

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

A SELF-LEARNING INTERSECTION CONTROL SYSTEM FOR CONNECTED AND AUTOMATED VEHICLES

by
Ardeshir Mirbakhsh

This study proposes a Decentralized Sparse Coordination Learning System (DSCLS) based on Deep Reinforcement Learning (DRL) to control intersections under the Connected and Automated Vehicles (CAVs) environment. In this approach, roadway sections are divided into small areas; vehicles try to reserve their desired area ahead of time, based on having a common desired area with other CAVs; the vehicles would be in an independent or coordinated state. Individual CAVs are set accountable for decision-making at each step in both coordinated and independent states. In the training process, CAVs learn to minimize the overall delay at the intersection. Due to the chain impact of taking random actions in the training course, the trained model can deal with unprecedented volume circumstances, the main challenge in intersection management. Application of the model to a single-lane intersection with no turning movement as a proof-of-concept test reveals noticeable improvements in traffic measures compared to three other intersection control systems.

A Spring Mass Damper (SMD) model is developed to control platooning behavior of CAVs. In the SMD model, each vehicle is assumed as a mass, coupled with its preceding vehicle with a spring and a damper. The spring constant and damper coefficient control the interaction between vehicles. Limitations on communication range and the number of vehicles in each platoon are applied in this model, and the SMD model controls intra-platoon and inter-platoon interactions. The simulation result for a regular highway section reveals that the proposed platooning algorithm increases the maximum throughput by 29%

and 63% under 50% and 100% market penetration rate of CAVs. A merging section with different volume combinations on the main section and merging section and different market penetration rates of CAVs is also modeled to test inter-platoon spacing performance in accommodating merging vehicles. Noticeable travel time reduction is observed in both mainline and merging lanes and under all volume combinations in 80% and higher MPR of CAVs.

For a more reliable assessment of the DSCLS, the model is applied to a more realistic intersection, including three approaching lanes in each direction and turning movements. The proposed algorithm decreases delay by 58%, 19%, and 13% in moderate, high, and extreme volume regimes, improving travel time accordingly. Comparison of safety measures reveals 28% improvement in Post Encroachment Time (PET) in the extreme volume regime and minor improvements in high and moderate volume regimes. Due to the limited acceleration and deceleration rates, the proposed model does not show a better performance in environmental measures, including fuel consumption and CO₂ emission, compared to the conventional control systems. However, the DSCLS noticeably outperforms the other pixel-reservation counterpart control system, with limited acceleration and deceleration rates. The application of the model to a corridor of four interactions shows the same trends in traffic, safety, and environmental measures as the single intersection experiment.

An automated intersection control system for platooning CAVs is developed by combining the two proposed models, which remarkably improves traffic and safety measures, specifically in extreme volume regimes compared to the regular DSCLS model.

**A SELF-LEARNING INTERSECTION CONTROL SYSTEM
FOR CONNECTED AND AUTOMATED VEHICLES**

**by
Ardeshir Mirbakhsh**

**A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Transportation**

John A. Reif Jr. Department of Civil and Environmental Engineering

May 2022

Copyright © 2022 by Ardeshir Mirbakhsh

ALL RIGHTS RESERVED

APPROVAL PAGE

**A SELF-LEARNING INTERSECTION CONTROL SYSTEM
FOR CONNECTED AND AUTOMATED VEHICLES**

Ardeshir Mirbakhsh

Dr. Joyoung Lee, Dissertation Advisor Date
Associate Professor of Civil and Environmental Engineering, NJIT

Dr. Steven Chien, Committee Member Date
Professor of Civil and Environmental Engineering, NJIT

Dr. Lazar Spasovic, Committee Member Date
Professor of Civil and Environmental Engineering, NJIT

Dr. Branislav Dimitrijevic, Committee Member Date
Assistant Professor of Civil and Environmental Engineering, NJIT

Dr. Guiling Wang, Committee Member Date
Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Ardeshir Mirbakhsh

Degree: Doctor of Philosophy

Date: May 2022

Undergraduate and Graduate Education:

- Doctor of Philosophy in Transportation, New Jersey Institute of Technology, Newark, NJ, 2022
- Master of Science in Highway and Transportation Engineering, Shahid Chamran University, Ahvaz, Iran, 2014
- Bachelor of Science in Civil Engineering, Malayer University, Malayer, Iran, 2012

Major: Transportation

Presentations and Publications:

Ardeshir Mirbakhsh, Jo Young Lee, and Dejan Besenski "A Spring-Mass-Damper-Based Platooning Logic for Connected and Automated Vehicles", The Transportation Research Board (TRB) 101st Annual Meeting, Washington DC, USA, January 2022.

Ardeshir Mirbakhsh, Jo Young Lee, and Dejan Besenski "A Self-learning Intersection Control System for Connected and Autonomous Vehicles", IEEE Transactions on Intelligent Vehicles, April 2022, Under Review.

Ardeshir Mirbakhsh, Joyoung Lee, "A Signal-free intersection control system for AVs", Planung Transport Verkehr (PTV) North America Annual User Meeting Talks, November 2021.

Ardeshir Mirbakhsh, Joyoung Lee, "The Stability comparison of Two Platooning algorithms: VISSIM 2020 Platooning System and the Spring-Mass-Damper System", Planung Transport Verkehr (PTV) Student Session Poster, August 2020.

Ardeshir Mirbakhsh, Joyoung Lee, "Collective Assessment of Active Traffic Management Policies", Planung Transport Verkehr (PTV) North America Annual User Meeting Talks, November 2020.

In dedication to my beloved mother and memory of my incredible father

تقدیم به وجود پر مهر مادر فداکارم و به یاد پدر عزیزم

ACKNOWLEDGMENT

Foremost, I would express my sincere gratitude to my dear advisor Dr. Jo Young Lee, for the appreciable support of my Ph.D. study and research, for his immense knowledge, enthusiasm, patience, positive thinking, communication, and motivation. He is one of the most influential individuals in my life.

I am also thankful to Dr. Steven Chien, Dr. Lazar Spasovic, Dr. Branislav Dimitrijevic, and Dr. Guiling Wang for serving as my dissertation committee members and their constructive input on this research.

My sincere thanks go to NJIT's Intelligent Transportation Systems Resource Center for involving me in several practical projects and for their financial support.

Finally, I am grateful to other researchers and friends for their technical and emotional support, particularly Dr. Ravi Jagirdar, Mr. Abdullah Traboulsi, Mr. Noujan Fakhri, Dr. Zijia Zhong, and my precious friend, Dr. Soheil Sohrabi.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement and Research Motivation	4
1.3 Research Scope and Objectives.....	6
1.4 Dissertation Organization.....	7
2 LITERATURE REVIEW.....	8
2.1 Rule-based Intersection Control Systems for CAVs.....	9
2.2 Optimization-based Intersection Control systems for CAVs	11
2.2.1 Online Optimization Intersection Control logic for CAVs.....	11
2.2.2 Trajectory Planning Intersection Control Logics for CAVs	14
2.3 Machine Learning-Based Intersection Control Logics for CAVs	16
2.4 Vehicles' Longitudinal Control Systems Background.....	20
2.4.1 Development and Assessment of CACC Models.....	22
2.4.2 SMD-based Longitudinal Vehicle's Control Systems.....	24
2.5 Chapter Summary.....	26
3 DEVELOPMENT OF A CAV-BASED INTERSECTION CONTROL.....	29
3.1 An Introduction to Reinforcement Learning.....	29
3.1.1 Finite Markov Decision Process.....	30
3.1.2 Returns, Episodes, And Learning Process.....	31

TABLE OF CONTENTS
(Continued)

Chapter	Page
3.2 Reinforcement Learning and Neural Networks.....	33
3.2.1 Deep Q-Networks.....	33
3.2.2 Multi-agent DQN.....	35
3.3 Decentralized Sparse Coordination Learning System for CAVs'.....	38
3.3.1 Testbed Settings and Reward Function Development.....	38
3.3.2 Training Settings and Learning Performance.....	42
3.4 Proof of Concept Test.....	45
3.4.1 Target Traffic Measures Comparison.....	46
3.5 Emission, Safety and Fuel Consumption.....	49
3.6 T-Test Result.....	52
3.7 Chapter Summary.....	54
4 DEVELOPMENT OF A PLATOONING POLICY FOR CAVs.....	55
4.1 SMD System and Vehicle Platooning.....	55
4.2 Characteristics of Spring Constant and Damping Coefficient.....	58
4.3 Developing a CAVs' Platooning Logic Based on SMD Model.....	59
4.4 Microsimulation Testbed Development.....	61
4.5 Proof of Concept Test.....	62
4.5.1 Spacing Error.....	62
4.6 Maximum Throughput Assessment.....	64

TABLE OF CONTENTS
(Continued)

Chapter	Page
4.7 Platoon Length Impact on Maximum Throughput.....	65
4.8 Travel Time at Merging Section.....	66
4.9 Chapter summary.....	70
5 EXPANSION OF DSCLS MODEL.....	71
5.1 Application of DSCLS to a Multilane Intersection.....	71
5.2 Evaluation of DSCL in the Multi-lane Intersection Setting.....	72
5.2.1 Traffic Measures Comparison.....	73
5.2.2 Other Measures Comparison.....	76
5.2.3 Statistical Analysis of the Simulation Results.....	79
5.3 Application of DSCLS to a Corridor of Four Intersections.....	81
5.3.1 Traffic Measures Comparison.....	81
5.3.2 Other Measures Comparison.....	83
5.3.3 Statistical Analysis of The Simulation Results.....	86
5.4 Development of a Platooning CAV-based Intersection Control System.....	88
5.4.1 Traffic Measures Comparison.....	89
5.4.2 Other Measures Comparison.....	92
5.4.3 Statistical Analysis of The Simulation Results.....	95
5.5 Chapter Summary.....	97
6 CONCLUSIONS AND FUTURE RESEARCH.....	99

TABLE OF CONTENTS
(Continued)

Chapter	Page
6.1 Conclusions.....	99
6.2 Research Contributions.....	101
6.3 Future Research.....	102
7 REFERENCES.....	104

LIST OF TABLES

Table	Page
3.1 Training Process Hyperparameters Setting.....	43
3.2 P-Value For T-Test Results – Proof of Concept Test.....	53
4.1 Ford Fusion 2019 Hybrid Specifications.....	57
4.2 Vissim Built-in Platooning System Constant Parameters Setting.....	63
5.1 P-values for T-test Results for DSCLS - Single Intersection.....	80
5.2 P-values for T-test Results for DSCLS – Corridor.....	87
5.3 P-values for T-test Results - DSCLS&SMD.....	96

LIST OF FIGURES

Figure	Page
3.1 The agent-environment interactions in MDP.....	31
3.2 The high-level training process of a DQN agent.....	34
3.3 Potential transition types in MADQL.....	37
3.4 The intersection environment.....	39
3.5 The transition from exploration to exploitation (decaying ϵ) in training.....	44
3.6 MAE change for four neural networks over learning progress.....	44
3.7 Global reward changes over learning progress.....	45
3.8 Average delay comparison for proof-of-concept test	47
3.9 Maximum throughput comparison for proof-of-concept test	48
3.10 Average travel time comparison for proof-of-concept test	49
3.11 Average fuel consumption comparison for proof-of-concept test	50
3.12 Average CO2 emission for proof-of-concept test comparison	50
3.13 Average PET comparison for proof-of-concept test.....	51
3.14 Average TTC comparison for proof-of-concept test.....	52
4.1 An n-body SMD system illustration	55
4.2 The SMD-based platooning system details.....	60
4.3 Integration of VISSIM and Python.....	61
4.4 Spacing error occurrence in harsh deceleration scenario.....	64
4.5 Maximum throughput under different MPR of platooning CAVs.....	65

LIST OF FIGURES
(Continued)

Figure	Page
4.6 Impact of platoon length on maximum throughput.....	66
4.7 Merging section scenario details.....	67
4.8 Travel time on the mainlane – mainlane volume = 1400 Veh/hr.....	68
4.9 Travel time on the mainlane – mainlane volume = 1800 Veh/hr.....	68
4.10 Travel time on the merging lane – mainlane volume = 1400 Veh/hr.....	69
4.11 Travel time on the merging lane – mainlane volume = 1800 Veh/hr.....	70
5.1 The intersection environment physical settings.....	72
5.2 Multi-lane intersection average delay comparison.....	74
5.3 Multi-lane intersection average travel time comparison.....	75
5.4 Multi-lane intersection maximum throughput comparison.....	76
5.5 Multi-lane intersection average fuel consumption.....	77
5.6 Multi-lane intersection average CO ₂ emission comparison.....	77
5.7 Multi-lane intersection average PET comparison.....	78
5.8 Multi-lane intersection average TTC comparison.....	79
5.9 Corridor’s average delay comparison.....	82
5.10 Corridor’s average travel time comparison.....	82
5.11 Corridor’s average fuel consumption comparison.....	83
5.12 Corridor’s average CO ₂ emission comparison.....	84
5.13 Corridor’s average PET comparison.....	85

LIST OF FIGURES
(Continued)

Figure	Page
5.14 Corridor’s average TTC comparison.....	86
5.15 Platooning CAV-based intersection control system.....	88
5.16 Average delay comparison, including DSCLS&SMD.....	90
5.17 Average travel time comparison, including DSCLS&SMD.....	91
5.18 Maximum throughput comparison, including DSCLS&SMD.....	91
5.19 Average fuel consumption comparison, including DSCLS&SMD.....	92
5.20 Average CO2 emission comparison, including DSCLS&SMD.....	93
5.21 Average PET comparison, including DSCLS&SMD.....	94
5.22 Average TTC comparison, including DSCLS&SMD.....	95

CHAPTER 1

INTRODUCTION

1.1 Background

Due to rapid population growth and the increased number of vehicles, traffic congestion, collisions, and pollution have become leading causes of decreased living standards. Only in 2017 and in the USA, traffic congestion has caused 8.8 billion hours of delay and 3.3 billion gallons of fuel waste, resulting in a total cost of 179 billion United States Dollars (USD) [1]. Traffic congestion has increased between 1 and 3 percent annually from 2008 to 2017 in the USA, and a national congestion cost of 237 billion USD is forecasted for 2025 [1]. Several studies prove that human errors play a pivotal role in traffic congestion and contribute to 75% of roadway crashes worldwide [2]. The global rate of road traffic death is 18.2 per 100,000 population. However, there is significant variation across different regions; the USA and Europe have the lowest regional rates with 15.6 and 9.3 death per 100,000 population, respectively [3]. Intersections play a crucial role in traffic delays and collisions among urban traffic facilities. In the United States, 44 percent of reported traffic accidents occur at urban intersections, leading to 8,500 fatalities and around 1 million injuries every year [4].

With advances in communication and sensing technologies within the last decade, vehicles are equipped with several driving assistant systems such as Adaptive Cruise Control (ACC), blind-spot monitors, back-up cameras, and lane centering. Some high-end cars produced within the last couple of years can perform most driving tasks without human interference, and the imminent widespread appearance of Connected and Autonomous Vehicles (CAVs) on the streets is expected. The most crucial feature of CAVs seems to be

the communication capability. Taking advantage of the CAVs' communication capability is promising in congestion relief and safety improvement at roadway facilities. Several previous studies have proved that the shorter following gap time between CAVs gained by Cooperative Adaptive Cruise Control (CACC) systems can increase the roadway segment capacity up to 100% [5]. Moreover, recent advances in computer science, data storage, and Artificial Intelligence (AI) have noticeably improved CAV-based intersection control systems. As a result, developing CACC models and CAV-based intersection control systems are hot topics in the transportation research world.

Considering CAVs as a revolution in the transportation industry, most public and private transportation-oriented agencies are involved with this phenomenon. In 2016 the U.S. Department of Transportation (USDOT) awarded a cooperative agreement collectively worth more than 45 million USD to three pilot Connected Vehicle (CV) sites in New York City, Wyoming, and Tampa [6]. Well-known car manufacturers, including Toyota, Honda, and Hyundai, are involved with the pilot program in Tampa [6]. In 2018, Waymo (Google) announced that its AVs have successfully driven 8 million miles on U.S. public roads [7]. Several companies such as TESLA and Cadillac have commercially produced level 3 to 4 CAVs [7]. General Motors is investing more than 20 billion USD until 2025 on its electric and CAV production [7]. In 2018 AUDI announced that the company would invest 16 billion USD on electric vehicles and AVs over the next five years. Ford has built an AVs testing facility in Miami and noticeably increased its partnerships and investments in technology companies; building fully automated vehicles by the end of 2021 is the primary goal of the Ford Company [7].

Several approaches, such as online optimization, trajectory planning, and rule-based intersection control logic for CAVs, have been around for the last couple of decades. Meanwhile, a revolution in decision-making optimization began in 2013 when researchers from a British startup called DeepMind developed a logic that could play almost any Atari game, such as “Space Invaders” and “Breakout” from scratch without prior knowledge of the rules and eventually outperforming humans. DeepMind researchers had combined Reinforcement Learning (RL) with Deep Learning (DL), and that was how Deep Reinforcement Learning (DRL) was born. Google bought DeepMind for 500 million USD in 2014. In 2017, a DeepMind robot called “AlphaGo” won the world champions of the “Go” game, an abstract strategy game for two people. Eventually, in 2019, Open AI’s robots won the “Dota2” match competition against the 2018 human world champions. Dota is a multiplayer battle arena game that requires strategy, teamwork, and quick decision to win. Building an AI to compete in such a game with numerous variables seems like an impossible challenge [8] [9]. If a DRL-based agent can compete in such extremely complicated circumstances, it can also be the CAVs’ mastermind at intersections too.

The development of CACC models dates back to the 1990s. However, field experiments have been accelerated within the last years due to technological advances and governmental encouragements. The highlights of deployments include the USDOT-approved CAV test sites, PATH (California Partner of Advanced Transportation Technology), and Energy ITS Project. Nevertheless, according to safety, technology, and financial limitations for large-scale field tests, the simulation is still one of the best approaches for mid to large deployment of CAVs’ evaluation [10].

1.2 Problem Statement and Research Motivation

In conventional intersection control systems, an intersection controller such as a traffic light dictates the rules to the vehicles. However, the recent advances in vehicles' communication systems demand communication from vehicles to controller systems to take full advantage of the CAVs' communication capabilities. Therefore, the CAV-based intersection control logic has been a point of interest for the last couple of decades. Several approaches such as trajectory planning, real-time optimization, and rule-based intersection control logics have been developed to establish a purposive vehicle to vehicle or vehicle to infrastructure communication at the intersections.

The most critical task in developing an intelligent intersection control system is to make the algorithm adjustable with stochastic or unprecedented circumstances. Since conventional optimization approaches are based on predefined models or fixed rules, making them work in a stochastic environment requires several adjustments and experiments. Finally, they are unlikely to produce a satisfactory outcome in extraordinary circumstances. Following by DRL's astounding performance in playing multi-player video games within the last few years, it is considered an outstanding Machine Learning (ML) technic for decision-making in stochastic environments. The chain impact of taking sequences of random actions in a DRL agent's learning process will expose the agent to enormous different circumstances, regardless of how likely they may happen in a real-world environment. Pieces of literature exist in using DRL to optimize traffic light phasing and timing or processing aerial images to optimize traffic flow at the intersections.

However, at an ultimate automation level, CAVs are expected to act as individual robots, and to the best of the author's knowledge, there is no DRL control system developed

controlling individual CAVs and making them accountable for decision-making in the traffic networks. A Decentralized Sparse Coordination Learning System (DSCLS) based on DRL is proposed in this study to control CAVs at the intersections. In this approach, vehicles try to reserve their desired cells ahead of time. Based on having a shared desired cell with other vehicles, they would be in an independent or coordinated state. Individual CAVs are set accountable for decision-making in both coordinated or independent states at each step. CAVs learn to minimize the overall delay and queue length at the intersection in the training process.

CACC is another promising technology that allows CAVs to be driven cooperatively. CACC introduces significant benefits to traffic flow and safety, and several CACC control systems have been developed within the last couple of decades. A noticeable portion of studies in this area is focused on the dynamics aspect of CACC, such as vehicle mass, tire friction, vehicle powertrain. These aspects are vital in bringing CACC to fruition yet providing limited insights into the impacts of CACC on the overall traffic network. Alternatively, most of the models developed explicitly for traffic assessment of CACC have missed several critical aspects of platooning such as platoon evolution process, communication range limitations, or interactions between platoons.

This study applies a classical physics-based model called Spring-Mass-Damper (SMD) that reflects the most critical dynamic aspect of vehicles, the mass, and covers the platoon evolution process for platooning CAV. A maximum communication range is reflected in the model to make it more compatible with real-world circumstances. Strings of vehicles are divided into sub-platoons to avoid lengthy platoons and accommodate potential merging vehicles, while the SMD model controls both inter-platoon and intra-

platoon interactions. The model is coded into commercial simulation software to facilitate traffic-oriented and potential macroscopic or mesoscopic assessments.

Considering platooning capability and automated intersection control systems as essential characteristics of CAVs, numerous studies have focused on these two areas. However, CACC platooning models are mainly developed and tested in uninterrupted flow circumstances, and the impact of platooning models on interrupted flow has not been examined. Few CAV-based intersection control systems can deal with platoons of vehicles. However, these models still lack a robust logic for platooning. This study also develops a platooning CAVs-based automated intersection control system for CAVs by simultaneously deploying the DSCLS and SMD models.

1.3 Research Scope and Objectives

The ultimate goal of this study is to develop a DSCLS based on DRL for platooning CAVs and compare its performance to other CAV-based and conventional intersection control systems. Toward this goal, the following objectives have been addressed.

- Develop a DSCLS to control individual CAVs at the intersection, and evaluate its performance in traffic, environmental, and safety measures compared to other intersection control systems.
- Develop an SMD-based platooning system for CAVs, evaluate its impact on maximum throughput at regular and merging roadway sections, and perform safety assessment by measuring spacing error between vehicles and comparing it with another platooning logic.
- Apply the DSCLS to a corridor, including several intersections, and compare its performance with other intersection control systems.
- Combine the proposed platooning logic and intersection control systems to develop a platooning CAVs-based intersection control system and compare its performance with conventional and CAV-based intersection control systems.

- Develop microsimulation testbed for all above scenarios.
- Develop a DSCLS to control individual CAVs at the intersection, and evaluate its performance in traffic, environmental, and safety measures compared to other intersection control systems.

1.4 Dissertation Organization

This dissertation is organized as follows; In Chapter 2, the CAV-based intersection control systems and CAVs' car-following models, also known as CACC, are reviewed. CAV-based intersection control systems are clustered into four categories: rule-based, optimization-based, trajectory planning-based, and ML-based; several prominent studies in each category are reviewed. Followed by that, the CAV car-following models are reviewed with a focus on the SMD model. Chapter 3 explains the general RL processes, formulation, and parameters. Specific RL approaches related to this study, including Deep Queue Networks (DQN) and Multi-Agent Deep Queue Networks (MADQN), are described. Next, the model's intersection environment design, training settings, and learning performance are presented. The proposed model's performance in controlling a single lane intersection is evaluated as a proof of concept test.

The SMD model is introduced in Chapter 4, including the model formulations and explaining the impact of hyperparameter settings on driving behavior. The development of the SMD-based CACC model is elaborated, and the model's performance is evaluated in several scenarios. In Chapter 5, the DSCLS is applied to a more realistic intersection, including several lanes and turning movements and a corridor of four intersections. A platooning CAV-based intersection control system is developed by combining the proposed platooning logic and intersection control system. The simulation results and

discussions are also presented in this chapter. Finally, Chapter 6 presents conclusions and a discussion of the future research.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews selected studies in two critical CAV-related research areas, including 1) CAV-based intersection control systems and 2) and Cooperative Adaptive Cruise Controls (CACC).

Based on the existing literature, CAV-based intersection control systems are clustered into three main groups, including 1) rule-based algorithms, 2) optimization-based algorithms, and 3) Machine Learning (ML)-based algorithms. Several studies in each group are reviewed, and it is clarified if any proposed intersection control logic can consider platoons of AVs instead of single vehicles. It should be noted that several studies have proposed a combination of the mentioned methodologies. In that case, they are included in the most related group.

Regarding the CACC systems, a brief history of vehicles' longitudinal control systems' evolution from basic Cruise Control (CC) systems to Cooperative Adaptive Cruise Control (CACC) is presented in this chapter. Several proposed CACC models and their impact on traffic flow measures are reviewed, and a brief background on the Spring-Mass-Damper (SMD) model is provided. It is also explained how previous studies have used the SMD model to reflect the impact of several variables such as driver aggressiveness, vehicle mass, and vehicle stability on CAVs' platooning behavior.

2.1 Rule-based Intersection Control Systems for CAVs

Most rule-based intersection logics are based on intersection space reservation approach. The space reservation approach was initially developed in 2004 by Dresner et al. [11]. This

method divides the intersection area into an $n \times n$ grid of reservation tiles. Each approaching vehicle to the intersection attempts to reserve a time-space block at the intersection area by transmitting a reservation request to the intersection manager. The reservation request includes information such as speed and arrival time. According to intersection control policy, the intersection manager decides whether to approve the request, provide more passing restrictions to the driver agent, or reject the reservation request. Dresner et al. adopted the First Come First Serve (FCFS) control policy, in which the passing priority is assigned to the vehicle with the earliest arrival time, and other vehicles have to yield to it. In a following study by Dresner et al. in 2008 [12], several complimentary regulations were added to FCFS policy to make it work more reliable, safe, and efficient. Simulation results revealed that FCFS policy noticeably reduces the intersection delay compared to traffic light and stop sign control systems. The reservation-based approach can be combined with various control policies and has been deployed by several researchers since 2008.

Zhang et al. proposed a state-action control logic based on a Priority First in First Out (PriorFIFO) [13]. This control model assumes autonomous motion with spatial-temporal and kinetic parameters based on a centralized scheduling mechanism. The target was to reduce control delay for vehicles with higher priority. The simulation results with a combination of high, average, and low priority vehicles showed that the algorithm works well for vehicles with higher priority. Meanwhile, causing some extra delay for regular vehicles compared to those with lower priority.

Carlin et al. developed an auction-based intersection control logic based on Clarke Groves tax mechanism and pixel reservation [14]. If commonly reserved tiles exist between vehicles, an auction is held between the involved vehicles. All vehicles in each direction

contribute to their leading vehicle to win the auction, and the control logic decides which leading vehicles receive a pass order first. The bid's winner and its contributors (followers) have to pay the runner-up bid amount with a proportional payment (based on their contribution value in the bid). A "system wallet" component was added to auction-based intersection control to ensure low-budget vehicles or emergency vehicles would not be over-delayed. Comparison of simulation results showed that the auction-based control logic outperforms the FIFO logic.

Vasirani et al. approached the intelligent intersection management problem in a mesoscopic scale and designed a competitive computational market to control a set of intersections in urban roadway networks [15]. In this algorithm, buyers are driver agents, the suppliers are the intersection managers, and the traded resource is the intersection capacity. Each vehicle communicates with the intersection manager and provides its desired route. The intersection manager adjusts the prices based on demand and supply values. Vehicles can reroute if the offered price is not desired, and the transactions are made as the equilibrium price is obtained. Mesoscopic simulation runs for a roadway network consisting of several intersections showed that deployment of the algorithm leads to travel time reduction for CAVs and density reduction at the network's critical sections.

Chen et al. developed the win-fit intersection control logic. In this algorithm, the "win" logic scores clusters of approaching vehicles to the intersection by the value of delay imposed on all other yielding clusters [16]. The cluster with the lowest delay impact on other clusters obtains the passing priority. The control algorithm's "fit" function assigns idle time slots (resulting from turning movements) to the vehicles in lower priority groups. Unlike the previously reviewed control system, the win-fit control logic could consider a

cluster of vehicles instead of a single-vehicle at each decision-making step. However, holding on to the platoon was not a priority. So the vehicles could leave their platoon to pass the intersection. A simulation run in SUMO revealed that the average delay at the intersection was improved compared to both FIFO and actuated traffic lights.

2.2 Optimization-based Intersection Control systems for CAVs

Several researchers have considered optimization approaches to develop CAV-based interaction control logic. A noticeable number of studies have focused on CAVs' trajectory planning to optimize the approach time. Meanwhile, other optimization logics, such as linear integer programming, are used by other researchers.

2.2.1 Online Optimization Intersection Control logic for CAVs

Yan et al. proposed a dynamic programming-based optimization system to find optimal vehicle passing sequence and minimize intersection evacuation time [17]. In this algorithm, the optimizer agent clusters vehicles into several groups so that each group of vehicles can pass the intersection simultaneously without a potential collision. Since vehicles are clustered into several groups, the conventional problem of finding an optimal passing order of vehicles is transformed into partitioning the vehicles into different groups and finding optimal group sequences to minimize the vehicle evacuation time. Approaching vehicles to the intersection provide their data to the controller agent, and the controller agent has to rerun the dynamic programming optimization if a new vehicle is detected. However, if a group of vehicles is authorized to pass the intersection, the recalculation process is delayed till the whole group passes the intersection.

Wu et al. deployed a timed Petri Net model to develop a simple intersection with two conflicting movement control logic [18]. The intersection control was considered a distributed system, with parameters such as vehicles' crossing time and time-space between successive vehicles. Two control logics, including 1) central controller and 2) car-to-car communication, were developed and tested. In central controller logic, approaching vehicles to the intersection provide their information to the controller center, and the controller center has to provide optimized passing order for the vehicle. While in car-to-car control logic, each leading vehicle collects its followers' data, and if the distance between them is shorter than a threshold, they would be considered a group. Leading vehicles of groups communicate with each other, and the passing order priority for each group is defined by its leader's proximity to the intersection. The optimization task was decomposed into chained sub-problems by a backtracking process, and the final optimal solution was found by solving single problems and applying forward dynamic programming to find the shortest path from the original problem to the last problem in the graph. The simulation results revealed that both algorithms have the same delay reduction performance, resulting in almost the same queue length. Wu et al. extended their solution to a more realistic 4-way intersection [19]. This problem was treated as a machine scheduling problem, considering vehicles as jobs, lanes as families, and the intersection control logic as a machine that can process multiple jobs. The optimization problem was solved by forward dynamic programming again. Simulation results revealed that the proposed approach outperforms traffic lights with fixed timing. In another following study, Wu et al. deployed Ant Colony System (ACS) to overcome combinatorial explosion in dynamic programming for a large number of vehicles [20]. The optimization problem was

converted to a graphical form, considering each vehicle as a node and then finding the shortest path covering all nodes using the ACS approach. The simulation results revealed that the ACS approach reduces the computation time and outperforms adaptive, FCFS, and fixed signals with increased throughput and decreased delay. An NXT robots-based prototype was also built to explore the feasibility of autonomous intersection management with the proposed method, and no collision was observed in the prototype testing process.

Fayazi et al. deployed Mixed-Integer Linear Programming (MILP) approach to optimize AVs' arrival time at the intersection [21]. In this approach, the intersection controller agent receives arrival time and departure time from approaching vehicles and optimizes the vehicles' arrival time. The optimization goal was to minimize the difference between the current time and the last vehicle's expected arrival time at the intersection. To ensure all vehicles are not forced to travel near the speed limit, the cost value was defined as the difference between each vehicle's assigned and desired crossing time. Several constraints, such as speed limit, maximum acceleration, minimum headway, and minimum cushion for conflicting movements, were applied to the model. A two-movement intersection simulation results showed that the MILP-based controller reduces average travel time by 70.5% and average stop delay by 52.4%. It was also approved that the control logic would encourage platooning under a specific gap setting.

Lee et al. proposed a Cumulative Travel-time Responsive (CTR) intersection control algorithm under CAVs' imperfect market penetration rate [22]. They considered the elapsed time spent by vehicles from when they entered the network to the current position as a real-time measure of travel time. The Kalman filtering approach was deployed to cover CAVs' imperfect market penetration rate for travel time estimation. Simulations were run

in VISSIM for an isolated intersection with 40 volume scenarios covering the volume capacity ratio ranging from 0.3 to 1.1 and different CAV market penetration rates. Simulation results showed that the CTR algorithm improves mobility measures such as travel time, average speed, and throughput by 34%, 36%, and 4%, compared to the actuated control system. The CO₂ emission and fuel consumption were also reduced by 13% and 10%. It was also revealed that the CTR would produce more significant benefits as the market penetration rate passes the threshold of 30%. More benefits are also expected as the total intersection volume increases.

2.2.2 Trajectory Planning Intersection Control Logics for CAVs

Lee et al. proposed a trajectory planning-based intersection control logic named Cooperative Vehicle Intersection Control (CVIC) [23]. The control algorithm provided a location-time diagram (trajectory) of individual vehicles and minimized the length of overlapped trajectories (conflicting vehicles). The optimization task was considered a Nonlinear Constrained Programming (NCP) problem. To ensure an optimal solution is achieved, three analytical optimization approaches, including the Active Set Method (ASM), Interior Point Method (IPM), and Genetic Algorithm (GA), were deployed. The three algorithms were implemented in parallel, and the first acceptable solution was implemented. Acceleration, speed range, and safe headway were considered constraints in the optimization problem. A simulation testbed in VISSIM was developed to compare the proposed algorithm's efficiency with actuated traffic signals under different volume conditions. The simulation results revealed that the proposed algorithm improves stopped delay, travel time, and total throughput by 99%, 33%, and 8%. A 44% reduction in CO₂

emission and fuel consumption was also achieved. It was also revealed that the CVIC is more advantageous when the intersection is operating under oversaturated conditions.

Based on the assumption that vehicle trajectories can be defined as cubic interpolated splines, having the flexibility to reflect delays from the given signal timing, Gutesta et al. developed a trajectory-driven intersection control for the CAVs [24]. Vehicle trajectories were developed for several vehicles, passing multiple intersections. The optimization task's goal was to minimize the sum of all trajectory curves (which reflect control delays) conditioned to meeting safety constraints. Traffic constraints such as speed limits were reflected in the model by adjusting the trajectory slopes. A Genetic Algorithm (GA) was deployed to evaluate all possible combinations of optimized single-vehicle trajectories. In addition, an Artificial Neural Network (ANN), trained with available traffic stream factors, was deployed to achieve short-term prediction of vehicle delays to be integrated with the optimization model. The simulation results revealed that the proposed control algorithm improves traffic measures under stable and unstable traffic conditions, even with CAVs' low market penetration rate.

Krajewski et al. proposed a decoupled cooperative trajectory optimization logic to optimize and coordinate CAVs trajectories at signalized intersections [25]. The optimization goal was to minimize delay by coordinating conflicting movements such as straight going and left turns. The state-space of each vehicle had three dimensions: position, speed, and time. The optimization task was transformed into a graph (nodes representing the states and edges representing possible transitions between pairs of states) with a combined cost function of delay and comfort. The "decoupled" term refers to splitting the optimization problem into two stacked layers, 1) Trajectory Layer (TL) and 2) Negotiation

Layer (NL). The TL layer's task was to calculate trajectories for individual vehicles, and the NL layer's task was to coordinate all trajectories to prevent potential collisions. The latter task was achieved by setting constraints for each vehicle's TL algorithm, and the cost function was a weighted sum of the individual cost functions. The weight could also prioritize specific vehicle types. Simulation results revealed that the proposed algorithm reduces the cost value by 28% compared to the intelligent driver model replicating human-driven vehicles.

Hajbabaie et al. formulated the trajectory optimization problem into a Mixed Integer Linear Programming (MILP) model [26]. Several constraints such as speed and acceleration range and safe headway were introduced to the model. The objective function was to minimize the difference between each vehicle's current location and its final destination. A stochastic look-ahead technique based on the Monte Carlo Tree Search (MCTS) algorithm was developed to solve the MILP problem with several variables, including individual vehicles' location and speed. The simulation results for an isolated intersection under different volume conditions revealed that the proposed algorithm reduces travel time by 59.4% and 83.7% compared to actuated and fixed traffic signal controls. Additionally, decreased speed variance was achieved, and the number of stops was dropped to zero.

2.3 Machine Learning-Based Intersection Control Logics for CAVs

Recent access to abundant and cheap computation and storage resources has made ML approaches very popular in solving stochastic problems in different fields. Over the last few years, some researchers have deployed ML techniques to optimize single traffic signal

controllers and corridors or develop signal-free control logics for CAVs. Several ML-based interaction control systems related to this study are reviewed in this section.

Joo et al. proposed an isolated n-way intersection control logic based on Deep Q-learning (DQL) [27]. Possible movements at the intersection determined the current states. The related state was called if a vehicle was stopped and needed to pass the intersection in a specific direction. The actions were defined as possible signal phases, and taking specific action shifted the agent to a specific new state. The reward was set as a function of two variables: 1) the standard deviation of the queue lengths in different directions to ensure a balanced distribution of different signals' queue lengths, and 2) the throughput to maximize throughput. The proposed model was simulated in SUMO. The simulation results revealed that the proposed model has a better performance in reducing queue length and its standard deviation and waiting time compared to the two previously developed DQL-based intersection control systems.

Lamouik et al. developed a multiagent control system based on Deep Reinforcement Learning (DRF) to coordinate CAVs' at the intersection. Each vehicle transfers five features to the controller agent in this system, including position, speed, dimension, destination, and priority [28]. The controller agent has three possible responses for each vehicle, including acceleration, deceleration, and keep-same-speed. Reward value is a function of speed, priority, and collision. In the training process, the controller agent was expected to learn to avoid collision and prioritize vehicles with higher priority or higher speed. The simulation results revealed that the agent was trained since the reward value increased after several training epochs. However, the simulations were not run in a traffic network setting, and no traffic measures were assessed in this study.

Tong Wu et al. developed a multiagent deep reinforcement learning algorithm to optimize several traffic lights in a corridor [29]. In This approach, each traffic light was considered an agent with green light as actions for different phases. Indicating green for different phases was considered as the state for each traffic light. The reward value was set as a function of real-time delay for each vehicle, weighted by its priority. The innovation in this study was information exchange between controller agents so that each agent could estimate the policy of the other agents. According to other agents' estimated policy, each agent could adjust the local policy to achieve the globally optimal policy. Two networks with different numbers of intersections were developed in SUMO for assessment purposes. Simulation results revealed that the proposed model outperforms independent deep Q-learning (no exchange of information between agents) and deep deterministic policy gradient (single-agent controlling all intersections). It was also shown that the model outperforms self-organizing and fixed-time traffic lights.

Touhbi et al. developed a Reinforcement Learning (RF)-based adaptive traffic control system [30]. The main goal was to find the impact of using different reward functions, including queue length, cumulative delay, and throughput, on intersection performance. Unlike previous studies, which considered queue or delay as states, in this study, the state value was defined as a maximum residual queue (queue length divided by lane length) to reflect the traffic load on each phase. The learning process for a four-way single-lane intersection took 100 epochs of 1hr simulation runs, and the simulation results revealed that the proposed algorithm remarkably outperforms the pre-timed traffic signal controller. Furthermore, the analysis of different reward function deployment showed that each function's efficiency is highly dependent on the traffic volume.

Wu et al. developed a decentralized coordination algorithm for CAVs' management at the intersection. Each vehicle was considered an agent in this approach, and the related state to each agent included its lane, speed, and moving intention [31]. The intersection area was divided into $n \times n$ grid, and each vehicle was supposed to reserve its desired pixel ahead of time. Vehicles could enter either a coordinated or an independent state, based on having a reserved pixel in common or not. A Conventional Q-learning was deployed in this study, meaning that new states and their Q values are added to a look-up table as they came up, and the controller agent has to refer to the same cell in the table to make a decision. Finally, each agent's effort to maximize its Q value results in an optimal global reward. Simulation results revealed that the proposed model outperforms FCFS, fixed time traffic light, and Longest Queue First algorithms in delay reduction.

Liang et al. dynamically optimized traffic light duration based on real-time traffic information and a deep reinforcement learning approach [32]. Traffic data collected by roadway sensors, including vehicle's speed and location, were assumed as states in this method. Duration of traffic lights was considered actions, and the reward value was set as the cumulative waiting time difference between the cycles. Multiple reinforcement learning booster methods such as target network, double Q-learning network, dueling network, and prioritized experience replay, were deployed in this study. The simulation results revealed that the proposed model could learn a good policy, either in rush hour or in normal traffic flow, by reducing the average delay time by over 20% in the training course. The model also outperformed other DRL-based models in learning speed and other measures.

Gong et al. developed a multi-objective reinforcement learning approach to improve traffic safety at the intersections controlled by adaptive signal controls [33]. Their

algorithm input was high-resolution real-time traffic data, and the output was an optimized traffic signal to reduce delay and crash risk at the intersection. The model was trained based on real-time data from an intersection to a simulation testbed. The proposed control system was compared with the real-world signal timing for the intersection provided by local jurisdictions. The simulation results revealed that the proposed model significantly improves traffic safety. Meanwhile, traffic efficiency slightly deteriorated.

2.4 Vehicles' Longitudinal Control Systems Background

The earliest vehicle longitudinal control systems, Cruise Control (CC), were developed in the late 1950s and 1960s. In the initial designs, the driver had to set the desired cruising speed by a dash-mounted dial, and the CC's role was to merely provide a full-throttle when the vehicle speed drops 6-10 mph below the desired speed. Since the 1980s, with the appearance of microprocessor technology, electronic-based controllers and interfaces were added to the CC system, improving their ability to provide driving comfort to some degree [34]. Still, no advantage in safety or capacity improvement was achieved.

Adaptive Cruise Control (ACC) can be considered an extension to CC systems. In an ACC system, vehicle velocity is adjusted automatically to provide a specified distance to the preceding vehicle by controlling the throttle or the brake. ACC systems development in the research world dates to the 1960s, even before the term ACC was adopted. However, the earliest efforts failed because computers, communication, and sensor technologies were not catching up with advances in the research area. By the 1980s, the transportation industry infrastructures were not meeting the rapid increment in the number of vehicles worldwide, causing more congestion, accidents, and air pollution. This issue forced

governmental, industrial, and research institutions to take severe measures to implement the ACC system. Coincided with technological advances, the implementation phase began almost simultaneously in Europe, the US, and Japan in 1986. Currently, ACC is no longer a technical term but a prevalent marketing term and most automakers have equipped their products with ACC systems [35]. Almost 20 years of research on ACC systems have revealed that ACC impact on traffic flow and operation is highly variable and strongly dependent on the vehicles' gap acceptance setting [36].

With the emergence of CAVs and taking advantage of communication capabilities within the last decade, conventional ACC systems have evolved to CACC systems [37]. CACC-equipped vehicles can share information with other vehicles and the controller unit; this enables the controller to provide safer, smoother, and more natural responses [38]. Despite being designed to give the driver more comfort and convenience, CACC can increase traffic throughput and safety by allowing a shorter headway between vehicles and platoon formation [39]. Since connectivity between vehicles will be mandatory for the new cars in the USA shortly, CACC systems are attracting noticeable attention from academia and industry [40]. A noticeable number of transportation engineering studies are involved with developing CACC longitudinal control system and assessing its impacts on the transportation system. A literature review of the existing CACC models and their impact on traffic measures is presented in the following sections.

2.4.1 Development and Assessment of CACC Models

One of the first studies assessing the impact of CACC on traffic flow was conducted by Shaldiver et al. [41]. In this study, the distribution of time gap between vehicles was collected from human-driven CACC and ACC-equipped vehicles in a field test. According

to the test results, in ACC mode, drivers were unlikely to adopt a gap smaller than what they chose in manual driving mode. A simulation platform was built in AIMSUN to compare ACC and CACC performance based on the car-following time gap derived from the field test. A simplified version of ACC and CACC car following models were deployed to reduce computational efforts. ACC car-following rules were complying with Nissan cars, and the CACC model was derived from [42], which has two main components: 1) speed control mode, which kept the vehicle's speed close to the speed limit, and 2) the gap control mode, which maintained the desired gap between pair of cars. Simulation results on a single-lane straight freeway under different Market Penetration Rates (MPR) of CACC and ACC revealed that ACC is unlikely to improve highway capacity significantly. However, CACC can noticeably increase highway capacity in moderate to high MPRs, as the higher dynamic response gives the driver confidence to adopt a shorter gap setting.

Van Arem et al. studied the impact of CACC on a four-lane highway merging section in the MIXIC microscopic traffic simulation model, which simulates traffic on link-level in a network [43]. The CACC acceleration and distance control functions were derived from [44]. The acceleration was calculated based on distance and speed difference with the preceding vehicle. The clearance length was a function of the speed and deceleration capability of the preceding vehicle and the deceleration capability of the target vehicle. This study revealed that a high MPR of CACC enhances highway capacity after a lane drop section with a relatively high volume. However, since the communication was limited to longitudinal control, with no control on the length and compactness of CACC platoons, CACC vehicles prevented other vehicles from cutting in.

Zonuzy et al. designed a CACC control logic with space-based model parameters to improve CACC platoons merging maneuver [45]. The idea was to assign each vehicle on the sub-stream to a pair of vehicles on the mainstream, then adjust the vehicle's time gap and velocity to accommodate the merging vehicle. The vehicles' longitudinal control logic was based on the nonlinear systems dynamic theory, which guarantees string stability and single-vehicle stability [46]. Time gap profiles were provided to ensure vehicles maintain a safe time gap and a minimum deceleration distance. The control model inputs were pre-designed time gap versus location and velocity versus location profiles. The numerical simulation results revealed that the proposed logic ensures string stability for platooning CACCs. However, vehicles from the sub-stream needed to be positioned at the correct location between the mainstream sub-platoons to ensure minor errors would not appear at the merging point.

Lee et al. developed a CACC control logic based on a Multi-Objective Optimization Problem (MOOP) approach [47]. Four objective functions were selected to be optimized, 1) target time headway deviation, to control platoon evolution time and make sure the platoon is stabilized under traffic disturbances, 2) unsafe condition, to make each pair of cars attain the minimum required headway and secure safety, 3) vehicular Jitter, to minimize switch between acceleration and deceleration and avoid a drastic change in any of them and improve comfort, 4) fuel consumption, to reduce fuel consumption and environmental pollution. A genetic algorithm was deployed to optimize four main objective functions and additional constraints. The optimization output was used as control logic for platooning CACC vehicles by controlling their acceleration. Simulation Results of a single lane 14.5 km freeway segment in VISSIM software with three different target

time headway values, revealed that MOOP CACCs with a longer target time headway has better performance in time headway deviations. However, a shorter time headway provided more throughput. It was also revealed that MOOP keeps a good balance between all objective functions compared to the Single Objective Optimization Problem (SOOP) since the latter was limited to specific search space for optimization, resulting in biased results.

2.4.2 SMD-based Longitudinal Vehicle's Control Systems

Developing car-following models based on Spring-Mass-Damper (SMD) system dates back to the late 1990s, and few researchers have deployed this model for the same purpose since then. In 1998 Eyre et al. used an SMD system with linear characteristics to evaluate platooning AVs' longitudinal string stability properties [48]. They considered two types of interactions: 1) unidirectional, each vehicle being connected only to its predecessor without being affected by its follower, and 2) bidirectional, each vehicle being coupled with both proceeding and following vehicles. Two different spacing policies were examined: 1) constant space policy and 2) speed-dependent spacing policy. Simulation results of autonomous commercial trucks revealed that: 1) the unidirectional controller only achieves stability if it is used with the speed-dependent policy, 2) bidirectional controller with the constant spacing policy achieves stability only for a specific range of SMD model hyperparameters, 3) in case of speed-dependent spacing policy, the bidirectional controller showed the most efficient performance if the space is only adjusted with predecessor vehicle.

Contet et al. studied a single platoon of three vehicles controlled by SMD logic, longitudinally and laterally [49]. Two simulation scenarios were developed to analyze trajectory spacing error: 1) while maneuvering to avoid an obstacle and 2) after

accommodating a merging vehicle. Simulation results revealed that the platooning vehicles could successfully avoid the obstacle, and no unsafe spacing error is produced in any scenario. A physical experiment was performed by running soccer robots in a prototype playground, and spacing errors were collected while facing speed constraints or moving in curved paths. The simulation results confirmed the flexibility and adaptability of the model. Real-world roadway circumstances were not considered in this study, either in the numerical simulation or the physical experiment.

Munigety et al. performed a sensitivity analysis on the SMD model to its three primary hyperparameters, including mass, spring constant, and damper coefficient. They proved that the spring constant and damper coefficient could represent driver aggressiveness and vehicle stability accordingly [50]. It was also demonstrated that speeding capability reduces as the vehicle's mass increases. By simulating a single lane 300 meters roadway in MATLAB, speed-flow diagrams for four different vehicle types (motorbike, auto-rickshaw, car, truck) were derived. It was revealed that as the vehicle size increases, the roadway capacity decreases.

Bang et al. developed a strategy for CAV platoon evolution by reflecting swarm intelligence theory to the SMD model [51]. The swarm intelligence theory describes animals' clustering behavior, such as bird flocking and fish schooling. It also describes the molecular behavior of materials in different phases, including gas, liquid, and solid [52]. In this study, the spring constant was defined as functions of traffic flow, and the damper coefficient was defined as a function of spring constant, mass, and vehicle response time functions. Simulation results revealed that the spring constant represents the tendency for platoon formation or clustering time, and the damping coefficient represents the stability

or oscillation of vehicles. Efficient spring constant and damping coefficient functions for minimizing clustering time in high flow circumstances or providing more freedom for lane changing in low volume circumstances were introduced.

Bang et al. also studied platoon stability at merging sections [53]. The idea was to adjust the spring constant and damping coefficient for the cut-in vehicle and its follower to minimize the cut-in movement's impact. The spring constant and damping coefficient for the cut-in vehicle and its immediate follower were defined as functions of spacing and speed difference between the two vehicles. Consequently, the cut-in vehicle and its follower would maintain a lower spring constant, allowing them to adopt a temporary shorter spacing and higher damper coefficient, which reduces speed disturbance since the vehicle matches the leader's speed more quickly. The simulation results for merging sections showed that the speed and spacing variation and recovery time diminishes significantly with the proposed control.

2.5 Chapter Summary

The literature reviewed in this section emphasizes the tremendous impact of CAV's communication capability in intersections control and CACC form. In this chapter, CAV-based intersection control systems were clustered into three categories, including ML-based approaches, and several studies in each cluster were reviewed. A systematic literature review on CAV-based intersection control systems proposed between 2008 and 2019 reveals that only 4% of the proposed models are ML-based; this study also finds ML-based approaches more promising than other methodologies [2]. Considering the RL as the third paradigm of ML and outstanding records of DRL in solving stochastic decision-

making problems, several RL-based intersection control systems proposed for CAVs were reviewed in this section. Some studies have deployed DRL to optimize traffic light phasing and timing or processing aerial images to optimize traffic flow at the intersections. CAVs are expected to act as individual robots at an ultimate automation level, we could not find any literature on controlling individual CAVs based on DRL in traffic roadway circumstances.

As the second section of the literature review, several developed CACC systems were reviewed, all approving the significant impact of CACC on traffic and environmental measures. Among existing CACC models, the SMD model is this study's point of interest. The SMD model is based on classical physics and reflects the most critical dynamic aspect of vehicles, the mass; it also covers the platoons evolution process. There have been few SMD-based CACC control systems proposed up to this date. However, the SMD model has never been coded into commercial traffic simulation software for a more realistic and comprehensive traffic-oriented assessment. Moreover, we believe limitations such as maximum communication range and maximum platoon length should be set to the SMD model to make it more compatible with real-world circumstances.

CACCs and CAV-based Intersection control systems play a pivotal role in CAVs' efficiency, and both are calling lots of attention in the CAV research area. And few CAV-based intersection control developers have considered controlling a group of vehicles instead single. None of them have deployed a firm platooning logic and an automated intersection control system simultaneously.

CHAPTER 3

DEVELOPMENT OF A CAV-BASED INTERSECTION CONTROL LOGIC AND PROOF OF CONCEPT TEST

In this chapter, a DSCLS is developed to manage CAVs at the intersections. Followed by an introduction to RL, the proposed model development is elaborated. A proof-of-concept test is performed by application of the model to a single lane intersection and comparing the proposed model performance with other intersection control systems.

3.1 An Introduction to Reinforcement Learning

RL is based on the idea of learning by interactions with the environment. For example, when humans learn how to talk, walk, or handle a conversation, no explicit teacher instructs them. However, the feedback from the environment provides a wealth of information about causes and effects, consequences of actions, and what to do to achieve goals. RL is learning how to map situations to actions to maximize the cumulative reward. The learner is not told which action to take; instead, it must discover which actions yield the most reward by trying them. An action may affect the immediate reward, the future states, and subsequent rewards. Considering supervised and unsupervised learning as two paradigms of ML, the RL is considered the third paradigm, the closest to humans and other animals' learning. RL algorithms were initially inspired by biological learning systems. Exploration (trial-and-error search) and delayed reward are RL's two most prominent features.

One of the challenges that arise in RL, and not the other two ML approaches, is the trade-off between exploration and exploitation. To maximize the reward, an agent or decision-maker has to take the actions that have been tried before and is found to be

effective in producing rewards (exploitation). However, the agent has to try new actions to discover such actions (exploration). The challenge is that neither exploration nor exploitation can be followed exclusively without failing the task. There is noticeable literature in mathematics in solving the exploration-exploitation dilemma, and the problem has remained unsolved yet. Another critical feature of RL is that it considers the ultimate goal of an agent in contrast to most other approaches that consider sub-problems without addressing how the solution might fit in a larger picture and be finally helpful [54].

Before offering any solution, an RL problem needs to be classically formulated in a precise theoretical form. Markov Decision Processes (MDPs) are a classical formalization of sequential decision making, capable of reflecting all RL characteristics, including delayed reward and the trade-off between immediate and delayed reward [8].

3.1.1 Finite Markov Decision Process

In the MDP, the agent takes actions, the environment responds to these actions, and presents new situations and rewards to the agent. The agent's goal is formalized in terms of unique signals called rewards. The agent seeks to maximize the reward over a series of interactions with the environment in discrete time steps (t). Interactions between the agent and environment in the MDP are shown in Figure 3.1.

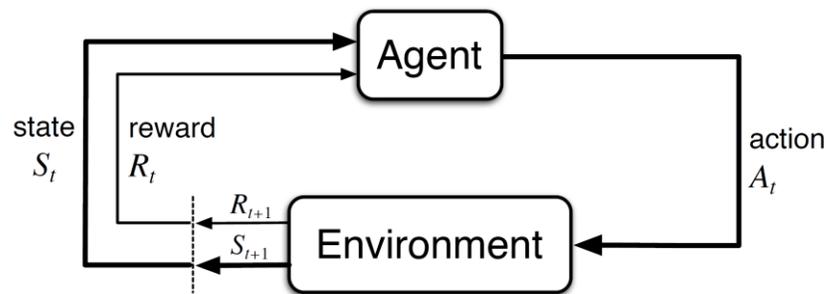


Figure 3.1 The agent-environment interactions MDP.

Source: [54].

In a finite MDP, the sets of States, Actions, and Rewards (S, A, R) have finite elements. So, as shown in Equation (3.1), the random variables R_t and S_t follow a discrete probability distribution, depending only on the preceding state and action.

$$p(s', r' | s, a) \doteq \Pr \{S_t = s', R_t = r', | S_{t-1} = s, A_{t-1} = a\} \quad (3.1)$$

3.1.2 Returns, Episodes, and Learning Process

Assuming the sequence of rewards received by an agent after several time steps (t) is denoted as $R_{t+1}, R_{t+2}, R_{t+3}, \dots, R_T$, the goal is to maximize the “expected return” where the return can be expressed as a specific function of rewards sequence and the simplest case, the sum of rewards, as shown in Equation (3.2).

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T \quad (3.2)$$

Some agent-environment interactions naturally break into episodes (episodic tasks), and T is the final time step of an episode which is called a “terminal state”. For example, a terminal state could be winning or losing a game, and the next episode starts independently of how the previous one ended. However, in some cases, the agent-environment interaction does not naturally break into episodes, such as an ongoing CAV control system, which is a “continuing task”. The return for a continuing task with $T = \infty$, is expressed in Equation (3.3).

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (3.3)$$

γ is a parameter called “discount rate” ranging from zero to one; γ defines the present value of future rewards. If $\gamma = 0$, the agent is myopic, only concerned with maximizing the immediate reward. As γ approaches one, the agent cares more about future rewards and becomes more farsighted. Return at each time step can be calculated as appears in Equation (3.4). According to this equation, the instant reward and the next step’s return value are required to calculate the return value at each step.

$$\begin{aligned}
 G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
 &\doteq R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots) \\
 &\doteq R_{t+1} + \gamma G_{t+1}
 \end{aligned} \tag{3.4}$$

The main idea behind RL is to run an experiment several times (episodes) to learn the return value for each action in a specific state. The requirement to achieve this goal is to balance between exploration and exploitation, which is applied by a parameter called epsilon (ϵ). Thus, the probability of taking a random action (exploration) at each step equals ϵ , and the probability of taking actions based on the learned return values of different actions equals $(1 - \epsilon)$. Therefore, the ϵ value needs to be decayed over episodes to make the agent gradually shift from exploration to exploitation.

3.2 Reinforcement Learning and Neural Networks

In conventional RL, the return values for possible actions in each state are saved in multi-dimension matrices. Each matrix cell is related to a specific state and a unique action taken at that state. Therefore, the matrix expands as the agent explores more in the environment.

However, this approach is not practical in complicated environments since lots of space is required to save all state-action combinations. Moreover, the agent will take a completely random action in case of facing an unprecedented state. In DRL, the look-up tables are substituted with neural networks. The neural network inputs are the state, and the output is an estimation of each possible action's value. The neural network weights are updated as the agent keeps exploring, and that's how the learning task is accomplished. Compared to RL, the DRL can deal with more complex environments since it will face unprecedented states by estimating the return value. Several methods, including Reinforce, SARSA, and DQN, are proposed to perform the DRL process. In this study, DQN is used.

3.2.1 Deep Q-Networks

The DQN is an algorithm proposed by Minh et al. [55] that estimates the expected return using Temporal Difference (TD) learning. DQN formulation for estimating return values, called Q-function, is presented in Equation (3.5).

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \quad (3.5)$$

Where:

$Q(s, a)$, is called Q-value and defines how valuable a state-action pair is.

In the DQN approach, states are forwarded to a neural network, and the neural network's output is an estimation of Q-values for all possible actions in the current state. The training occurs by updating the neural network weights based on having a batch of recent historical data (states and actions) as features, and target Q-values for each of them,

calculated based on Equation (3.5). A high-level training process of a DQN agent is shown in Figure 3.2. When the agent is trained, the leftmost loop in Figure 3.2 would be eliminated.

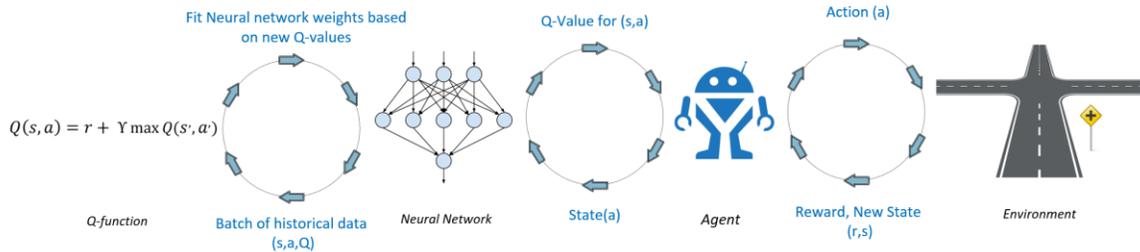


Figure 3.2 High-level training process of a DQN agent.

The training minimizes the difference between the last estimated Q-values and the target Q value. However, performing this task is difficult as the target Q-value constantly changes during the training course. To solve this issue, a lagged copy of the Q-network called the target network calculates the target Q-values to solve this issue and make the neural network's training process more stable. The target neural network parameters will be updated with a specific frequency [8].

The Pseudocode presented below helps to understand the general process of training an agent in the DQN approach. Several hyperparameters need to be adjusted in the DQN training process, and finding a good combination of all parameters requires several training trials.

```

Initialize the number of epochs (EP)
Initialize the number of steps per epoch (STP)
Initialize discount ( $\gamma$ )
Initialize ( $\epsilon$ )
Initialize  $\epsilon$ -decay Factor (DF)
Initialize replay memory size (RMS)
Initialize minimum replay memory size (MRMS)
Initialize mini-batch size (MBS)
Initialize neural networks
Initialize target neural networks
Initialize target neural network update frequency (F)
for i = 1 ... EP:
    Gather and store RPL experiences
    for j = 1 ... STP:
        if (0 < random value < 1) >  $\epsilon$ :
            take an action based on trained network
(exploitation)
        else:
            take a random action (exploration)
            if the number of stored experiences > MRMS:
                Sample a random MBS size of experiences from RMS
                for k = 1 ... MBS:
                    Calculate target Q-values for each example
                    (Equation (3.5))
                    calculate the loss (Mean Absolute Error)
            if a terminal state (Each epoch's last state):
                Update the neural networks' weights
 $\epsilon = \epsilon * DF$ 
    if EP % F == 0:
        Update the target neural networks' weight

```

3.2.2 Multi-agent DQN

Most real-world RL applications such as autonomous driving involve more than a single agent, which falls into the realm of Multi-Agent Reinforcement Learning (MARL). The MARL addresses the sequential decision-making problem of multiple agents operating in a common environment, optimizing the long-term return by interacting with other agents and the environment. MARL algorithms could be either fully cooperative, fully competitive, or mixed. Each agent tries to optimize the common long-term return in a cooperative setting. However, in a competitive setting, each agent tries to maximize its

own return, and the summation of all agents' returns would be zero [56]. Considering an intersection as an environment and vehicles approaching it as agents leads us to a cooperative setting to optimize the overall return at the intersection.

This study deploys DQN as the optimization tool. In Multi-Agent Deep Queue Learning (MADQL) approach, an agent can be either in an independent state (which does not include any interaction with other agents) or a coordinated state (which requires interactions with other agents). In the case of being in an independent state, the agent receives an individual reward (Equation (3.6)). In a coordinated state, the agent would receive a joint reward (Equation (3.7)), and the total distributed reward at each step is called global reward (Equation (3.8)).

$$r(s_i, a_i) = r_i \quad (3.6)$$

$$r(s_j, a_j) = \sum_{b=1}^h r_{jb} \quad (3.7)$$

$$R(S, A) = \sum_{b=1}^N r_{(i \text{ or } j) b} \quad (3.8)$$

Where:

Subscripts i refer to an agent being in an independent state,

Subscripts j refer to an agent being in a joint state,

h is the number of agents involved in a joint state,

N is the total number of agents in an environment,

R is the global reward.

Referring to the nature of RL, which considers future states and rewards, while updating the current state's Q-value, a transition from an independent state to a coordinated state or vice versa requires specific settings for updating Q-value in MADQL. The transition could also be from a joint state to another joint state, but with different agents involved, as shown in Figure 3.3.

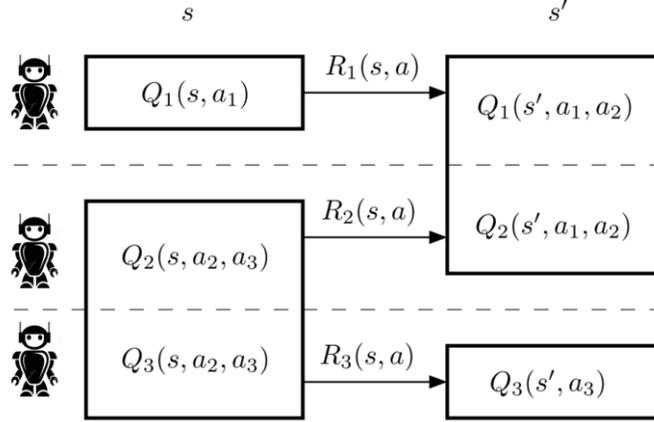


Figure 3.3 Potential transition types in MADQL.
Source: [57].

Kok et al. proposed formulations for updating Q-values for possible transition types in a conventional Q-learning approach [57]. However, the formulas need to be adjusted to be compatible with deep Q-learning. The adjustment includes removing the learning rate factor from the formulas since the learning rate is already included in neural network settings. In the following, the adjusted updating Q-values formulas for three possible types of transitions are presented.

- Type 1: when an agent moves from a coordinated state to another coordinated state or from an independent state to another independent state, the coordinated or independent Q-values are updated by Equation (3.9).

$$Q(s, a)_{i \text{ or } j} = r(s, a)_{i \text{ or } j} + \gamma \max Q(s', a')_{i \text{ or } j} \quad (3.9)$$

- Type 2: when an agent moves from a coordinated state to an independent state, the joint Q-values are updated by Equation (3.10).

$$Q(s, a)_j = \sum_j^h [r(s, a)_j + \gamma \max_{a'} Q(s', a')_i] \quad (3.10)$$

- Type 3: when an agent moves from an independent state to a coordinated state, the independent Q-values are updated by Equation (3.11).

$$Q(s, a)_i = [r(s, a)_i + \gamma \frac{1}{h} \max_{a'} Q(s', a')_i] \quad (3.11)$$

3.3 Decentralized Sparse Coordination Learning System for CAVs' Management at the Intersections

The testbed for model development is a single-lane intersection with no turning movements. Traffic volume production and roadway configurations are set up in the VISSIM software. The DSCLS application is developed in Python, and communication between the Python application and the simulation software is established through VISSIM Component Object Model (COM) Interface. The developed control system is called DSCLS since each vehicle has its own control logic (decentralized), and coordination with other agents is not required in all steps (sparse coordination).

3.3.1 Testbed Settings and Reward Function Development

The intersection and approaching sections are divided into 2.5×2.5 m grid areas, as shown in Figure 3.4. At the beginning stages of the training process, agents' exploration-oriented actions would cause very long queues. Therefore, each intersection leg is extended 2 km upstream, so all demanding vehicles would have a chance to enter the network and form a queue. The DSCLS controls vehicles' manoeuvres from 50 meters upstream of the

intersection. Assuming a vehicle length of 4.90 meters (Ford Fusion 2019), each vehicle would occupy either two or three cells at each time step. For instance, vehicles 1 and 2 in Figure 3.4 occupy 2 and 3 cells accordingly.

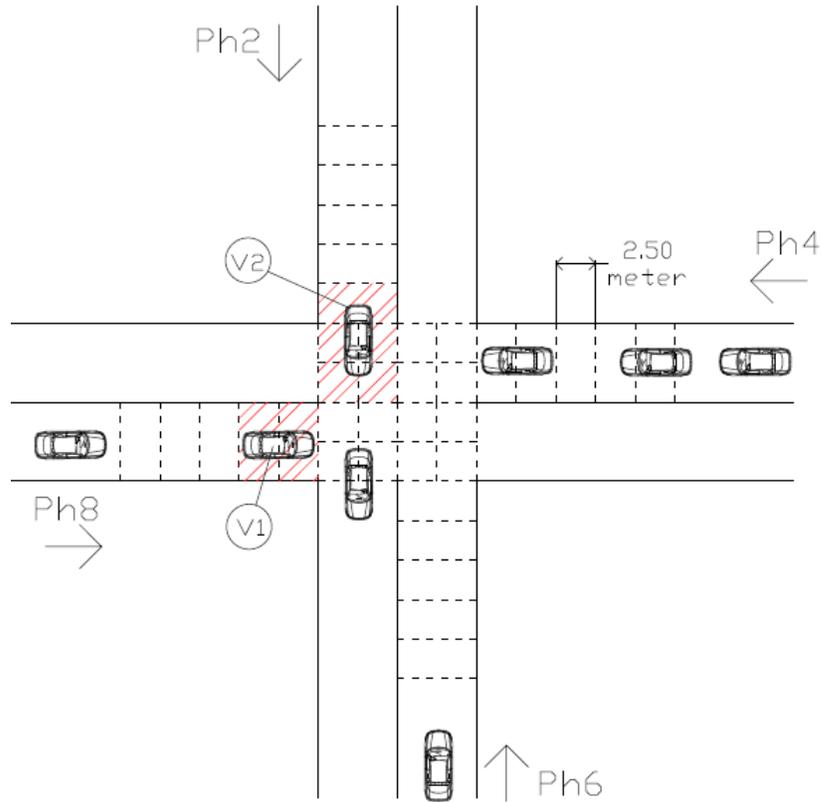


Figure 3.4 The intersection environment.

The first vehicle in each approaching direction tries to reserve its desired cell for the next step. The number of desired cells describes the moving intention of each vehicle at each time step and is calculated based on the current speed and stopping distance of vehicles. Based on classical physics formulas, as shown in Equation (3.12), the survey distance equals $(V\Delta t + \frac{1}{2}a\Delta t^2)$ if the vehicle is moving slower than the maximum speed and can still accelerate, or it equals $(V\Delta t)$ if the vehicle is already moving with the

maximum speed, and the stop distance equals $\frac{V^2}{2a}$. The acceleration and deceleration rates are set to 3.5 and 7 m/sec^2 .

$$C_s = \begin{cases} \max\left(\left[\frac{V^2}{2a}\right], \left[V\Delta t + \frac{1}{2}a\Delta t^2\right]\right) & V < V_m \\ \max\left(\left[\frac{V^2}{2a}\right], [V\Delta t]\right) & V = V_m \end{cases} \quad (3.12)$$

Where:

Δt : each step's time length

V : current speed

a : acceleration or deceleration capability

If a common desired cell is detected between vehicles, they will enter a coordinated state; otherwise, they would be in an independent state. For example, vehicles 1 and 2 in Figure 3.4 are in a coordinated state since they try to reserve common cells ahead of them. The state of each leading vehicle includes its current speed, current cell, desired cells, and queue length behind it. Then, the state of each vehicle is forwarded to the Q-network. The output Q-network is either acceleration or deceleration for a leading vehicle to eliminate any collision and optimize the global reward. Equations (3.13) and (3.14) present the state and possible actions.

$$State_i = (Speed_i, Current\ Cells_i, Desired\ Cells_i, Queue_i) \quad (3.13)$$

$$Action_i = (-acc, +acc, maintain\ curent\ speed) \quad (3.14)$$

If an idle vehicle receives a declaration order or a vehicle moving with maximum speed receives an acceleration demand, it just keeps idling or moving with the maximum speed; the Q-network is expected to learn this behaviour. The maximum speed is 40.2 km/hr, and the simulation resolution is set to 1sec/step, meaning that each vehicle would reserve a maximum of 6 pixels for the next step. However, the communication range is set to 50-meter upstream in each direction, so the vehicles would take advantage of being in independent states at the upstream. As mentioned earlier in this study, the RL optimization logic is not based on the instant reward but the total return. Therefore, a vehicle in an independent state could receive a deceleration order to optimize future sequences and maximize the overall return, and being in an independent state in this environment does not necessarily result in maintaining the maximum speed

The return for each leading vehicle at each time step is the summation of the delay of the leading vehicle and its followers. The delay value for each approach is calculated based on Equation (3.15).

$$r(step, approach) = - \sum_{i=1}^n (\Delta t - \frac{L_i}{V_m}) \quad (3.15)$$

Where:

Δt : Step size (time passed at each step)

L_i : Surveyed distance

V_m : Maximum allowable speed

3.3.2 Training Settings and Learning Performance

Four neural networks with different layer structures are developed to take actions in all possible states, including 1) coordinated state of 4 vehicles, 2) coordinated state of 3 vehicles, 3) coordinated vehicle of 2 vehicles, and 4) independent state. In addition, a lagged copy of each neural network is also developed to estimate target Q-values (as explained in Section 3-3-1. Since the multi-agent Q-network needs information not only about current states but also the future states, the replay memory is updated with more detailed data about each leading vehicle at each time step, as shown below:

(Current state, Current action, Reward, Future joint or independent state)

Several combinations of hyperparameters and neural network structures are examined by running numerous training trials. The optimal values of hyperparameter settings are presented in the following. Traffic volume is set to 1,800 and 1,200 Veh/hr on major and minor streets. Vehicles are stochastically generated with a constant random seed by VISSIM software for 900 seconds. Initial Epsilon (ϵ) value is set to 1, meaning that the agents are in a full exploration mode by taking random actions, which may cause long queues. Therefore, intersection legs are extended 2 km in each direction to reflect the potential queue length. The total epoch length is set to 1500 seconds to ensure that all generated vehicles have evacuated the intersection at the end of each epoch. The ϵ decays at the end of each epoch by being multiplied by the ϵ – decay factor, which gradually reduces the chance of taking random actions (transition from exploration to exploitation).

The simulation resolution is set to 1 step/sec, and the random seed is kept constant during all training iterations. As mentioned in the methodology section, to improve the DQN performance, a lagged copy of the neural network estimates the target Q-values. The

target neural network parameters are updated every five epochs (target network update frequency). The replay memory size, which defines how many of the latest experiences will be stored in the memory to be used for training, is set to 10,000. The minibatch size, which defines the size of random data set from replay memory to be forwarded to the neural network, is 32. The training process hyperparameter settings appear in Table 3.1.

Table 3.1 Training Process Hyperparameters Setting

Hyperparameter	Value
Number of steps per epoch	1,500
Step size (Δt), equal to simulation resolution	1 sec
Replay Memory Size (RMS)	10,000
Mini-Batch Size	32
Target network update frequency (F)	5 epochs
Initial ϵ	1
ϵ – Decay Factor (DF)	0.999
Discount (γ)	0.9

The ϵ , Mean Absolute Error (MAE), and global reward changes in the training process are shown in Figure 3.5 to Figure 3.7, all three graphs' Values are smoothed by 0.98 to remove noises and help readers understand the general training process.

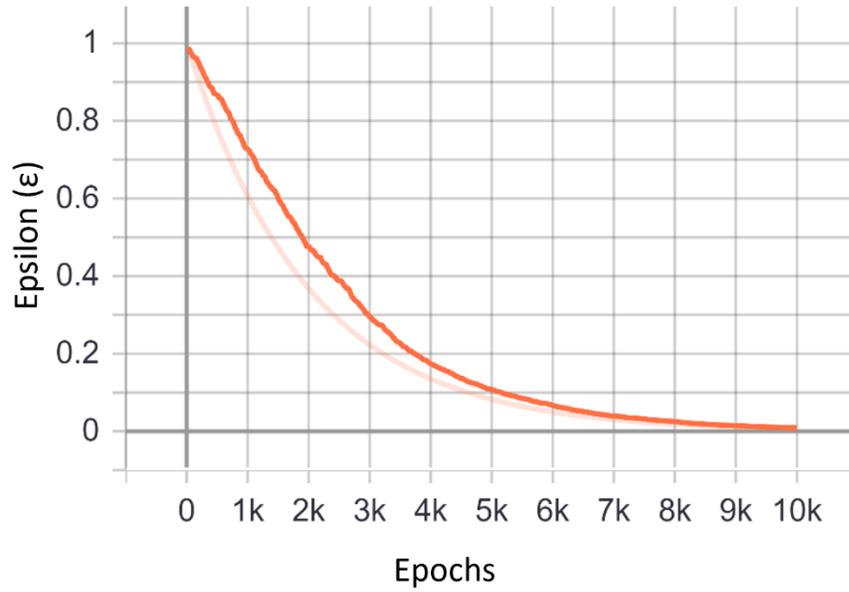


Figure 3.5 Transition from exploration to exploitation (decaying ϵ) in the training process.

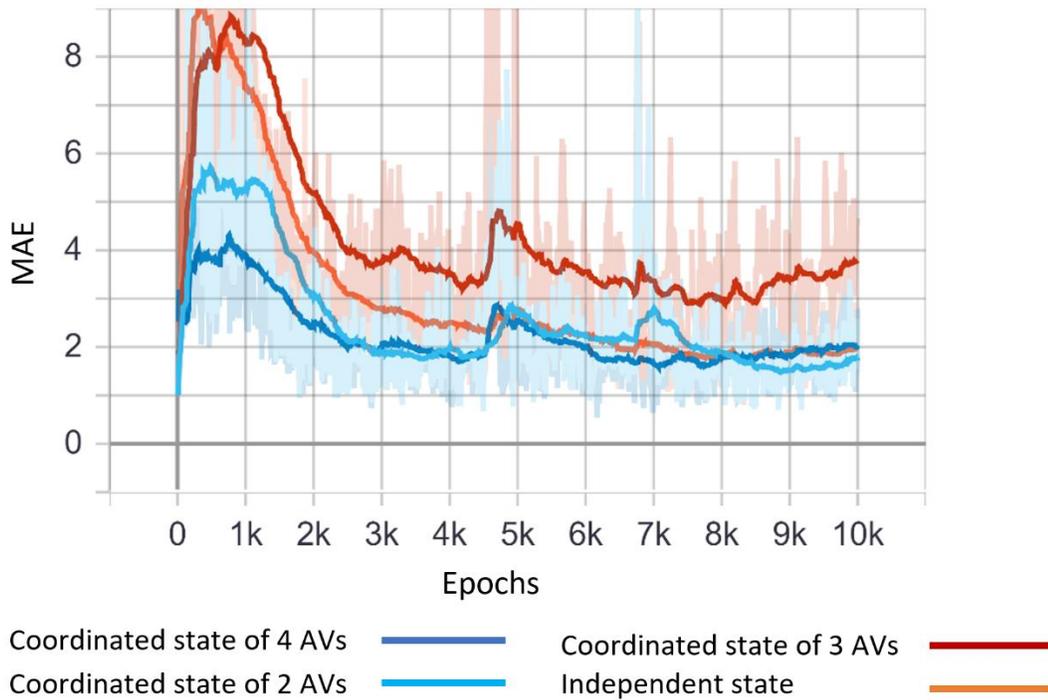


Figure 3.6 MAE change for four neural networks over learning progress.

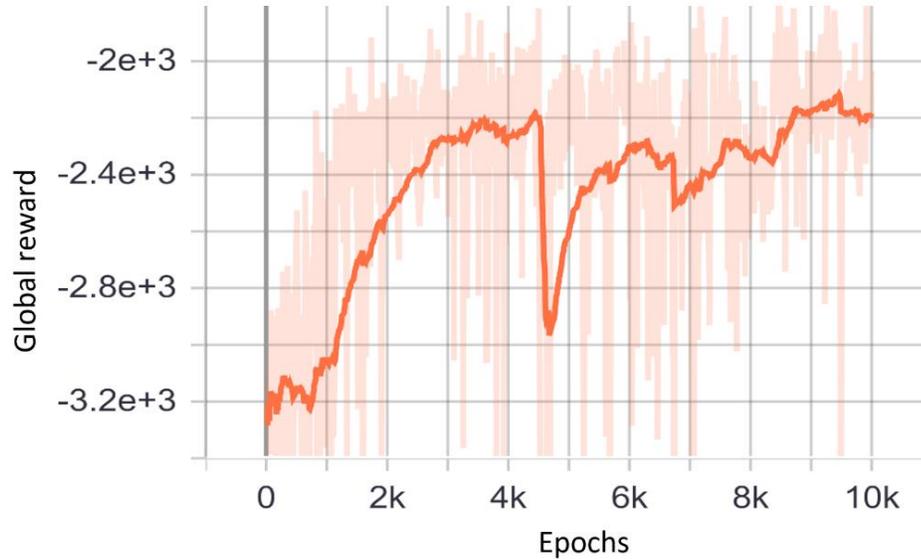


Figure 3.7 Global reward changes over learning progress.

According to Figure 3.6 and Figure 3.7, the MAE for all four neural networks is reduced over the training process. Furthermore, the global reward is decreased in the same way, meaning that the agent is being trained.

3.4 Proof of Concept Test

The trained model is compared with three other control systems, including:

1) Fixed traffic signal, 2) actuated traffic signal, and 3) Longest Queue First (LQF) control logic. The LQF control algorithm is developed by Wunderlich et al. [58]. In this algorithm, the phases have no particular order and are triggered only based on queue length. The simulation network is a 4-way intersection, having a single lane at each approach, as shown in Figure 3.4. The speed limit is set to 40.2 km/hr, and the intersection legs are extended 2km upstream at each direction so that the DSCLS can monitor potential long queues at the upstream. Three different volume regimes mentioned below are considered for evaluation purposes:

1. Moderate volume: consists of 600 and 400 Veh/hr on the major and minor streets accordingly. This volume combination leads to Level Of Service (LOS) B for the major street and LOS C for the minor street at the testbed intersection.
2. High volume: consists of 800 and 600 Veh/hr on the major and minor streets accordingly. This volume combination leads to LOS D for both major and minor streets.
3. Extreme volume: consists of 1000 and 700 Veh/hr on the major and minor streets accordingly. This volume combination leads to LOS F or congestion for both major and minor streets.

The LOS is calculated based on HCM 6th edition in VISTRO software, assuming the intersection operates under fully actuated traffic signals. 20 simulation runs with different random seeds, and a length of 20 minutes is run for each scenario. The simulation results are presented in the following sections.

3.4.1 Target Traffic Measures Comparison

The reward value in DSCLS in this study is expected to reflect the delay value (Equation (3.15)). Therefore, the delay is expected to be improved compared to the other control systems. All traffic measures' data collection section is extended from the intersection area to 200 meters upstream for each approach. According to Figure 3.8, the DSCLS is noticeably effective in delay reduction in all volume regimes compared to all other control systems, specifically in lower volume regimes. Delay reductions of 79%, 32%, and 19% are gained compared to the second-best control system in moderate, high, and extreme volume regimes accordingly.

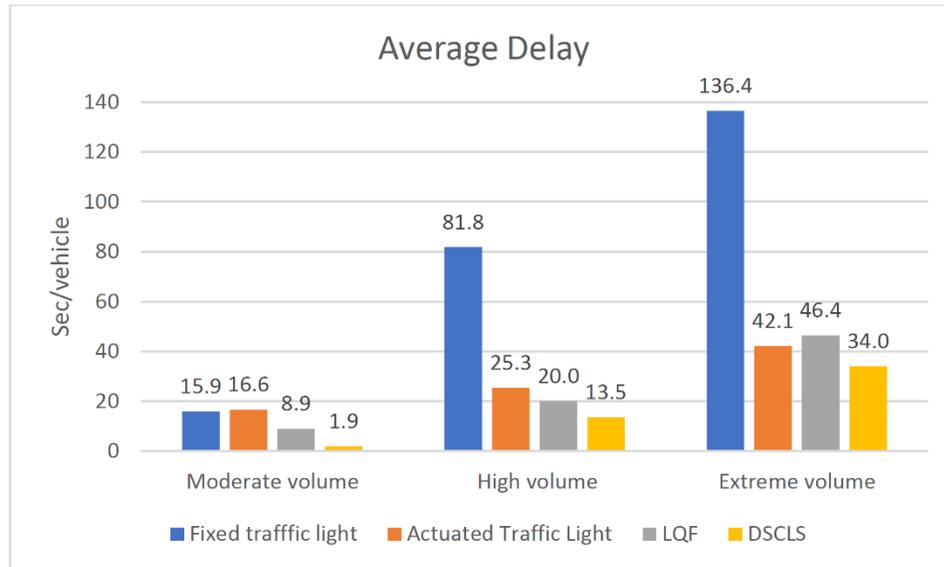


Figure 3.8 Average delay comparison for proof-of-concept test.

Followed by delay reduction, the travel time is expected to be improved. According to the average travel time values shown in Figure 3.9, 35% and 17% travel time reductions are gained compared to the LQF control system in moderate and high-volume circumstances. The proposed control system outperforms the actuated control system with a 13% travel time reduction in the extreme volume regime.

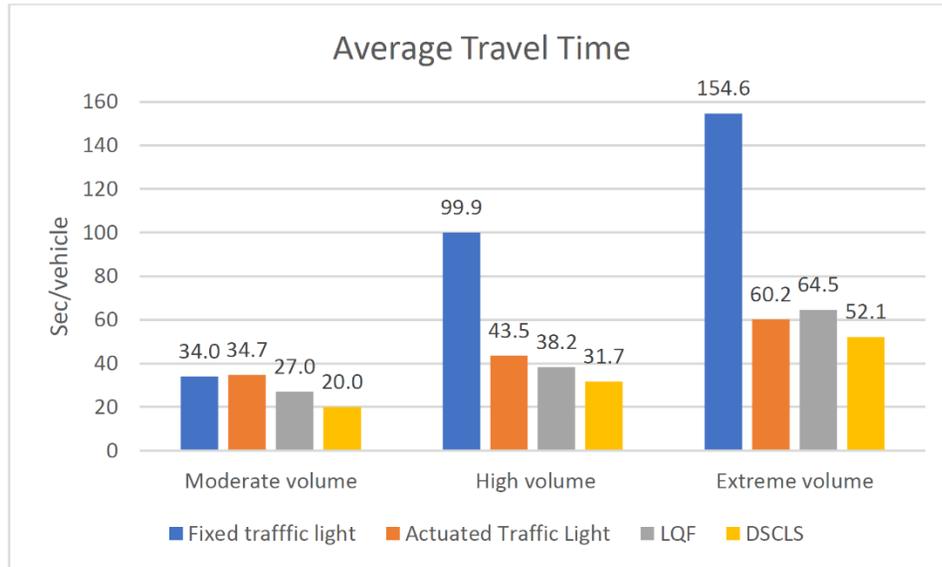


Figure 3.9 Average travel time comparison for proof-of-concept test.

The impact of the DSCLS model on maximum throughput is also examined. In this experiment, volumes of 2400 Veh/hr and 1200 Veh/hr are applied on the major and minor streets. The simulation time is set to 1 hr, and volume input remains active till the end of the simulation period. Detectors are placed at intersection discharge sections, and the total discharged volume is recorded. According to the results appearing in Figure 3.10, the DSCLS outperforms all other intersection control systems in this measure. A maximum throughput increment of 9% is gained compared to the second-best intersection control system in this experiment.

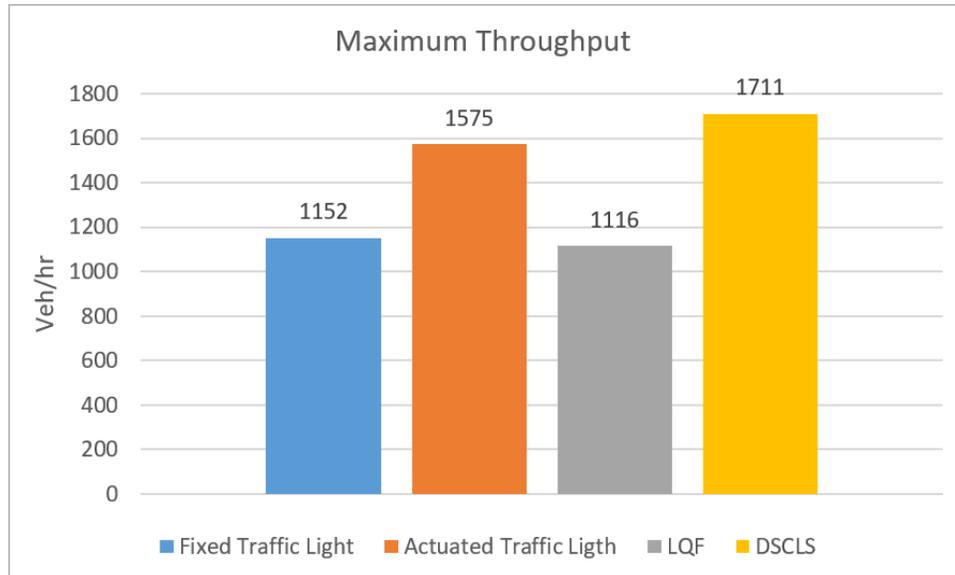


Figure 3.10 Maximum throughput comparison for proof-of-concept test.

3.5 Emission, Safety and Fuel Consumption

Along with targeted traffic measures, emission, safety, and fuel consumption are also evaluated. Fuel consumption and CO₂ emission are calculated based on the VT-Micro model. This model can estimate emission and fuel consumption for individual vehicles based on instantaneous acceleration and speed [59]. The average fuel consumption results are shown in Figure 3.11; fuel consumption reduction of 17%, 8%, and 5% is gained compared to the second-best control systems in moderate, high, and extreme volume regimes accordingly. The average CO₂ emission results are shown in Figure 3.12; an emission reduction of 16%, 7%, and 5% is gained compared to the second-best control systems in moderate, high, and extreme volume regimes accordingly. Despite acceleration and deceleration rates for DSCLS being limited to the maximum values only, fuel consumption and emission are still less than other control systems.

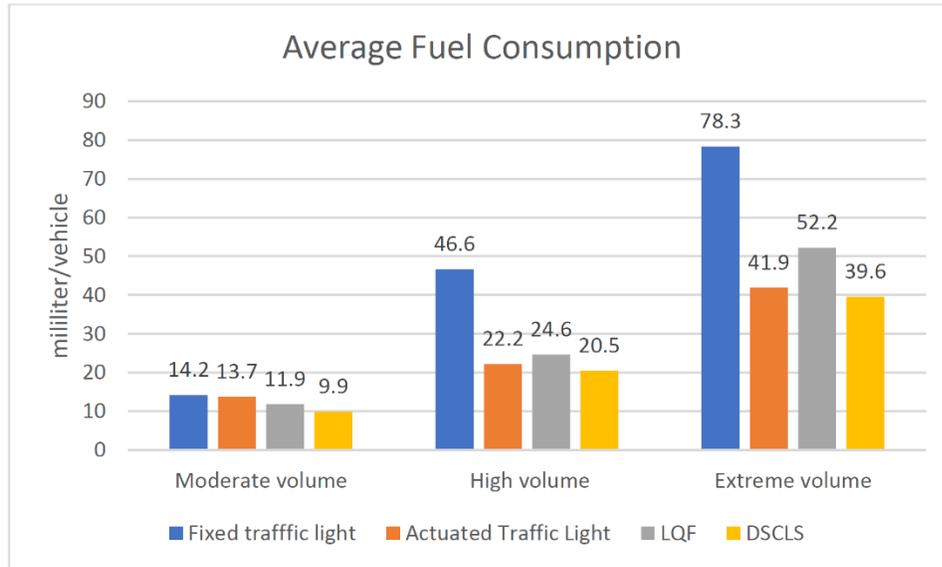


Figure 3.11 Average fuel consumption comparison for proof-of-concept test.

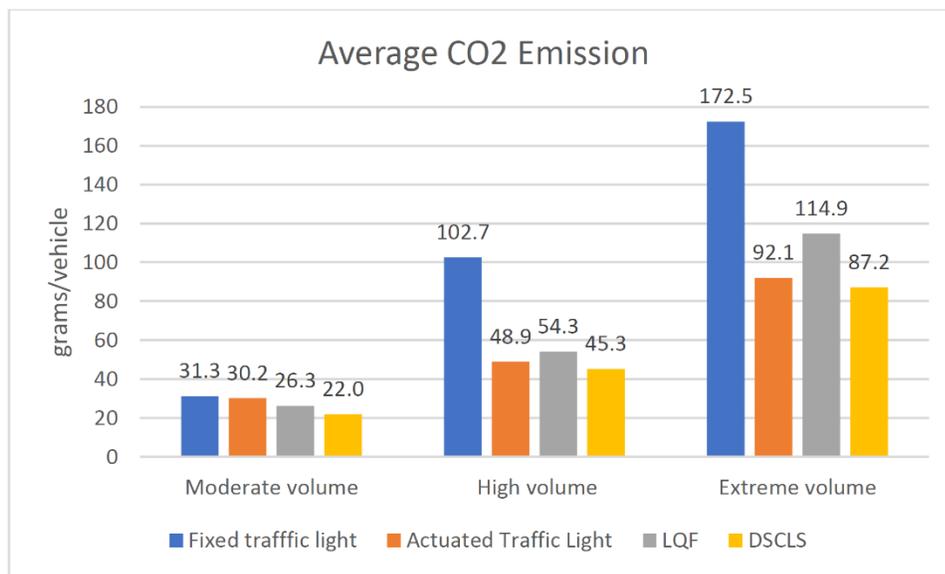


Figure 3.12 Average CO₂ emission comparison for proof-of-concept test.

Among different Surrogate Safety Measures (SSM), the Post Encroachment Time (PET) is a well-fitting measure to identify safety threats for crossing vehicles at the intersection since it represents the time between the departure of the encroaching vehicle

from the conflict point and the vehicle's arrival with the right-of-way at the conflict point. For rear-end crash risk assessment, another SSM named Time To Collision (TTC) has been used [60]. Safety analysis is performed in SSAM software which executes a statistical analysis of vehicle trajectory data output from microscopic simulation models.

The comparison of PET for different traffic control systems is presented in Figure 3.13. According to the figure, DSCLS and LQF have almost equal PET, which is expected since both are pixel reservation-based. The PET increases by 23% in DSCLS and LQF compared to the actuated traffic lights in extreme volume conditions. However, less improvement is observed in moderate and high-volume regimes.

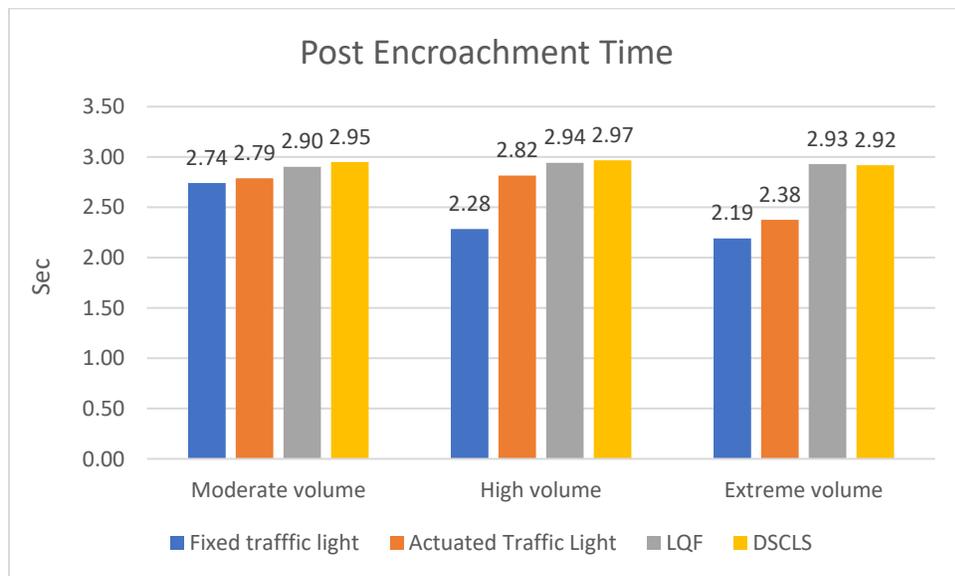


Figure 3.13 Average PET comparison for proof-of-concept test.

According to the TTC results, shown in Figure 3.14, the TTC is decreased by 10% in all volume regimes for pixel reservation-based control systems, including LQF and DSCLS, compared to the actuated traffic light. However, the increment in TTC may not

result in the chance of rear-end accident increment due to the shorter processing time of CAVs compared to human-driven vehicles.

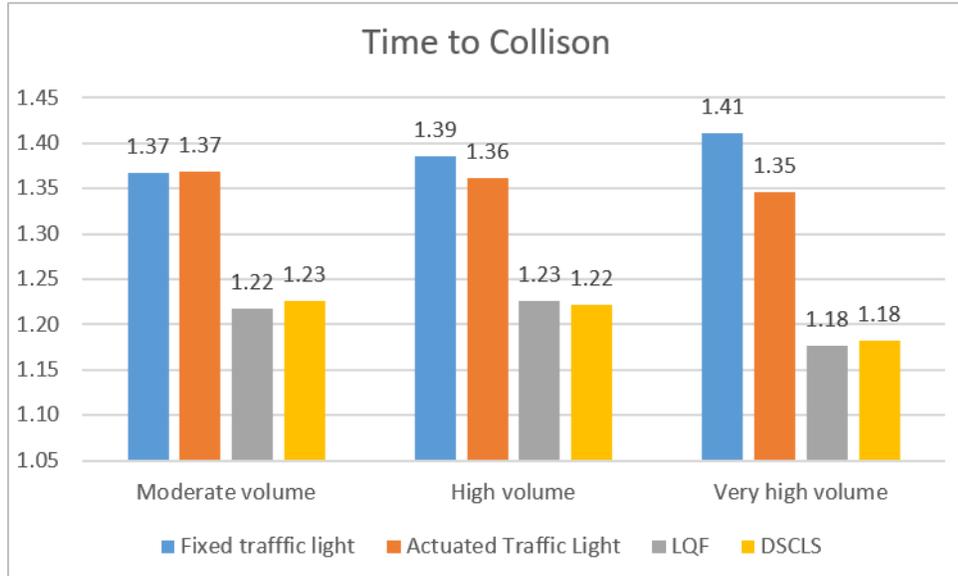


Figure 3.14 Average TTC comparison for proof-of-concept test.

3.6 T-Test Result

A t-test is performed between DSCLS and the other three control systems for all collected measures. According to the t-test results shown in Table 3.2, all measures between DSCLS and other three intersection control systems in all volume regimes are statistically significant except safety measures for both pixel reservation-based control systems, including TTC in all volume regimes and PET in the extreme volume regime.

Table 3.2 P-value for T-test Results – Proof of Concept Test

Fixed Traffic Light								
Moderate volume								
	Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	
	High Volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	<0.05
	Extreme volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
Actuated Traffic Light								
Moderate volume								
	Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	
	High Volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	<0.05
	Extreme volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
LQF								
Moderate volume								
	Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	-	<0.05	<0.05	<0.05	>0.05	
	High Volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	>0.05
	Extreme volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	>0.05	>0.05	

3.7 Chapter Summary

A DSCLS based on DRL was developed in this chapter to control CAVs at the intersections. Followed by elaboration of the methodology and development process of the model, a simulation testbed was set up for proof-of-concept test purposes. The developed model was compared with three counterpart intersection control systems: fixed traffic light, actuated traffic light, and LQF in three different volume regimes. The simulation results revealed that the proposed model outperforms other control systems in traffic measures, specifically in moderate and high volume regimes. Despite improvements in traffic measures, the proposed model shows close performance to other control systems (except fixed traffic lights) in environmental and safety measures.

In Chapter 5, the DSCL will be applied to a more realistic intersection, including several lanes and turning movement, which involves reconfiguring the model and running a new training course. The model will also be applied to a corridor of four intersections.

CHAPTER 4

DEVELOPMENT OF A PLATOONING POLICY FOR CAVs AND PROOF OF CONCEPT TEST

In this chapter, an SMD-based platooning logic is developed for CAVs. The model development process is elaborated by describing the SMD model formulations, parameters, and their impacts on driving behavior. Simulation testbeds with different scenarios are developed for proof of concept test purposes.

4.1 SMD System and Vehicle Platooning

The SMD model describes how objects reduce their oscillations based on spring constant, damping coefficient, and object mass [61]. Figure 4.1 illustrates the SMD system for n objects or vehicles.

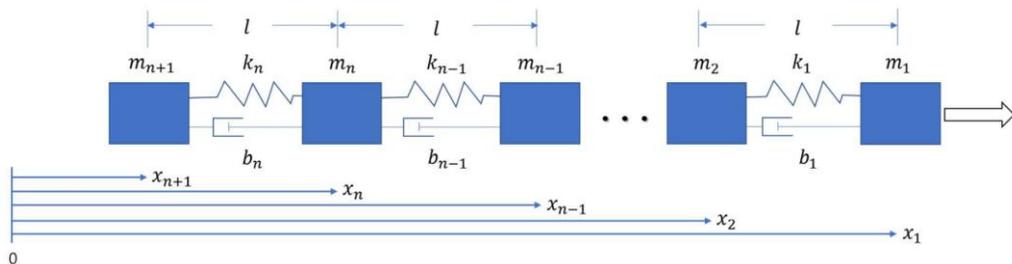


Figure 4.1 A n -body SMD system illustration.
Source: [61].

It is assumed that the friction between masses and surface is negligible, and each mass is connected only to its predecessor and not being affected by its follower. Based on these assumptions, the forces acting on the leading mass and following masses in a string are expressed in Equations (4.1) and (4.2) [61].

$$m_1 \ddot{x}_1 = c(v_d - \dot{x}_1) \quad (4.1)$$

$$\begin{aligned} m_n \ddot{x}_n &= k_{n-1}(x_{n-1} - x_n - l) + b_{n-1}(\dot{x}_{n-1} - \dot{x}_n) \\ &= k_{n-1}\Delta x_n + b_{n-1}\Delta \dot{x}_n \end{aligned} \quad (4.2)$$

Where:

m_i : mass of i^{th} vehicle in the platoon ($i = 1$ for the lead vehicle)

$x_i, \dot{x}_i, \ddot{x}_i$: position, speed, and acceleration of i^{th} vehicle

v_d : desired speed of the lead vehicle

c : acceleration coefficient

k_i, b_i : spring constant and damping coefficient for the i^{th} vehicle

l : Original, unstretched spring length

Parameter c controls the acceleration or deceleration rate of the leading vehicle.

Based on the assumption that the maximum acceleration is attained when $\dot{x}_1 = 0$, and the maximum deceleration is attained when the lead vehicle wants to stop ($v_d = 0$); constraints (4.3) and (4.4) are derived for the c parameter.

$$c \leq \frac{m_1 a_{max}}{v_d} \quad (4.3)$$

$$c \leq \frac{m_1 d_{max}}{x_1} \quad (4.4)$$

The variables a_{max} and d_{max} are the maximum acceleration and deceleration capability of vehicles. According to the characteristics of Ford Fusion, 2019 Hybrid, presented in Table 4.1 [62], a_{max} equals to 3.7 m/sec^2 , and d_{max} equals to 9.023 m/sec^2 . Inserting a_{max} , d_{max} , m , and assuming desired speed equals to 30 m/sec, c_{max} will be

equal to 229.67. To achieve the desired speed in the shortest possible time, the maximum value for c , based on Equations (4.3) and (4.4) is assumed in this study.

Table 4.1 Ford Fusion 2019 Hybrid Specifications

Specification	Value
Weight	1676 kg (3695 lb)
Length	4.87 meter (191.7 inches)
Acceleration time from zero to 60 mph	7.3 sec
Braking distance 70 mph to zero mph	178 ft

Source: [62].

The unstarched spring length (l) represents the minimum neutral spacing for speed v and is calculated based on Newell's simplified car-following model [63]. The l value's function is presented in Equation (4.5). Considering l as unstarched spring length or desired spacing for a specific speed, the term $(x_{n-1} - x_n - l)$ in the model represents deviation from desired spacing.

$$l = s_0 + \tau \times v \quad (4.5)$$

Where:

s_0 : minimum spacing between two vehicles

τ : response/processing time

The CAVs processing time (τ) ranges from 0.5 to 1.0 sec [5], while Human-Driven Vehicles (HDV) reaction time ranges from 1.5 to 2.0 sec [64]. The parameter s_0 represent minimum required spacing and is assumed to be 7 meters, which provides an approximate bumper to bumper space of 2 meters for medium-size sedan cars in idling condition.

4.2 Characteristics of Spring Constant and Damping Coefficient

The spring constant, k_i represents the spring stiffness. A spring with a larger k is harder to stretch or shrink, but once stretched or shrunk, there is a greater force to recover its original length. Thus, a large k represents a stiff spring with a high body acceleration (high frequency). In contrast, a small k represents a limp spring with low acceleration (low frequency). The spring constant in the SMD model is coefficient of Δx_n and is regarded as a CAV's sensitivity to deviation from its desired spacing with the ago vehicle.

The damping coefficient, b_i represents the degree of resistance that alleviates the spring force. The damping force defines how a following CAV approaches its predecessor and adjusts its speed based on the ago vehicle's speed. For example, with a large b , it takes longer for a following vehicle to attain the desired distance, but the speed might be adjusted faster. Meanwhile, with a smaller b , a CAV attains the desired distance faster by oscillating in location and speed. The damping coefficient in the SMD model is the coefficient of deviation from the preceding vehicle's speed and can be regarded as a CAV's sensitivity to speed deviation from its ago vehicle.

Harmony between the spring constant and damping coefficient is essential for the stability and efficiency of CAV platooning. These parameters' critical values lead to the shortest time to attain L without collision or oscillations, called critical damping. Two more possible damping types are 1) under-damping, which is not desirable since vehicles would oscillate back and forth before reaching l , and 2) over-damping, which takes longer to reach L , but there is no oscillation. According to [51], condition of critical-damping and over-

damping can be achieved if $b \geq \max\left(\frac{m}{\tau}, \sqrt{\frac{k}{m}}\right)$, and the system would be critical-damping

$$\text{if } b = \max\left(\frac{m}{\tau}, \sqrt{\frac{k}{m}}\right).$$

If we assume that $\Delta x' = 0$, meaning that the force is only coming from the spring, the upper bound for k , which provides the greatest spring force, and makes vehicles cluster quickly is obtained as follows:

$$m\ddot{x} = ma_{max} = k\Delta x$$

$$k = \frac{1}{\Delta x} ma_{max}$$

Therefore,

$$0 \leq k \leq \frac{1}{\Delta x} ma_{max} \quad (4.6)$$

4.3 Developing CAVs' Platooning Logic Based on SMD Model

In this study, a maximum communication range equal to $4l$ is assumed for CAVs. So, as the speed increases, the communication range increases too, and a CAV only establishes platoon-oriented communication with its predecessor vehicle detected in the communication range. Each platoon is divided into several sub-platoons with a maximum length of four vehicles to avoid very long platoons and accommodate potential merging vehicles. The leading vehicle of each platoon communicates with the last vehicle of the predecessor platoon, and inter-platoon desired spacing is set equal to $3l$.

The maximum value for spring constant and critical value for damper coefficient, presented in Equations (4.7) and (4.8), are used in the model to minimize clustering time.

$$k = \frac{1}{\Delta x} ma_{max} \quad (4.7)$$

$$b = \max\left(\frac{m}{\tau}, \sqrt{\frac{k}{m}}\right) \quad (4.8)$$

Different vehicle types in the model based on their platooning circumstances are shown in Figure 4.2.

Vehicle 1: leading vehicle of a platoon with having no ago vehicle in the communication range. Equation (2.1) defines this vehicle's acceleration, and it keeps accelerating to maintain the desired speed as long as it does not detect any other vehicle in the communication range.

Vehicle 2 and 4: following vehicles in a sub-platoon, maintaining the intra-platoon desired spacing (l), and the acceleration is derived from Equation (4.2).

Vehicle 3: leading vehicle of a sub-platoon, either its sub-platoon or its predecessor sub-platoon, has reached the maximum platoon length. So, this vehicle maintains inter-platoon spacing ($3l$) with its ago vehicle within the communication range.

Vehicle 5: having no vehicle in the communication range and maintaining its desired speed is based on Equation (4.1).

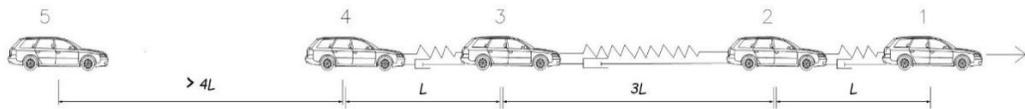


Figure 4.2 The SMD-based platooning system details.

4.4 Microsimulation Testbed Development

The SMD-based CAVs' platooning logic is developed as a Python application integrated with VISSIM traffic microsimulation software. The VISSIM stochastically produces vehicles and directs them through specified routes. It also controls potential human-driven vehicles based on the Wiedemann 99 car-following model [65], and collects traffic measures such as delay and travel time. The python program collects vehicle data such as location and speed, establishes communication between vehicles, and controls CAVs' acceleration based on the SMD model. The Common Object Model (COM) interface allows the python program to access the simulation network's data and objects during simulation. Adjusting editable objects and variables such as traffic volume and individual vehicle's speed is also possible in the COM interface. Integrating VISSIM and Python through the COM interface is shown in Figure 4.3.

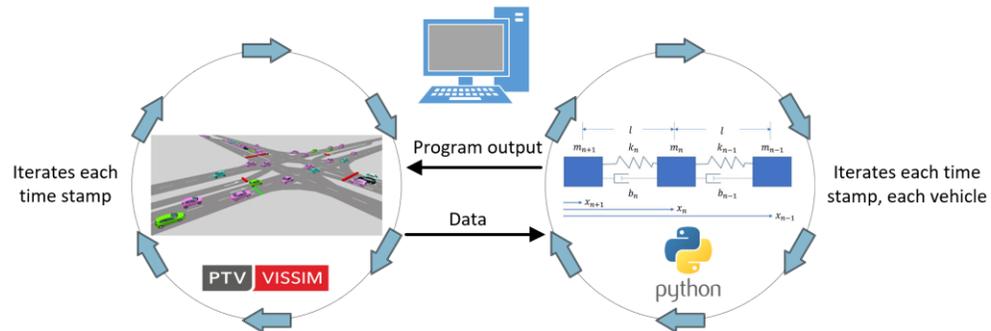


Figure 4.3 Integration of VISSIM and Python.

4.5 Proof of Concept Test

Several scenarios are developed for proof of concept test purposes. Developed scenarios and their results are described in the following sections.

4.5.1 Spacing Error

For initial evaluation of the model, platooning vehicles are exposed to a sudden deceleration scenario, and the average spacing error between CAVs is collected. In this scenario, a platoon of CAVs is following an HDV which suddenly decelerates with a rate of 5.5 m/sec^2 and decrease its speed from 120 km/hr to 30 km/hr. The average spacing error between platooning vehicles is collected over time. The VISSIM's built-in CAVs platooning model added to VISSIM since the 2020 version is adopted as a counterpart of the SMD model. However, the details of this model are unknown, considering that modeling another platooning logic in VISSIM is highly time-consuming and requires lots of programming; the VISSIM built-in model is selected as a counterpart of SMD for initial proof of concept.

The adjustable platooning parameters for the VISSIM built-in model are shown in Table 4.2; these parameters are set to be complying with SMD model hyperparameters. Except for the maximum communication range, which is set to 50 meters in the VISSIM built-in model, since defining this value as a speed function, like what we did for the SMD model, is not an option in VISSIM.

Table 4.2 Vissim Built-in Platooning System Constant Parameters Setting

VISSIM 2020 built-in platooning parameter setting	
Maximum number of vehicles in each platoon	Four vehicles
Maximum communication range	50 meters
Platoon follow up gap time	0.5 sec
Minimum clearance distance	2 meters
Maximum platooning desire speed	120 km/hr
Maximum acceleration	3.7 m/sec ²
Maximum deceleration	9.023 m/sec ²

According to the VISSIM2021 manual, vehicles' safety distance is calculated based on Equation (4.9) [65].

$$PlatoonMinClear + vReference * PlatoonFollowUpGapTm \quad (4.9)$$

Where:

vReference: The lead vehicle's speed in the platoon

The simulation is run five times with a 10 step/sec resolution. The simulation's average result is shown in Figure 4.4.

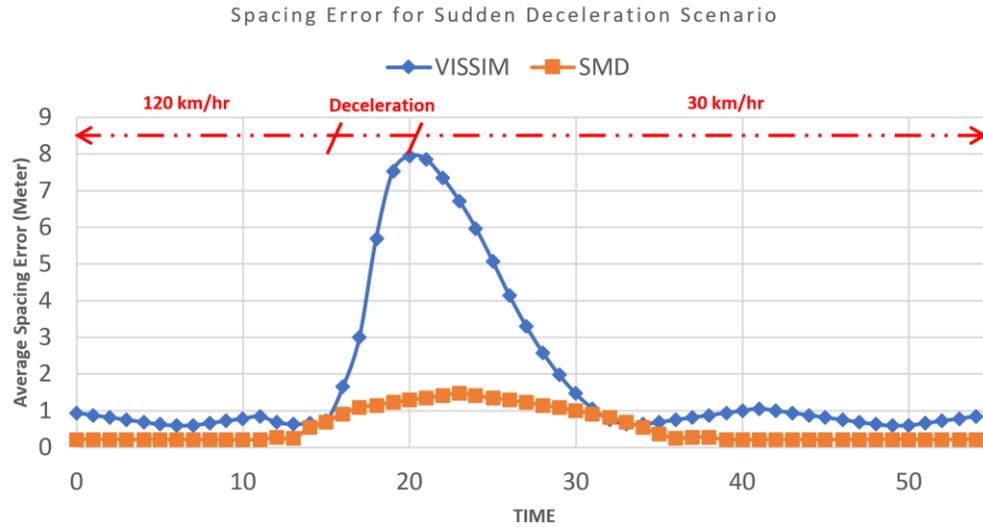


Figure 4.4 Spacing error occurrence in harsh deceleration scenario.

According to the results, minor positive errors (less than one meter) are constantly observed for the VISSIM built-in model while following an HDV driver. An increasing error appears as soon as the deceleration phase starts and reaches the climax of 8 meters at the end of the deceleration phase. After 10 seconds of moving with constant speed, the spacing error is reduced to the minimum value. However, no error is observed regularly in the SMD model, and a maximum error of 1.5 meters is observed at the end of the deceleration phase. The advantage of both models is that no negative spacing error is observed, which is equivalent to a safety threat. However, noticeable positive errors are observed in VISSIM built-in model, which results in a maximum capacity drop.

4.6 Maximum Throughput Assessment

A single-lane highway with a length of 4 kilometers is modeled in VISSIM to evaluate the proposed model's impact on the roadway's maximum throughput under different CAVs' MPR and processing time (τ). HDVs are controlled by VISSIM's car-following model

(Wiedemann 99), and it is assumed that all HDVs are equipped with Vehicle Awareness Device (VAD). VAD is an aftermarket positioning and onboard communication unit proposed by the US Department of Transportation's CVs Initiatives to improve CACC systems' performance in low MPR. A VAD-equipped vehicle broadcasts a Basic Safety Message (BSM), including its location and speed, so its follower CAV can use CACC capability [66].

Simulation results revealed that the maximum throughput increases as the CAVs' MPR increases. A maximum throughput increment of 63% is achieved in a full MPR of AVs with $\tau = 0.5 \text{ sec}$. However, a maximum increment of 23% is achieved if $\tau = 1 \text{ sec}$. Simulation results for various CAV's MPR are presented in Figure 4.5.

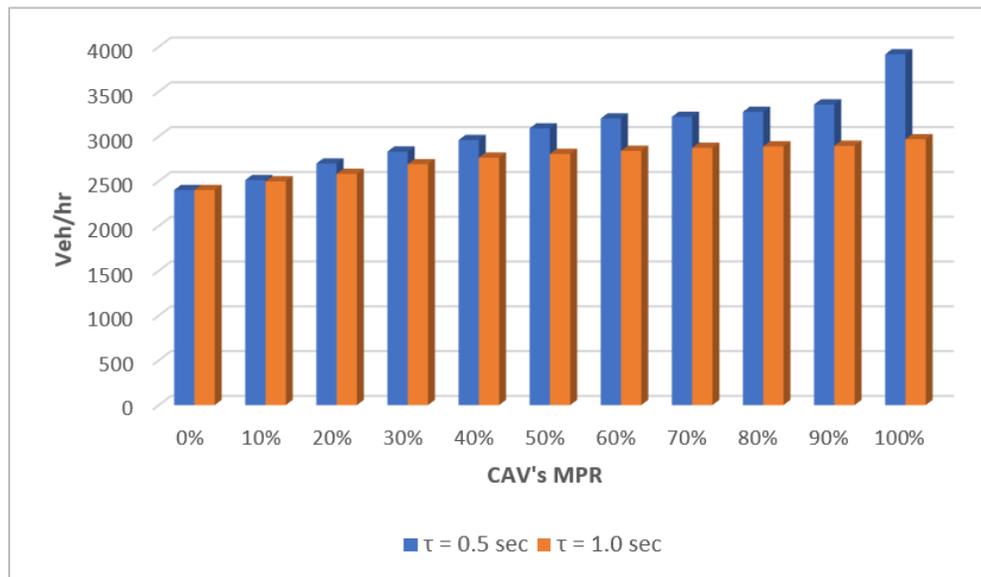


Figure 4.5 Maximum throughput under different MPR of platooning CAVs.

4.7 Platoon Length Impact on Maximum Throughput

Assuming a processing time of 0.5 sec for all CAVs, the impact of platoon length on maximum throughput is assessed in this experiment. Several scenarios are run, increasing the platoon length from 4 vehicles to 12 vehicles with an interval of two vehicles. The simulation results are shown in Figure 4.6; the maximum throughput slightly increases as with platoon length increment. For instance, a maximum throughput increment of 3% is gained if the platoon length increases from 4 vehicles to 12 vehicles with a 0.5 sec processing time.

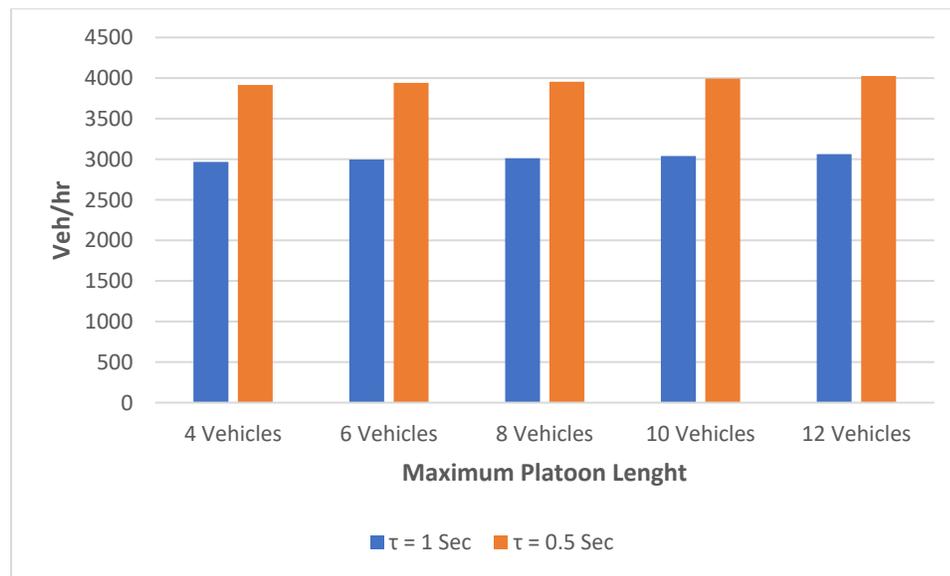


Figure 4.6 Impact of platoon length on maximum throughput.

4.8 Travel Time at Merging Section

As shown in Figure 4.7, a merging section is modeled in VISSIM. In this scenario, CAVs on the mainlane are in platooning mode, and CAVs on the merging lane will change lane as soon as enough space on the mainlane is observed. This scenario aims to assess the impact of inter-platoon spacing policy in accommodating merging vehicles. Controlling

the merging maneuver is not a point of interest in this study, and lane-changing maneuvers are performed by VISSIM software. Vehicles from the merging lane start platoon-oriented communication as soon as they enter the mainlane to join the existing platoons on the mainlane.

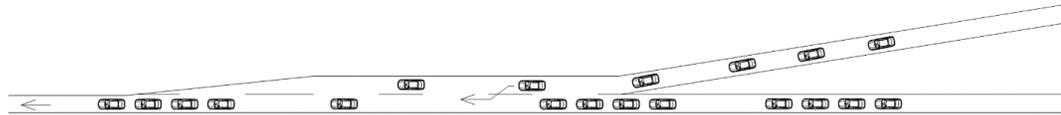


Figure 4.7 Merging section scenario details.

The maximum speed on the main and merging lanes is 30 meters/second. Several scenarios with combinations of high and extreme volumes on the mainlane and moderate, high, and extreme volumes on the merging lane, with different MPR of CAVs, are run. The CAVs' processing time is assumed to be 0.5 sec in all scenarios, and the maximum platoon length on the mainlane is limited to four vehicles. Simulation time is set to 15 minutes, and the resolution is set to 10 steps/sec. The presented results are the average of five simulation runs with different random seeds. The travel time's data collection length is 280 meters for both merging lane and mainlane. The travel time values for the vehicle on the mainlane are presented in Figure 4.8 and Figure 4.9 for high and extreme volume regimes on the mainlane. According to the results, the travel time on the mainlane decreases as the CAV's MPR increases, specifically if the mainlane volume is extreme. For instance, assuming extreme volume regimes on the mainlane and merging lane, travel time reductions of 36% and 47% are observed in the case of CAVs' MPR shifting from 0% to 40% and from 40% to 100% accordingly (Figure 4.9).

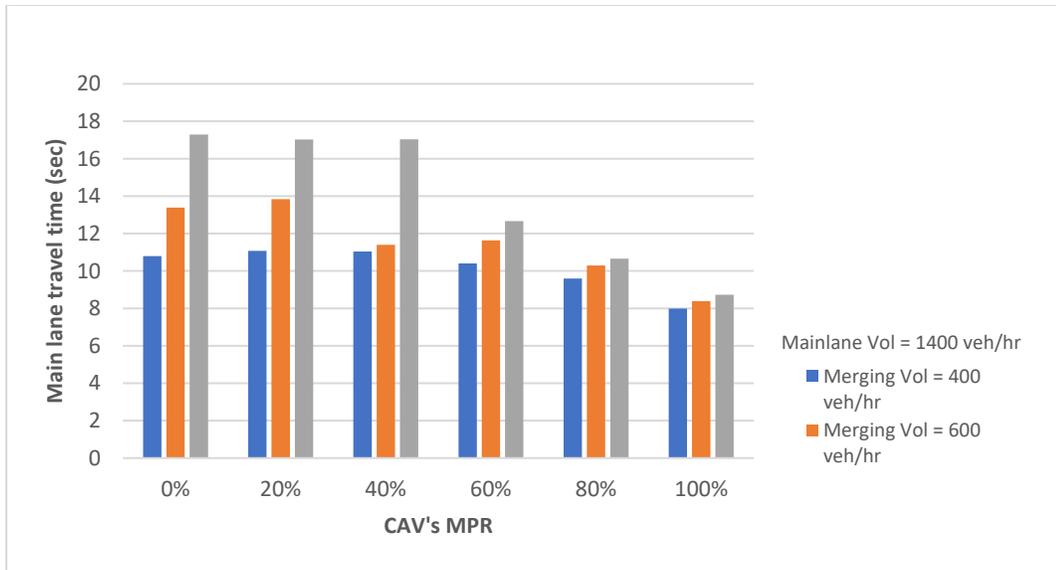


Figure 4.8 Travel time on the mainlane – mainlane volume = 1400 Veh/hr.

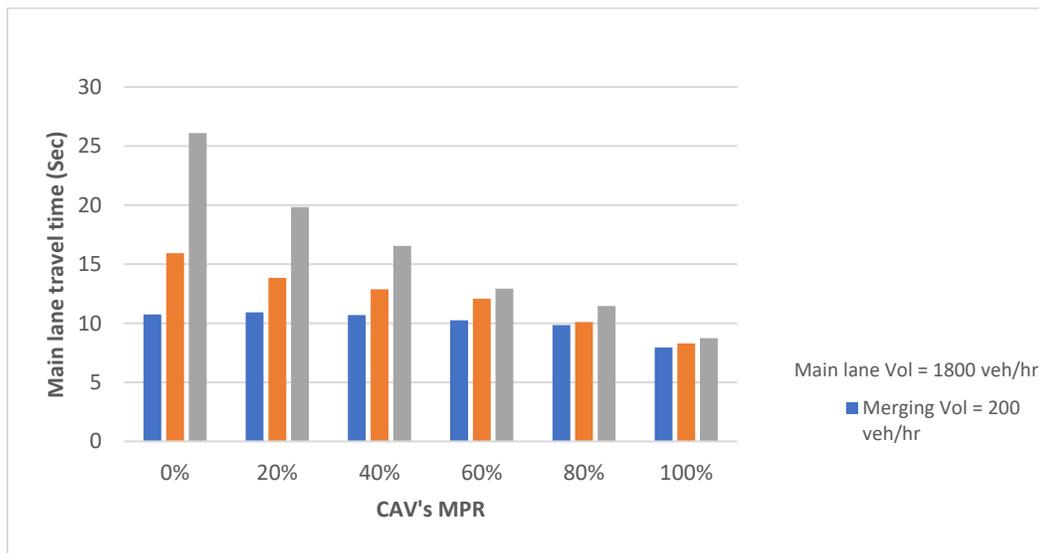


Figure 4.9 Travel time on the mainlane – mainlane volume = 1800 Veh/hr.

The travel time values for vehicles on the merging lane are presented in Figure 4.10 and Figure 4.11. According to the results, deploying CACC on the mainlane remarkably decreases travel time on the merging lane if the CAVs' MPR is 80% and higher and the

merging volume is high. For instance, a travel time reduction of 33% is gained on the merging lane in case of extreme volume on the mainlane and merging lane and a full MPR of CAVs.

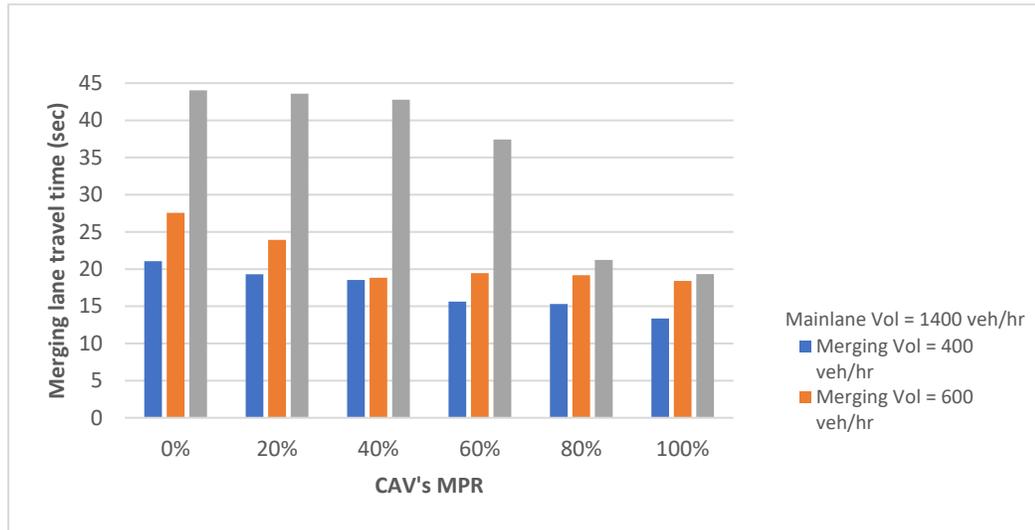


Figure 4.10 Travel time on the merging lane – mainlane volume = 1400 Veh/hr.

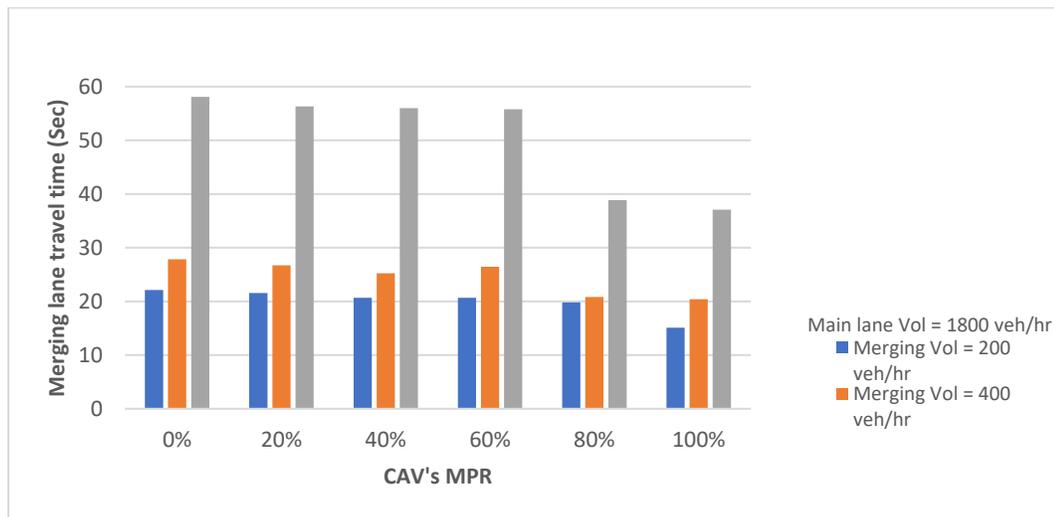


Figure 4.11 Travel time on the merging lane – mainlane volume = 1800 Veh/hr.

4.9 Chapter summary

In this chapter, a car following logic for platooning CAVs was developed based on an existing SMD model. Adjustments such as setting limitations on maximum CAVs' communication range and maximum platoon length were applied to the model to make it more aligned with real-world circumstances. The proposed model was coded into VISSIM software for proof-of-concept test purposes and future traffic-oriented assessments. The proof-of-concept test results revealed that the SMD model produces a smaller spacing error than a counterpart platooning logic in a harsh deceleration scenario. It was proved that the maximum throughput noticeably increases with an increment in CAVs' MPR or CAVs' processing time. The platoon length increment has a very slight impact on maximum throughput. The proposed model can noticeably improve travel time for the mainlane in the merging section even with low CAVs' MPR in the case of extreme volume regimes. Travel time improvement on the merging lane was also observed in the high MPR of CAVs.

CHAPTER 5

EXPANSION OF DSCLS AND DEVELOPMENT OF A INTERSECTION CONTROL LOGIC FOR PLATOONING CAVS

In Chapter 3, an RL-based intersection control system called DSCLS was developed to manage CAVs at the intersections. As a proof-of-concept test, the model was applied to a single-lane intersection with no turning movement; the simulation results revealed noticeable improvements in traffic measures. In Chapter 4, a SMD-based car following model was developed to control CAVs' platooning behavior; the initial evaluations of the model confirmed traffic measures and safety improvements in uninterrupted traffic flow circumstances. In this chapter, the proposed intersection control system will be applied to a more realistic intersection, requiring model reconfiguration and running a new training course. The DSCLS is also applied to a corridor of 4 intersections. Finally, the SMD model and DSCLS are combined, and an intersection control system for platooning CAVs is developed.

5.1 Application of DSCLS to a Multilane Intersection

The testbed in this experiment is an urban intersection, including three lanes on each approach and turning movements. The intersection's environment settings appear in Figure 5.1.

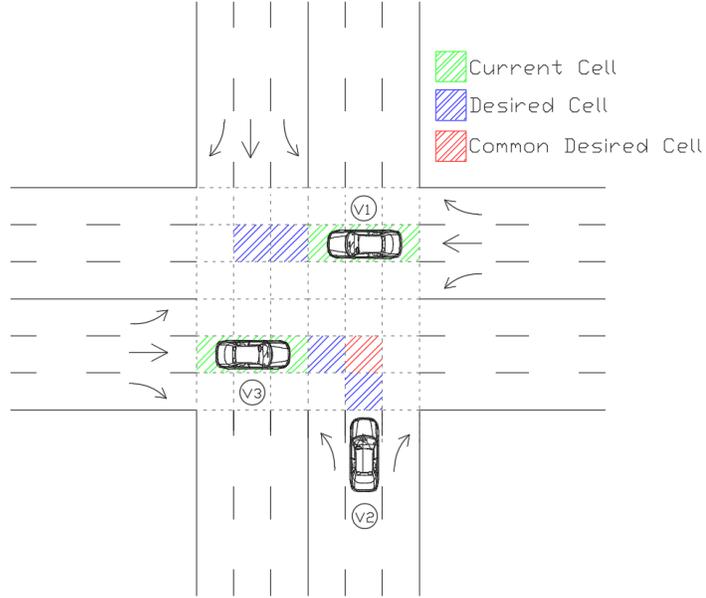


Figure 5.1 The intersection environment physical settings.

The following adjustments have been applied to the model in the new experiment: Colliding vehicles in the training course would receive a negative reward equal to $(-10 \times \Delta t)$. According to Equation (3.15), the collision reward is equal to reward values for ten stopped vehicles at the intersection ($L_i = 0$). In the proof of concept test experiment, independent states and coordinated states of two, three, and four vehicles were forwarded to different neural networks. In this experiment, followed by proper adjustments to the model, a single neural network performs the reward estimation task for all states.

5.2 Evaluation of DSCL in the Multi-lane Intersection Setting

Same as DSCLS's proof of concept test experiments, presented in Chapter 3, the multi-lane intersection trained model is compared with three other control systems, including 1) fixed traffic signal, 2) actuated traffic signal, and 3) Longest Queue First (LQF). The simulation settings such as simulation resolutions, simulation time, number of simulation

runs, and vehicles' maximum speed are the same as proof-of-concept test experiments. The testbed environment complies with Figure 5.1. Three different volume regimes described below are considered for evaluation purposes:

1. Moderate volume: consists of 1,150 and 850 Veh/hr on the major and minor streets, respectively. This volume combination leads to LOS B for the major street and LOS C for the minor street.
2. High volume: consists of 1,600 and 1,100 Veh/hr on the major and minor streets, respectively. This volume combination leads to LOS D for both major and minor streets.
3. Extreme volume: consists of 2,000 and 1,300 Veh/hr on the major and minor streets, respectively. This volume combination leads to LOS E to F or congestion for major and minor streets.

The LOS is calculated based on HCM 6th edition in VISTRO software, assuming the intersection operates under fully actuated traffic signals.

5.2.1 Traffic Measures Comparison

Since the proposed control system's optimization goal is to minimize the delay, the delay is specified as the target traffic measure. A comparison of the average delay between different control systems appears in Figure 5.2. According to the figure, the DSCLS effectively reduces delay in all volume regimes, specifically in moderate volume, with a 58% delay reduction compared to the second-best control system. A delay reduction of 19% and 13% is achieved compared to the second-best control systems in high and extreme volume regimes.

One reason for the poor performance of LQF in extreme volume regimes is that, unlike the DSCLS, this model does not consider approaching vehicles' speed in decision making. The model strictly asks the vehicles in the direction with a shorter queue length to yield to vehicles in the other direction, regardless of the current speed of vehicles in both

directions. The LQF model could also cause occasional switching between giving passing orders to vehicles. However, the DSCL logic has definitely learned to avoid this behavior if it causes a delay increment.

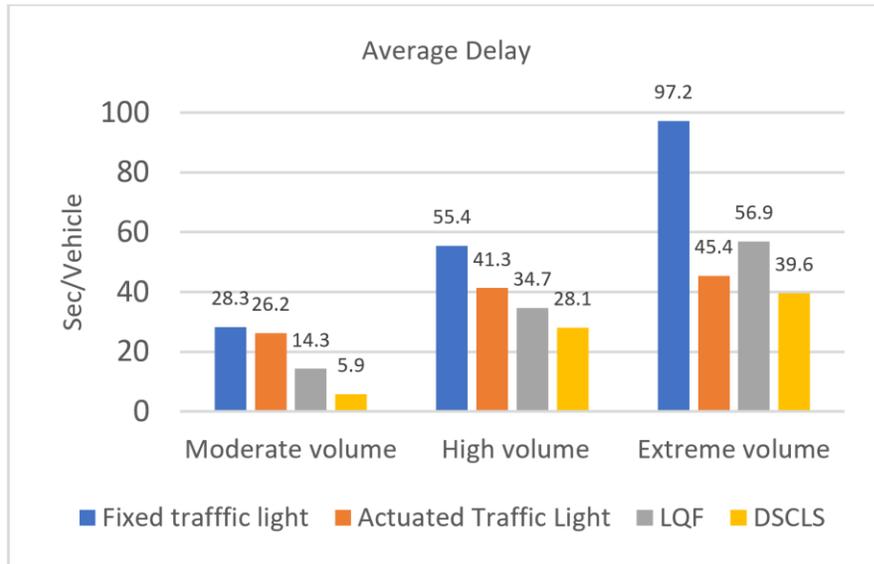


Figure 5.2 Multi-lane intersection average delay comparison.

Followed by delay reductions, travel time improvement is also expected. The average travel time values shown in Figure 5.3 reveal that the proposed model gains 26% and 10% travel improvement in moderate and high-volume regimes compared to the LQF control system. The DSLS outperforms the actuated control system with a 9% travel time reduction in the extreme volume regime.

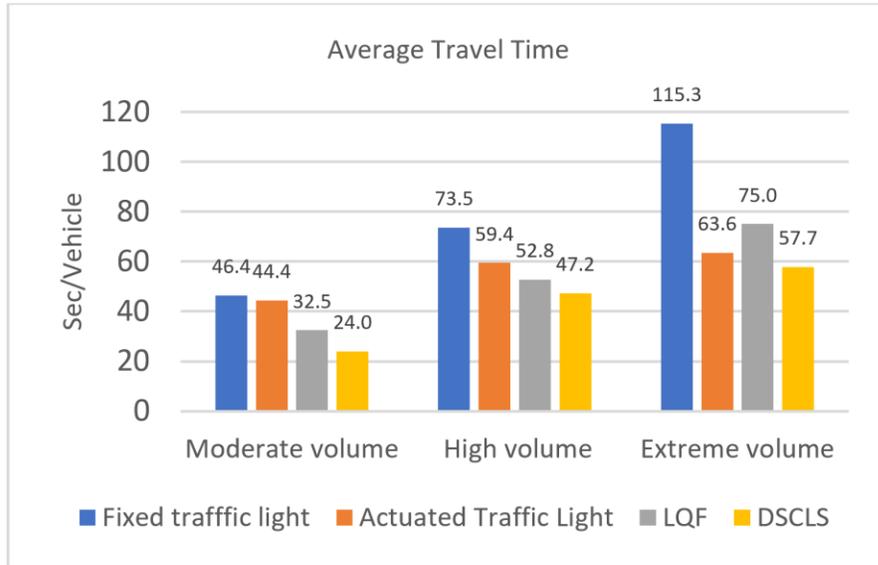


Figure 5.3 Multi-lane intersection average travel time comparison.

Volumes of 4000 Veh/hr and 2400 Veh/hr are applied on the major and minor streets to assess the impact of model on maximum throughput. According to the results appearing in Figure 5.4, the DSCLS outperforms all other intersection control systems in this measure. A maximum throughput increment of 10% is gained compared to this experiment's second-best intersection control system.

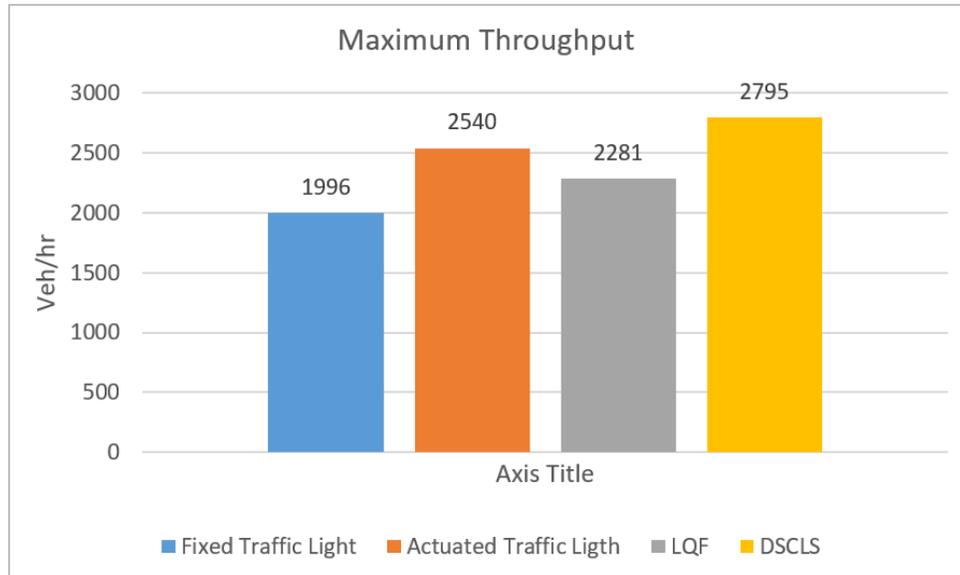


Figure 5.4 Multi-lane intersection maximum throughput comparison.

5.2.2 Other Measures Comparison

Along with target traffic measures, the environmental and safety impacts of the model are also evaluated. Fuel consumption and CO₂ emission are calculated based on the VT-Micro model. The average fuel consumption results appear in Figure 5.5; 13% and 4% reductions are gained compared to the second-best control systems in moderate and high-volume regimes. However, in the extreme volume regime, the fuel consumption is 4% higher than the actuated traffic lights. Despite noticeable improvements in delay, travel time, and maximum throughput, the proposed model does not significantly improve environment measures since the model assumes constant acceleration and deceleration rates for CAVs. However, the DSCLS noticeably outperforms LQF in all volume regimes, which is also based on constant acceleration and deceleration rates for CAVs.

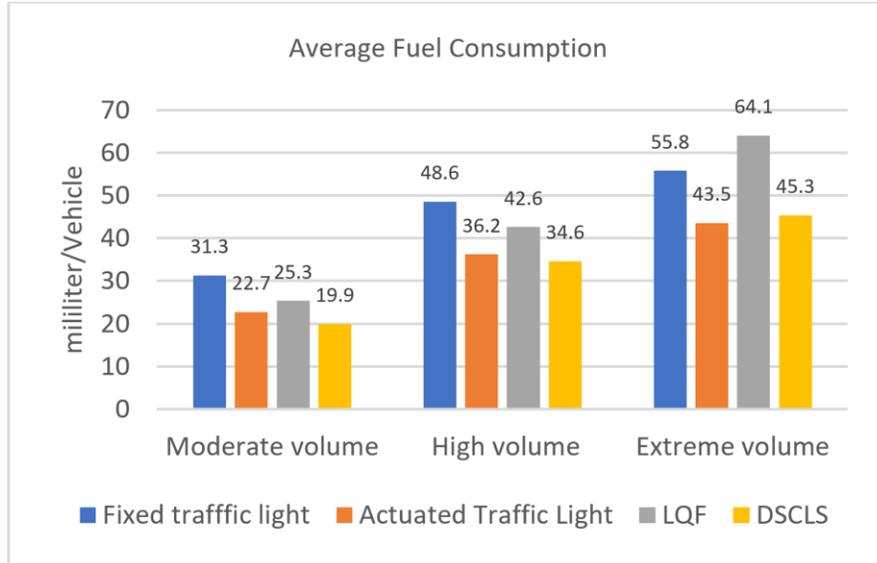


Figure 5.5 Multi-lane intersection average fuel consumption.

As shown in Figure 5.6, average CO₂ emission follows the same pattern as average fuel consumption. An emission reduction of 12%, 4% gained compared to the actuated traffic signal in moderate and high volume regimes. However, the actuated traffic light outperforms DSCLS by 4% in CO₂ emission in the extreme volume regime.

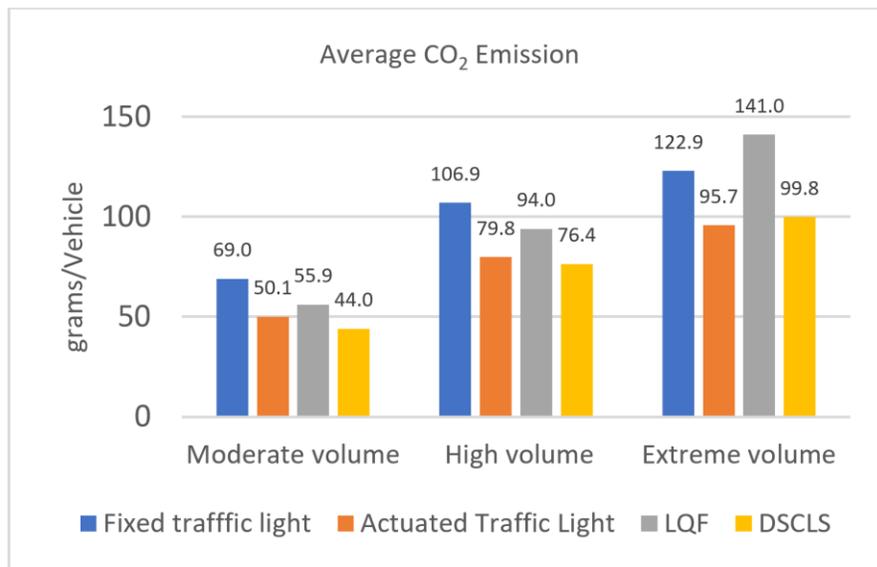


Figure 5.6 Multi-lane intersection average CO₂ emission comparison.

The comparison of PET for different traffic control systems appears in Figure 5.7. According to the figure, both pixel reservation-based logics, including DSCLS and LQF, have close PET values. The PET increases by 28% in DSCLS and LQF compared to the actuated traffic lights in extreme volume conditions; 5% and 7% improvements are recorded in moderate and high-volume regimes. Pixel-reservation-based control systems consist of more stop and go, which increases the chance of side-impact accidents in regular intersections. Therefore, even minor improvement in PET compared to conventional control systems is noticeable.

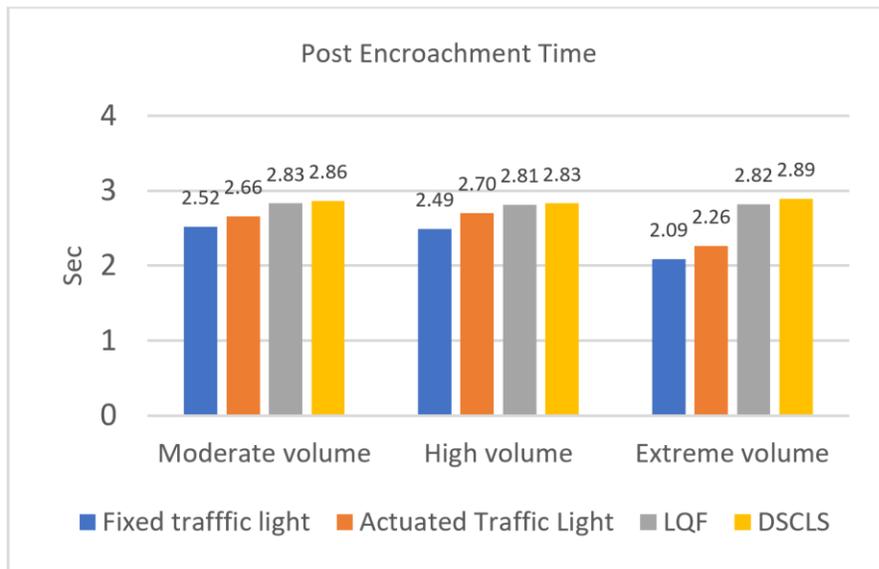


Figure 5.7 Multi-lane intersection average PET comparison.

According to the TTC measures comparison shown in Figure 5.10, both pixel reservation models show poor performance compared to the conventional control systems with 10%, 7%, and 6% decrement in TTC in moderate, high, and extreme volume regimes accordingly. However, the increment in TTC may not result in the chance of rear-end accident increment due to the shorter processing time of CAVs compared to human-driven vehicles.

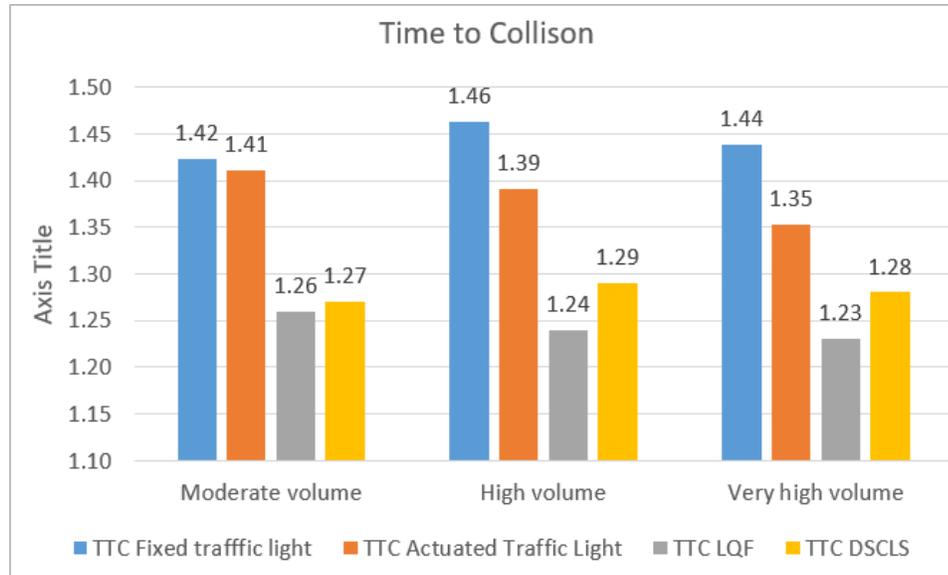


Figure 5.8 Multi-lane intersection average TTC comparison.

5.2.3 Statistical Analysis of the Simulation Results

According to the t-test results appearing in Table 5.1, fuel consumption and CO2 emissions results between DSCLS and actuated traffic lights are not statistically significant in high and extreme volume regimes. Safety measures, including PET and TTC, are not statistically significant between DSCLS and LQF, which is expected since both models are based on the pixel reservation approach.

Table 5.1 P-values for T-test Results for DSCLS - Single Intersection

Fixed Traffic Light								
Moderate volume								
	Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	
	High Volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	<0.05
	Extreme volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
Actuated Traffic Light								
Moderate volume								
	Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	-	<0.05	<0.05	<0.05	<0.05	
	High Volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	-	>0.05	>0.05	<0.05	<0.05	
	Extreme volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	>0.05	>0.05	<0.05	<0.05		
LQF								
Moderate volume								
	Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	-	<0.05	<0.05	>0.05	>0.05	
	High Volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	-	<0.05	<0.05	>0.05	>0.05	
	Extreme volume							
		Delay	Travel Time	Maximum Throughput	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	>0.05	>0.05	

5.3 Application of DSCLS to a Corridor of Four Intersections

The DSCLS is applied to a corridor of four intersections to assess its impact on traffic flow in a more extensive network. The simulation results are compared with three other control systems under moderate, high, and extreme volume regimes, resulting in LOS E to F, LOS D to E, and LOS A to B for most intersections, if they are controlled by conventional control systems. In this experiment, all intersections have the same lane configuration as in Figure 5.1, and they are 500 meters apart. The simulation settings are set the same as proof-of-concept test experiments, and the results are presented in the following sections.

5.3.1 Traffic Measures Comparison

A comparison of the average delay between different control systems appears in Figure 5.9. The DSCLS effectively reduces delay compared to the second-best control system in moderate, high, and extreme volume regimes by 50%, 29%, and 23% accordingly. The DSCLS noticeably outperforms LQF in all volume regimes. Application of the model to a corridor shows more delay improvement than the single intersection experiment, specifically in high and extreme volume regimes.

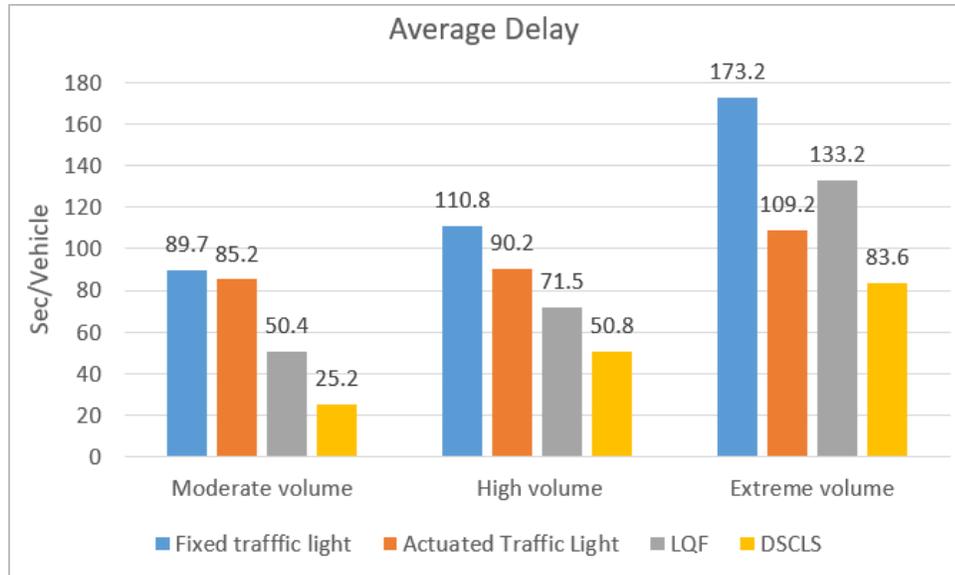


Figure 5.9 Corridor’s average delay comparison.

The average travel time values follow the same trend as the delay. Travel time improvements of 22%, 16%, and 11% are gained compared to the actuated traffic light in moderate, high, and extreme volume regimes.

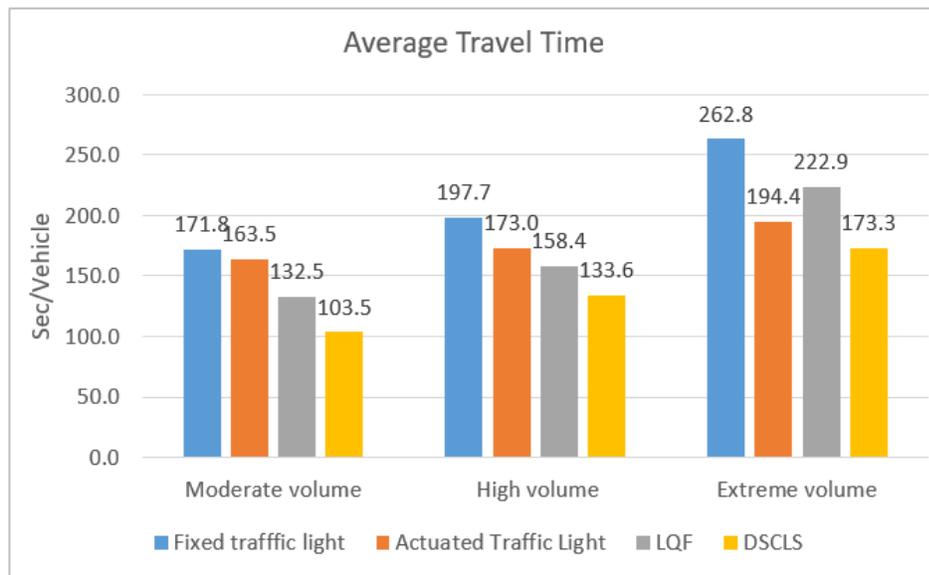


Figure 5.10 Corridor’s average travel time comparison.

5.3.2 Other Measures Comparison

According to the average fuel consumption results presented in Figure 5.11, the DSCLS gains a 9% fuel consumption reduction in the moderate volume regime compared to the actuated control system. However, the actuated traffic light outperforms DSCLS with 2% and 6% fuel consumption reductions in high and extreme volume regimes. The proposed control system noticeably outperforms the other pixel reservation-based model (LQF) in all volume regimes. The fuel consumption measures follow a similar trend as the single intersection experiment.

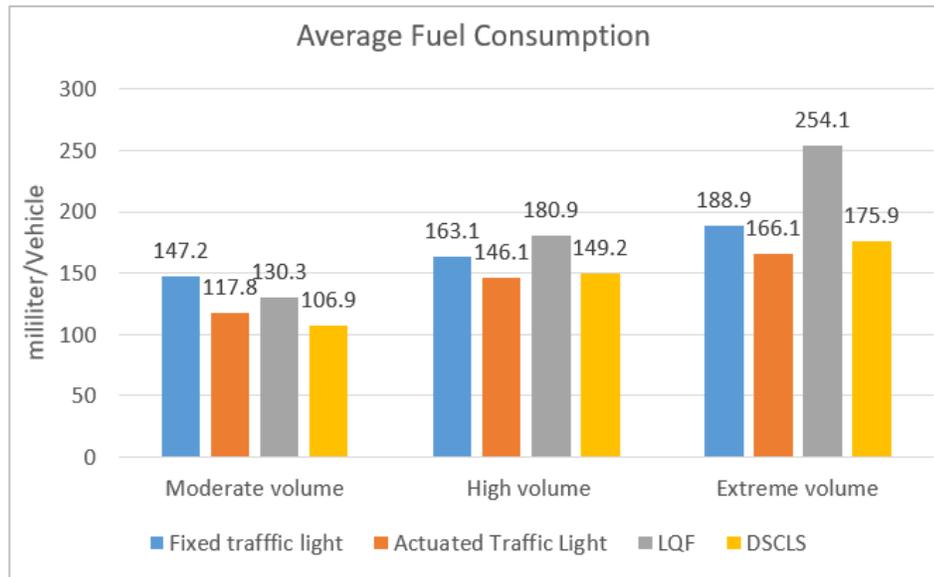


Figure 5.11 Corridor’s average fuel consumption comparison.

The average CO₂ emission results are shown in Figure 5.12. The DSCLS outperforms actuated traffic lights with a 9% CO₂ emission improvement. However, the actuated traffic light performs better than DSCLS with 4% and 7% less CO₂ emission in high and extreme volume regimes.

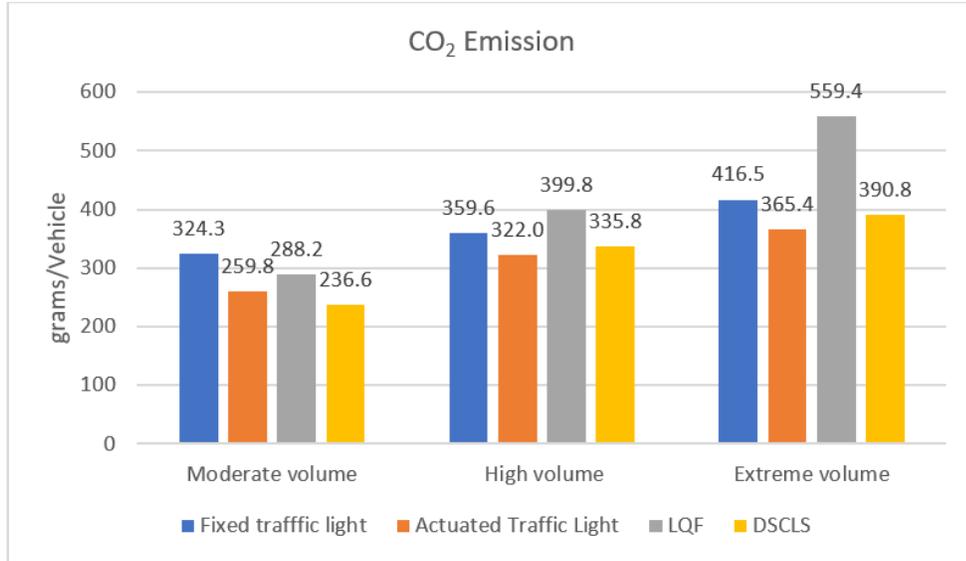


Figure 5.12 Corridor’s average CO₂ emission comparison.

The comparison of PET for different traffic control systems appears in Figure 5.13. According to the figure, pixel reservation-based logics, including DSCLS and LQF, have almost equal PET. Both have a better performance than the actuated traffic light, specifically in extreme volume regimes with a 22% improvement in PET. The results comply with the single intersection experiment.

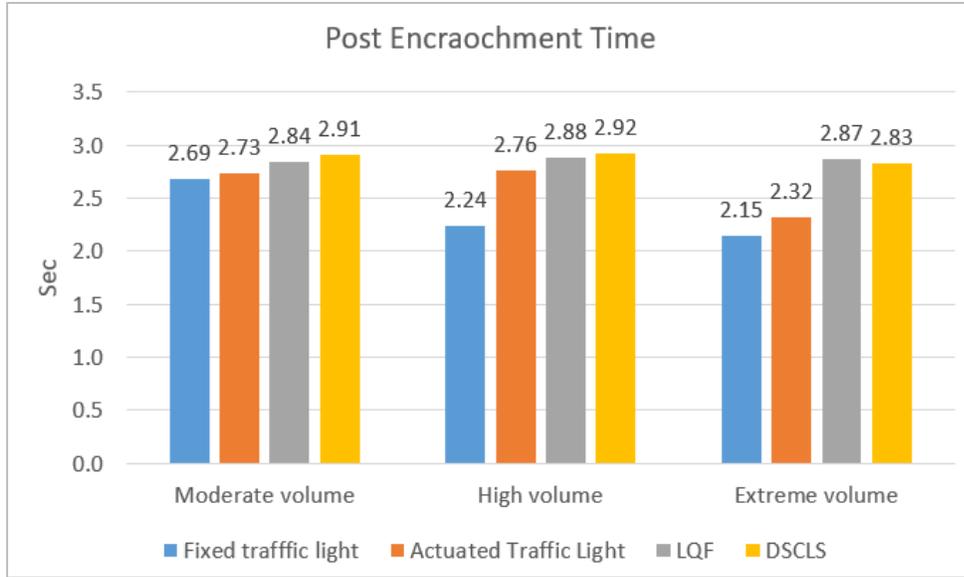


Figure 5.13 Corridor’s average PET comparison.

According to the TTC measures shown in Figure 5.14, both pixel reservation models show poor performance compared to the conventional control systems, with around a 10% TTC decrement in all volume regimes. However, the increment in TTC may not result in the chance of rear-end accident increment due to the shorter processing time of CAVs compared to human-driven vehicles.

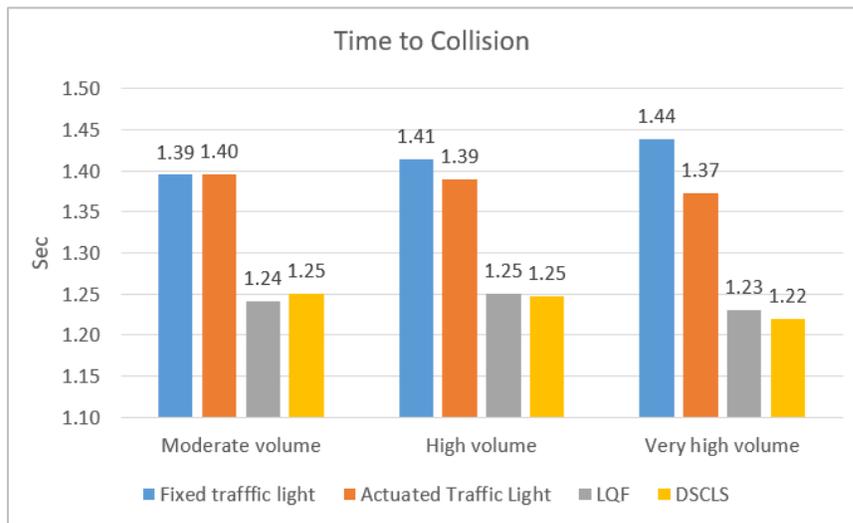


Figure 5.14 Corridor’s average TTC comparison.

5.3.3 Statistical Analysis of the Simulation Results

According to the t-test results, TTC between LQF and DSCLS is not significant in any volume regime. PET is not significant in high and extreme volume regimes. The environmental measures, including Co₂ emission and fuel consumption between DSCLS and the actuated traffic light in the high volume regime. All other measures are statistically significant. T-test results are shown in Table 5.2.

Table 5.2 P-values for T-test Results for DSCLS - Corridor

Fixed Traffic Light							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
Actuated Traffic Light							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	>0.05	>0.05	<0.05	<0.05	
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
LQF							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS	<0.05	<0.05	<0.05	<0.05	<0.05	>0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	<0.05	<0.05	<0.05	>0.05	>0.05
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	>0.05	>0.05	

5.4 Development of a Platooning CAV-based Intersection Control System

One of the main goals of this study is to develop a platooning CAV-based intersection control system, which includes the simultaneous deployment of SMD platooning logic and the DSCLS. It also requires making adjustments to the models to be compatible. A general layout of platooning CAV-based intersection control system based on the SMD model and DSCLS is shown in Figure 5.15. This model is called DSLCS&SMD.

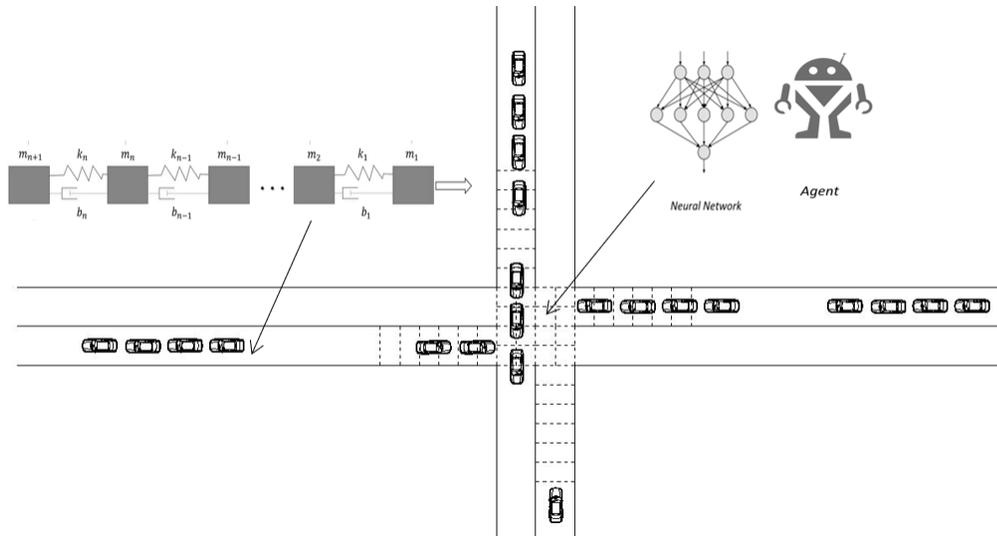


Figure 5.15 Platooning CAV-based intersection control system.

In this experiment, a full MPR of CAVs is assumed, meaning that the SMD car-following model controls all vehicles in the network. The leading vehicles of the first platoon approaching the intersection will be forwarded to the DSCLS to define passing order priority. Once the leading vehicle receives an acceleration or deceleration command from DSCLS, all the following vehicles in its platoon follow that command till the platoon passes the intersection. In this case, the DSCLS needs to be reconfigured to consider a platoon of vehicles instead of a single-vehicle. To this end, the platoon length has been

added to each leading vehicle's state, as appears in Equation (5.1). The training course has been rerun with the new settings.

$$State_{Li} = (Speed_{Li}, Current\ Cell_{Li}, Queue_{Li}, Platoon\ Length_{Li}) \quad (5.1)$$

Where:

Speed_{Li}: platoon's leading vehicle speed

Current Cell_{Li}: platoon's leading vehicle current cell

Queue_{Li}: Queue length behind platoon's leading vehicle

Platoon Length_{Li}: Leading vehicle's platoon length

Since the simulation results for the single lane intersection, multi-lane intersection, and the corridor of four intersections comply in the aspect of gains or losses of the proposed model. The simulation testbed in this experiment is the same as the proof-of-concept test experiment presented in Figure 3.4. The proposed model is compared with four other intersection control systems, including fixed traffic light, actuated traffic light, LQF, and DSCLS, in three different volume regimes. The simulation settings are set the same as proof-of-concept test experiments, and the simulation results are presented in the following sections.

5.4.1 Traffic Measures Comparison

According to the delay results comparison shown in Figure 5.16, the DSCLS&SMD model noticeably improves the delay in the extreme volume regime with a 31% delay reduction compared to the DSCLS. A delay reduction of 10% is observed in moderate and high volume regimes. Since fewer platoons are formed in the moderate and high volume regimes due to lower occupancy, the model is less effective in delay reduction than in the extreme

volume regime. The delay results confirm that clustering vehicles into platoons can improve DSCLS decision-making efficiency.

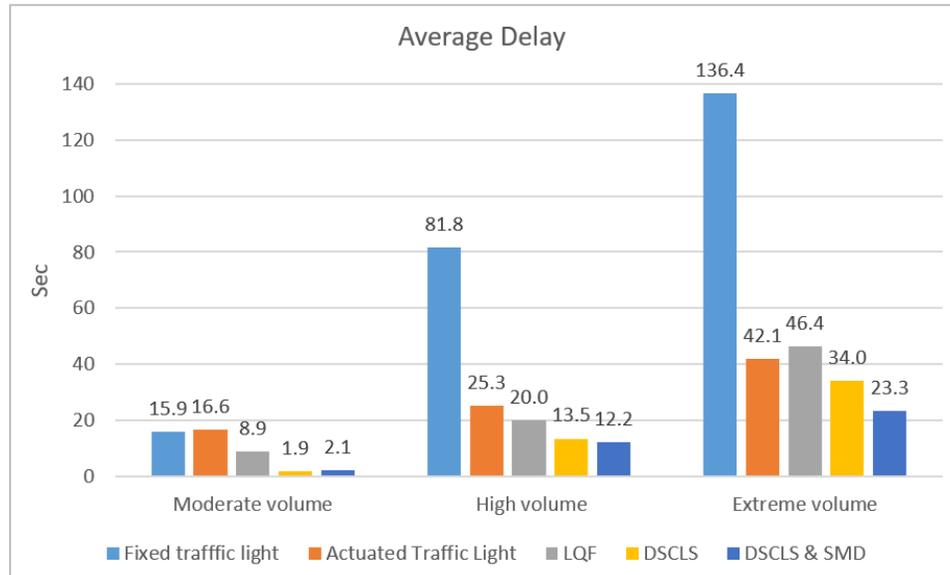


Figure 5.16 Average delay comparison, including DSCLS&SMD.

The DSCLS&SMD gains a 10.7% travel time reduction compared to the DSCLS in the extreme volume regimes. However, the impact of the model on travel time is not noticeable in moderate and high volume regimes. The travel time results appear in Figure 5.17.

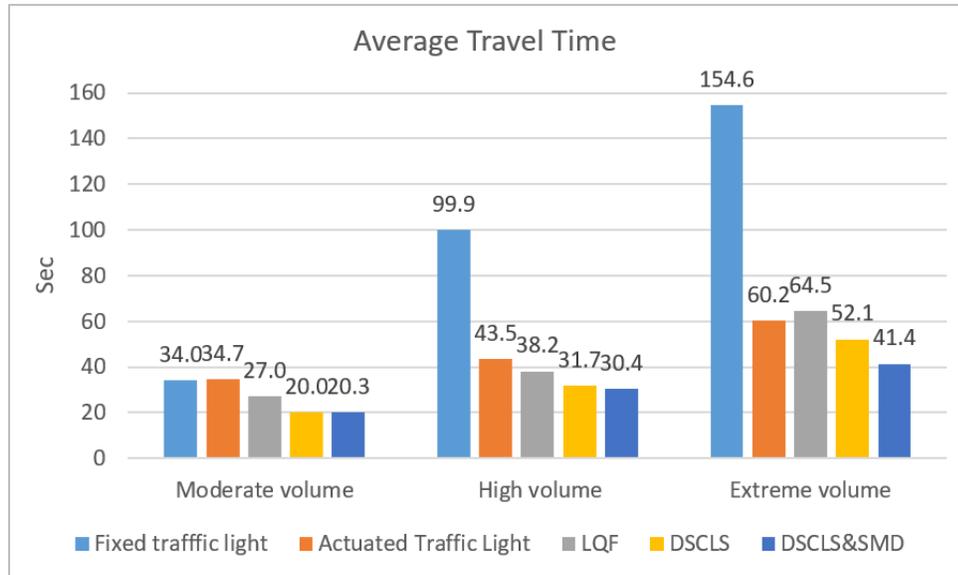


Figure 5.17 Average travel time comparison, including DSCLS&SMD.

According to the maximum throughput results in Figure 5.18, the DSCLS&SMD noticeably outperforms the DSCLS by an 18% increment in the maximum throughput.

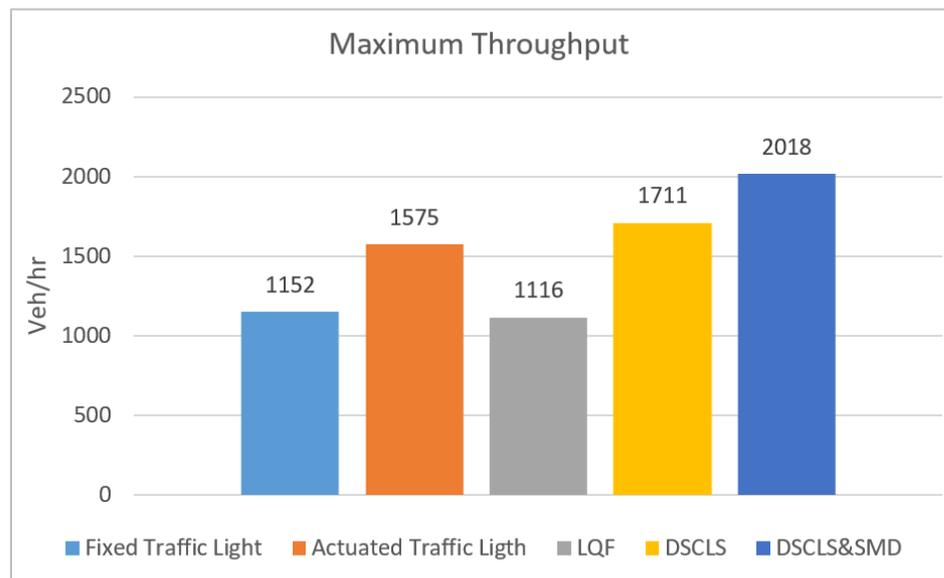


Figure 5.18 Maximum throughput comparison, including DSCLS&SMD.

5.4.2 Other Measures Comparison

The average fuel consumption results appear in Figure 5.19. The DSCL&SMD increases the fuel consumption by 21%, 25%, and 31% compared to the DSCLS model in moderate, high, and extreme volume regimes. The fuel consumption increment results from the SMD model pushing all vehicles to strictly follow the leading vehicle, which is controlled by DSCLS with limited acceleration and deceleration rates. However, the DSCLS&SMD still has almost equal or better performance in fuel consumption compared to the conventional control systems and the LQF.

The average CO₂ emission shown in Figure 5.20 follows the same pattern as fuel consumption with 20%, 24%, and 31% fuel consumption increment for platooning CAVs compared to the DSCLS.

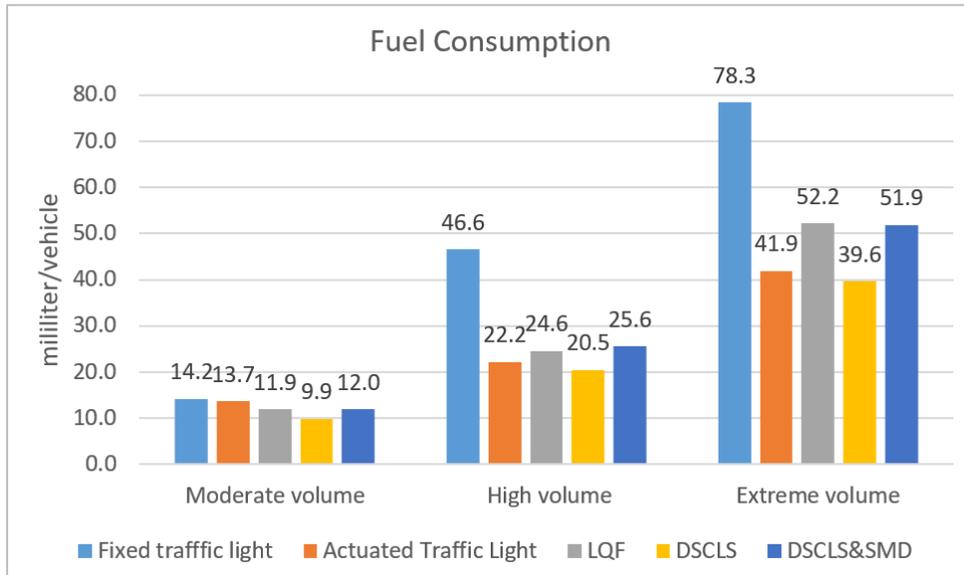


Figure 5.19 Average fuel consumption comparison, including DSCLS&SMD.

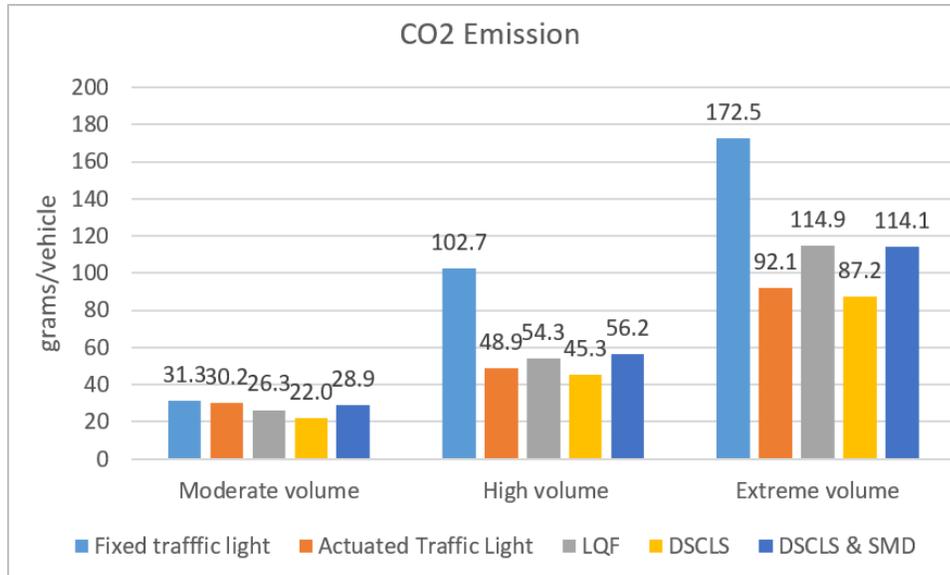


Figure 5.20 Average CO₂ emission comparison, including DSCLS&SMD.

The comparison of PET, shown in Figure 5.21, reveals that the platooning CAVs have a safer crossing maneuver at the intersections. The PET value is improved by 1%, 6%, and 10% in moderate, high, and extreme volume regimes. The reason is that instead of single vehicles being involved in a crossing maneuver, platoons of vehicles are involved, reducing the number of conflicts and the chance of side collisions.

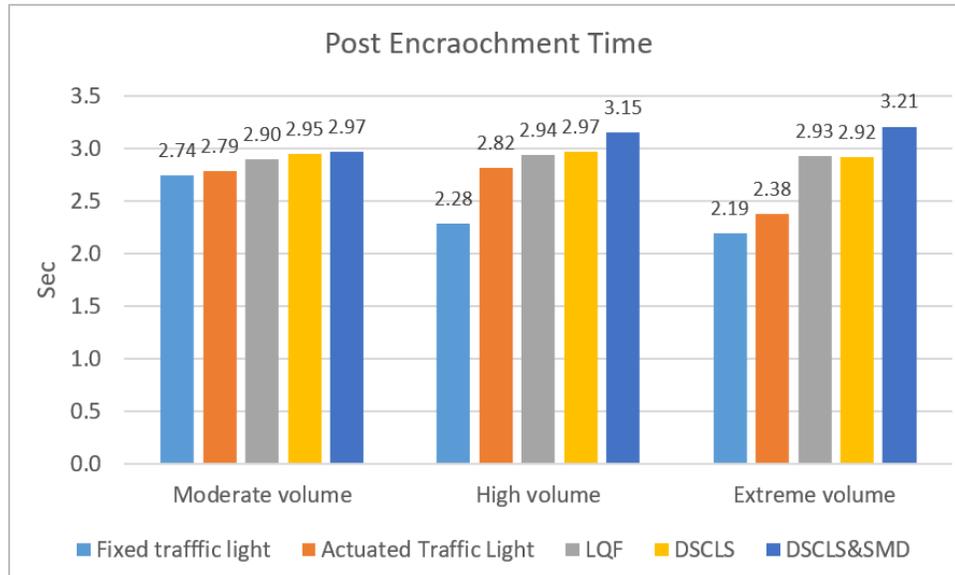


Figure 5.21 Average PET comparison, including DSCLS&SMD.

The TTC measure comparison appears in Figure 5.22. The platooning CAVs have a better performance than the DSCLS in the TTC measure, with around a 45% increment in all volume regimes. The reason is that in DSCLS&SMD, all vehicles in the network are controlled by the SDM model, catching up with the leading vehicle’s acceleration and deceleration smoothly.

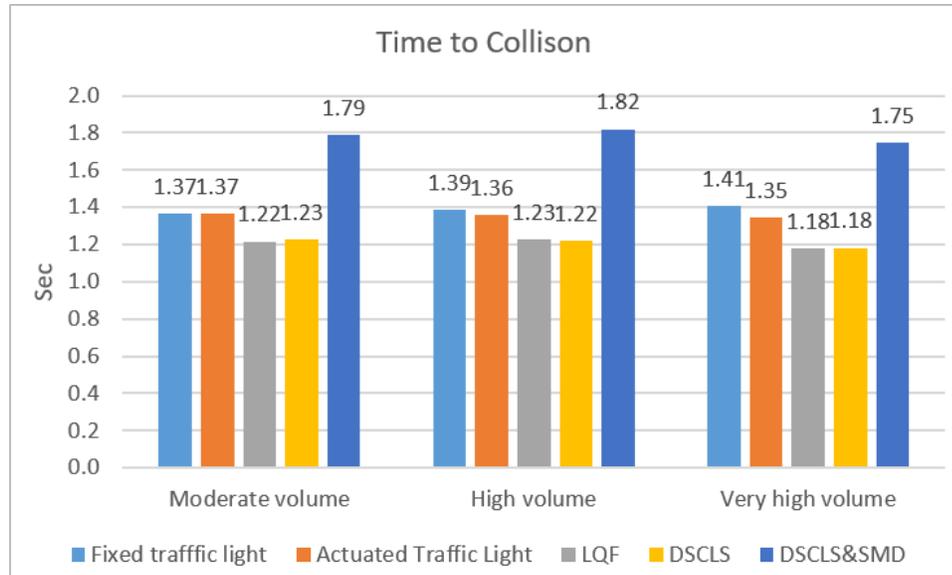


Figure 5.22 Average TTC comparison, including DSCLS&SMD.

5.4.3 Statistical Analysis of the Simulation Results

The t-test results between DSCLS&SMD and four other control systems are shown in Table 5.3. Differences in fuel consumption are not statistically significant between DSCLS&SMD and LQF in the high volume regime, and differences in CO₂ emissions are not statistically significant in the extreme volume regime for the two models.

Table 5.3 P-values for T-test Results - DSCLS&SMD

Fixed Traffic Light							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS & SMD	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
Actuated Traffic Light							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS & SMD	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
LQF							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS & SMD	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	>0.05	<0.05	<0.05	<0.05	
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	>0.05	<0.05	<0.05	<0.05	
DSCLS							
Moderate volume							
	Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC	
DSCLS & SMD	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	High Volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	
	Extreme volume						
		Delay	Travel Time	Fuel Consumption	CO2 Emission	PET	TTC
<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	

5.5 Chapter Summary

Followed by developing a CAV-based intersection control system and a platooning logic for CAVs in Chapters 3 and 4, the developed models were expanded as the following in Chapter 5.

The DSCLS was applied to a more realistic intersection consisting of three lanes and turning movements, which required reconfiguring the model and rerunning the training course. Comparing the proposed model's performance with other control systems revealed that the DSCLS noticeably improves traffic measures, including delay and travel time, specifically in moderate and high-volume regimes. However, a maximum throughput increment of 10% was gained in the extreme volume regime. Due to the limited acceleration and deceleration rates assumed for the vehicles in the DSCLS, the model does not show a noticeable improvement in environmental measures compared to the conventional control systems. However, the DSCLS outperforms the other pixel reservation-based models with limited acceleration and deceleration rates (LQF). Both pixel reservation-based control systems gain noticeable improvements in PET measure, referring to the side collision risk. The TTC measure is decreased compared to the conventional intersection control systems, however due to CAVs' shorter processing time compared to HDVs, this may not be considered as a rear-end collision risk increment.

The developed automated intersection control system and platooning logic were combined, and a new intersection control system was developed to control platooning CAVs at the intersection, called DSCLS&SMD. Development of this model required adjustments on both models and the rerun of the training course. Simulation results revealed that the DSCLS&SMD noticeably improves traffic measures in the extreme

volume condition with a 31% delay reduction and 18% maximum throughput increment. However, minor improvements were observed in moderate and high volume regimes in traffic measures since fewer platoons had been formed due to lower occupancy. Since the SMD model pushes all vehicles to strictly follow the leading vehicle, which is being controlled by DSCLS with limited acceleration and deceleration ranges, poor performance was observed in environmental measures. Involving platoons of vehicles instead of single vehicles in crossing maneuvers has improved the PET measure by 6% and 10% in high and extreme volume regimes. The TTC measure was improved by around 45% in all volume regimes due to all vehicles being controlled by either SMD or DSCLS model.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

This chapter summarizes findings, research contributions, and recommendations for further research.

6.1 Conclusions

This dissertation addressed two critical aspects of the CAVs research world, including platooning behavior and coordination at the intersections. An advanced machine learning method called Deep Reinforcement Learning and a famous conflict point detection logic in CAVs' literature, known as the pixel reservation system, were deployed to detect conflicting maneuvers and minimize overall delay at the intersections. Deep Reinforcement Learning is proven to have a stunning performance in decision-making in complex stochastic circumstances, and the pixel reservation logic guarantees collision-free coordination. A platooning system for CAVs was developed based on a classical physics logic known as the Spring-Mass-Damper model, which adjusts the acceleration and deceleration of vehicles to maintain the desired headway. Finally, a platooning-CAV-based intersection control system was developed by combining the two models and applying the required adjustments.

Two conventional intersection control systems, including fixed traffic light and actuated traffic light, and a CAV-based control system called LQF were selected as counterparts of the proposed intersection control system. The simulation results for a multilane intersection revealed that the proposed model noticeably outperforms other control systems in traffic measures, with 58%, 19%, and 13% delay reduction in

moderate, high, and extreme volume regimes. The travel time was improved accordingly, and a 10% maximum throughput increment was also recorded compared to the second-best intersection control system. Compared to conventional control systems, the proposed model does not improve environmental measures, including CO₂ emission and fuel consumption. Meanwhile, it noticeably outperforms the LQF model by 20% to 30% improvement in environmental measures in different volume regimes. Regarding the safety parameters, both pixel reservation-based models improve PET measures by around 20% and the TTC measure was slightly increased, however due to CAVs' shorter processing time, this may not be considered as a rear-end accident risk increment. The application of the control system to a corridor of four intersections has produced compatible results with the single intersection experiment.

Application of the developed SMD-based platooning logic to a single lane highway revealed that the model improves the maximum throughput by 63% and 27% in the full and 50% MPR of CAVs with a response time of 0.5 *sec*. A maximum throughput increment of 23% and 17% was achieved in the full and 50% MPR of CAVs with a response time of 1.0 *sec*. Assessment of CAV's travel time at a merging section proved the effectiveness of the inter platoon spacing policy with 50% and 67% travel time reduction on the mainlane in 60% and 100% CAVs' market penetration rates. Noticeable travel time reduction on the merging lane was observed in 80% and higher CAV's MPR and high merging demand. The SMD platooning logic produced 75% less positive average spacing error than the VISSIM 2020 platooning module in a harsh declaration scenario, which guarantees stable throughput in unstable driving circumstances. No negative error was recorded, which guarantees a safe deceleration maneuver.

As the last step of this study, an automated intersection control system for platooning CAVs was developed, called DSLCS&SMD. This model outperformed the DSCLS in traffic measures with a 9% delay reduction and 18% maximum throughput increment. The safety measures were also remarkably improved since all vehicles in the network were controlled by the SMD model. However, limited acceleration and deceleration rates assumed in DSCLS resulted in poor performance in environmental measures.

6.2 Research Contributions

The main contributions of this research are as follows:

- The intersection control system proposed in this study assumes individual vehicles as independent agents, coordinating with other agents only if required, which is most likely how highly automated vehicles will act.
- The developed intersection system can improve its performance or adjust to the new circumstances during its life cycle.
- Based on the existing literature and experiments, deep reinforcement learning is an outstanding decision-making logic in decision making in stochastic environments. And to the best of the author's knowledge, this research is the first try in deploying deep reinforcement learning to manage CAVs' conflicting maneuvers in a roadway network setting.
- The platooning logic developed in this study reflects real-world platooning circumstances and requirements such as maximum communication range, inter platoon spacing, vehicle acceleration and deceleration capability, vehicle mass, and platoon evolution process.
- Two crucial features of CAVs, including platooning behavior and coordination at the intersections, were aligned together. Platooning CAVs were exposed to interrupted traffic flow in an arterial network setting, including several intersections, and the automated intersection control system managed the platoons' conflicting maneuvers.

- All developed models were coded in a commercial traffic microsimulation software, and their performance was compared with conventional and CAV-based intersection control systems through real-world scenarios.

6.3 Future Research

As mentioned in the conclusion section, due to limited acceleration and deceleration ranges assumed for the DSCLS model in this study, no improvement is gained in environmental measures compared to the conventional intersection control systems. To better assess the proposed models' impact on environmental measures, considering different acceleration and deceleration rates for the DSCLS is recommended in future studies.

Passengers' throughput optimization-based intersection control systems have been a point of interest for a long time, and noticeable literature exists on related systems such as Transit Signal Priority (TSP), which give priority to transit vehicles. However, in an ultimate level of automation and communication, the control system can optimize individual passenger delay, regardless of vehicle type. The DSCLS model is flexible enough to reflect passenger delay in the optimization process, which could be considered in subsequent studies. The DSCLS model developed in this dissertation only controls the leading vehicle at each approach, which has resulted in rear-end collision risk increment compared to the conventional control systems in some scenarios. However, this issue was resolved in the DSCLS&SMD model; controlling all vehicles in the network based on the pixel reservation approach if the DSCLS is deployed by itself is recommended in future studies.

Other recommendations for future studies are as follows:

- 1- Adding a lane changing logic to the SMD model or expanding the model to control CAV's lateral interactions,
- 2- Improving the performance of DSCLS by deploying other DRL booster methods such as Double DQN and Prioritized Experience Replay in the training course,
- 3- Developing a more generic DSCLS to be applicable to different intersections without adjustment or re-running the training course.

REFERENCES

- [1] D. Schrank, B. Eisele and T. Lomax, "Urban Mobility Report," Texas A&M Transportation Institute, College Station, TX, 2019.
- [2] E. Namazi, J. Li and C. Lu, "Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic Literature Review," IEEE Access, vol. 7, pp. 91946-91965, 2019.
- [3] "Global Status Report on Road Safety," World Health Organization, Geneva, Switzerland, 2018.
- [4] A. Reza, B. Gaurav, R. Rajkumar and P. Mudalige, "STIP: Spatio-temporal Intersection Protocols for Autonomous Vehicles," ACM/IEEE International Conference on Cyber-Physical Systems, pp. 1-12, 2014.
- [5] C. Nowakowski and S. Shladover, "Cooperative Adaptive Cruise Control: Testing Drivers' Choices of Following Distances," California PATH Program, Institute of Transportation Studies, University of California at Berkeley, Berkeley, 2011.
- [6] "<https://www.its.dot.gov/pilots/>," U.S. Department of Transportation . [Online]. [Accessed April 2021].
- [7] "Autonomous Vehicle Market Report," GreyB, May 2021. [Online]. Available: <https://www.greyb.com/autonomous-vehicle-companies/>. [Accessed July 2021].
- [8] L. Graesser and W. L. Keng, Foundations of Deep Reinforcement Learning, New York, NY: Pearson Education Inc, 2020.
- [9] A. Geron, Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow, Sebastopol, California: Orielly, 2019.
- [10] Z. Zhong, Assessing the Effectiveness of Managed Lane Strategies for the Rapid Deployment of Cooperative Adaptive Cruise Control Technology, Doctoral Dissertation: New Jersey Institute of Technology, 2018.
- [11] K. Dresner and P. Stone, "Multiagent Traffic Management: A Reservation-based Intersection Control Mechanism," in Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, USA, 2004.

- [12] K. Dresner and P. Stone, "A Multiagent Approach to Autonomous Intersection Management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591-656, 2008.
- [13] K. Zhang, A. Fortelle, D. Zhang and X. Wu, "Analysis and Modeled Design of One State-driven Autonomous Passing-through Algorithm for Driverless Vehicles at Intersections," in *16th IEEE International Conference on Computational Science and Engineering*, Sydney, NSW, Australia, 2015.
- [14] D. Carlino, S. Boyles and P. Stone, "Auction-based Autonomous Intersection Management," in *IEEE Conference on Intelligent Transportation Systems*, The Hague, Netherlands, 2013.
- [15] Vasirani.Matteo and S. Ossowski, "A Computational Market for Distributed Control of Urban Road Traffic Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 313-321, 2011.
- [16] G. Chen and D.-K. Kang, "Win-fit: Efficient Intersection Management via Dynamic Vehicle Batching and Scheduling," in *International Conference on Connected Vehicles and Expo (ICCVE)*, henzhen, China, 2015.
- [17] F. Yan, M. Dridi and A. El Moudni, "Autonomous Vehicle Sequencing Algorithm at Isolated Intersections," in *IEEE Conference on Intelligent Transportation Systems*, St. Louis, MO, USA, 2009.
- [18] J. Wu, A. Abbas-Turki and A. El Moudni, "Discrete Methods for Urban Intersection Traffic Controlling," in *IEEE Vehicular Technology Conference*, Barcelona, Spain, 2009.
- [19] J. Wu, A. Abbas-Turki and A. El Moudni, "Intersection Traffic Control by a Novel Scheduling Model," in *IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, Chicago, IL, USA, 2009.
- [20] J. Wu, A.-T. Abdeljalil and A. El Moundi, "Cooperative Driving: An Ant Colony System for Autonomous Intersection Management," *Applied Intelligence*, vol. 37, pp. 207-222, 2012.
- [21] A. Fayazi, A. Vahidi and A. Luckow, "Optimal Scheduling of Autonomous Vehicle Arrivals at Intelligent Intersections via MILP," in *the American Control Conference*, Seattle, WA, USA, 2017.
- [22] J. Lee, B. Park and i. Yun, "Cumulative Travel-time Responsive Real-time Intersection Control Algorithm in the Connected Vehicle Environment," *Journal of Transportation Engineering*, vol. 139, no. 10, pp. 1020-1029, 2013.

- [23] J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicle Environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81-90, 2012.
- [24] G. Slobodan and J. Lee, *Development and Evaluation of Cooperative Intersection Management Algorithm under Connected Vehicles Environment*, Doctoral Dissertaion, New Jersey Institute of Technology, 2018.
- [25] R. Krajewski, P. Themann and L. Eckstein, "Decoupled Cooperative Trajectory Optimization for Connected Highly Automated Vehicles at Urban Intersections," in *IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016.
- [26] A. Mirheli, L. Hajibabai and A. Hajibabai, "Development of a Signal-head-free Intersection Control Logic in a Fully Connected and Autonomous Vehicle Environment," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 412-425, 2018.
- [27] H. Joo, S. H. Ahmed and Y. Lim, "Traffic Signal Control for Smart Cities Using Reinforcement Learning," *Computer Communications*, vol. 154, pp. 324-330, 2020.
- [28] I. Lamouik, A. Yahyaouy and M. A. Saberi, "Smart Multi-agent Traffic Coordinator for Autonomous Vehicles at Intersections," in *International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Fez, Morocco, 2017.
- [29] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang and D. Wu, "Multi-Agent Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 5243-8256, 2020.
- [30] S. Touhbi, M. Ait Babram and T. Nguyen-Huu, "Adaptive Traffic Signal Control: Exploring Reward Definition For Reinforcement Learning," in *The 8th International Conference on Ambient Systems, Networks and Technologies*, Caen, France, 2017.
- [31] Y. Wu, H. Chen and F. Zhu, "DCL-AIM: Decentralized Coordination Learning of Autonomous Intersection Management for Connected and Automated Vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 246-260, 2019.

- [32] X. Liang, X. Du, G. Wang and Z. Han, "A Deep Reinforcement Learning Network for Traffic Light Cycle Control," *IEEE Transactions on Vehicular Technology*, vol. 68, p. 1243, 2019.
- [33] Y. Gong, M. Abdel-Aty, J. Yuan and Q. Cai, "Multi-Objective Reinforcement Learning Approach for Improving Safety at Intersections with Adaptive Traffic Signal Control," *Accident Analysis and Prevention*, vol. 144, p. 105655, 2020.
- [34] A. Shaout and M. Jarrah, "Cruise Control Technology Review," *Computers Electrical Engineering*, vol. 23, pp. 259-271, 1997.
- [35] L. Xiao and F. Gao, "A Comprehensive Review of the Development of Adaptive Cruise Control Systems," *Vehicle System Dynamics*, vol. 48, no. 10, pp. 1167-1192, 2010.
- [36] R. M. James, C. Melson, J. Hu and B. Joe, "Characterizing the Impact of Production Adaptive Cruise Control on Traffic flow: an Investigation," *Transportmetrica B*, vol. 7, no. 1, pp. 992-1012, 2019.
- [37] Z. Wang, G. Wu and M. J. Barth, "A review on Cooperative Adaptive Cruise Control (CACC) Systems: Architectures, Controls, and Applications," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, 2018.
- [38] B. Van Arem, C. Driel and R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-flow Characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429-436, 2006.
- [39] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe and M. Nakamura, "Cooperative Adaptive Cruise Control in Real Traffic Situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296-305, 2014.
- [40] K. C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen and M. Chowdhury, "A Review of Communication, Driver Characteristics, and Controls Aspects of Cooperative Adaptive Cruise Control (CACC)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 491-509, 2016.
- [41] S. E. Shladover, D. Su and X.-Y. Lu, "Impacts of Cooperative Adaptive Cruise Control on Freeway Traffic Flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2324, no. 1, pp. 63-70, 2012.

- [42] F. Bu, H.-S. Tan and J. Huang, "Design and Field Testing of Cooperative Adaptive Cruise Control," in American Control Conference, Baltimore, MD, USA, 2010.
- [43] B. v. Arem, C. J. G. v. Driel and R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 4, pp. 429-436, 2006.
- [44] J. Vanderwerf, N. Kourjanskaia, S. Shladover, M. Miller and H. Krishnan, "Modeling Effects of Driver Control Assistance Systems on Traffic," Transportation Research Record, no. 1748, pp. 167-174, 2001.
- [45] A. Hasanzadezonuzy, S. Arefizadeh, A. Talebpour, S. Shakkottai and S. Darbha, "Collaborative Platooning of Automated Vehicles Using Variable Time-Gaps," in Annual American Control Conference (AAC), Milwaukee, WI, USA, 2018.
- [46] K. Hassan K, Nonlinear systems, Prentice Hall, 1996.
- [47] Z. Zhong, J. Lee and L. Zhao, "Multiobjective Optimization Framework for Cooperative Adaptive Cruise Control Vehicles in the Automated Vehicle Platooning Environment," Transportation Research Record: Journal of Transportation Research Board, vol. 2625, pp. 32-42, 2017.
- [48] J. Eyre, D. Yanakiev and I. Kanellakopoulos, "A Simplified Framework for String Stability Analysis of Automated Vehicles," International Journal of Vehicle Mechanics and Mobility, no. 5, pp. 375-405, 1998.
- [49] J.-M. Contet, F. Gechter, P. Gruer and A. Koukam, "Multiagent System Model for Vehicle Platooning with Merge and Split Capabilities," in 3rd International Conference on Autonomous Robots and Agents (ICARA), 2006.
- [50] C. R. Munigety, "A Spring-mass-damper System Dynamics-based Driver-vehicle Integrated Model for Representing Heterogeneous Traffic," International Journal of Modern Physics B, vol. 32, no. 11, 2018.
- [51] S. Bang and S. Ahn, "Platooning Strategy for Connected and Autonomous Vehicles: Transition from Light Traffic," Transportation Research Record: Journal of the Transportation Research Board, vol. 2623, no. 1, pp. 73-81, 2017.
- [52] R. W. Craig, "Flocks, Herds, and Schools: A Distributed Behavioral Model Craig," in ACM SIGGRAPH, Anaheim, California, USA, 1987.

- [53] B. Soohyuk and A. Soyoung, "Control of Connected and Autonomous Vehicles with Cut-in Movement Using Spring Mass Damper System," *Transportation Research Record*, vol. 2672, no. 20, pp. 133-143, 2018.
- [54] R. S. Sutton and A. G. Barto, *Reinforcement Learning, An Introduction*, Cambridge, MA: The MIT Press, 2018.
- [55] V. Mnih, K. Kavukcuoglu, D. Silver and A. Graves, "Playing Atari with Deep Reinforcement Learning," in *NIPS Deep Learning Workshop*, 2013.
- [56] L. Busoniu, R. Babuska and B. D. Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics*, part C, pp. 156-172, 2008.
- [57] J. R. Kok and N. Vlassis, "Sparse Cooperative Q-learning," in *Twenty-First International Conference on Machine Learning*, Banff, Canada, 2004.
- [58] R. Wunderlich, I. Elhanany and T. Urbanik, "A Stable Longest Queue First Signal Scheduling Algorithm for an Isolated Intersection," in *IEEE International Conference on Vehicular Electronics and Safety, ICVES*, Beijing, China, 2006.
- [59] H. Rakha, K. Ahn and A. Trani, "Development of VT-Micro Model for Estimating Hot Stabilized Light Duty Vehicle and Truck Emissions," *Transportation Research Part D: Transport and Environment*, vol. 9, no. 1, pp. 49-74, 2004.
- [60] D. Gettman and L. Head, "Surrogate Safety Measures from Traffic Simulation Models," *Transportation Research Record*, no. 1840, pp. 104-115, 2003.
- [61] S. Bang, *Traffic Flow Control for Connected and Automated Vehicles Using Spring Mass Damper System*, Doctoral Dissertation, University of Wisconsin-Madison, 2017.
- [62] "<https://www.caranddriver.com/ford/fusion>," *Car and Drivers*, [Online]. Available: <https://www.caranddriver.com/ford/fusion>. [Accessed March 2021].
- [63] G. Newell, "A Simplified Car-following Theory: A Lower Order Model," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195-205, 2002.
- [64] S. Ahn, M. Cassidy and J. Laval, "Verification of a Simplified Car-following Theory," *Transportation Research Part B: Methodological*, vol. 38, no. 5, pp. 431-440, 2004.

- [65] VISSIM Manual, Karlsruhe, Germany: PTV Group, 2021.
- [66] "Safety Pilot Model Deployment," U.S. Department of Transportation, Washington DC, 2015.
- [67] H. Rakha, K. Ahn and A. Trani, "Development of VT-Micro model for estimating hot stabilized light duty vehicle and truck emissions," Transportation Research Part D: Transport and Environment, vol. 9, no. 1, pp. 49-74, 2004.
- [68] J. R. Taylor, Classical Mechanics, Sausalito, CA, USA: University Science Books, 2005.
- [69] "Top 30 Self Driving Technology and Car Companies," GreyB, 25 05 2021. [Online]. Available: <https://www.greyb.com/autonomous-vehicle-companies/>. [Accessed 20 07 2021].
- [70] K. M. Gurumurthy, C. R. Munigety, T. Mathew and A. Gowri, "An Integrated Pedestrian Crossing Behavioural Model Using Spring-Mass-Damper Dynamics," in Transportation Research Board, Washington DC, 2017.
- [71] D. Schrank and B. Eisele, "2015 URBAN MOBILITY SCORECARD," Texas A&M Transportation Institute and INRIX, Texas, USA, 2015.
- [72] S. Touhbi, M. A. Babram, T. Nguyen Huu and N. Marilleau, "Traffic Signal Control: Exploring Reward Definition For Reinforcement Learning," Procedia Computer Science, vol. 109, pp. 513-520, 2017.
- [73] E. Ahmadi, C. Caprani, S. Živanović and A. Heidarpour, "Experimental Validation of Moving Spring-mass-damper Model for Human-structure Interaction in the Presence of Vertical Vibration," Structures, vol. 29, pp. 1274-1285, 2021.
- [74] Y. Cai, L. Chen, W. Yu, J. Zhou, F. Wan, M. Suh and D. Chow, "A Piecewise Mass-spring-damper Model of the Human Breast," Journal of Biomechanics, vol. 67, pp. 137-143, 2018.
- [75] H. Xu, S. Wang and J. Hu, "Mass-spring-damper Modeling and Stability Analysis of Type-4 Wind Turbines Connected into Asymmetrical Weak AC Grid," in 7th International Conference on Power and Energy Systems Engineering, Fukuoka, Japan, 2020.
- [76] G. Gürsel, A. Frijns, E. Homburg and A. v. Steenhoven, "A Mass-spring-damper Model of a Pulsating Heat Pipe with a Non-uniform and Asymmetric Filling," Applied Thermal Engineering, vol. 91, pp. 80-90, 2015.

- [77] B. Hirzinger, C. Adam and P. Salcher, "Dynamic Response of a Non-classically Damped Beam with General Boundary Conditions Subjected to a Moving Mass-spring-damper System," *International Journal of Mechanical Sciences*, vol. 185, 2020.
- [78] P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang and D. O. Wu, "Multi-Agent Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, 2020.