parameters when shock waves are detected downstream.

Contribution

We explore the complications involved in shock wave mitigation in multi-lane highway environments, with a focus on the use of V2V communication to ameliorate this problem. We propose a communication-based cooperative control method for shock wave dissipation in a mixed traffic scenario. This method takes proactive control actions when compared to most methods which are reactive to the presence of shock waves. Here the CAVs, dispersed among other human driven vehicles (HDVs), act as the sensing and control agents. Our method was tested and its performance was validated using the realistic SUMO traffic simulation system.

5.2 Problem overview

Shock waves are one of the major preventable causes of traffic buildup in highway networks. Once a shock wave forms, if left unchecked, it usually resolves only when the demand (vehicles entering the highway) reduces. As such, in high-density periods, shock waves can last for hours on end. The major issue in dealing with shock waves lies in the difficulties in sensing accurate highway conditions and subsequently actuating vehicles in a suitable manner to dissipate the shock wave. Traditionally, these were carried out by infrastructure-based sensing and infrastructure-based actuation such as variable speed limits.

However, with the advent of CAVs, many more options for sensing, actuation, and control open up. Even at low percentages of CAVs to HDVs (hereafter referred to as CAV penetration levels), the CAVs can act as Lagrangian sensors, collecting local information about the state of the highway traffic. This local information can then be shared and combined in order to get an accurate

picture of the state of traffic. Furthermore, CAVs can also act as Lagrangian actuators, allowing the control of themselves and their surrounding vehicles via inter-vehicle interactions. Thus, the slowing down and speeding up of human driven vehicles can be achieved by suitable actuation of the CAVs dispersed among the HDVs.

In order to find methods that can resolve this issue, the first step is the modeling and recreation of a realistic highway network with realistic driver behaviors.

A three-lane highway is selected in order to accurately represent the effects of vehicle lane changing on shock wave creation and dissipation. The resulting simulation is more realistic, highlights the effects of interactions between vehicles, and provides a more accurate representation of the conditions faced by vehicles under highway shock wave conditions.

5.3 Modeling the network and vehicles

In this section, we model the highway infrastructure used in this research and the characteristics of the autonomous vehicles used as both sensors and actuators for shock wave dissipation. A section of the highway model is shown in figure 5.1, in which lane structure, HDVs, CAVs, and CAV sensor ranges are highlighted.

5.3.1 Modeling the physical highway structure

The highway stretch is modeled in the form of a loop. This design decision was made in order to simulate an infinite stretch of a multi-lane highway. We set the length of the highway loop to 1 km and the number of lanes to 3. The length of the loop and the number of lanes are both

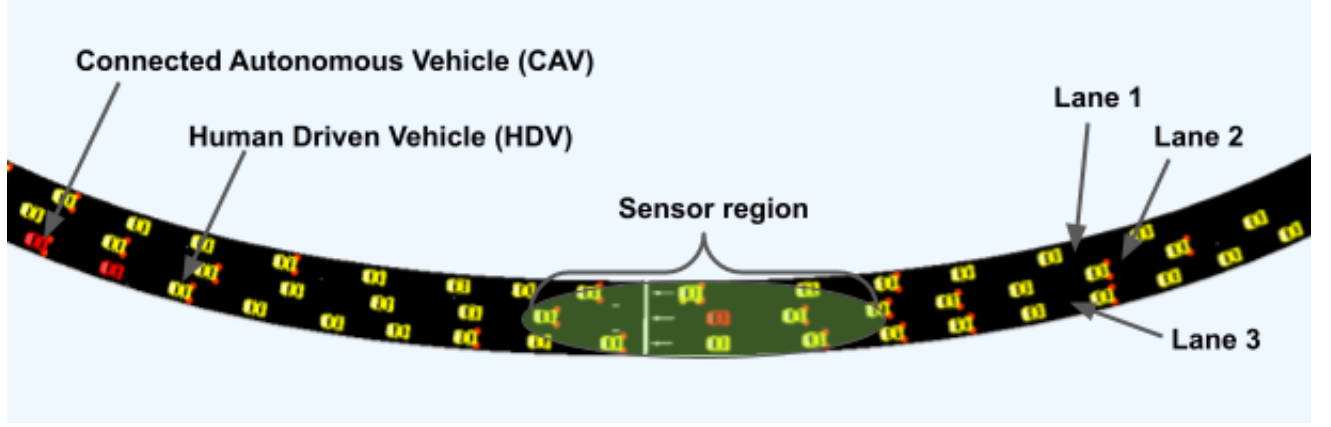


Figure 5.1: Modeling a highway section including vehicles

parameters that can be modified. It should also be possible to change the number of vehicles on this highway loop as necessary.

5.3.2 Vehicle Model

Two different types of vehicles are used in this research. Human driven vehicles whose motion is modeled according to section 5.4, and CAVs modeled as follows.

For the high-level controller proposed for shock wave mitigation, we find that it is unnecessary to consider the highly non-linear nature of the dynamics of real-world vehicles. In this research, we assume that the low-level control of each CAV is managed by a local controller c_i , which is also responsible for the control of lateral motion that keeps the vehicle in the lane. This allows the i^{th} vehicle to be modeled as a point object moving along the center of the lane according to a non-linear differential equation:

$$\dot{s}_i = f(t, s_i, c_i), \quad s_i(t_i^0) = s_i^0 \quad (5.1)$$

where t^0 is the initial time the vehicle enters the highway. Therefore, we can define the high-level longitudinal vehicle dynamics by the following velocity control scheme:

$$\begin{aligned}\dot{s}_i &= v_i \\ v_i(t) &= u_i(t)\end{aligned}\tag{5.2}$$

where $s_i(t)$, and $v_i(t)$ denote the position and velocity of each vehicle i respectively, along the direction of the lane, for $i \in \{1, \dots, n\}$. n is the number of CAVs on the modeled highway stretch. Here, 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 system (5.1). 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.

Each CAV i , is assigned an integer variable $b^i \in \{1, \dots, m\}$ which signifies which lane it is on, with m denoting the number of lanes. It also has a length l^i parameter, and bounded acceleration capabilities characterized by its maximum acceleration a_{max} and maximum braking a_{min} capability. Therefore, all the CAVs are completely defined by the state vectors:

$$x_i(t) = [s_i(t), v_i(t), b^i, l^i, a_{max}^i, a_{min}^i]^T\tag{5.3}$$

for $i \in \{1, \dots, n\}$.

With regards to the sensing capabilities of the CAVs, each CAV is assumed to have the minimum required onboard sensing capabilities to detect the positions and velocities of surrounding vehicles within a realistic sensor range. Each CAV can track the positions of up to eight adjacent

non-occluded vehicles surrounding it. The surrounding vehicles that a CAV could track are highlighted in figure 5.2. In practice, the actual number of vehicles tracked may be lower due to factors such as density level and the lane the CAV occupies.

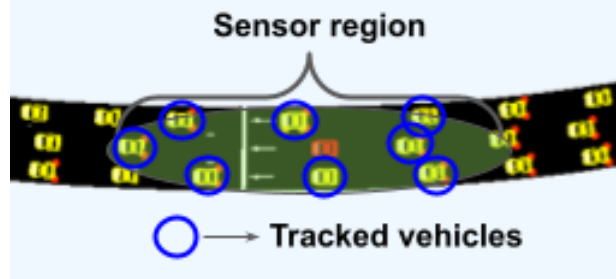


Figure 5.2: Modeling sensing capability of CAVs

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. In this work, we do not consider network delay and packet loss during transmission. Vehicles within a realistic communication range of each other are assumed to communicate shared information in real-time.

5.4 Microscopic traffic models

The process of modeling vehicle behavior in large-scale highway networks usually involves two separate models. A car-following model is used to define the longitudinal movement of a vehicle, considering both its interactions with a lead vehicle and safety parameters. A lane change model is used to determine the appropriate choice of lane, and the parameters necessary for a safe lane change maneuver in multi-lane highways. Some of the widely accepted car-following models adopted in traffic modeling and simulation include the Krauss model [25], the Wiedemann model [76], and the Intelligent Driver Model (IDM) [77]. In this research, we opt to use the Krauss

model due to the simplicity of the model, its accuracy, and the ease with which parameters relating to human reaction speed can be adjusted.

5.4.1 Krauss Car-following Model

The Krauss model [25] developed in 1997, allows for the direct computation of the target control velocity needed to achieve safe car following behavior. The behavior of the vehicle in this model depends on if it is in free motion or interacting motion. In free motion, there is no lead vehicle affecting the ego vehicle and the command velocity $u(t)$ is governed by the road speed limits $\bar{v}$ as shown in equation (5.4). Additionally, $u(t)$ is also constrained by the acceleration capabilities of the ego vehicle, characterized by its maximum acceleration a_{max} and maximum braking a_{min} capabilities as shown in equation (5.5). Here $v(t)$ represents the current velocity of the ego vehicle and Δt is the time step.

$$0 \leq u(t) \leq \bar{v} \quad (5.4)$$

$$a_{min}\Delta t \leq u(t) - v(t) \leq a_{max}\Delta t \quad (5.5)$$

In the case of interacting motion, the ego vehicle is affected by the behavior of the lead vehicle. Therefore, in addition to the constraints in equations (5.4) and (5.5), a safe velocity $v_s(t)$ is computed considering the lead vehicle velocity $v_l(t)$, gap to lead vehicle $\Delta s(t)$ and the driver

reaction time τ_r as shown in equation (5.6). Here, $b(v(t))$ represents the deceleration function.

$$v_s(t) = v_l(t) + \frac{\Delta s(t) - v_l(t)\tau_r}{\frac{v(t)}{b(v(t))} + \tau_r} \quad (5.6)$$

Therefore, the desired command velocity $v_d(t)$ is computed as the minimum of the speed limit, acceleration bound and safe following velocity as shown in:

$$v_d(t) = \min(\bar{v}, v(t) + a_{max}\Delta t, v_s(t)) \quad (5.7)$$

Finally, the command velocity $u(t)$ is set by considering the random perturbations η that occur due to both the imperfection in human drivers and vehicle actuation as shown in:

$$u(t) = \max(0, v_d(t) - \eta) \quad (5.8)$$

A key factor for the choice of the Krauss model in this research is the accessibility of parameters τ_r and η which allow the overall modeling of human driving imperfection and reaction time. These factors are essential in shock wave research as they are usually the major contributors to spontaneous shock wave generation.

5.4.2 Lane change model

When we consider a large-scale simulation of a multi-lane highway, the modeling of lane change behavior plays an important role. This model involves the computation of speed adjust-

ments necessary to change lanes, the choice of the best lane to change to, and the safety criteria in choosing suitable gaps for a lane change maneuver. Lane change can be necessitated by strategic requirements such as taking the next off-ramp, speed gain requirements, or cooperative requirements which involve changing lanes to allow other vehicles to pass. In this research, we use the model developed by Erdmann [26], as it provides many parameters for fine-tuning each vehicle's lane-changing behavior. This model carries out the tasks of selecting the best lane in order to maximize vehicle requirements, choosing when to merge depending on certain safety criteria, and the execution of the actual lane change maneuver. It is also capable of ensuring traffic rules such as always overtaking on the left are followed.

5.5 V2V Communication-based Shock wave dissipation method

The process of designing a control algorithm for shock wave dissipation involves multiple sub-steps. In order to apply control for shock wave prevention, we first need to detect the presence of a shock wave. The accurate detection of shock waves requires a decent estimate of the traffic conditions at various locations on the highway. We will now look at the methods used in our algorithm at each of these sub-steps.

5.5.1 Traffic state estimation

Accurate traffic state estimation is the first step in solving the problem of shock wave generation. Here we define traffic state as the aggregate state over all lanes and not the state of a single lane. The main desired components of the traffic state are the density, throughput, and mean veloc-

ity of vehicles along the length of the highway. Traditionally, data for traffic state estimation had to be gathered from static, sparsely located infrastructure nodes. Therefore, unless a large number of sensors were used, the resulting state estimation was quite inaccurate. However, with the presence of Lagrangian sensors in the form of CAVs interspersed among the human driven vehicles, we can now obtain a far more accurate depiction of the state of traffic on a highway link. Here, accuracy is defined as the similarity between the estimated traffic state and the ground truth value. It is also important to note that when estimating the traffic state using CAVs, we see that the accuracy of state prediction achieved increases with the number of CAVs present on the highway. In this research, we consider two separate methods for highway state reconstruction. CAVs could gather data based on their internal state or based on the information it gathers about surrounding vehicles.

5.5.1.1 Estimation based on ego vehicle data

While density and throughput can be computed based on the number of vehicles in a highway section at each time instance, it is more difficult to compute the average velocity of vehicles at a specific point on a highway section. The estimation methods presented try to compute the average vehicle velocity at the CAV's location on the highway. We assume each CAV maintains a memory of its own past k velocities, where k is a parameter defining the length of the memory needed. Then the average velocity estimate $V_i^e(t)$ at position $s_i(t)$ of CAV i is computed by a rolling time average as follows,

$$V_i^e(t) = \frac{1}{k+1} \sum_{\tau=0}^k v_i(t-\tau) \quad (5.9)$$

5.5.1.2 Estimation based on surrounding vehicle data

The average velocity estimate $V_i^e(t)$ computed in equation (5.9) may not be accurate in the case of a multi-lane highway, since the lane containing the CAV may be moving slower or faster than its surrounding lanes. In order to address this issue, we leverage the advanced sensor suite on-board modern CAVs to collect data regarding the vehicles surrounding the CAV. The sensor range and tracked vehicles are discussed in Section 5.3.2. Let the number of vehicles tracked be m , the maximum length of memory be k and $v_j^i(t)$ represent the velocity at time t of the j^{th} vehicle tracked by CAV i . Also, let $k_j \leq k$ be the number of time steps the j^{th} vehicle is tracked. Then the average velocity estimate $V_i^e(t)$ at position $s_i(t)$ of CAV i is computed by a rolling time average considering all tracked vehicles as follows,

$$V_i^e(t) = \frac{1}{m+1} \sum_{j=0}^m \frac{1}{k_j+1} \sum_{\tau=0}^{k_j} v_j^i(t-\tau) \quad (5.10)$$

Here, $v_0^i(t)$ represents the velocity of the CAV under consideration i , at time t . As data is gathered from multiple lanes, this method leads to a much more accurate representation of the average vehicle velocity at position $s_i(t)$ at time t . It is important to note that, k_j for every tracked vehicle j will vary depending on how long it stays within the field of view of CAV i .

5.5.2 Shock wave detection

Shock waves are characterized by a sudden change in the traffic state on a highway. These shock waves propagate either upstream or downstream on the highway in the form of a wave. The

rate at which the shock wave moves V_λ , can be computed using the Rankine-Hugoniot condition which guarantees the conservation of mass of traffic flow as follows,

$$V_\lambda = \frac{Q_c - Q_f}{\rho_c - \rho_f} \quad (5.11)$$

Here, density and throughput are given by Q_c and ρ_c for the congested region at the shock wave, and Q_f and ρ_f for the free-flow region outside the shock wave. While this parameter characterizes the shock wave, the actual detection process is carried out by the evaluation of the speed distribution of CAVs on the highway. In the case where CAVs do not communicate with each other, shock waves are detected by comparing the current velocity of the CAV to its long-term average velocity data. This method only enables the CAV to estimate whether or not it is currently facing a shock wave scenario. This process is vastly improved when communication among CAVs is considered. A CAV can compare its own current and temporal average velocity with the velocities communicated to it from downstream CAVs within communication range. This allows CAVs to not only detect the presence of the shock wave in advance but also the exact location of the shock wave on the highway. This then paves the way for the use of proactive control algorithms, which start affecting highway conditions upstream of the actual shock wave location. This is shown to have a strong impact on the shock wave dissipation process.

The algorithm for shock wave detection and the subsequent application of suitable control is carried out by every CAV present on the highway. The first step of the algorithm for shock wave detection using V2V communication involves gathering local traffic state data, as discussed in Section 5.5.1, from all downstream CAVs within communication range C_i of the ego vehicle i .

The ego vehicle then builds up an estimate of the traffic conditions in front of it and selects the worst-case scenario as shown in equation (5.12). We observe that the best performance is obtained by identifying the worst-case scenario downstream of the CAV and adapting the control to suit this.

$$v_{rel}^i(t) = \max\{0, v_i(t) - \min_{j \in C_i} V_j^e(t)\} \quad (5.12)$$

Here, $V_j^e(t)$ is obtained from either equation (5.9) or (5.10) depending on whether data is collected from surrounding vehicles. The algorithm also takes note of the position $s_j^i(t)$ of the vehicle j which corresponds to the worst-case minimum average velocity $v_{det}^i(t) = V_j^e(t)$ ahead of ego vehicle i . When the value of $v_{rel}^i(t)$ exceeds a tunable threshold value V_{sw} , a shock wave is considered detected, and control will then be applied to CAV i for shock wave dissipation.

5.5.3 Control for shock wave dissipation

The control algorithms defined in this section aim to dissipate shock waves as soon as possible with minimal control effort. While the main algorithm (*comm_method*), focuses on cooperative control and information sharing, for comparison we also implement a control algorithm (*ind_method*) for the case of no cooperation, similar to the method in [72].

The output desired target velocity $v_d^i(t)$ generated for ego vehicle i will be based on the type of control algorithm used.

5.5.3.1 Independent control method

For the simple case (*ind_method*) that does not consider information sharing among vehicles, $v_d^i(t)$ is computed based on a rolling average of the velocity of vehicle i as shown in equation (5.13). Here, m represents the length of memory used and needs to be set to a high value ($m \gg k$) to achieve suitable results.

$$v_d^i(t) = \frac{1}{m+1} \sum_{\tau=0}^m v_i(t-\tau) \quad (5.13)$$

5.5.3.2 Communication-based control method

In contrast, for the cooperative control approach (*comm_method*) with V2V information sharing, when a shock wave is detected downstream, we set the value of $v_d^i(t)$ based on $v_{det}^i(t)$ obtained from equation (5.12). This allows the ego vehicle i to react in advance to the shock wave traffic conditions ahead of it. Note that this control will only be applied until the CAV reaches the position $s_j^i(t)$, detected as the worst-case conditions ahead of the CAV. However, the CAV can always adapt its control if traffic conditions worsen ahead of it. The process of setting the target velocity $v_d^i(t)$ is explained in Algorithm 1.

Additionally, it is important to ensure that safety constraints are considered when choosing a control for the vehicle. Therefore, we compute the maximum safe following velocity of vehicle i denoted by $v_s^i(t)$, using the Krauss car following model described in Section 5.4.1. This method uses information pertaining to the distance gap to lead vehicle $\Delta s(t)$ and lead vehicle velocity $v_l(t)$ which are measured using the onboard sensors of the CAV. Note that constraints in equations (5.4) and (5.5) also need to be satisfied.

Algorithm 1: Communication based control

Result: Obtain control $u_i(t)$

```
1 if  $v_{rel}^i(T) > V_{sw}$  (Shock wave detected at  $t = T$ ) then
2    $detPos = s_j^i(T);$ 
3    $v_d^i(t) = v_{det}^i(T);$ 
4   if  $s_i(t) < detPos$  then
5     if  $v_{rel}^i(T') > v_{rel}^i(T)$  (New det. at  $t = T'$ ) then
6        $T = T';$ 
7        $detPos = s_j^i(T');$ 
8        $v_d^i(t) = v_{det}^i(T');$ 
9     end
10    Compute  $v_s^i(t);$ 
11     $u_i(t) = \max(0, \min(v_s^i(t), v_d^i(t)));$ 
12  end
13 end
```

5.6 Experimental Setup and Results

The performance of the proposed approach is evaluated on a circular three-lane highway loop simulation, implemented on the SUMO [22] traffic simulation platform. The simulation setup for the highway loop is shown in figure 5.3. 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.

The length of the circular three-lane highway loop simulated is 1 km long and all tests were carried out with a density of 200 vehicles in the loop. The vehicles used were a mix of CAVs and HDVs at varying proportions (CAV penetration levels). Two simulation parameters play a major role in simulating realistic driving behavior which results in the natural formation of shock waves. *Sigma* is a parameter that allows the specification of driver imperfection and is set to its maximum value of 1. The parameter *actionStepLength* which handles the reaction time involved

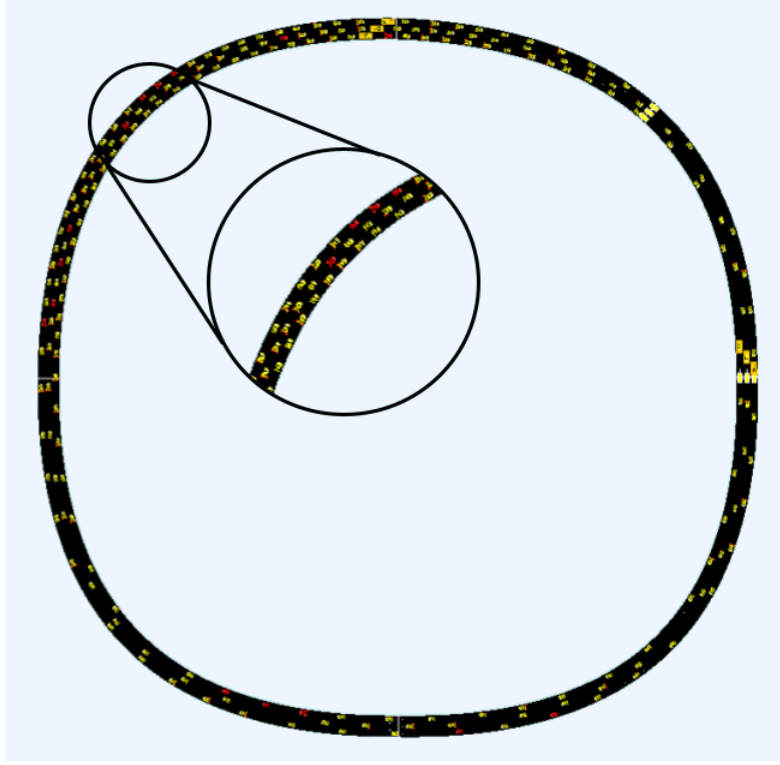


Figure 5.3: Circular multi-lane highway loop simulation

in the decision-making process of vehicles is set to $1sec$. These parameters lead to natural traffic shock wave formation over time, similar to that observed in human driving data. Additionally, as safety was a critical component built into all control algorithms used, no collisions were observed in any of the experiments.

5.6.1 Proposed method Performance

In this section, we discuss the performance achieved by the proposed V2V communication-based method (*comm_method*) which allows cooperative information sharing among CAVs to achieve better shock wave dissipation performance. The presence of continuing shock waves is evident in the case in which no control was applied as seen in figures 5.4a and 5.5a. While the

vehicle trajectories showcase the repeating shock waves, from the velocity curves we observe that the majority of vehicles constantly switch between higher velocities and very low velocities which is characteristic of a traffic shock wave. Note that while the experiment contains 200 vehicles,

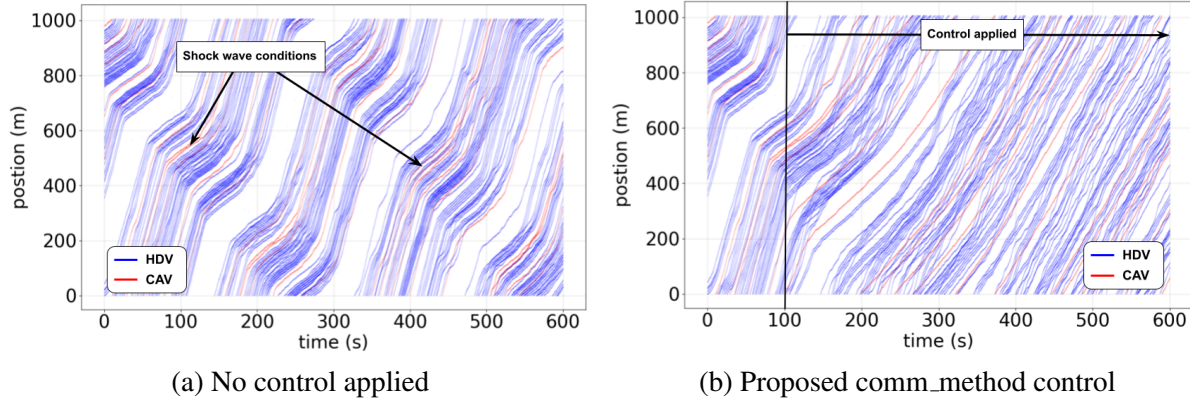


Figure 5.4: Variation in trajectories of vehicles

figures 5.4 and 5.5 plot only the information of 50 vehicles (6 CAVs and 44 HDVs). This results in clear uncluttered graphs which showcase information representative of the traffic state in the overall simulation. In all the experiments conducted, the CAV penetration level is set as 7.5%. Furthermore, control is applied to CAVs starting at time $t = 100$ s in all experiments.

In figure 5.4b, based on vehicle trajectories, we observe that the shock wave is fully dissipated within 2 minutes of activating the control strategy. From figure 5.5b, we observe that the velocities of all vehicles converge to a common average value and note that this average value gradually increases with time.

Remark 2. In figure 5.4, trajectories of individual vehicles occasionally cross one another. Due to the fact that we simulate a multi-lane highway system, it is understood that these intersections represent a vehicle being overtaken by vehicles in other lanes.

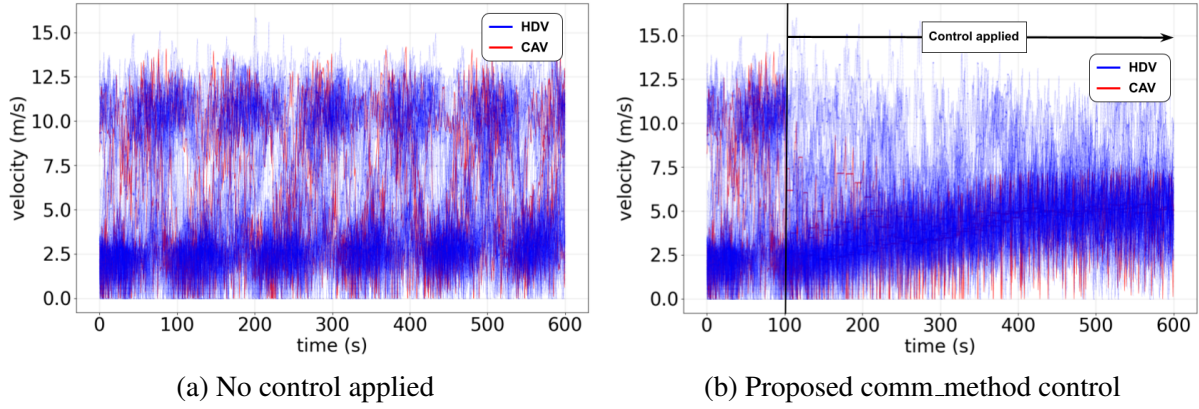


Figure 5.5: Variation in velocities of vehicles

5.6.2 Comparison between Methods

The shock wave dissipation performance of the proposed *comm_method* is compared to two other methods. We use a baseline case (*no_control*), which does not apply any shock wave dissipation control to the CAVs. We also implement a method (*ind_method*) that applies a control similar to that used in [72], to each CAV on the highway independently. In the *ind_method*, vehicles do not communicate and control is computed based on equation (5.13).

The main performance indicator in evaluating different control schemes for shock wave dissipation is the variation in the velocities of the vehicles involved. A good control scheme should be able to reduce this variation within a short time span without affecting the overall system throughput. The change in the standard deviation in vehicle velocities for the different methods are shown in figure 5.6. We observe that the proposed V2V communication-based method results in a 40% lower standard deviation in velocity than the independent control method. In figure 5.7, we observe a greater reduction in the average velocity of vehicles when control is activated in the proposed *comm_method*. This highlights how fast the proposed algorithm begins to clear up the shock

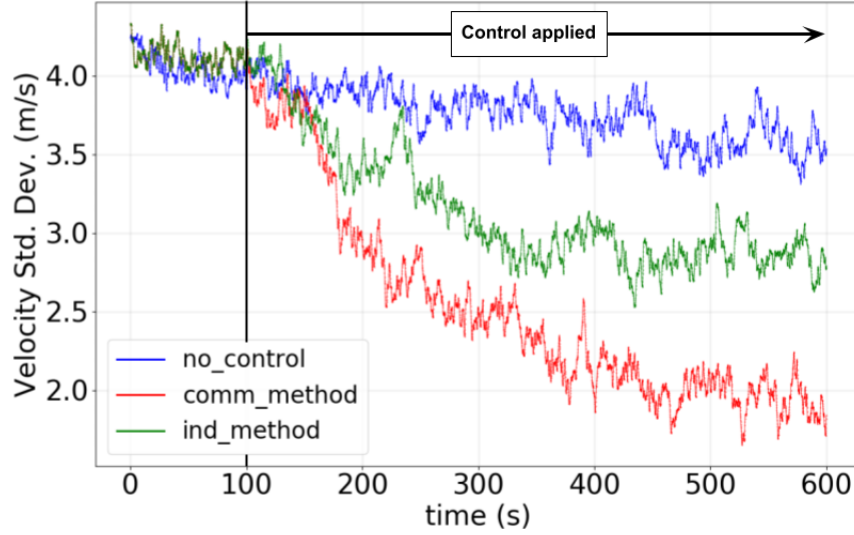


Figure 5.6: Velocity Standard Deviation for different methods

wave when compared to other methods. The reduction in average velocity is due to vehicles in the zones outside the shock wave being preemptively slowed down in anticipation of downstream shock wave conditions. We see that by the 10-minute (600sec) mark all methods reach an average velocity close to $6m/s$, but the standard deviation in these velocities is drastically lower in the proposed *comm_method*.

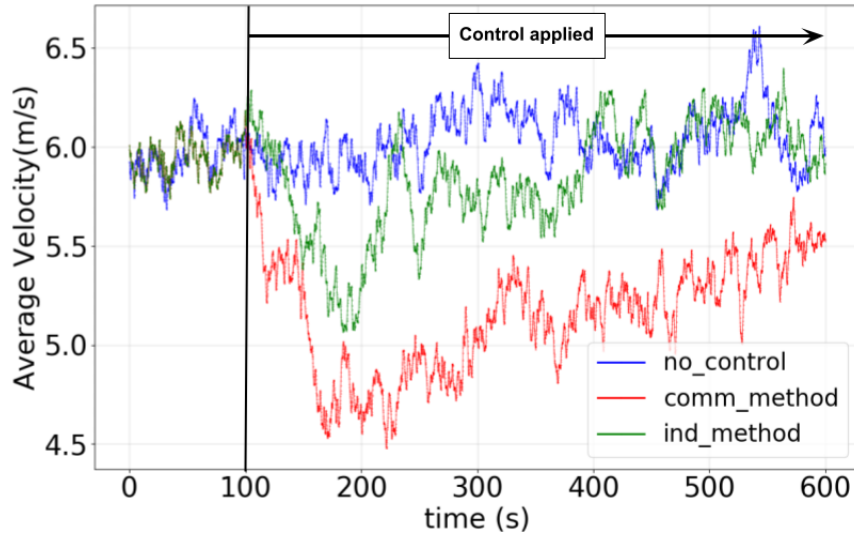


Figure 5.7: Average velocity comparison for different methods

Remark 3. It is important to note that the performance of the *ind_method* is quite dependent on the circular highway model. This is due to each vehicle needing to collect adequate information about shock wave conditions before control can be applied. The proposed *comm_method* does not have this limitation and is therefore independent of the highway structure.

5.6.3 Impact of CAV penetration levels

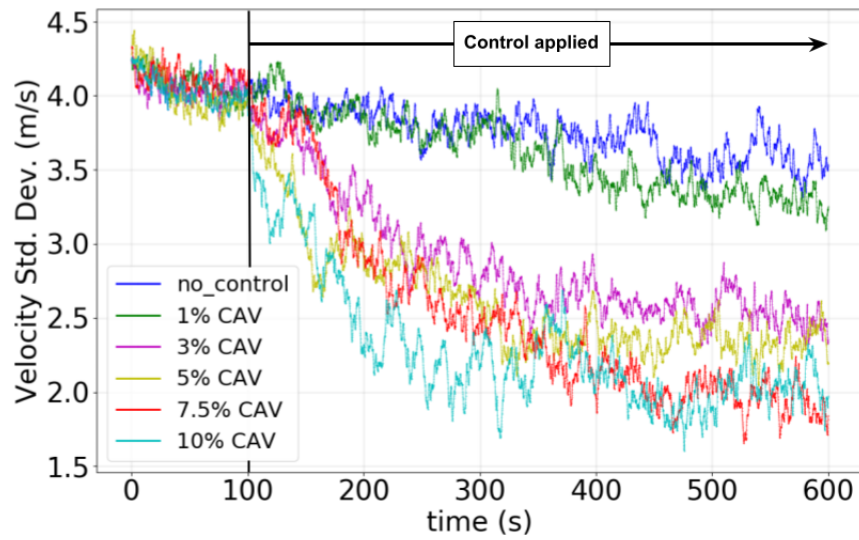


Figure 5.8: Velocity std. dev. for varying CAV penetration levels

In this section, we explore the impact of different CAV penetration levels on the performance of the proposed *comm_method*. In figure 5.8, we see that very low penetration levels like 1% and 2% have minimal impact on shock wave mitigation. This is mainly due to the fact that on a multi-lane highway, other vehicles will simply overtake the few CAVs attempting to apply control. As the number of lanes increases, the penetration level needed for good performance also increases. For our experiments in a 3-lane highway, we observe that around 7.5% or higher CAV penetration leads to effective shock wave dissipation in a timely manner. However, even at lower penetration

levels, there is still a positive impact on overall shock wave dissipation, even though full dissipation may require a longer duration of time.

5.7 Improvements to Shock-Wave Dissipation Control

The control methods thus far developed, focus on longitudinal speed control by creating a moving bottleneck. This control also takes the form of an on-off control that requires vehicles to perform rapid acceleration and deceleration. This leads to uncomfortable experiences for passengers and is therefore a less practical implementation. To overcome these issues we show that the detection information can be used to design a class of proactive shock wave mitigating CAV controllers. The choice of the controller can then depend on design parameters such as the aggressiveness of the driving behavior allowed. We also demonstrate the importance of the positioning of autonomous agents in multi-lane scenarios by introducing a new lane-changing controller which aims to distribute the CAVs equally among all lanes of the highway.

5.7.1 Proactive Shock-Wave Dissipation Controller

Once a shock wave has been detected, a control strategy should be implemented by each CAV to dissipate the effects of the shock wave as soon as possible, and with minimal control effort. It has been shown that a shock-wave mitigating controller should reduce the velocity of the CAV in an attempt to regulate traffic and avoid stop-and-go behavior [72]. In a naive approach, a velocity control scheme for CAV (i) is computed based on the worst case minimum average velocity $v_{det}^i(t)$

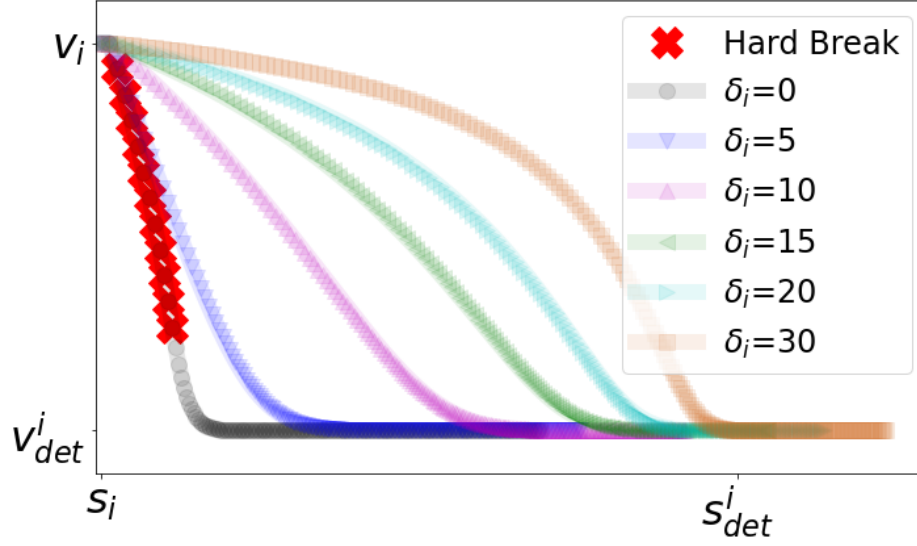


Figure 5.9: Closed-loop velocity-position profiles for different parameter values $\delta_i \in [0, 5, 10, 15, 20, 30]$ ($a_i = 1$, $d_c = 300$). Here v_i , s_i , are the initial velocity and position of a CAV that detects a shock wave at a distance s_{det}^i from the worst case minimum average velocity $v_{det}^i(t)$.

in (5.22), as shown in equation (5.14) below:

$$v_i(t+1) = v_i(t) + dt \min\{\alpha_{min}^i, v_{det}^i(t) - v_i(t)\} \quad (5.14)$$

where it is assumed that $v_{det}^i(t) < v_i(t)$, α_{min}^i represents the maximum braking capacity of the vehicle, and dt is the timestep used which depends on the operation frequency of the controller.

While this is indeed a proactive controller that allows the ego vehicle i to react in advance to the shock wave traffic conditions, it is easy to see (see Fig. 5.9, case $\delta_i = 0$) that it corresponds to a largely aggressive braking profile. In designing a better controller, we have to take into account both the velocity gap $e_v = v_{det}^i(t) - v_i(t)$, and the actual distance $e_s = |s_{det}^i(t) - s_i(t)|$ between the ego vehicle and the vehicle which corresponds to the worst case minimum average velocity $v_{det}^i(t)$ ahead of ego vehicle i . In particular, the shock-wave dissipation scheme should be proportional

to the velocity gap, and should be less aggressive if the distance from the shock-wave e_s is large, i.e. when the CAV has detected the shock wave in a timely manner. This contributes to reducing fuel consumption and maximizing the passenger's comfort. In addition, the controller should also be appropriately parametrized, such that it is able to model different driving behaviors, for example, the aggressiveness of the CAV with respect to exogenous metrics, e.g. vehicle priority or emergency. In view of all these, we propose the following adaptive controller:

$$\begin{aligned}
v_i(t+1) = v_i(t) + \min\{ & \alpha_{min}^i dt, \\
& a_i \gamma_i(s_i(t), s_{det}^i(t))(v_{det}^i(t) - v_i(t))dt \\
& + b_i(v_i(t-1) - v_i(t))\}
\end{aligned} \tag{5.15}$$

where the term $b_i(v_i(t-1) - v_i(t))$ contributes to a smooth velocity trajectory (notice also its connection to the derivative of the error $\dot{e}_v \approx \frac{v_i(t-1)-v_i(t)}{dt}$), and the position-dependent coefficient $\gamma_i(s_i(t), s_{det}^i(t))$ is given by:

$$\gamma_i(s_i(t), s_{det}^i(t)) = \frac{1}{1 + (\frac{\delta_i}{d_c})^2 |s_{det}^i(t) - s_i(t)|^2} \tag{5.16}$$

where d_c is the maximum communication range of the CAVs.

The parameters a_i and δ_i control the closed-loop velocity trajectory as shown in Fig. 5.9. We observe that for different δ_i coefficients the velocity-position (and velocity-time) profiles of the closed-loop system can vary from sub-linear, to linear, and to super-linear, while, in the end, converging to the detected worst-case minimum average velocity $v_{det}^i(t)$. The case $\delta_i = 0$ corre-

sponds to the naive control scheme (5.14). An increase in the parameter a_i induces larger deceleration steps and is interpreted as a tendency to hard break (increased fuel consumption, decreased driver's comfort). On the other hand, a low value of a_i results in smaller deceleration steps, but may not result in convergence to $v_{det}^i(t)$. A typical value is $a_i = 1$, which represents the maximum coefficient of the proportional term $(v_{det}^i(t) - v_i(t))$ of the controller.

The optimal controller parameters a_i and δ_i , are estimated from simulation data in Section 5.7.3, based on exhaustive search in the parameter space aiming to minimize the objective function:

$$\begin{aligned} \min J &= \hat{V}[v] \\ \text{s.t. } \hat{E}[v] - \bar{v} &> \nu \end{aligned} \tag{5.17}$$

where $\hat{E}[v] = \frac{1}{N} \sum_{i=1}^N \|v_i(T)\|^2$ represents the average velocity of the N cars in the highway section, and $\hat{V}[v] = \frac{1}{N} \sum_{i=1}^N \|v_i(T) - \hat{E}[v]\|^2$ the average variance of the velocity of the cars, at the end of the simulation time $t = T$. Minimizing the variance of the velocity of the cars results in mitigating the stop-and-go effect of the shock wave. The threshold ν represents how much we are willing to reduce the average velocity of the vehicles on the highway (with respect to the speed limit $\bar{v}$) to dissipate the shock wave effect. Notice that the optimization problem (5.17) can be alternatively written as:

$$\max J = c_1 \hat{E}[v] - c_2 \hat{V}[v] \tag{5.18}$$

which represents the scalarization method in constrained optimization, and results in different Pareto optimal solutions, depending on the values of c_1 and c_2 which can be chosen depending on the priorities given by the designer. We note that the simulation time T should represent a

reasonable time frame to resolve the shock wave phenomenon. Note that the constraints placed on the vehicle by the speed limit $\bar{v}$ and each vehicle's dynamics and acceleration capabilities also need to be satisfied.

5.7.2 CAV's Lane-Changing Controller

The proactive shock-wave dissipation controller described in Section 5.7 constituted a high-level longitudinal controller for the velocity control of CAVs once a shock wave is detected. In parallel to this controller, we also implement a lane-changing controller which identifies the best lane a CAV should be in and facilitates the lane-changing maneuver to achieve this. In this work, the lane-changing controller is based on maximizing the entropy of the distribution of the CAVs along the multiple lanes, i.e., on maintaining a uniform distribution for the CAVs at any given highway segment.

In multi-lane highways when CAVs attempt to apply a forced bottleneck control, what often occurs is that the neighboring HDVs would simply move to a different lane and overtake the CAV. This reduces the corrective effect of the shock wave dissipation control applied by the CAV. In order to prevent this and maximize this correctional effect, it is important that the CAVs are evenly distributed among the lanes of the highway. To achieve this uniform distribution, each ego CAV calculates the distribution of other CAVs among lanes downstream of its location. This information is gathered via V2V communication with all CAVs within range. At this point, our high-level lane-changing controller computes the lane with the lowest CAV occupancy and sets this as the target lane. The system then performs a safety check to identify if the ego vehicle can safely

execute a lane change maneuver in the direction of the target lane. If this safety check is passed then this controller issues the lane change command to the low-level lane-changing controller for execution. For our research, we adopt the low-level controller found in [26], which provides several parameters for fine-tuning a vehicle's lane-changing behavior.

5.7.3 Improved proactive control results

The performance of the proposed approach (Section 5.7) which includes a proactive shock-wave dissipation controller designed using the V2V communication-based shock wave detection algorithm is compared against the baseline case where no shock wave mitigating control is applied by the CAVs.

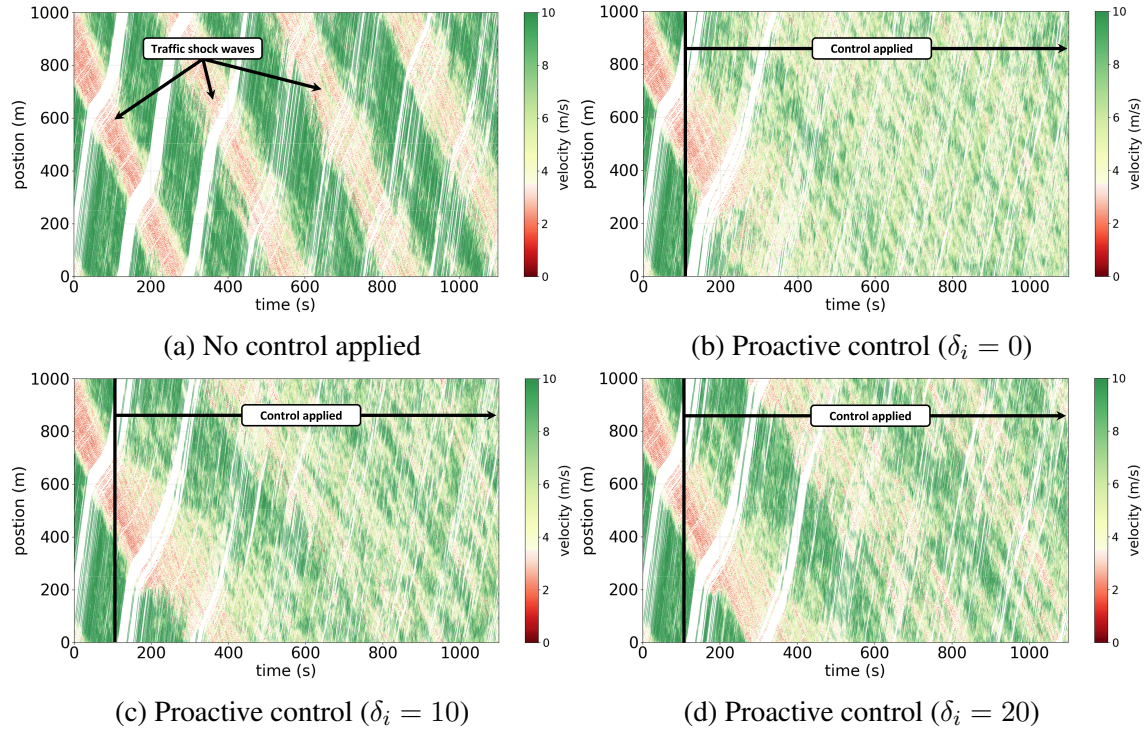


Figure 5.10: Variation in trajectories of vehicles.

In Fig. 5.10 and Fig. 5.11, we compare the performance of the proposed approach for

different control profiles, i.e., different parameters δ_i (see Section 5.7).

We illustrate the trajectories of the vehicles over time and their corresponding velocities over time in Fig. 5.10 and Fig. 5.11 respectively. In both these figures the presence of continuing shock waves is evident in the case in which no control was applied as indicated by the waves of red regions depicting very slow-moving traffic in 5.10a and the continued high-low discontinuous velocity profile in 5.11a.

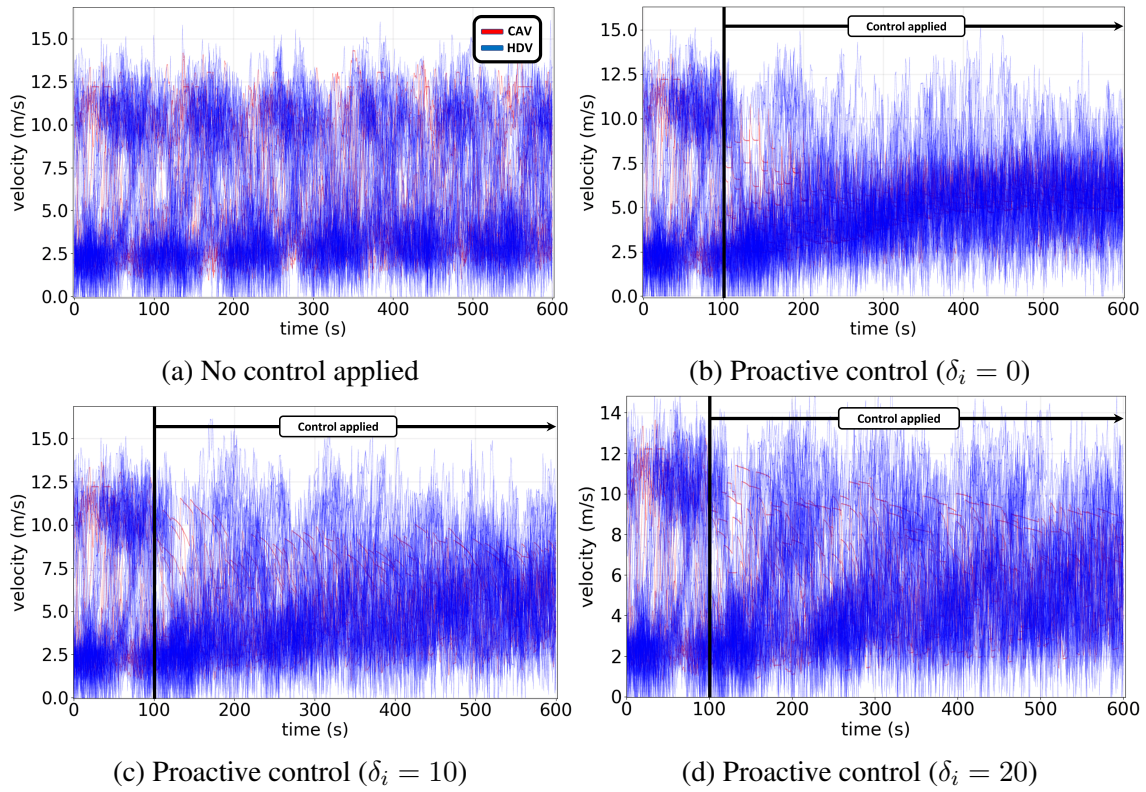


Figure 5.11: Variation in velocities of vehicles.

We also explore the impact of applying proactive control with varying control profiles with parameter $\delta_i = 0, 10, 20$. We find that lower values of δ_i lead to very effective shock wave dissipation but comes at the cost of requiring CAVs to perform very steep deceleration tasks, which leads to very uncomfortable conditions for passengers and also possibly hazardous conditions if

the following HDVs cannot react in time. This is evident in Fig. 5.10b, where we observe that the shock wave is fully dissipated within 2 minutes of activating the control strategy. Higher δ_i values on the other hand result in slower shock wave dissipation but allow for smoother deceleration tasks, which is safer and more comfortable to passengers. In Fig. 5.10c, we observe that it takes more than 4 minutes in order to fully dissipate the shock wave. Our testing showed that a value of around $\delta_i = 4$ provided a good trade-off between shock wave dissipation performance, safety, and passenger comfort.

Remark 4. Fig. 5.10 and Fig. 5.11 plot only 50 of the total 200 vehicles (6 CAVs and 44 HDVs) in order to provide less cluttered graphs. In the experiments, the CAV penetration level is set as 7.5% and control application is initiated at time $t = 100$ s. With safety being built into all algorithms, no collisions were observed.

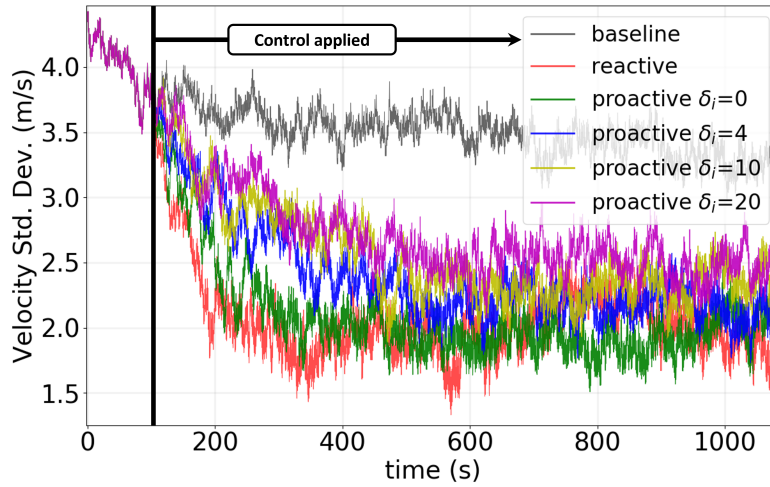


Figure 5.12: Velocity Standard Deviation for different methods.

In Fig. 5.12 and Fig. 5.13, the proposed approach (hereby referred to as “proactive” control method) is compared against the baseline case (no shock wave dissipation control) and an independent method (hereby referred to as “reactive” control method) which applies a control that does

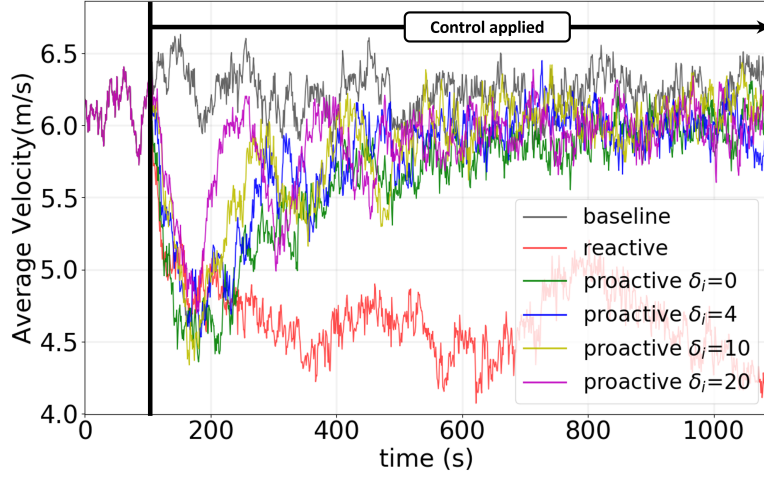


Figure 5.13: Average velocity comparison for different methods.

not consider cooperation among CAVs, similar to that used in [72], in terms of optimizing the objective function mentioned in (5.17) and (5.18). In other words, we are investigating the performance of the different control strategies with respect to the average velocity achieved (higher average velocity corresponds to higher overall throughput) and the average variation of the velocities of each vehicle (low variation corresponds to dissipation of the shock wave). A good control scheme should be able to reduce the velocity variation within a short time span without affecting the overall system throughput. It is also important to stress the connection between shock wave dissipation and fuel consumption. As the variations in velocities reduce, this implies that there are much fewer acceleration and braking tasks leading to reduced fuel consumption.

As the performance of our proactive control approach can be tuned based on the control parameter δ_i , to get a better understanding of the impact of this parameter we plot the performance in terms of average velocity (Fig. 5.13) and velocity standard deviation (Fig. 5.12) with varying control parameter values of $\delta_i = 0, 4, 10, 20$. The standard deviation in velocities is a key indicator of the smoothness of traffic flow on the highway. Fig. 5.12 shows the rate and level of reduction in

standard deviation across the different control strategies in contrast to a baseline approach where no control is applied. This figure needs to be analyzed along with the average velocity of vehicles on the highway which is a measure of the overall throughput and is depicted in Fig. 5.13. From these two figures, we observe that all the proactive control strategies have a strong effect in reducing the velocity standard deviation while ensuring that the average velocity is not impacted in the long run. We find that the strategies using lower δ_i values result in overall faster shock wave dissipation and the ability to reach lower values of standard deviation in a shorter time. However, this comes at the cost of passenger comfort and safe breaking behavior. Therefore in practice, it would be more suitable to use a control profile with slightly higher δ_i values (eg. $\delta_i = 4$) which would provide a balance between these factors.

In terms of the average velocity achieved, as shown in Fig. 5.13 all the proactive strategies achieve similar overall velocities in the long run. Here, the initial reduction in average velocity when control is applied ($t = 100s$) is due to vehicles in the zones outside the shock wave being preemptively slowed down in anticipation of downstream shock wave conditions. Overall we see that all the proposed proactive control strategies once activated lead to around a 50% improvement in standard deviation while leaving the overall average velocity unchanged. In comparison, we observe that while the *reactive method* appears to show the same improvement in standard deviation this comes at the cost of greatly reduced average velocities. Applying this type of reactive non-cooperative control in multi-lane highways can therefore lead to all the vehicles being forced to slow down thus severely impacting overall highway throughput. This does not solve the shock wave problem and can often lead to increased congestion.

Remark 5. *The reactive method requires a circular ring highway structure due to each vehicle needing to face shock wave conditions multiple times before control can begin to be applied. The proposed proactive method applies control based on downstream conditions and is therefore independent of the highway structure.*

5.8 Deep Reinforcement Learning Shock-Wave Dissipation Control

We propose the use of a multi-agent shared policy reinforcement learning algorithm that leverages communication between CAVs for improved observability of downstream traffic conditions. A key feature of this method is the ability to perform shock wave dissipation control without the need for global information and the applicability of this method to multi-lane mixed traffic highways of arbitrary structure. We use the shared-parameter Proximal Policy Optimization (PPO) reinforcement learning strategy for obtaining the controls for each CAV in the simulation. We also built a custom SUMO-Gym wrapper for the multi-lane highway simulation with custom-designed observation space, action space, and rewards for each agent. The shock wave dissipation efficiency is evaluated on a three-lane circular highway loop using realistic traffic simulation software and low CAV penetration levels.

Some initial work in this area explored the use of learning-based techniques, specifically deep reinforcement learning, to address the shock wave dissipation problem [78]. However, this approach makes the assumption of the availability of global traffic state information, which is unrealistic in a real-world situation.

In this work, we develop a shock-wave detection and dissipation approach that does not re-

quire an assumption that global traffic information is available to all CAVs at any time. Instead, we utilize the local sensing and communication capabilities of CAVs to build up a sufficiently data-rich observation space for each CAV from which we can use a DRL algorithm to detect shock-wave conditions and obtain a safe control strategy that would lead to shock wave dissipation. We implemented a custom-built SUMO-gym wrapper for multi-agent learning in a multi-lane highway environment. This allowed us to use a custom-designed observation space, action space, and reward structure for each agent in the simulation. The system also allows for multi-agent handling using the multi-agent environment architecture of the RLLIB platform. We use a Proximal Policy Optimization (PPO) based reinforcement learning strategy with parameter sharing to learn a policy that shows positive improvements in terms of safe controls and shock wave dissipation performance even at low CAV penetration levels. A rule-based lane-changing controller is also implemented to ensure that the CAVs are uniformly distributed among the lanes of the highway, leading to effective control application on each of the lanes. We find that communication-based methods can operate with high levels of performance without the need for global information and show that the proposed learning-based method shows a positive trend towards matching the performance of a rule-based communication control strategy presented in our previous work [31, 79]. We compare our proposed approach against a baseline no-control approach as well as our previous rule-based approach to highlight the advantages and disadvantages of the learning-based method.

5.8.1 Deep Reinforcement Learning Formulation

The shock wave problem with N ego vehicles can be modeled as a Stochastic (Markov) Game and reinforcement learning algorithms can be used to equip the CAVs with a data-driven algorithmic decision mechanism. Starting from some initial global state drawn from some distribution p_0 , at each time-step t , the i -th ego vehicle perceives its local state s_t^i and executes an action a_t^i according to its local policy. The system transitions to a successor state s' under the joint action $a_t := (a_t^1, a_t^2, \dots, a_t^N)$ with some probability, i.e., $p(s'|s, a)$. Then each agent receives a reward $r_t^i := r^i(s_t, a_t)$. The RL-agent is characterized by its policy. We consider randomized policies. Randomized policy $\pi_i(\cdot|s^i)$ is a probability distribution over action space given the state, parameterized by θ_i , which prescribes the probability of taking an action a^i when in state s^i . Echo ego vehicle seeks a policy that maximizes its cumulative discounted reward, i.e.

$$\pi_i^* = \arg \max_{\pi_i} J(\theta_i) := \mathbb{E}_{\pi, P} [R^i] \quad (5.19)$$

where $\pi := (\pi_1, \pi_2, \dots, \pi_N)$, $R^i = \sum_{t=0}^{T-1} \gamma^t r_t^i$ is the γ -discounted cumulative reward over the system trajectory; γ is a discount between 0 and 1. The expectation is taken over the space of trajectories generated by following the policy, i.e. $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim p(\cdot|a_t, s_t)$.

We assume limited sensing and communication range. The state space of each ego vehicle is a finite-dimensional vector of six continuous variables: the velocity and average velocity estimate of the ego vehicle itself, the relative position and velocity of the leading vehicle in the sensing range of its sensors, and the relative position and average velocity estimate of the slowest (facing

worst conditions) CAV in its communication range (further details in section 5.8.1.1). The action space of the ego vehicle is a discrete space with values $\{-3, -2, -1, 0, 1, 2\}$. This limitation is set for simplicity and can easily be extended to allow for a larger action space which could result in smoother velocity trajectories. The positive values indicate acceleration and the negative values indicate deceleration. The reward received at each time step r_t^i by ego vehicle i is a weighted sum of the average velocity (*avg_vel*) of the cars in the ego vehicles' communication range (weighted with β_1) and the standard deviation (*std_dev*) of the velocity of the cars in the ego vehicle's communication range (weighted with β_2); this weighted sum gets augmented by a high negative value if there is a collision (*coll*) (weighted with β_3) or if a non-executable action (*imp*) (weighted with β_4) is selected, i.e.

$$r_t^i = \beta_1 * avg_vel + \beta_2 * std_dev + \beta_3 * coll^i + \beta_4 * imp^i$$

5.8.1.1 Observation space

The two main tasks of each ego CAV involves choosing a control that leads to safe navigation (car-following behavior) and also leads to the eventual dissipation of the high-speed variance conditions witnessed during shock waves. To achieve safe following behavior, the CAVs observation space includes information about the leading vehicle, such as the relative distance to the leading vehicle and the current tracked velocity of the leading vehicle. The leading vehicle is defined as the vehicle directly in front of the ego vehicle in the same lane. The observation state also includes information on the ego vehicles' own velocity.

A key feature needed for shock wave dissipation is a good understanding of the traffic state surrounding a CAV. The presence of a shock wave can be detected by a sudden change in this traffic flow state from one point of the highway to another. This is especially important in a multi-lane scenario where the CAV needs to compute an accurate estimate of the traffic flow conditions in its vicinity. This change in conditions along with the characteristics of the shock wave can be computed using the Rankine-Hugoniot condition which provides the rate at which the shock wave moves V_λ . This is related to the conservation of mass of traffic flow and is given by,

$$V_\lambda = \frac{Q_c - Q_f}{\rho_c - \rho_f} \quad (5.20)$$

Here, throughput and density are given by Q_c and ρ_c for the congested region at the shock wave, and Q_f and ρ_f for the free-flow region outside the shock wave. Based on this information shock wave detection can be carried out by a systematic comparison of the estimated traffic state near each of the CAVs on the highway. In the case where the CAVs cannot communicate with each other, shock waves can only be detected by comparing the long-term average velocity data of a CAV with its current velocity [72]. Here, shock waves can only be detected once the CAV is already facing high congestion conditions. Furthermore, this form of detection is often inaccurate in multi-lane highways, due to the fact that traffic conditions in different lanes are often significantly different.

In contrast by leveraging the communication capabilities onboard modern CAVs, the shock wave detection process can be vastly improved by allowing CAVs to collect information downstream of their actual location via communication with other CAVs. The information communicated for this process involves the average traffic conditions at each CAV's location.

Using the sensor suite on board of CAVs, the immediate neighboring vehicles in the vicinity of a CAV can be tracked fairly accurately. This allows CAVs to obtain an accurate estimate of the traffic conditions in their vicinity by aggregating the information collected from all these tracked vehicles. More specifically, as discussed in [31], let the number of vehicles tracked be m , the maximum length of memory be k and $v_j^i(t)$ represent the velocity at time t of the j^{th} vehicle tracked by CAV i . Also, let $k_j \leq k$ be the number of time steps the j^{th} vehicle is tracked. Then the average velocity estimate $V_i^e(t)$ at position $s^i(t)$ of CAV i is computed by a rolling time average considering all tracked vehicles as follows,

$$V_i^e(t) = \frac{1}{m+1} \sum_{j=0}^m \frac{1}{k_j+1} \sum_{\tau=0}^{k_j} v_j^i(t-\tau) \quad (5.21)$$

Here, $v_0^i(t)$ represents the velocity of the ego CAV under consideration i , at time t . As information on vehicles across all lanes is aggregated, this leads to a much more accurate representation of the average velocity traffic conditions in a multi-lane highway. Note that, depending on how long the tracked vehicle j stays within the field of view of CAV i , the value of k_j can vary.

The state space of the i^{th} CAV then is extended to include the computed average velocity estimate $V_i^e(t)$ which provides information on the traffic state in the vicinity of the ego CAV.

The next important stage in enabling accurate shock wave detection is to provide information about downstream traffic conditions in the ego vehicle's observation space.

This process is carried out via communication with all downstream CAVs j within the communication range of ego CAV i , in order to obtain their current traffic condition estimates $V_j^e(t)$. The ego CAV can then compare this data with its own computed temporal average velocity $V_i^e(t)$.

Based on this comparison, the ego CAV should be able to detect the presence of the shock wave in advance.

In order to streamline this process, the information received from the downstream CAVs C^i of the ego vehicle i , via V2V communication is first sent through a sorting step which identifies the downstream CAV that is facing the worst-case traffic conditions, as shown in equation (5.22). Once identified, the information regarding these worst-case downstream velocity conditions, along with the relative distance to the ego vehicle from the location of these worst conditions, are included in the observation state for the ego CAV i . This simplification through sorting is possible due to the fact that a control computed to handle the worst-case bottleneck traffic conditions provides the best performance, i.e., the information from other CAVs facing better conditions is redundant and would not affect the control applied.

$$v_{wc}^i(t) = \min_{j \in C^i} V_j^e(t) \quad (5.22)$$

Thereafter, by combining all this information, the observation space of CAV i is defined as [Ego CAV velocity, Ego CAV average velocity, Lead vehicle position gap, Lead vehicle velocity, Position gap to CAV with worst-case conditions, Average velocity of CAV with worst case conditions].

5.8.1.2 PPO

To find the optimal policy, the ego vehicles use the PPO algorithm. PPO is an iterative first-order optimization procedure that attempts to find the modified KL-constrained objective function

given by

$$\begin{aligned} \max_{\pi} \mathbb{E}_{\pi, P} \Big[& \min \left(r(\theta_t) \hat{A}_{\theta_t}(s, a), \right. \\ & \left. \text{Clip}(1 - \epsilon, 1 + \epsilon, r(\theta_t) \hat{A}_{\theta_t}(s, a)) \right) \Big] \\ \mathbb{E}_{\pi, P} \Big[& D_{KL}(\pi_{\theta_{t+1}}(a|s), \pi_{\theta_t}(a|s)) \Big] \leq \delta \end{aligned}$$

where

$$r(\theta_t) = \frac{\pi_{\theta_t}(a|s)}{\pi_{\theta_{t-1}}(a|s)}$$

is the ratio of successive policies; D_{KL} is the Kullback-Libeler (KL) divergence between the two distributions. The Clip function restricts the ratio of successive policies to a range between $1 - \epsilon$ and $1 + \epsilon$, i.e., $r(\theta) \in [1 - \epsilon, 1 + \epsilon]$ and helps with stability and convergence speed. For more details on the algorithm, please see [80].

5.8.1.3 Multi-agent DRL Architecture

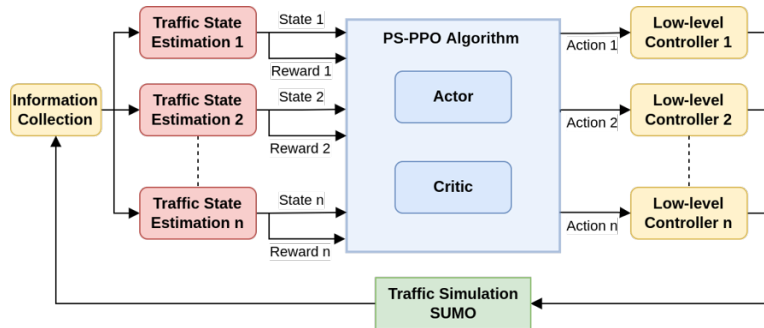


Figure 5.14: Flow diagram for Deep RL shock wave control architecture.

The overall architecture of our proposed system is shown in figure 5.14. Here, we see the location of the traffic state estimation nodes that compute the observation space and rewards for each CAV agent i . The parameter sharing PPO model that computes the actions for each agent, and the low-level controllers onboard each CAV that executes these target actions.

We use PPO with full parameter sharing, that is, all policies are represented by a single neural network with the same shared parameters, for more details please refer to [81]. Each agent observes the environment (the agents have different observations), chooses an action, and then receives a reward. A shared policy gets updated according to the collective experiences of the agents in the environment. It has been shown that Parameter Sharing results in more efficient learning by decreasing the number of trainable parameters, which in turn shortens the training times.

5.8.1.4 SUMO DRL Wrapper Interface

In this work, we developed a custom RLLIB [82] multi-agent environment wrapper based on the OpenAI Gym structure, for the SUMO [50] traffic simulation environment. As such multiple different reinforcement learning algorithms can be tested in the same environment. This also allows for the customization of the simulation environment in terms of the number of agents, physical highway structure, individual vehicle parameters, and initialization points. It also provides the ability to customize the observation spaces, action spaces, and reward functions of each agent in the environment. This module is responsible for converting the actions provided by the learning algorithm into SUMO vehicle speed commands, and collecting data from the SUMO environment in

order to build the observation spaces and rewards for each agent.

5.8.1.5 CAV's Lane-Changing Controller

The system introduced in our method involves two parallelly implemented control structures. A longitudinal controller for the velocity control of CAVs based on Deep RL and a rule-based lane-changing controller which identifies the best lane a CAV should be in and carries out the needed lane-changing maneuvers. The implementation of this lane-changing controller is based on maintaining a uniform distribution of CAVs across all the lanes of the multi-lane highway.

In order to ensure the maximum impact of a shock wave dissipation control in a multi-lane highway, we need the CAVs to be evenly distributed among the lanes of the highway, thus ensuring that control is applied equally on all the lanes. To achieve this, each ego CAV computes the distribution of other downstream CAVs within the communication range. Then the lane-changing controller identifies the lane with the lowest CAV occupancy as the target lane. The system then checks if the ego vehicle can safely execute a lane change maneuver into the target lane. If it is safe to change lanes, the controller issues a lane change command to the low-level controller. In this research, we use the low-level lane-change controller [26].

5.8.2 DRL Experimental Setup and Results

We implemented a circular three-lane highway loop simulation on the SUMO [50] traffic simulation platform as shown in Fig. 5.15. Our Gym-based SUMO wrapper uses the TraCI traffic controller interface to communicate with the SUMO simulator. The RL subsection uses the Ray

RLLIB framework for implementing the multi-agent shared policy PPO learning algorithm. A personal computer with an Intel i7-8750H CPU and 32GB of RAM was used to run all the simulations and algorithms.

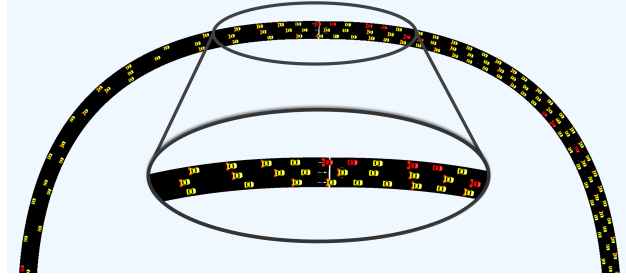


Figure 5.15: Circular multi-lane highway simulation

5.8.2.1 Modeling the physical highway structure

The highway is modeled in the form of a loop in order to simulate an infinite stretch of multi-lane highway. The length of the highway loop is set to 1km and the number of lanes to 3.

The resulting multi-lane simulation is more realistic, highlights the effects of interactions between vehicles, and provides a more accurate representation of the conditions faced by vehicles under shock wave conditions in a multi-lane highway. All tests were carried out with $N = 200$ vehicles in the loop with 15 of these vehicles being CAVs. Note that, it is possible to change the number of vehicles as well as the CAV penetration level as needed.

5.8.2.2 Parameters for shock wave generation

In order to simulate realistic driving behavior which results in the natural formation of shock waves, we modify two key simulation parameters related to SUMO and the Krauss car-following

model. The parameter σ allows the specification of driver imperfection and is set to its maximum value of 1. The parameter $actionStepLength$ handles the reaction time involved in the decision-making process of HDVs and is set to $1sec$. We found that these parameters lead to natural traffic shock wave formation over time, similar to that observed in human driving data.

5.8.2.3 Performance Evaluation

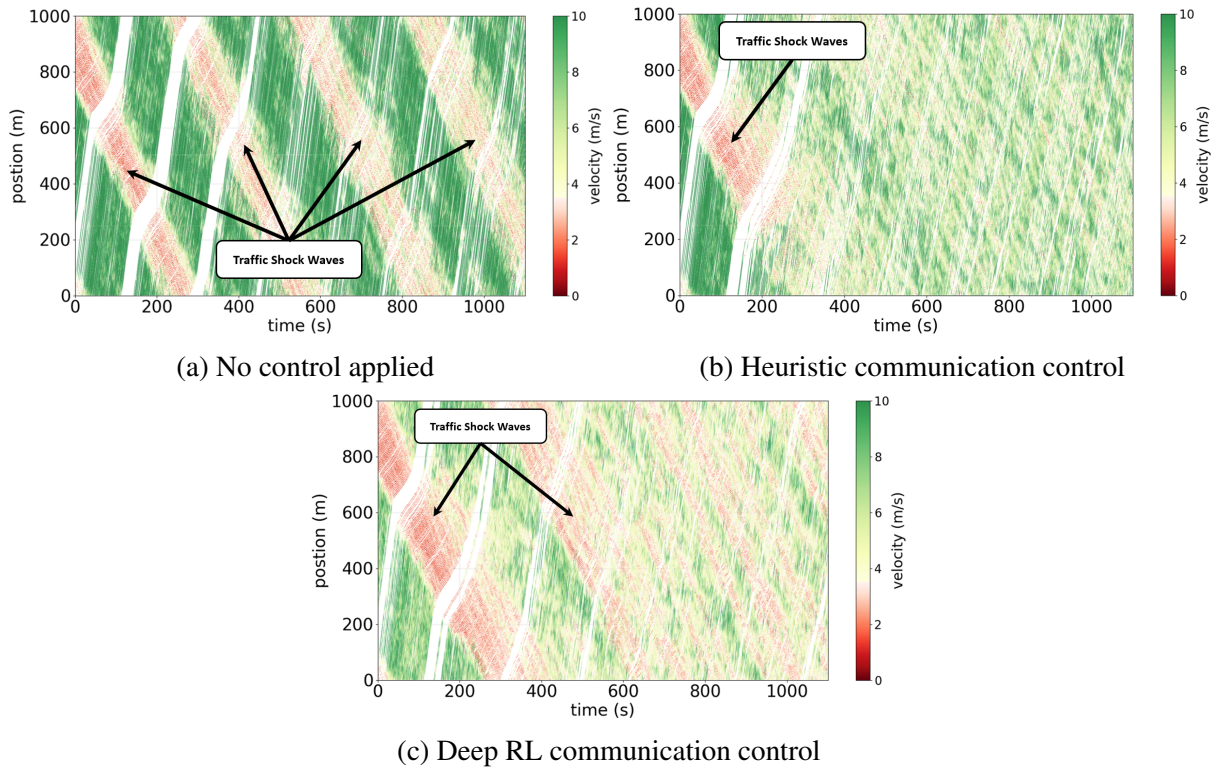


Figure 5.16: Variation in trajectories of vehicles.

We use a comparison with a method in which no control is applied (Fig. 5.16a) and a rule-based communication-based method from our previous work [31, 79] (Fig. 5.16b) in order to evaluate our proposed trained PPO algorithm's (Fig. 5.16c) performance. For better comparability, the previous rule-based method [31, 79] was updated to include the parallelly implemented lane-

changing controller so that the focus of this comparison is purely based on the performance of the proposed learning method. In Fig. 5.16, we observe that in the case where no control is applied the shock wave conditions continue over time indefinitely. Our previous rule-based method [31] is capable of dissipating the shock wave rapidly within a few minutes, but comes at the cost of hard braking controls resulting in an uncomfortable experience for passengers. In contrast, we find that after 100 episodes of shared policy training for 15 agents, our proposed learning-based method is capable of reducing the harsh stop-go behavior of the shock wave. However, in terms of overall average velocity reached it is unable to perform at the same level as the rule-based method. It is possible that given more training episodes and better algorithmic tuning the performance of the learning algorithm can be improved further.

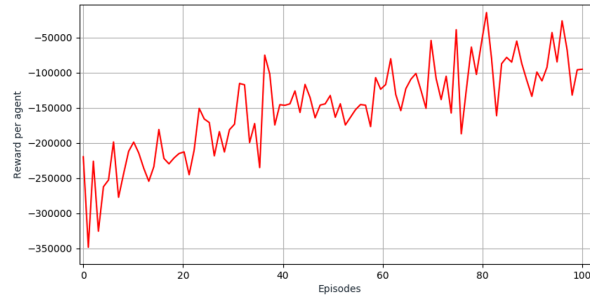


Figure 5.17: Deep RL training performance

The proposed PPO-based learning algorithm was trained over 100 episodes with 15 CAV agents in the simulation learning a shared policy. As shown in Fig. 5.17, we observe that the average reward per agent shows a positive trend as learning episodes proceed. The agents learn to provide safe actions and begin to dissipate the shock wave. It is possible that further modifications in terms of hyperparameter tuning and learning duration could improve the overall performance of the algorithm.

5.8.2.4 Conclusion

We propose the use of V2V communication among CAVs to design a Deep Reinforcement Learning based traffic shock wave detection and dissipation controller for multi-lane highways. This implementation involves a multi-agent learning algorithm with a shared parameter space. In order to implement this we build a custom multi-agent SUMO-Gym wrapper with a custom-designed observation space, action space, and rewards. Our cooperation-based learning method for shock wave dissipation is evaluated in a multi-lane simulation with comparisons to existing communication-based approaches that do not use learning. A lane-changing controller is also implemented to provide a uniform distribution of CAVs among the multiple lanes of the highway. While the performance of the learning algorithm shows promising results, future work in this area involves more intensive training with hyperparameter tuning to extract the best performance of this learning method. Additionally, other reinforcement learning algorithms can be explored using the same custom SUMO-Gym environment.

5.9 Future Work

The work described in this chapter, published in [31], has many possible extensions and improvements. This section focuses on the future direction of the communication-based shock wave method proposed.

5.9.1 Accurate modeling of V2V communication

One of the major assumptions in the results presented in this chapter thus far is the availability of perfect communication. We assumed that vehicles within a determined communication range can communicate with each other with zero delays and packet loss. However, in most practical scenarios this is not a suitable assumption. Multiple factors affect the network quality and thereby how effectively data can be communicated between vehicles. Given this requirement algorithms that require V2V communication need to have suitable tolerances built-in in order to overcome the issues caused by transmission latency and packet loss. It is also important to accurately model the V2V communication protocols and methods used. For this modeling task, there are a number of open source communication modeling software that interfaces well with the existing SUMO based traffic simulation software. Some examples of these network simulators are Veins, NS-3 and OMNeT++.

Once the inter-vehicle communication network has been modeled the next step is to adjust the algorithms used to perform robustly even in the presence of network delays and packet loss. The changes needed may take the form of increased redundancy in data transmission and also increased tolerances in the control algorithms to account for possible data loss or delays. Algorithms can also use time-stamped data from multiple sources in order to build up an accurate up-to-date representation of its required state space. In other words, the time the data was generated prior to transmission becomes an important consideration for the state estimation process. After this is carried out it is also important to stress test the algorithm and identify its worst-case operation capabilities with deteriorating network conditions. A trade-off between robustness to network conditions and subsequent performance loss can also be carried out.

5.9.2 Improvements to Reinforcement learning based shock wave control

In the results presented in this chapter, the control section for deep-RL-based shock wave dissipation achieved promising preliminary results. However, it was unable to outperform the best heuristic shock wave control algorithm. This performance gap may be filled with improvements to the input observation space, output action space, reward structure, and overall RL model architecture.

With regard to the observation space, we may consider the need to stack some of the most recent state space frames in order to capture some of the temporal relationships present in the data. The action space for the DRL algorithm currently consists of the longitudinal acceleration or deceleration commands given to the ego vehicle. We would also explore the benefits of incorporating lane change commands into the action space of the ego CAV vehicle. This would allow the ego vehicle to have more influence on the behavior of HDVs in other lanes. The action space can be designed as a continuous or discrete space as required. We may also experiment with the use of Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) network units in order to capture the temporal relationships related to a single vehicle's behavior.

Chapter 6: V2X communication modeling and applications

6.1 Introduction

Communication has become a core pillar of vehicle automation and the evolution of intelligent transportation systems. The introduction of 5G networks and advanced dedicated short-range communication protocols have enabled vehicles to communicate large amounts of data with very low latency. These advances in vehicular communication are expected to translate into safer road conditions and more efficient transportation networks.

These high bandwidth capable low latency communication systems have now opened the door to many interesting applications in the automated vehicle domain. Accident reporting systems, obstacle early warning systems, blind spot warning systems, large-scale traffic data collection systems, intersection assist systems, and emergency braking detection systems are some of the applications currently being developed and tested. Most of these applications focus on either data collection via Vehicle to Infrastructure (V2I) communication or one or two-hop data sharing via Vehicle to Vehicle (V2V) communication.

When we focus on modeling V2X communication systems multiple factors need to be considered. In the controls research domain, a large portion of research always assumes the presence of

perfect communication. This is instantaneous data transmission with zero delays and zero packet loss. However, this assumption is not valid in almost all practical scenarios. Real-world communication links face network delays, data loss, and physical limitations such as communication range degradation due to interference. The communication capability between two nodes depends on the communication method, the physical capabilities of the transmitter and receiver as well as the physical properties of the communication medium. For example, the communication range of a CAV operating inside a large city with many buildings is vastly different from that of a CAV operating on a large open highway.

In order to build control algorithms that can operate robustly in real-world communication conditions it is very important to be able to accurately model and simulate the different types of links in vehicular communication. This chapter focuses on the diverse needs of these different links, the currently available simulation tools, and the introduction of a new simulation tool that can be used easily and seamlessly alongside existing control algorithms.

Accurate V2X communication modeling and simulation also opens the door to many new exciting research directions in the domain of traffic state estimation and traffic control in mixed-traffic (CAVs and HDVs) settings. We highlight an example (cooperative bidirectional overtaking) in which sharing information among vehicles can significantly increase the safety of CAV operations. This chapter also discusses an interesting application of accurate V2V modeling involving the use of dynamic multi-graphs for representing the micro-level traffic state. Furthermore, we also explore the possibility of GNN-based behavior prediction on graphs.

6.2 Communication technology and simulation overview

Different forms of V2X communication often have diverse requirements and may require different communication methods/protocols. The two main protocols currently considered for V2X communication include the IEEE 802.11p wireless protocol and 5G wireless connectivity. The use of these technologies in conjunction with other communication links is often application dependent and most modern vehicles use a combination of the available communication methods. In this section, we discuss the diverse needs of different vehicular communication links and the available tools for simulating these links.

6.2.1 V2I communication capabilities

Vehicle to infrastructure communication involves the communication between a fixed location base station (central unit) with multiple mobile agents (connected vehicles). This communication can be unidirectional in which the central unit provides information to vehicles in range or the central unit collects information from individual vehicles. The communication could also be bidirectional involving a request and response. This communication could also be either synchronous or asynchronous depending on the individual application.

In V2I when the central unit sends messages to the vehicles it could be in the form of point-to-point communication in which the central unit forms a link with an individual vehicle and can thereafter send a customized message directly to that vehicle. The communication can also take the form of a limited-range broadcast. Here, all the vehicles in range would receive the same information and would perform some computation to determine the portion of the message relevant

to its operation. This type of broadcast messaging system has been shown to be very efficient in highly dense situations involving multiple communicating vehicles. There is also less communication overhead needed for broadcast-based communication since receiving agents can simply wait for the next message if there was some issue in receiving the current message.

The main network characteristics needed for V2I communication are the availability of a high bandwidth connection for collecting large amounts of data, the capability of simultaneously handling multiple incoming and outgoing connections, and the application domain-specific need for low latency communication. For example in the situation of data gathering for traffic state estimation, high bandwidth is more desirable than low latency. While in applications where motion controls are provided to vehicles, low network latency, and low packet loss take priority over the bandwidth availability of the communication link.

6.2.2 V2V communication capabilities

Vehicle to vehicle communication involves communication between multiple mobile agents. These V2V networks have a rapidly evolving topology and communication links may only be formed for a limited amount of time. These V2V connections are often modeled in the form of a Vehicular Ad hoc Network (VANET). There are many challenges faced in VANETs caused due to the high mobility of nodes (agents), the random topology of the network, and the heterogeneous nature of agents in the network. In the US, the VANET protocol stack builds on IEEE 802.11p WLAN operating on seven reserved channels in the 5.9 GHz frequency band and is referred to as WAVE (Wireless Access in Vehicular Environments).

The main difficulty in handling V2V communication networks involves maintaining multiple mobile connections in which agents may be moving at high speeds. In other words, the environment in which V2V communications are used is highly dynamic and constantly changing. Since most V2V applications involve using immediately available data for control decisions the latency in the network plays a major role. The availability of low latency and minimal packet loss communication links often play a vital role in the safety-critical applications which use V2V communications. Hence, age of information is a very important factor to consider in such a fast-paced dynamically changing environment. If data is received too late it may not be of any use at all.

6.2.3 Communication modeling software

In the V2X modeling domain there are a number of open source communication modeling software that interfaces well with the existing SUMO based traffic simulation software. Some examples of these network simulators are Veins, NS-3 and OMNeT++. These software offer highly robust and detailed communication modeling capability. We find that they are quite suitable for modeling the communication between a few vehicles. However, when large-scale simulations are considered, these software are often too bulky and cause unnecessary slowdowns in simulation. For most control or optimization-based formulations, it becomes very difficult to interface with the described communication simulation software. This is due to the fact that when these V2X simulation software are used, they take over the task of moving the simulation forward in time. This makes it difficult to interface with control methods running in parallel or optimization methods running in an event-triggered manner.

6.3 Simplified communication modeling for control applications

In most traffic-related research using optimization, control, and learning algorithms, the role of communication is often supplementary. Due to the complexities in modeling V2X research, in most control-related research, an assumption of perfect communication is often made in order to model the problem. Therefore in order to couple these divided domains of controls research and communications research, a new software needs to be developed. A software that can easily interface with control algorithms and provide the necessary tools needed to model the communication links involved.

6.3.1 Algorithm features and structure

The proposed software is python based and provides tunable parameters which are used to model the communication link and communication medium. The software would provide multiple APIs allowing for many functions such as computing network delay, packet loss, packet transmission probability, and adapting to network topology changes. The software would be provided in the form of a python library.

The initial version of this software library would emulate a single communication link between two agents, either the vehicle and control tower (V2I) or between two vehicles (V2V). Once the dynamic state of each agent (position, velocity, etc.) is provided to the communication modeling node, the node will initialize the expected channel conditions for this upcoming data transfer. Once this is completed the communication node is expected to provide a binary decision on packet loss or transmission for a specific command sent through the created channel between the specified

agents. In these control algorithms, sensing information and control information is continuously being shared between agents at every time step. Therefore, initially, it is adequate if we can provide information on whether a data packet is successfully transmitted or not in the form of a binary decision. This type of binary decision for each data transfer at each time step is also suitable due to the importance of age of information in the applications we consider. Past sensor data and controls related to prior time steps hold little to no value in most of the applications considered. In the world of fast-moving vehicles, a large portion of the environment conditions can change in a short period of time. Furthermore, it is often more beneficial to transmit the updated sensor or control data/information than try to re-transmit data packets from previous time steps that failed to be sent over the network. Due to these factors, our initial implementation of the communication modeling node aims to provide a binary decision on whether a data packet sent between two agents is received within the relevant time step.

6.3.2 Incorporating communication limitations in algorithms

Once the inter-vehicle communication network has been modeled the next step is to adjust the algorithms used, to perform robustly even in the presence of network delays and packet loss. The changes needed may take the form of increased redundancy in data transmission and also increased tolerances in the control algorithms to account for possible data loss or delays. Algorithms can also use time-stamped data from multiple sources in order to build up an accurate up-to-date representation of its required state space. In other words, the time the data was generated prior to transmission becomes an important consideration for the state estimation process. After this

is carried out it is also important to stress test the algorithm and identify its worst-case operation capabilities with deteriorating network conditions. A trade-off between robustness to network conditions and subsequent performance loss can also be carried out.

6.3.3 Trade-off in communication bandwidth vs system performance

It is often the case that bandwidth available for V2V and V2I communication is very limited. Generally, CAVs collect enormous quantities of data constantly, and trying to share all this data will result in unnecessary network congestion as well as lead to large amounts of latency in data transmission. To address this problem, for each application domain, it is important to identify the minimum amount of data needed to be communicated in order to achieve the desired level of overall system performance. This study also needs to factor in the overall quality of the network, since, a network providing a higher bandwidth will allow for sharing of more data and thereby possibly an increase in system performance.

6.4 Cooperative bidirectional overtaking application

A key goal of communicating sensor information among CAVs is to improve the overall safety of the system. One of the major issues faced by a single agent is the limited visibility due to sensor range and sensor occlusion (objects blocking line of sight). This often creates many blind spots in which the agent has very minimal information which can then lead to either overly conservative inefficient controls or overly aggressive dangerous controls. Our proposed approach to overcome this is to allow CAVs within communication range to share progressively collected

information about their local traffic conditions with each other. An ego CAV can then use this additional information it receives via V2V communication to fill the blind spots in its vision and therefore have a better understanding of its surroundings in order to carry out safer maneuvers. We showcase these improvements to safety in the application domain of bidirectional overtaking where the possibility of head-on collisions heightens the need for inherently safe maneuvers.

6.4.1 Overview

In the bidirectional overtaking scenario, an ego vehicle is required to overtake the leading vehicle(s) by moving into the adjacent lane while avoiding incoming traffic. This scenario involves three different types of vehicles: Autonomous ego vehicle which carries out the overtaking maneuver, the vehicles traveling ahead in the same lane as the ego vehicle, and the vehicles approaching in the incoming lane. The non-ego vehicles can be either HDVs or CAVs. A simple three-vehicle overtaking scenario with a sample trajectory is shown in Fig. 6.1, with the ego vehicle depicted in red and the lead vehicle and incoming vehicle both depicted in yellow.

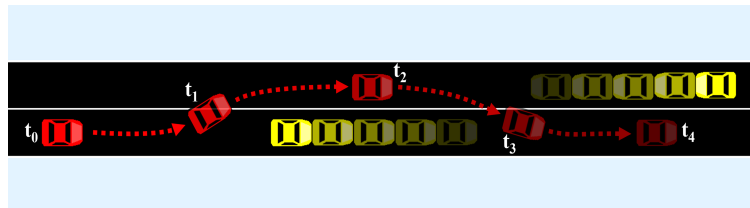


Figure 6.1: Bidirectional traffic overtaking scenario.

This scenario is further complicated by 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.

Our proposed method in this section aims to combine the benefits of cooperation-driven communication-based CAV control systems with the safety guarantees of a MI-MPC optimization-based controller for the domain of bidirectional overtaking in mixed traffic. The main contribution of this work involves the implementation of a V2V 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 method was tested using the realistic SUMO traffic simulation system, and the results and comparisons are presented. We show that our 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 CAV penetration levels and lane density on the performance of our method.

6.4.2 Modeling sensor occlusion

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 ahead. Additionally, depending on the proximity of the lead vehicle to the ego vehicle, visibility will also be reduced in the incoming lane. The resultant visible regions are shown in Fig. 6.2.

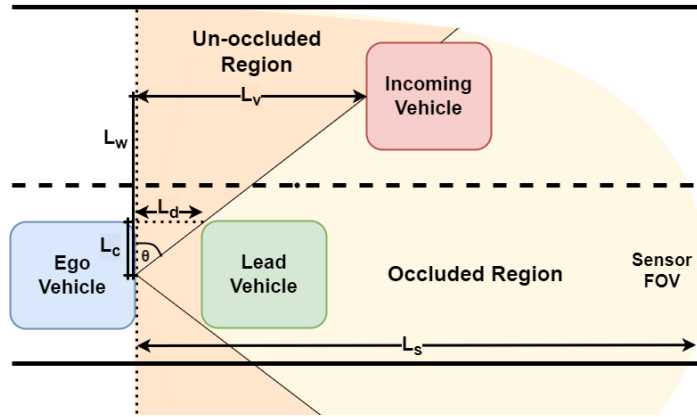


Figure 6.2: Sensor Occlusion Model.

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. It is clear that in the situation where the CAV is closely following a lead vehicle a large portion of the incoming lane will be occluded. For an ego CAV i , the set of observed vehicles at time instant k , is denoted by $\mathbb{O}_i(k)$.

6.4.3 Information sharing using V2V communication

Using the onboard sensor suite and computation capabilities, we know that each CAV can track a group of vehicles in its ‘line of sight’ as discussed in 6.4.2. This tracked vehicle information, including predicted velocities and positions of neighboring vehicles, can then be shared with other CAVs within V2V communication range. Therefore, the ego CAV i obtains 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 incoming 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 $\mathbb{I}_i(k)$.

$$\mathbb{W}_i(k) = \mathbb{I}_i(k) \cap (\mathbb{O}_i(k) \cup \mathbb{C}_i(k)) \quad (6.1)$$

The MI-MPC controller of CAV i will then be provided the predicted states of the vehicles in $\mathbb{W}_i(k)$. The details of the MI-MPC can be found in [83] and [84]. Overall the MI-MPC controller generates safe overtaking trajectories and the associated steering and throttle controls based on the

final observation state $\mathbb{W}_i(k)$ provided to it.

6.4.4 Experiments and results

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.

In order to highlight the benefits of the proposed communication-based 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 information case, CAVs make the unrealistic assumption of having access to the states of all its neighboring vehicles irrespective of sensor occlusion status. We utilize two key metrics to compare the performances of these three algorithms: The number of overtakes attempted per CAV and the Success ratio of completed overtakes to attempted overtakes. Here, overtakes attempted tracks the number of times the CAV moves into the incoming 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 regard to these two evaluation metrics with varying traffic flow conditions is depicted in Fig. 6.3 and Fig. 6.4. Here, the flow rate represents the vehicles per minute entering into 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.

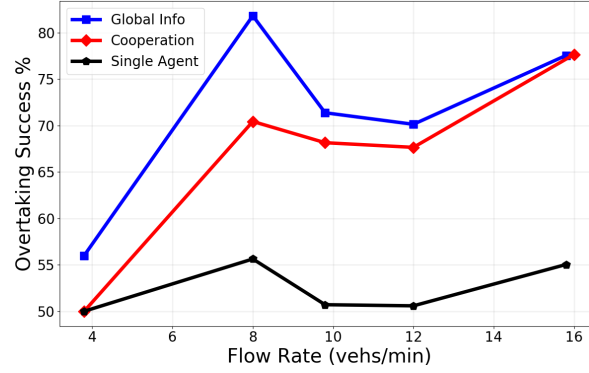


Figure 6.3: Overtaking success over varying flow rate.

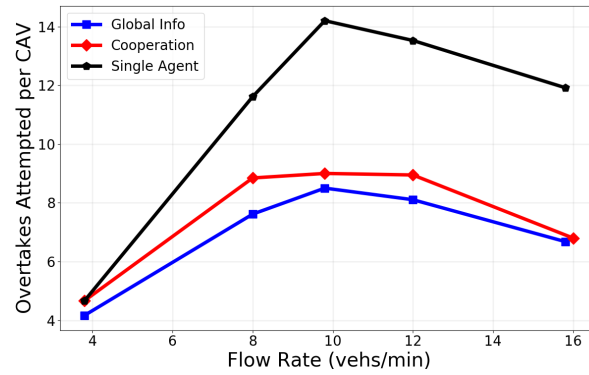


Figure 6.4: Overtakes attempted over varying flow rate.

From Fig. 6.3, we observe that the cooperation-based 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 increasing input flow 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 and leads to less risky overtaking attempts and more overtaking successes. As expected the unrealistic global information method outperforms our cooperative method. However, we observe that this difference reduces with increasing

input flow rate for the same reason involving an increased number of CAVs in the communication range. From Fig. 6.4, 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 single agent and global info strategies in Fig. 6.4.

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

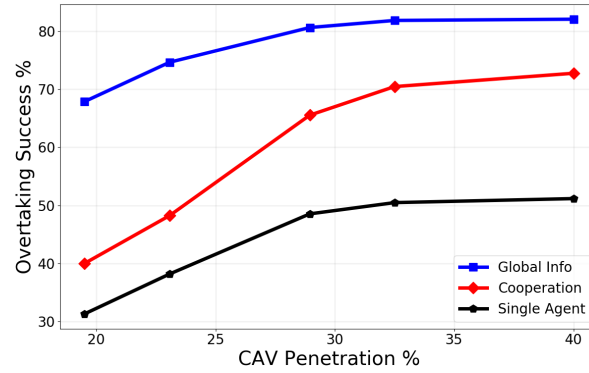


Figure 6.5: Overtaking success over CAV penetration level.

Fig. 6.5 shows an increase in successful overtakes in the cooperative method with increasing CAV penetration. This is due to the increase in the number of CAVs within the communication range providing information to the ego CAV (i.e. increased CAV penetration leads to increased information available for decision-making).

In Fig. 6.5 and 6.6 we observe that the overall performance of our cooperative algorithm

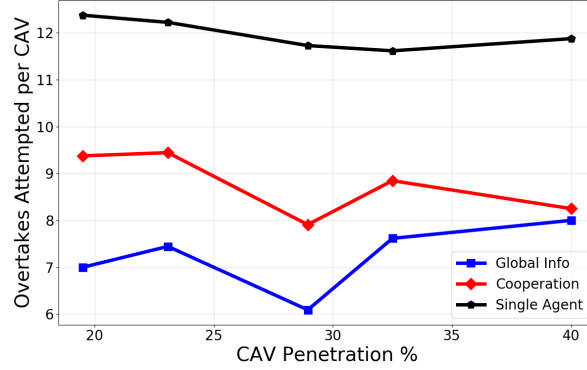


Figure 6.6: Overtakes attempted over CAV penetration level.

lies in between that of the global info and single agent methods. On average we achieve around 40% improvement over the single agent method in terms of successful overtakes and reduced unnecessary overtaking attempts. We also find that as the CAV penetration increases our method approaches the performance of the global information approach.

6.5 Graph Networks for Highway Traffic State Estimation in Mixed Traffic Settings

Modern CAVs often have a plethora of onboard sensing and communication devices. These devices generate huge amounts of data every second which provide the CAV with situational awareness and information about its environment. However, onboard data storage available in CAVs is often limited and this leads to data being discarded after it is used. While this is adequate for certain applications, some other applications require data to be accumulated over extended periods of time in order to achieve suitable performance levels. Examples of these applications include reliable trajectory prediction of on-road entities, human driving behavior classification, long-term traffic prediction, and determining trigger factors for potentially hazardous situations. The lack

of information in an adequately long time horizon is the primary performance bottleneck faced by these systems. This issue highlights the need for suitable knowledge representation and knowledge compaction, a process that involves extracting all the data points relevant to these applications and representing the extracted data in a space-efficient data structure. Some of the more commonly used data structures for these tasks are graphs. Graphs offer flexibility in terms of adding and removing information (nodes), and relationships (edges) and provide rapid methods for sorting and searching through the represented information.

The data collected by CAVs regarding on-road agents and environmental points of interest have both spatial and temporal information. Due to this even after graph-based methods are selected, there are still multiple ways in which the data can be organized within the graph. For example, the on-road agents detected can be considered as nodes on the graph with temporal information related to each agent stored within each node. The edges can then be based on which vehicle observes the agent represented as a node. Edges may also be used to represent the spatial location of each of the agents depicted in the graph. Similarly, many other possible representations exist each with its own advantages and disadvantages.

6.5.1 Dynamic multi-graphs for representing traffic state

The goal of this section is to define a suitable structure to store the information related to the traffic state in the vicinity of a CAV in a compact, efficient, and descriptive manner. When a graph is used to represent the micro-level information and interactions of on-road agents, a key concern is the decision on when to add or remove agents from this graph. As vehicles move in

and out of CAV sensor regions the nodes of the graph that represent these vehicles need to change accordingly. This results in a dynamically changing graph structure in which nodes and edges may be added or removed at any time.

Furthermore, as discussed earlier, when dealing with on-road agents there are many different types of information and interconnectivity that can be represented by the nodes and edges of the graph. For example, edges can represent the spatial distribution of agents, the availability of communication among agents, the willingness of agents to collaborate, or even the influence individual vehicles have on one another. One concept that can be used to represent all this information is the use of co-evolving multi-graphs [85] as shown in figure 6.7. These multi-graph involves multiple layers of interlinked graphs which evolve alongside one another. An example implementation of these co-evolving multi-graphs in the CAV domain could involve having the nodes of each graph layer represent the individual on-road agents and the edges between these nodes in each graph layer would represent the spatial distribution of agents, communication links between agents, influence network among agents or trust levels between agents. However, as this CAV domain is rapidly changing with vehicles entering and exiting the sensor regions, an important extension is to allow for dynamic changes to the graph structure. The resulting model would then be a dynamic co-evolving multi-graph for traffic state representation.

6.5.2 Global State Graphs by combining local CAV state graphs

While we discussed how to generate and utilize a dynamic co-evolving multi-graph representation of the traffic state in the local region around a CAV, an important extension involves the

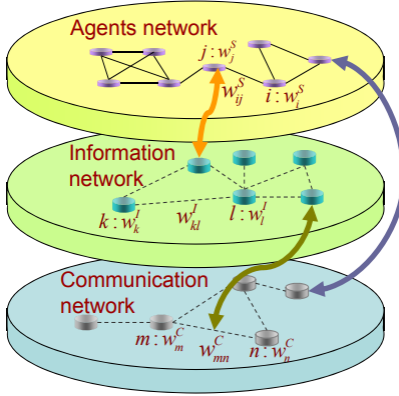


Figure 6.7: Co-evolving Multi-graph Model

ability of CAVs to communicate and share this information in order to get a better understanding of the traffic states beyond its sensor coverage region. Many factors need to be considered when combining multiple local graphs communicated into a global state graph. First, we need to identify what information needs to be communicated. Is it viable to send the entire local graph or do we need to condense the information and send a select portion of the available information? This may be of increased interest in the case of poor communication network conditions with low bandwidth availability. Once the local graphs are communicated, the process of combining the graphs also needs to be handled carefully. Nodes in the same layer of two different local graphs need to be connected with appropriate edges (connections), In this phase, additional problems may arise such as whose information should be trusted if the same node/edge information is available with conflicting data in multiple different local graphs. Another important consideration is whether this global graph is formed in a decentralized manner on board each CAV or if the global graph should be created at one centralized location and then shared with the relevant agents.

Once a global graph of traffic states has been created, this opens the door to many real-world applications. Individual CAVs can detect and avoid hazardous objects/road conditions downstream

of their location. Can perform accurate detection and prediction of unwanted conditions such as traffic shock waves and impending bottlenecks. Once detected/predicted proactive control can be applied to upstream CAVs often leading to even the complete prevention of the occurrence of the unwanted traffic condition. This vast collection of micro-level traffic information can also be used by macro-level centralized path planning and route guidance systems in order to achieve far superior performance to current systems often based on random or point sampling of traffic streams [86]. Additionally, these existing systems rely heavily on fixed infrastructure facilities such as cameras and induction loops which limits the data collection to the monitored regions which represent a very small fraction of the overall road network. Our proposed system of information sharing and graph-based representation would allow these macro-level algorithms to collect and use data from almost all points in highway and road networks.

6.5.3 GNN-based behavior and traffic prediction on graphs

The use of graph-based algorithms related to vehicles is a recently explored topic with a focus placed on macro-level traffic prediction [87]. Here, the nodes and edges of the graph often represent the traffic conditions present on links of the highway network [88], [89]. As discussed previously, our proposed approach overcomes many shortcomings of existing methods and allows for the gathering of data at any point of the road network. Additionally, as information from increasingly many CAVs is communicated and added to the global traffic state graph, this information can then be condensed and represented as the evolving traffic state on each of the links in the road network. This would allow existing algorithms to be directly used on top of the pro-

posed novel data collection and representation method. Furthermore, as this new method contains a lot more in-depth data related to the traffic state future algorithms can utilize this information to make better predictions and compute better controls for applications such as route guidance, ramp metering, and variable speed limits.

There is also attention placed on representing micro-level information in graphs using fixed interaction graphs [90] and graph self-attention networks [91]. This information can then be used to make predictions regarding the individual trajectories of the tracked vehicles. These graph-based methods have the advantage of being able to capture the interactions between vehicles in addition to the dynamics of each of the vehicles. Our proposed approach separates out different relationships among vehicles into separate layers. When we use graph neural network (GNN) methods on these individual layers we can discover the impact communication, interaction, influence, cooperation, and spatial dynamics have on the trajectory prediction process. By incorporating this additional information we expect to achieve high-accuracy trajectory prediction.

Chapter 7: Conclusion and Future Work

This Ph.D. dissertation focuses on some of the multi-faceted research opportunities in the domain of connected autonomous vehicles (CAVs). We focus on creating inherently safe control methods to solve some of the persistent problems in modern road networks, such as highway merge junction bottlenecks, traffic shock waves, and urban driving with complex intersections. In doing this we attempt to harness the advanced sensing, computation, and communication capabilities onboard CAVs. We introduced a novel constrained optimization-based approach to cooperative highway on-ramp merging of heterogeneous CAVs. We then used the capability of auction theory to extend this work to dynamic unsignalized intersections. Building on this we proposed a mixed integer linear programming solution for a large-scale network with multiple signalized intersections and mixed traffic. We also introduced a traffic shock wave dissipation method for multi-lane highways involving V2V communication-based cooperative control of CAVs.

In solving these described problems, we address multiple different strategies involving both centralized or decentralized control which may require different levels of V2I or V2V communication. We also explore the use of optimization-based control, heuristic control, and learning-based control. Thus, our proposed methods cover a wide array of application scenarios and we use control methods whose performance is best suited to the given environment. It is important to note

that safety guarantees and real-time operation capability are fundamental features built into all our proposed methods.

A primary goal of our research involves the creation of algorithms that can function well under realistic conditions. Therefore we develop methods that can perform real-time computations while addressing details such as heterogeneous traffic, multiple lanes, imperfections in communications, and the presence of mixed traffic. With our focus on communications, we attempt to create robust control algorithms which can function adequately even in the presence of imperfect communication networks. This also involves finding the relationships between network capability, minimum information shared, the importance of knowledge representation, and overall system performance achieved. Finally, the performance of our proposed methods is verified through realistic traffic simulation software such as the SUMO platform.

Future Work

Future research directions have been outlined within the chapters of this dissertation and can be summarized as follows.

Future directions in highway merging

The extension of our work on real-time cooperative highway merging to utilize a decentralized strategy for a multi-lane highway, that can be implemented anywhere without the need for infrastructure. The implementation of this strategy also involves the search for the best method for vehicle grouping for decentralized control.

A learning-based control algorithm can also be adopted for this decentralized merging approach. A deep reinforcement learning solution to this problem may be able to capture information and relationships among vehicles that are not considered in the optimization-based formulations.

Comparison of our centralized hybrid rule and optimization approach against a full-scale optimization formulation. We did not implement a full-scale optimization due to the difficulty in getting such a system to be real-time operation capable. But with advancements in computation, this may be an area that deserves to be revisited.

Exploring the macroscopic effects of the proposed merging method is an exciting future direction where the centralized merging method can be used to carry out tasks such as ramp metering. For example, a governing body could impose constraints that limit the number of ramp vehicles allowed onto the highway at a given time of day.

Future directions in unsignalized intersection control

A learning-based control algorithm could possibly be adopted for this centralized intersection coordination task. This could be applied in the form of an all-encompassing system that takes in the overall observation state and computes the control actions for each CAV or it can be implemented only for the order selection process while the actual computation of CAV controls to achieve this sequence is left to a separate optimization-based controller.

Future directions in large-scale signalized multi-intersection control

This work may need to be extended to handle heterogeneous traffic needs before it can be practically implemented. Different classes of traffic often have different needs which need to be catered to individually.

This method may also benefit from learning-based approaches, especially in the computation of usage probabilities in the central traffic light command portion of the hierarchical controller. The ability to over-time learn the usage patterns in a network and use this to provide guidance to each intersection's control center may prove invaluable in terms of both performance and efficiency gains.

Future directions in shock wave dissipation control

The work on shock wave dissipation can also be extended with a focus on the use of deep reinforcement learning methods to capture more details/features present in the information available to the CAVs. While we provide an initial deep-RL solution to this problem, as it stands it is unable to outperform state-of-the-art heuristic-based methods. Further exploration of possible observation spaces, action spaces, reward structures, and network models could vastly improve results. We may also experiment with the use of Recurrent Neural Networks (RNN) and LSTM units in order to capture the temporal relationships related to a single vehicle's behavior.

Future directions in V2X networks and robust control algorithms

With a focus on the communication capabilities of CAVs, we can build an uncomplicated communication framework in order to test the robustness of our methods to imperfect communication conditions. All the application domains discussed above can benefit from the ability to test the developed methods under realistic imperfect network conditions. Achieving a suitable level of robustness may then require changes in the existing and proposed communication-based control methods. We can also develop better graph-based traffic state representation methods, with an emphasis on sharing these representations in order to generate a global traffic state graph with multiple application possibilities.

Bibliography

- [1] United States Department of Transportation. 2023 progress report on the national roadway safety strategy, 2023 [Online].
- [2] United States National Highway Traffic Safety Administration. Early estimates of motor vehicle traffic fatalities and fatality rate by sub-categories through june 2022. DOT HS 813 405, 2022 [Online].
- [3] United States National Highway Traffic Safety Administration. The economic and societal impact of motor vehicle crashes, 2019 (revised). DOT HS 813 403, 2019 [Online].
- [4] INRIX. Global traffic scorecard 2022, 2022 [Online].
- [5] L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016.
- [6] Koohong Chung, Jittichai Rudjanakanoknad, and Michael J. Cassidy. Relation between traffic density and capacity drop at three freeway bottlenecks. *Transportation Res. Part B: Methodological*, 41(1):82–95, 2007.
- [7] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017.
- [8] M. Papageorgiou and A. Kotsialos. Freeway ramp metering: an overview. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):271–281, 2002.
- [9] H. Hadj-Salem, J. M. Blosseville, and M. Papageorgiou. Alinea: a local feedback control law for on-ramp metering; a real-life study. In *Third International Conference on Road Traffic Control*, pages 194–198, 1990.
- [10] M. Sarvi and M. Kuwahara. Microsimulation of freeway ramp merging processes under congested traffic conditions. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):470–479, 2007.
- [11] R. Scarinci, B. Heydecker, and A. Hegyi. Analysis of traffic performance of a merging assistant strategy using cooperative vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2094–2103, 2015.

- [12] T. Awal, L. Kulik, and K. Ramamohanrao. Optimal traffic merging strategy for communication- and sensor-enabled vehicles. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1468–1474, 2013.
- [13] J. Ding, L. Li, H. Peng, and Y. Zhang. A rule-based cooperative merging strategy for connected and automated vehicles. *IEEE Trans. on Intelligent Transp. Systems*, 21(8):3436–3446, 2020.
- [14] J. Rios-Torres, A. Malikopoulos, and P. Pisu. Online optimal control of connected vehicles for efficient traffic flow at merging roads. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2432–2437, 2015.
- [15] J. Rios-Torres and A. A. Malikopoulos. Automated and cooperative vehicle merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):780–789, 2017.
- [16] Y. Zhou, M. E. Cholette, A. Bhaskar, and E. Chung. Optimal vehicle trajectory planning with control constraints and recursive implementation for automated on-ramp merging. *IEEE Trans. on Intelligent Transportation Systems*, 20(9):3409–3420, 2019.
- [17] N. Chen, B. van Arem, T. Alkim, and M. Wang. A hierarchical model-based optimization control approach for cooperative merging by connected automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14, 2020.
- [18] S. Li, C. Wei, and Y. Wang. A ramp merging strategy for automated vehicles considering vehicle longitudinal and latitudinal dynamics. In *2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, pages 441–445, 2020.
- [19] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill. On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 900–906, 2012.
- [20] A. Uno, T. Sakaguchi, and S. Tsugawa. A merging control algorithm based on inter-vehicle communication. In *Proceedings 199 IEEE/IEEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, pages 783–787, 1999.
- [21] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S. Azuma. Multi-vehicle coordination on merging roads based on pseudo-perturbation-based broadcast control. In *2018 Annual American Control Conference (ACC)*, pages 4008–4013, 2018.
- [22] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Eva-marie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, pages 2575–2582. IEEE, November 2018.
- [23] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

- [24] 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.
- [25] S. Krauss, P. Wagner, and C. Gawron. Metastable states in a microscopic model of traffic flow. *Phys. Rev. E*, 55:5597–5602, May 1997.
- [26] Jakob Erdmann. Lane-changing model in sumo. In *SUMO2014*, volume 24 of *Reports of the DLR-Institute of Transportation Systems Proceedings*, pages 77–88. Deutsches Zentrum für Luft- und Raumfahrt e.V., May 2014.
- [27] Nilesh Suriyarachchi, Rien Quirynen, John S. Baras, and Stefano Di Cairano. Optimization-based coordination and control of traffic lights and mixed traffic in multi-intersection environments. In *2023 American Control Conference (ACC)*, 2023.
- [28] H. Hadj-Salem, J.M. Blosseville, and M. Papageorgiou. Alinea: a local feedback control law for on-ramp metering; a real-life study. In *Third International Conference on Road Traffic Control, 1990.*, pages 194–198, 1990.
- [29] Ioannis Papamichail, Katerina Kampitaki, Markos Papageorgiou, and Albert Messmer. Integrated ramp metering and variable speed limit control of motorway traffic flow. *IFAC Proceedings Volumes*, 41(2):14084–14089, 2008. 17th IFAC World Congress.
- [30] Offer Grembek, Alex A Kurzhanskiy, Aditya Medury, Pravin Varaiya, and Mengqiao Yu. Introducing an intelligent intersection. *Intelligent Transportation Systems Reports*, 2018.
- [31] 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.
- [32] Rohan Chandra and Dinesh Manocha. Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging. *IEEE Robotics and Automation Letters*, 2022.
- [33] Dustin Carlino, Stephen D Boyles, and Peter Stone. Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 529–534. IEEE, 2013.
- [34] Muhammed O Sayin, Chung-Wei Lin, Shinichi Shiraishi, Jiajun Shen, and Tamer Basar. Information-driven autonomous intersection control via incentive compatible mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):912–924, 2018.
- [35] David Rey, Michael W Levin, and Vinayak V Dixit. Online incentive-compatible mechanisms for traffic intersection auctions. *European Journal of Operational Research*, 2021.

- [36] Noam Buckman, Alyssa Pierson, Wilko Schwarting, Sertac Karaman, and Daniela Rus. Sharing is caring: Socially-compliant autonomous intersection negotiation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6136–6143, 2019.
- [37] 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.
- [38] Nan Li, Yu Yao, Ilya Kolmanovsky, Ella Atkins, and Anouck R Girard. Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections. *IEEE Trans. on Intelligent Transportation Systems*, 2020.
- [39] Ran Tian, Nan Li, Ilya Kolmanovsky, Yildiray Yildiz, and Anouck R Girard. Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation. *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [40] Andreas A. Malikopoulos and Liuhui Zhao. Optimal path planning for connected and automated vehicles at urban intersections. In *IEEE Conference on Decision and Control (CDC)*, pages 1261–1266, 2019.
- [41] Yougang Bian, Shengbo Eben Li, Wei Ren, Jianqiang Wang, Keqiang Li, and Henry X. Liu. Cooperation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control. *IEEE Transactions on Industrial Electronics*, 67(12):10744–10754, 2020.
- [42] Huaxin Pei, Yuxiao Zhang, Yi Zhang, and Shuo Feng. Optimal cooperative driving at signal-free intersections with polynomial-time complexity. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2021.
- [43] Seyed Alireza Fayazi and Ardalan Vahidi. Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing. *IEEE Transactions on IV*, 3(3):287–299, 2018.
- [44] Alessandro Paolo Capasso, Paolo Maramotti, Anthony Dell’Eva, and Alberto Broggi. End-to-end intersection handling using multi-agent deep reinforcement learning. *arXiv preprint arXiv:2104.13617*, 2021.
- [45] Junha Roh, Christoforos Mavrogiannis, Rishabh Madan, Dieter Fox, and Siddhartha S Srinivasa. Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections. *arXiv preprint arXiv:2011.03894*, 2020.
- [46] Tim Roughgarden. *Twenty lectures on algorithmic game theory*. Cambridge University Press, 2016.
- [47] Mingyu Wang, Negar Mehr, Adrien Gaidon, and Mac Schwager. Game-theoretic planning for risk-aware interactive agents. *IROS*, 2020.

- [48] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [49] GameOpt. Gameopt project supplementary material, 2022.
- [50] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Eva-marie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
- [51] Mohamad Hafizulazwan Mohamad Nor and Toru Namerikawa. Optimal motion planning of connected and automated vehicles at signal-free intersections with state and control constraints. *SICE Journal of Control, Measurement, and System Integration*, 13(2):30–39, 2020.
- [52] Nilesh Suriyarachchi, Rohan Chandra, John S. Baras, and Dinesh Manocha. Gameopt: Optimal real-time multi-agent planning and control for dynamic intersections. In *2022 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2022.
- [53] Xiaomeng Du, Meng Su, Xiaoquan Zhang, and Xiaona Zheng. Bidding for multiple keywords in sponsored search advertising: Keyword categories and match types. *Information Systems Research*, 28(4):711–722, 2017.
- [54] Riccardo Colini-Baldeschi, Stefano Leonardi, Monika Henzinger, and Martin Starnberger. On multiple keyword sponsored search auctions with budgets. *ACM Transactions on Economics and Computation (TEAC)*, 4(1):1–34, 2015.
- [55] Daniel Krajzewicz, Elmar Brockfeld, Jürgen Mikat, Julia Ringel, Christian Feld, Wolfram Tuchscheerer, Peter Wagner, and Richard Woesler. Simulation of modern traffic lights control systems using the open source traffic simulation sumo. In *Proceedings of the 3rd Industrial Simulation Conference*, pages 299–302, 06 2005.
- [56] Myungeun Eom and Byung-In Kim. The traffic signal control problem for intersections: a review. *European Transport Research Review*, 12(1):50, Sep 2020.
- [57] Tulga Ersal, Ilya Kolmanovsky, Neda Masoud, Necmiye Ozay, Jeffrey Scruggs, Ram Vasudevan, and Gábor Orosz. Connected and automated road vehicles: state of the art and future challenges. *Vehicle system dynamics*, 58(5):672–704, 2020.
- [58] Jackeline Rios-Torres and Andreas A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017.
- [59] Anurag Kanungo, Ayush Sharma, and Chetan Singla. Smart traffic lights switching and traffic density calculation using video processing. In *Recent Advances in Eng. and Computat. Sciences (RAECS)*, pages 1–6, 2014.

- [60] Bart De Schutter and Bart De Moor. Optimal traffic light control for a single intersection. *Europ. J. Control*, 4(3), 1998.
- [61] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. A deep reinforcement learning network for traffic light cycle control. *IEEE Trans. Vehicular Technology*, 68(2):1243–1253, 2019.
- [62] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 2496–2505. Association for Computing Machinery, 2018.
- [63] Di Liu, Wenwu Yu, Simone Baldi, Jinde Cao, and Wei Huang. A switching-based adaptive dynamic programming method to optimal traffic signaling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11):4160–4170, 2020.
- [64] Behrang Asadi and Ardalan Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Trans. Control Systems Technology*, 19(3):707–714, 2010.
- [65] Giovanni De Nunzio, Carlos Canudas De Wit, Philippe Moulin, and Domenico Di Domenico. Eco-driving in urban traffic networks using traffic signals information. *Int. J. Robust Nonlinear Control*, 26(6), 2016.
- [66] G. Lee, J. Guo, K. J. Kim, P. Orlik, H. Ahn, S. Di Cairano, and W. Saad. Edge computing for interconnected intersections in internet of vehicles. In *Proc. IEEE Intell. Veh. Symp.*, pages 480–486, 2020.
- [67] F. Ye, J. Guo, K. J. Kim, P. V. Orlik, H. Ahn, S. Di Cairano, and M. J. Barth. Bi-level optimal edge computing model for on-ramp merging in connected vehicle environment. In *Proc. IEEE Intell. Veh. Symp.*, pages 2005–2011, 2019.
- [68] Shreejith Ravikumar, Rien Quirynen, Akshay Bhagat, Eyad Zeino, and Stefano Di Cairano. Mixed-integer programming for centralized coordination of connected and automated vehicles in dynamic environment. In *Proc. IEEE Conf. Control Technology and Applications*, 2021.
- [69] Roya Firoozi, Rien Quirynen, and Stefano Di Cairano. Coordination of autonomous vehicles and dynamic traffic rules in mixed automated/manual traffic. In *2022 American Control Conference (ACC)*, pages 1030–1035, 2022.
- [70] Andrés F. Acosta, Jorge E. Espinosa, and Jairo Espinosa. TraCI4matlab: Enabling the Integration of the SUMO Road Traffic Simulator and Matlab® Through a Software Re-engineering Process. In *Modeling Mobility with Open Data*, pages 155–170. Springer, 2015.

- [71] Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiro Nishinari, Shin ichi Tadaki, and Satoshi Yukawa. Traffic jams without bottle-necks—experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10(3):033001, Mar 2008.
- [72] Raphael E. Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, R’mani Haulcy, Hannah Pohlmann, Fangyu Wu, Benedetto Piccoli, Benjamin Seibold, Jonathan Sprinkle, and Daniel B. Work. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221, 2018.
- [73] Amr Ibrahim, Mladen Čičić, Dip Goswami, Twan Basten, and Karl Henrik Johansson. Control of platooned vehicles in presence of traffic shock waves. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1727–1734, 2019.
- [74] Yang Zheng, Jiawei Wang, and Keqiang Li. Smoothing traffic flow via control of autonomous vehicles. *IEEE Internet of Things Journal*, 7(5):3882–3896, 2020.
- [75] Nassim Motamedidehkordi, Martin Margreiter, and Thomas Benz. Shockwave suppression by vehicle-to-vehicle communication. *Transportation Research Procedia*, 15:471–482, 2016. International Symposium on Enhancing Highway Performance (ISEHP), June 14-16, 2016, Berlin.
- [76] R. Wiedemann and U. Reiter. Microscopic traffic simulation: the simulation system mission, background and actual state. *Project ICARUS (V1052) Final Report. Brussels, CEC. 2: Appendix A*, 1992.
- [77] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, 62:1805–1824, Aug 2000.
- [78] Abdul Rahman Kreidieh, Cathy Wu, and Alexandre M Bayen. Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1475–1480, 2018.
- [79] Nilesh Suriyarachchi, Christos Mavridis, and John S. Baras. Cooperative multi-lane shock wave detection and dissipation via local communication. In *2022 30th Mediterranean Conference on Control and Automation (MED)*, pages 1080–1086, 2022.
- [80] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [81] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.
- [82] Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning, 2017.

- [83] 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, 2022.
- [84] 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)*, pages 3132–3139, 2022.
- [85] John S. Baras. A fresh look at network science: Interdependent multigraphs models inspired from statistical physics. In *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 497–500, 2014.
- [86] Guillaume Leduc. Road traffic data: Collection methods and applications. *Working Papers on Energy, Transport and Climate Change*, 1(55):1–55, 2008.
- [87] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2020.
- [88] Ken Chen, Fei Chen, Baisheng Lai, Zhongming Jin, Yong Liu, Kai Li, Long Wei, P. Wang, Yandong Tang, Jianqiang Huang, and Xiansheng Hua. Dynamic spatio-temporal graph-based cnns for traffic flow prediction. *IEEE Access*, 8:185136–185145, 2020.
- [89] Kan Guo, Yongli Hu, Zhen Qian, Hao Liu, Ke Zhang, Yanfeng Sun, Junbin Gao, and Baocai Yin. Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):1138–1149, 2021.
- [90] Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3960–3966, 2019.
- [91] Luyao Ye, Zezhong Wang, Xinhong Chen, Jianping Wang, Kui Wu, and Kejie Lu. Gsan: Graph self-attention network for learning spatial-temporal interaction representation in autonomous driving. *IEEE Internet of Things Journal*, pages 1–1, 2021.