ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

APPLICATIONS OF MULTI AGENT SYSTEMS IN TRANSPORTATION

Ph.D. THESIS

İlhan TUNÇ

Department of Mechatronics Engineering

Mechatronics Engineering Programmme

MARCH 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

APPLICATIONS OF MULTI AGENT SYSTEMS IN TRANSPORTATION

Ph.D. THESIS

İlhan TUNÇ
(518152012)

Department of Mechatronics Engineering

Mechatronics Engineering Programmme

Thesis Advisor: Prof. Dr. Mehmet Turan SÖYLEMEZ

MARCH 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

ULAŞIMDA ÇOKLU AJAN SİSTEMLERİNİN UYGULAMALARI

DOKTORA TEZİ

İlhan TUNÇ
(518152012)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mehmet Turan SÖYLEMEZ

MART 2023

İlhan TUNÇ, a Ph.D. student of ITU Graduate School student ID 518152012, successfully defended the dissertation entitled “APPLICATIONS OF MULTI AGENT SYSTEMS IN TRANSPORTATION”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Mehmet Turan SÖYLEMEZ** Istanbul Technical University	....
Jury Members :	**Assoc. Prof. Dr. Gülay ÖKE GÜNEL** Istanbul Technical University	....
	Prof. Dr. Tankut ACARMAN Galatasaray University	....
	Assoc. Prof. Dr. Volkan SEZER Istanbul Technical University	....
	Prof. Dr. Şeref Naci ENGİN Yıldız Technical University	....

Date of Submission : 27 December 2022
Date of Defense : 21 March 2023

To my spouse and daughters,

FOREWORD

I would like to express my special thanks to my supervisor, Prof. Dr. Mehmet Turan SÖYLEMEZ, for his contributions, interest, and guidance. I would also like to thank committee members Assoc. Prof. Dr. Gülay ÖKE GÜNEL and Prof. Dr. Tankut ACARMAN for their valuable comments and guidance.

I am grateful to my wife, Büşra, and my lovely daughters, Zeynep and Ahsen, as well as my mother, Şaziye, my father, Naci, and my sisters for their persistent and tireless support. Additionally, I am thankful to Abdurrahman YILMAZ, Atakan Yasin YEŞİLYURT, Nuri KORHAN, and Osman ERVAN for sharing their experiences and supporting me throughout my doctoral process. Finally, I am lucky to have worked with Ali Emre EDEM, Ali Osman KÖROĞLU, Halil AYDIN, Koralp SERT, Özge ELMAS, Serra Nur AKMEŞE, and Zeynep GÜMÜŞ, who have each provided me with valuable support and guidance.

March 2023 İlhan TUNÇ
(Research Assistant)

TABLE OF CONTENTS

Page

FOREWORD ix
TABLE OF CONTENTS xi
ABBREVIATIONS xiii
SYMBOLS xv
LIST OF TABLES xvii
LIST OF FIGURES xix
SUMMARY xxiii
ÖZET xxvii
1. INTRODUCTION 1
1.1 Purpose of Thesis 6
1.1.1 Unique aspect 7
1.1.2 Impact 8
1.2 Literature Review 9
1.3 Hypothesis 18
2. TRAFFIC LIGHT CONTROL SYSTEMS 19
2.1 Fuzzy Logic and PI Control for Traffic Lights 20
2.1.1 System overview 20
2.1.2 Traffic light design with fuzzy logic controller 22
2.1.3 Traffic light design with PI controller 24
2.1.4 Simulation results 25
2.2 State Feedback Control for Traffic Light Systems 28
2.2.1 System overview 28
2.2.2 Traffic light design with fuzzy logic controller 29
2.2.3 State feedback control for intelligent traffic light systems 32
2.2.3.1 State-space equations 32
2.2.3.2 Ackermann's formula 33
2.2.4 Simulation results 34
2.3 Traffic Light Control System Simulation for Different Strategies with Fuzzy Logic Controller 38
2.3.1 System overview 38
2.3.2 Traffic light design with fuzzy logic controller 41
2.3.3 Simulation results 44
2.4 Fuzzy Logic Control Strategies for Traffic Signal Timing Control with State Inputs 48
2.4.1 System overview 48
2.4.2 Traffic light controller design 49
2.4.2.1 Fuzzy logic controller with queue length input 50
2.4.2.2 Fuzzy Logic traffic signal timing with states input 51

2.4.3 Simulation results 54
2.5 Traffic Light Control for Multi Intersection Model in Istanbul/Altunizade 58
2.5.1 System overview 58
2.5.2 Traffic light control system 60
2.5.2.1 Actuated traffic light control 61
2.5.2.2 Fuzzy logic traffic light control 63
2.5.3 Simulation results 67
3. FUZZY LOGIC AND DEEP Q LEARNING BASED CONTROL FOR TRAFFIC LIGHTS 71
3.1 Reinforcement Learning 72
3.2 System Architecture 74
3.2.1 State 74
3.2.2 Action 79
3.2.3 Reward 79
3.3 Deep Q Learning with Fuzzy Logic 80
3.3.1 Deep neural network 81
3.3.2 Deep Q learning with FLSI 83
3.4 Simulation results 87
4. AGENT-BASED ROUTE PLANNING WITH DEEP Q LEARNING 103
4.1 Introduction 103
4.2 System Overview 104
4.2.1 Deep Q learning model 105
4.2.1.1 States 105
Position data 106
Traffic density of neighboring roads data 106
Traffic lights of neighboring intersections data 107
4.2.1.2 Actions 108
4.2.1.3 Reward 110
4.2.2 Simulation results 110
5. A PLATOON ORDERING ALGORITHM ON RESERVATION-BASED INTERSECTION MANAGEMENT SYSTEMS 115
5.1 Reservation-Based Intersection Management 116
5.1.1 Intersection agent 119
5.1.2 Vehicle agent 120
5.1.2.1 Vehicle agent cost 121
5.1.2.2 Vehicle agent reservation request 121
5.1.2.3 Vehicle agent action 122
5.1.2.4 Intelligent connected adaptive cruise control 122
5.2 Platoon Ordering Algorithm 123
5.3 Simulation results 124
5.3.1 Case 1 126
5.3.2 Case 2 127
6. CONCLUSIONS AND RECOMMENDATIONS 131
REFERENCES 137
CURRICULUM VITAE 149

ABBREVIATIONS

ACC	: Adaptive Cruise Control
AIM	: Autonomous Intersection Management
ANN	: Artificial Neural Network
AZ	: Action Zone
CACC	: Cooperative Adaptive Cruise Control
CVIC	: Cooperative Vehicle Intersection Control
CZ	: Communication Zone
DQN	: Deep Q- Network
DSS	: Decision Support System
ES	: East to South
EWN	: East to West or North
FLC	: Fuzzy Logic Control
FLQL	: Fuzzy Logic Queue Length
FLSI	: Fuzzy Logic State Input
GP	: Green Phase
IA	: Intersection Agent
IIM	: Intelligent Intersection Management
LQF	: Longest-Queue-First
LSD	: Lane Space Discretization
MADARP	: Multi Agent Dial-a-ride Problem
MAS	: Multi Agent System
MPC	: Model Predictive Control
MSIM	: Multi-Agent Intersection Management
NE	: North to East
NSW	: North to South or West
NVD	: Number of Vehicles Difference
PI	: Proportional Integral
RA	: Robot Agents
RL	: Reinforcement Learning
RP	: Red Phase
SLAM	: Simultaneous Localization and Mapping
SNE	: South to North or East
SUMO	: Simulation of Urban Mobility
SW	: South to West
TLC	: Traffic Light Control
TNV	: Total Number of Vehicles
VA	: Vehicle Agent
WES	: West to East or South
WN	: West to North

SYMBOLS

CO_2	: Carbon dioxide
CO	: Carbon monoxide
d_i	: The number of vehicles leaving the intersection
h	: Hour
kg	: Kilogram
km	: Kilometer
q_i	: The number of vehicles entering the intersection
m	: Meter
s	: Second
$\boldsymbol{\mu}$	: Service Rate
t	: Time
$\boldsymbol{W}$	: Waiting time
$\boldsymbol{\lambda}$	: Arrival Rate
$\boldsymbol{\rho}$	: Utilization Factor

LIST OF TABLES

Page

Table 2.1 : Order of phases **21**
Table 2.2 : Rule table for fuzzy logic **25**
Table 2.3 : Simulation results **26**
Table 2.4 : Order of phases **29**
Table 2.5 : Rule table for fuzzy logic **29**
Table 2.6 : Strategy 1 pass permission topology **40**
Table 2.7 : Strategy 2 pass permission topology **41**
Table 2.8 : Rule table for fuzzy logic **44**
Table 2.9 : Rule table for fuzzy logic **44**
Table 2.10 : Simulation results **46**
Table 2.11 : Pass permission table **49**
Table 2.12 : Rule table for FLQL and FLSI **52**
Table 2.13 : Phase, cycle, minimum and maximum green light times **61**
Table 2.14 : Rule table for fuzzy logic **67**
Table 3.1 : Rule table for FLSI **86**
Table 3.2 : Abbreviations of rule table **88**
Table 3.3 : Simulation results **89**
Table 3.4 : Stability analysis **97**
Table 4.1 : Sample state vector **107**
Table 5.1 : Vehicle routes and the regions they occupy. **119**
Table 5.2 : Vehicle routes and the regions they occupy. **120**
Table 5.3 : Routes of the vehicles in different scenarios **126**
Table 5.4 : Intersection time and average speed data in simulations with ordering and without ordering for case 1. .... **127**
Table 5.5 : Routes of the vehicles in different scenarios **128**
Table 5.6 : Intersection time and average speed data in simulations with ordering and without ordering for case 2. .... **129**

LIST OF FIGURES

		Page
Figure 2.1	: Traffic intersection model	**21**
Figure 2.2	: Membership function of the total number of vehicles from input value for FLC.	**22**
Figure 2.3	: Membership function of the number of vehicle differences from input value for FLC.	**23**
Figure 2.4	: Traffic light green phase time member function.	**24**
Figure 2.5	: Average speed values according to changes in the number of vehicle density.	**27**
Figure 2.6	: Total emission values according to changes in the number of vehicle density.	**27**
Figure 2.7	: Traffic intersection simulation model.	**29**
Figure 2.8	: Membership function of the total number of vehicles.	**30**
Figure 2.9	: Membership function of the number of vehicle differences.	**31**
Figure 2.10	: Traffic light green phase time member function.	**31**
Figure 2.11	: Simulation results based on cumulative CO_2 emission.	**35**
Figure 2.12	: Simulation results based on average speed.	**36**
Figure 2.13	: Simulation results based on total waiting time.	**37**
Figure 2.14	: Traffic intersection model general architecture.	**39**
Figure 2.15	: Traffic intersection simulation model.	**39**
Figure 2.16	: Strategy 1 pass permission topology	**40**
Figure 2.17	: Strategy 2 pass permission topology.	**41**
Figure 2.18	: Membership function of the total number of vehicles from input values for Strategy 1 and Strategy 2.	**42**
Figure 2.19	: Membership function of the number of vehicles on the busiest road from input values for Strategy 1.	**42**
Figure 2.20	: Membership function of the number of vehicles difference from input values for strategy 2.	**43**
Figure 2.21	: Traffic light green time member function.	**45**
Figure 2.22	: Simulation results based on average speed.	**46**
Figure 2.23	: Simulation results based on cumulative CO_2 emission.	**47**
Figure 2.24	: TLC simulation environment representation scheme.	**49**
Figure 2.25	: Membership function of queue length in GP input for FLQL	**50**
Figure 2.26	: Membership function of queue length in RP input for FLQL.	**51**
Figure 2.27	: Traffic light green time member function for FLC.	**52**
Figure 2.28	: Separation of a leg of a traffic junction into cells.	**53**
Figure 2.29	: Membership function of state input in GP input for FLSI.	**53**
Figure 2.30	: Membership function of state input in RP input for FLSI.	**54**
Figure 2.31	: Simulation results based on total queue length (2000 vehicles).	**55**
Figure 2.32	: Simulation results based on waiting time (2000 vehicles).	**56**

Figure 2.33 : Simulation results based on total queue length (3000 vehicles). **56**
Figure 2.34 : Simulation results based on waiting time (3000 vehicles)................. **57**
Figure 2.35 : Traffic density map and display of traffic lights. **59**
Figure 2.36 : Detailed presentation of traffic lights, intersections and phases. **60**
Figure 2.37 : Membership function of the total number of vehicles in red phase.... **63**
Figure 2.38 : Membership function of total number of vehicles in green phase..... **64**
Figure 2.39 : Membership function of green time duration. **65**
Figure 2.40 : Detailed representation of fuzzy logic traffic light control method diagram. **66**
Figure 2.41 : Simulation results based on CO_2 emission for each methods............ **68**
Figure 2.42 : Simulation results based on average speed for each method. **68**
Figure 2.43 : Simulation results based on average speed for each methods with median values.. **69**
Figure 3.1 : TLC simulation environment... **74**
Figure 3.2 : Actions for traffic light agent... **75**
Figure 3.3 : The process of obtaining the state matrix **78**
Figure 3.4 : Block diagram of DQ FLSI.. **84**
Figure 3.5 : Division of an arm at a traffic intersection into cells **84**
Figure 3.6 : Trafik stream L_i values and cell compartments at the traffic intersection. **85**
Figure 3.7 : Traffic light control process with deep Q learning with FLSI method.**87**
Figure 3.8 : Simulation results based on reward values according to fixed green light time values. **88**
Figure 3.9 : Simulation results based on total queue length values according to fixed green light time values. **90**
Figure 3.10 : Simulation results based on cumulative delay values according to fixed green light time values. **90**
Figure 3.11 : Simulation results based on CO_2 emission output values according to fixed green light time values. **91**
Figure 3.12 : Simulation results based on reward values according to different states.. **91**
Figure 3.13 : Simulation results based on total queue length values according to different states.. **92**
Figure 3.14 : Simulation results based on cumulative delay values according to different states. **92**
Figure 3.15 : Simulation results based on CO_2 emission output values according to different states. **93**
Figure 3.16 : Simulation results based on reward values for DQ FLSI and DQ FLSI eqst.. **93**
Figure 3.17 : Simulation results based on total queue length values for DQ FLSI and DQ FLSI eqst. **94**
Figure 3.18 : Simulation results based on cumulative delay values for DQ FLSI and DQ FLSI eqst. **94**
Figure 3.19 : Simulation results based on CO_2 emission output values for DQ FLSI and DQ FLSI eqst. **95**
Figure 3.20 : For arrival rate of 937 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst **97**
Figure 3.21 : For arrival rate of 1887 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst.. **98**

Figure 3.22 : For arrival rate of 2831 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst....... **98**
Figure 3.23 : For arrival rate 3762 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst **99**
Figure 3.24 : For arrival rate 4713 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst **100**
Figure 3.25 : Total queue length change according to different arrival rates for the DQ FLSI method....... **100**
Figure 3.26 : Vehicle generation distribution for testing scenario....... **101**
Figure 3.27 : Simulation results based on total queue length values for DQ FLSI and DQ FLSI eqst for test scenario....... **101**
Figure 4.1 : Traffic flow scenario....... **104**
Figure 4.2 : Sample traffic scenario....... **106**
Figure 4.3 : Acquisition of traffic density information on neighbouring roads..... **107**
Figure 4.4 : Acquisition of traffic light information at neighbouring intersections....... **108**
Figure 4.5 : Actions for a 4-way intersection. **109**
Figure 4.6 : Actions for a 3-way intersection. **109**
Figure 4.7 : The scenario of route planning. **111**
Figure 4.8 : Passenger average waiting times....... **112**
Figure 4.9 : Reward value of the agent for the number of episodes. **113**
Figure 5.1 : Intersection zones....... **117**
Figure 5.2 : Intersection regions. **118**
Figure 5.3 : Communication architecture of multi-agent system. **120**
Figure 5.4 : Vehicle agent control algorithm. **121**
Figure 5.5 : Ordering of the vehicles in a platoon **125**
Figure 5.6 : Intersection times with ordering and without ordering for case 1. .... **127**
Figure 5.7 : Average speeds with ordering and without ordering for case 1......... **128**
Figure 5.8 : Intersection times with ordering and without ordering for case 2. .... **129**
Figure 5.9 : Average speeds with ordering and without ordering for case 2......... **130**

APPLICATIONS OF MULTI AGENT SYSTEMS IN TRANSPORTATION

SUMMARY

Traffic density is a growing drawback of the crowding of cities in contemporary societies. As a consequence of financial and technological innovations, the living standards of people are improving yet this increases the number of cars and traffic density accordingly. Thus, the density of traffic is reducing the quality of life for individuals in metropolitan areas in particular. Traffic is an important factor affecting human life quality in crowded cities. The increasing population and increasing individual vehicle ownership lead to an increase in traffic density. This causes an increase in loss of time and pollution. Traffic density in big cities is an important factor that reduces the quality of human life. Due to the growing population in metropolitan areas and the inadequate infrastructure to accommodate this density, traffic problems are on the rise. As a result, passengers waste more time in traffic, and the amount of emissions, and hence air pollution, also increases. The issue of traffic congestion is a significant concern for numerous metropolitan areas across the globe, as it causes delays, increases commuting time, and contributes to air pollution. Controlling the flow of traffic is problematic in terms of many complexities and uncertainties. Despite this situation, this problem needs to be solved as it reduces productivity and living standards. Modern traffic control methods offer a more effective solution, unlike traditional methods. As traffic congestion continues to increase rapidly in the world, the need to research and apply more effective methods of traffic control than the traditional method is increasing. Solving traffic congestion is one of the most important and complex problems, as it causes chaos in metropolitans, especially during heavy traffic hours. Traditional methods that continue to be used have proven to be inadequate, and as a result, the developing technology has affected all areas as well as the solutions to the traffic control problem. With the emergence of Intelligent Transportation Systems (ITS), utilizing artificial intelligence and communication technologies, a more effective and efficient solution to traffic congestion is possible.

Transportation techniques are improving day by day with the pace of growing technology. Intelligent Transportation Systems (ITS) provide advanced services such as high-tech traffic controllers and various transportation modes, reducing the burden on drivers and thus enabling them to meet the need for complex decision-making while on the road. Intelligent transportation solutions have enabled an unprecedented level of data collection within the industry, leading to significant advancements in transportation system management and operation. With the increasing demand and rate of data collection, ITS has also been advancing day by day and increasing the speed of progress of smart transportation systems.

ITS can be described as systems consisting of technologies such as electronics, data processing and wireless networks that provide security and efficiency in the transportation network. ITS provides communication and information exchange between each transport unit. These units can be centres that provide information to pedestrians, vehicles, infrastructure, transportation and other peripherals such as traffic lights and other communication and control units.

The application of MAS (Multi-Agent Systems) techniques, as a new development in information technology, can help to increase interest in traffic and promote energy-efficient transportation. ITS-based multi-agent technology is an important approach to solving complex traffic problems. The complexity of the elements of the traffic makes them convenient for multi-agent structures. ITS-based multi-agent technology provides us with safer controllers and makes us feel more comfortable in our daily lives. It increases the quality of our lives by decreasing the amount of time spent in traffic and by lowering the amount of emission gases released by our vehicles.

The structurally dispersed nature of components in heterogeneous environments causes application difficulties, such as interoperability between agents forming a demand for a unified software platform as an underlying infrastructure. Therefore, it is preferable to use centralized solutions for relatively simple problems such as the one considered in this paper. For both transport decision-makers and drivers, ITS have a great potential for efficient and intelligent traffic management, threat identification, driving comfort and safety. ITS can also provide a flexible approach for the effective management of complex networked transportation systems letting traffic management decision-makers to control signal changes, regulate route flows, and broadcast real-time traffic information. In addition to providing route scheduling, weather forecasting, and emergency services for drivers, ITS (Intelligent Transportation Systems) can also help to reduce driving loads and improve safety.

The implementation of ITS (Intelligent Transportation Systems) can generate positive outcomes across a range of areas, spanning from environmental and national security issues to emergency management and transportation. ITS applications can reduce time spent on the road. Short travel times provide economic savings for both individual and commercial vehicles and usually mean less environmental pollution. Intelligent Intersection Management (IIM) technology has started to develop in traffic intersections as part of Traffic Light Control (TLC) systems.

Intersections are some of the busiest parts of roads. Therefore, the control of traffic lights plays an important role in decreasing the density. In this thesis, particular attention is given to the control of intersections in order to find solutions to decrease traffic density leading to an increased quality of life in big cities. Intelligent traffic control methods, the use of which is increasing with the development of new methods, are used especially in traffic intersections with high traffic density in order to provide efficient solutions.

Control of a single intersection with traffic lights is considered first in the thesis. Various methods, including Fuzzy Logic Control (FLC), Proportional Integral (PI) control and State Space Model Control techniques, have been proposed and compared for a better traffic light controller architecture so as to increase the traffic flow and reduce the overall waiting time of the cars and the emissions released by them. It is demonstrated that the proposed architectures give better results compared to the traditional fixed-time traffic light control method. Different types of traffic intersections are considered in the study. Initially, a simple single-lane traffic intersection with no left or right turn is taken into consideration. Later on, intersections on which three-lane (or four-lane) roads meet with vehicles turning left and right are considered. Finally, a realistic case study, in which the Altunizade Region of Istanbul, is examined to demonstrate the efficiency of some of the proposed methods. The results of simulations indicate that the FLC, in which the positions of the vehicles are used as the state variables, gives superior results in comparison to the other classical methods.

In order to increase the efficiency of the FLC further, a built-in learning algorithm is proposed to be used in addition to the FLC. A deep Q-learning method is employed for this purpose as a part of the agent-based traffic light controller. Hence, the resulting intelligent traffic light controller is named DQ-FLSI. In this method, a state matrix which divides the arms of the traffic intersection into cells is used. The traffic light durations are determined using fuzzy logic, and traffic light actions are determined by the help of deep Q-learning. A stability analysis is also carried out for this newly proposed method.

Another important traffic problem is route planning. This is particularly important in large cities with complex traffic networks. In order to address this problem, an agent-based traffic route planning method has also been proposed as part of this thesis with the motivation of vehicles choosing the fastest route. In this method, route planning is made by deciding at traffic intersection points. Vehicle agents make decisions when they reach traffic intersections. In this way, dynamic route planning becomes possible for the vehicles.

Another solution for the traffic intersection problem is multi-agent reservation-based traffic intersection control. With this method, all vehicles (called agents) can pass the intersection without the need for a traffic light thanks to a traffic intersection agent. A platoon method, which can work in harmony with reservation-based traffic intersection management, is proposed as an improvement in this part of the study. The proposed method aims to reduce the slowdowns that occur when approaching the traffic intersection by properly lining up the vehicles approaching the traffic intersection. It is shown by simulations that the proposed platoon method reduces energy consumption and gas emissions while increasing the average speed of the vehicles, especially as the density of the traffic increases.

Work environments for all studied traffic problems are designed and simulated using the SUMO program. Simulation of Urban MObility (SUMO) is an open-source simulation package that works on networks imported from maps, provides various workspaces at micro levels, also allows pedestrian simulation, and has a sufficient set of tools that makes it more reachable.

ULAŞIMDA ÇOKLU AJAN SİSTEMLERİNİN UYGULAMALARI

ÖZET

Trafik yoğunluğu, günümüzde şehirlerin kalabalıklaşmasıyla önemli bir problem haline gelmektedir. Finansal ve teknolojik yeniliklerin bir sonucu olarak insanların yaşam standartları gelişmekte ancak bu durum araç sayısını ve buna bağlı olarak trafik yoğunluğunu artırmaktadır. Dolayısıyla trafik yoğunluğu özellikle metropollerde bireylerin yaşam kalitesini düşürmektedir. Kalabalık şehirlerde trafik, insanın yaşam kalitesini etkileyen önemli bir faktördür. Artan nüfus ve artan bireysel araç kullanımı, trafik yoğunluğunun artmasına neden olmaktadır. Bu da yolcular için trafikte kaybedilen zamanın ve hava kirliliğinin artmasına neden olur. Büyük şehirlerdeki trafik yoğunluğu insan yaşam kalitesini düşüren önemli bir faktördür. Büyükşehirlerin artan nüfusu ve altyapılarının bu yoğunluğu kaldıramaması ile birlikte trafik yoğunluğu da giderek artmaktadır. Sonuç olarak, yolcular daha fazla trafikte zaman kaybetmekte ve emisyon miktarı dolayısıyla hava kirliliği de artmaktadır. Trafik sorunu, dünyadaki birçok büyükşehir için önemli bir endişe kaynağıdır. Trafik akışını kontrol etmek, birçok karmaşıklık ve belirsizlik nedeniyle zordur. Bu duruma rağmen üretkenliği ve yaşam standartlarını düşürdüğü için bu sorunun çözülmesi gerekmektedir. Modern trafik kontrol yöntemleri, geleneksel yöntemlerden farklı olarak daha etkili bir çözüm sunmaktadır. Dünyada trafik sıkışıklığı hızla artmaya devam ederken, geleneksel yöntemden daha etkili trafik kontrol yöntemlerinin araştırılması ve uygulanması ihtiyacı artmaktadır. Özellikle trafiğin yoğun olduğu saatlerde büyükşehirlerde kaosa neden olan trafik sıkışıklığının çözülmesi en önemli ve karmaşık sorunlardan biridir. Halen kullanılmaya devam eden geleneksel yöntemlerin yetersiz kaldığı ortaya çıkmış ve bunun sonucunda gelişen teknoloji, trafik kontrol sorununa getirilen çözümlerin yanı sıra tüm alanları etkilemiştir. Yapay zeka ve iletişim teknolojilerinin gelişmesiyle birlikte Akıllı Ulaşım Sistemleri (AUS) ortaya çıkmıştır.

Gelişen teknolojinin hızı ile ulaşım teknikleri her geçen gün gelişmektedir. Bu nedenle, yüksek teknolojili trafik kontrolörleri ve farklı ulaşım yöntemleri gibi yenilikçi hizmetler sunarak sürüş konusunda düşünme veya karar verme sorumluluğunu azaltmak için AUS ortaya çıktı. Ulaşım sistemindeki akıllı çözümler sayesinde ulaşım sistemlerinde benzeri görülmemiş veri toplanmasına yol açmıştır. Artan talep ve veri toplama hızı ile AUS her geçen gün gelişmekte ve bu sistemlerinin ilerleme hızı da buna paralel olarak artmaktadır.

AUS, ulaşım ağında güvenliği ve verimliliği sağlayan elektronik, bilgi işlem ve kablosuz ağlar gibi teknolojilerden oluşan sistemler olarak tanımlanabilir. AUS, her bir taşıma birimi arasında iletişim ve bilgi alışverişini sağlar. Bu birimler, yayalara,

araçlara, altyapıya, ulaşıma ve trafik ışıkları gibi diğer çevre birimlerine ve diğer iletişim ve kontrol birimlerine bilgi sağlayan merkezler olabilir. AUS uygulaması, çevre sorunlarından ulusal güvenlik sorunlarına, acil durum yönetiminden ulaşıma kadar pek çok alanda olumlu sonuçlar üretebilir.

Bilgi teknolojisinde yeni bir gelişme olarak Çok Etmenli Sistem (ÇES) teknikleri, trafiğe olan ilginin artması ve verimli ulaşımın daha fazla enerji tasarrufu sağlaması için yardımcı olabilir. AUS tabanlı çok etmenli teknolojisi, karmaşık trafik problemlerinin çözümünde önemli bir yaklaşımdır. Trafiğin öğelerinin karmaşıklığı, onları çok etmenli sistemler için uygun hale getirir. AUS tabanlı çok etmenli teknolojisi, bize daha güvenli kontrolörler sağlar ve günlük hayatımızda daha rahat hissetmemizi sağlayabilir. Trafikte geçirilen süreyi azaltarak ve araçlarımızın saldığı emisyon gazlarının miktarını düşürerek yaşam kalitemizi yükseltebilir.

Heterojen ortamlardaki bileşenlerin yapısal olarak dağınık yapısı, altyapı olarak birleşik bir yazılım platformu talebi oluşturan aracılar arasındaki birlikte çalışabilirlik gibi uygulama zorluklarına neden olur. Bu nedenle, nispeten basit problemler için merkezi çözümlerin kullanılması da tercih edilebilir. Hem ulaşım karar vericileri hem de sürücüler için AUS, verimli ve akıllı trafik yönetimi, tehdit belirleme, sürüş konforu ve güvenliği için büyük bir potansiyele sahiptir. AUS ayrıca, trafik yönetimi karar vericilerinin sinyal değişikliklerini kontrol etmesine, rota akışlarını düzenlemesine ve gerçek zamanlı trafik bilgilerini yayınlamasına izin vererek, karmaşık ağ bağlantılı ulaşım sistemlerinin etkili yönetimi için esnek bir yaklaşım sağlayabilir. Sürücüler için rota planlama, hava durumu tahmini, acil durum hizmetleri vb. kadar, AUS de sürüş yüklerini azaltmayı kolaylaştırabilir ve güvenliği artırabilir.

AUS uygulamaları, çevre sorunlarından ulusal güvenlik sorunlarına, acil durum yönetiminden ulaşıma kadar pek çok alanda olumlu sonuçlar verebilir. AUS uygulamaları yolda geçirilen süreyi azaltabilir. Kısa seyahat süreleri hem bireysel hem de ticari araçlar için ekonomik tasarruf sağlamakta ve genellikle daha az çevre kirliliği anlamına gelmektedir. Akıllı Kavşak Yönetimi (IIM) teknolojisi, Trafik Işık Kontrol (TLC) sistemleri kapsamında trafik kavşaklarında da yaygın olarak kullanılmaya başlanmıştır.

Karayollarında trafik sıkışıklığının en fazla olduğu kısımlardan birisi trafik kavşaklarıdır. Dolayısıyla trafik ışıklarının kontrolü yoğunluğun azaltılmasında önemli rol oynamaktadır. Bu tezde, büyükşehirlerde yaşam kalitesini azaltan trafik yoğunluğunu azaltacak çözümler bulmak için kavşakların kontrolüne özel önem verilmektedir. Yeni yöntemlerin geliştirilmesiyle kullanımı artan akıllı trafik kontrol yöntemleri, özellikle trafik yoğunluğunun yüksek olduğu trafik kavşaklarında etkili çözümler sunmak amacıyla kullanılmaktadır.

Tezde ilk olarak trafik ışığına sahip tek bir kavşağın kontrolü ele alınmıştır. Trafik akışını artırmak ve genel bekleme süresini ve araçlar tarafından salınan emisyon gazlarını azaltmak için daha iyi bir trafik ışığı denetleyici mimarisi olarak bulanık mantık kontrol (FLC), Oransal İntegral (PI) Kontrolü ve durum uzay model kontrolü teknikleri dahil olmak üzere çeşitli yöntemler önerilmiş ve karşılaştırılmıştır. Önerilen mimarilerin geleneksel sabit zamanlı trafik ışığı kontrol yöntemine göre daha iyi sonuçlar verdiği gösterilmiştir. Çalışmada farklı tipteki trafik kavşakları ele alınmıştır. Başlangıçta, sola veya sağa dönüşü olmayan basit bir tek şeritli trafik kavşağı dikkate alınır. Daha sonra üç şeritli (veya dört şeritli) yolların sağa ve sola dönen araçlara izin veren kavşaklar ele alınmıştır. Son olarak, önerilen yöntemlerin bazılarının etkinliğini göstermek için İstanbul'un Altunizade bölgesindeki trafik ışıklarının incelendiği bir çalışma yapıldı. Benzetim sonuçları gösterdi ki, giriş değeri araçların konum bilgilerinin kullanıldığı FLC yöntemi diğer klasik yöntemlere göre daha üstün sonuçlar verdi.

FLC'nin verimliliğini daha da artırmak için FLC'ye ek olarak yerleşik bir öğrenme algoritmasının kullanılması önerilmiştir. Bu amaçla, etmen tabanlı trafik ışığı kontrolörünün bir parçası olarak bir derin Q-öğrenme yöntemi kullanıldı. Bu nedenle, ortaya çıkan akıllı trafik ışığı kontrolörü DQ-FLSI olarak adlandırılır. Bu yöntemde trafik kavşağının kollarını hücrelere ayıran bir durum matrisi kullanılmaktadır. Bulanık Mantık ile trafik ışığı süreleri, derin Q-öğrenme vasıtası ile trafik ışığı yönleri belirlenir. Bu yeni önerilen yöntem için bir kararlılık analizi de yapılmıştır.

Bir diğer çalışılan önemli trafik problemi ise rota planlamasıdır. Bu, özellikle karmaşık trafik ağlarına sahip büyük şehirlerde önemlidir. Bu sorunu çözmek için, araçların en hızlı rotayı seçme motivasyonu ile bu tez kapsamında etmen tabanlı bir trafik rota planlama yöntemi de önerilmiştir. Bu yöntemde trafik kavşak noktalarında karar verilerek rota planlaması yapılır. Araç etmenleri, trafik kavşaklarına ulaştıklarında karar verirler. Bu sayede araçlar için dinamik rota planlaması mümkün hale gelmektedir.

Trafik kavşağı problemine bir diğer çözüm önerisi ise çok etmenli rezervasyon tabanlı trafik kavşak kontrolü yöntemidir. Bu yöntemle tüm araçlar (etmen adı verilen) bir trafik kavşağı etmeni sayesinde trafik ışığına ihtiyaç duymadan kavşaktan geçebilmektedir. Çalışmanın bu bölümünde bir iyileştirme olarak, rezervasyona dayalı trafik kavşak yönetimi ile uyumlu çalışabilecek bir platoon yöntemi önerilmiştir. Önerilen yöntem, trafik kavşağına yaklaşan araçları düzgün bir şekilde sıralayarak trafik kavşağı yaklaşırken oluşan yavaşlamaları azaltmayı amaçlamaktadır. Önerilen platoon yönteminin özellikle trafik yoğunluğu arttıkça araçların ortalama hızlarını artırırken enerji tüketimini ve gaz emisyonlarını azalttığı benzetim sonuçları ile gösterilmiştir.

Çalışılan tüm trafik problemleri için çalışma ortamları SUMO programı kullanılarak tasarlanmış ve benzetimleri de yine bu program ile gerçekleştirilmiştir. Simulation of Urban MObility (SUMO), haritalardan içe aktarılan ağlar üzerinde çalışan, mikro düzeyde çeşitli çalışma alanları sağlayan, yaya simülasyonuna da izin veren ve daha erişilebilir hale getiren yeterli araç setine sahip açık kaynaklı bir benzetim yazılım paketidir.

1. INTRODUCTION

An agent can be defined as a system that perceives and interacts with its environment through sensors and actions. In general, an agent is a system of detecting the environment and reacting to the purpose of the ability to make changes in the environment [1]. The choice of action of the agents at any given moment may depend on the entire set of perceptions observed but is not dependent on anything that is not perceived. There is usually more than one agent in agent-based systems. Such systems are called Multi-Agent Systems (MAS).

The concept of MAS is the designed modelling approach to represent systems that exhibit assets, intelligence, autonomy and interactions, both with each other and with the environment. MAS are the systems having different information or different interests or both, with multiple autonomous entities [2]. Besides, MAS is a collaborative intelligent system consisting of an interactive set of computing units that can solve complex problems based on minimal or reduced data processing resources. These systems consist of a set of homogeneous or heterogeneous smart software or hardware agents that can exchange information, and coordinate and negotiate activities. MAS can be used in areas like economics, technology, mathematics, computing, networking, artificial intelligence, robotics, collaborative decision support systems, data mining, and social sciences. MAS proposes a distributed control definition based on cooperative and autonomous agents to perform a task. The structure of MAS allows the processing of significant amounts of data due to the scalability of these systems. MAS can be expanded by adding new agents or new behaviours, thus they can be appropriate in the context of decentralized and heterogeneous environments where major changes may occur. One of the areas where MAS are widely used is traffic problems. Mostly known applications in intelligent transportation systems are route planning and traffic intersection problems.

Intelligent Transportation Systems (ITS) [3] based on multi-agent technologies have become an important approach to solving complex transportation problems. The structurally dispersed nature of components in heterogeneous environments causes application difficulties, such as interoperability between agents forming a demand for a unified software platform as an underlying infrastructure. For both transport decision-makers and drivers, Intelligent transportation systems (ITS) have a great potential for efficient and intelligent traffic management, threat identification, driving comfort, and safety [4]. ITS can provide a flexible approach for the effective management of complex networked transportation systems letting traffic management decision-makers to control signal changes, regulate route flows, and broadcast real-time traffic information. As much as route scheduling, weather forecasting, emergency services, etc. for drivers, ITS can also facilitate reducing driving loads and improve safety.

With the development of technology, the development and advancement of transportation technology is inevitable [5]. Intelligent Transportation Systems (ITS) have been developed to reduce people's thinking or decision-making responsibility by providing innovative services such as high technology, traffic control and different modes of transport [6]. Technological advances have enabled transport systems to collect unprecedented amounts of data. With the help of such data, the development of intelligent transportation systems is increasing rapidly.

Intelligent Transportation Systems (ITS) can be called as systems consisting of technologies such as electronic, data processing and wireless networks that provide a level of security and efficiency in the transportation network. ITS enable communication and exchange of information between each unit of transportation. These units can be centers that provide control of people, vehicles, infrastructure and transportation. As the development process of ITS systems continues, it is thought that the expectations and benefits of these systems may change over time or focus on different areas.

As the use of the Internet of Things (IoT) becomes widespread, a large number of complex systems, networks, or social infrastructures can be used in existing systems and generate massive amounts of data by connecting multiple devices [7]. As Artificial Intelligence (AI) and IoT systems evolve, the functionality of AI-based Intelligent Transportation Systems is becoming worth considering. Intelligent Transportation Systems (ITS) help to make transportation more environmentally friendly and safe

[8]. ITS applications can also provide constructive solutions in many areas from environmental problems, national security problems, and emergency management to transportation. In an active traffic scenario, traffic intersections are the most critical components that will slow down or speed up traffic flow [9].

Traffic control and optimization are challenging topics for researchers and engineers. Traffic control is a compelling issue, as transportation systems have low predictability and are often dispersed and complex [10]. According to the Texas Transport Institute's (TTI) urban mobility report, the delay per passenger is approximately 34 hours. As a result, the approximate cost of traffic problems for the US alone is around $350 billion in 2017 [11].

Nowadays, road traffic is widely used at critical points of vital operation such as logistics and transportation. Especially increasing population in big cities causes an increase in traffic problems, which is an important problem for daily life [12]. As a result of this high demand, it is possible to encounter some negative consequences such as high waiting times, wasted time and high CO_2 emissions [13]. Increasing the number of vehicles causes loss of time in traffic and fuel wastage in cities that do not have sufficient infrastructure. Environmental problems such as air pollution and noise pollution occur, as well as health problems and traffic accidents. Traffic control not only reduces environmental problems but also benefits human psychology by reducing the time people spend in traffic [14]. Real-time (adaptive) Traffic Light Control (TLC) techniques use real-time measurements to determine appropriate traffic light times. The control update time will vary from one second to a single traffic light cycle, depending on the TLC strategy used.

Many studies have been carried out on the management of traffic intersections. Some of them are reservation-based [15] studies, and some are related to the control of traffic lights [16]. Such Real-time (adaptive) Traffic Light Control (TLC) techniques use real-time measurements to determine appropriate traffic light periods [17]. Adaptive traffic light systems based on waiting time give advantageous results compared to common fixed-time traffic light systems today [18]. In these systems, the control update period could vary from one second to a single traffic light cycle, depending on the TLC strategy used.

Considering the non-linear nature of traffic lights, the Fuzzy Logic Control (FLC) method is a useful method for traffic light control [19]. However, in these studies, no learning algorithms were used to determine the traffic phase. In [19], queue length and waiting times are selected as fuzzy logic input values, and the output of the fuzzy logic output value is given as the green light time. However, a method for determining the phase sequence is not used.

In traffic intersection management, with the Reinforced Learning (RL) technique, reward functions can be defined to reduce values such as waiting time or emission from traffic problems, and the defined reward function can be optimized with possible actions according to current situations.

The important advantage of RL is that it can learn the optimal action by trying methods according to the information it receives from the environment [20]. In the reinforcement learning method, the agent is expected to be able to choose the sequence of actions that can reach the maximum reward value in various situations. It typically has three components: environmental states, the agent's action space, and the reward for each action [21]. The key to agent-based traffic light control is the proper selection of these three components in the traffic intersection system so that they can be calculated. Inappropriate choices can cause an extra computational load or inaccurate results for traffic light control.

The reinforcement learning method has been applied in many applications [22–25]. Many studies have been conducted with the reinforcement learning method in order to control traffic lights dynamically. In earlier studies, the states were defined according to the sum of vehicles with near-zero speed [26]. However, the sum of vehicles with near-zero speed cannot accurately represent the real-world traffic situation [27]. With the proliferation of vehicle networks and cameras, it has been possible for more information, such as vehicle speed and standby time to be collected and transmitted over the network [28]. Using more information is crucial to solving the problem, but the number of states increases, and with it, the complexity of the conventional reinforcement learning system increases exponentially. Deep neural networks are used to overcome the problem that becomes more complex due to the increasing number of states [29]. Some recent studies proposed that deep enhancement learning to be applied in the issue of traffic light control [30]. Previous works generally divide traffic signals into fixed times, and this makes an important limitation.

The route planning problem is a very important issue, especially in sectors such as logistics and transportation. Inefficient time spent on the road disrupts transportation and the logistics supply chain. Therefore, it will cause undesirable costs in terms of economic and environmental pollution.

Many criteria will affect the performance of the current agent for the route planning problem. The agent must learn the balance between the shortest path and the fastest path to reach the destination. It would be wrong to consider the factors affecting performance in some traffic scenarios as choosing only the shortest and quickest route. Another factor that affects the time it takes to reach the destination is the traffic lights that will restrict the movement in the network of roads it will move. Understanding the phase sequences of traffic lights with the status information it will receive from the environment, will be an essential criterion for its chosen action on the way to the target.

In the near future, the spread of unmanned vehicles in the flowing traffic and communication with each other or with other infrastructures seems normal thanks to scientific and technological developments. Studies into congestion control were conducted in order to prevent future collisions and traffic obstructions by ensuring that vehicles function in harmony with one another and with the environment [31]. Many different methodologies have been proposed to control unmanned vehicles at crossing points. One of them includes Cooperative Adaptive Cruise Control (CACC) which is modified for vehicle-based scenarios [32]. In addition to such micro-organizational approaches, reservation-based methods, auction-based methods or platooning methods have also been presented as macro-regulatory approaches. Another proposed method is Autonomous Intersection Management (AIM), a traffic intersection management method based on reserving a specific section of the intersection for a specific vehicle to avoid collisions [33]. One of the recommended intersection management methods is the platooning method, in which a group or car team moves in close order under fully automatic, longitudinal and lateral control. During cooperative driving, autonomous vehicles mimic migrating birds or a group of dolphins [34]. It can be clearly observed that the AIM method significantly increases the traffic flow compared to traditional intersection management systems. The scheduled auction method including offering is used to choose an optimal path and based on some AIM methods [35].

In this thesis, various problems are examined using theoretical knowledge of MAS. The most significant of these problems is traffic problems. The simulation of these methods is performed using Simulation of Urban MObility (SUMO), and the results are compared. A simulation environment is designed using SUMO. The SUMO program is an open-source, highly portable, microscopic road traffic simulation package designed to handle large road networks [36].

1.1 Purpose of Thesis

ITS has an extensive structure that performs information, communication, and control of traffic items. A more secure structure can be with ITS structures. Especially by reducing the load on drivers, accidents due to fatigue and carelessness can be avoided. In addition, with the optimum suggestions offered by ITS, traffic jams can be reduced, and as a result, environmental pollution is reduced.

ITS which is based on MAS appears as a solution approach for complex transportation problems. The structural disintegration of components in heterogeneous environments leads to application difficulties, such as interoperability among other factors that require a unified software platform as basic infrastructure. Therefore, it is preferred to use centralized solutions for relatively simple problems such as those discussed in this publication. For transport decision-makers and drivers, ITS, smart traffic management, has a high potential for detecting possible threats and ensuring drivability and safety. ITS can also provide a flexible approach to the efficient management of complex networked transport systems, enabling traffic management decision-makers to control signal changes, regulate route flows, and broadcast real-time traffic information. ITS can reduce drivers' driving difficulties and improve safety by helping drivers in different ways such as route planning, weather forecasting and organizing emergency services. ITS implementation can give positive results and improvements in many areas such as from environmental issues to national security issues, as well as from emergency management to transportation. ITS can decrease the consumption of travelling time for drivers and pedestrians. Shorter travel times result in economic savings for both individual and commercial vehicles and often mean less environmental pollution.

Through the gains brought by MAS, ITS can offer solutions to numerous traffic problems, including intelligent traffic light control, intelligent intersection management, and intelligent route planning using reservation-based traffic intersection control methods.

1.1.1 Unique aspect

In this thesis, control of a single intersection with traffic lights is considered first. Various methods, including Fuzzy Logic Control (FLC), Proportional Integral (PI) Control and State Space Model Control techniques, have been proposed and compared in terms of the overall waiting time of the cars and the emissions released by them. It is shown that the proposed architectures give better results compared to the traditional fixed-time traffic light control method. The results are tested for different types of traffic intersections including a simple single-lane intersection as well as for junctions on which three-lane (or four-lane) roads meet, and a realistic case study with several junctions.

As the main contribution of this thesis, a deep Q-learning algorithm is proposed to be used in addition to the fuzzy logic controller in order to increase efficiency. This newly introduced method, which is named DQ-FLSI, employs fuzzy logic for determining the duration of traffic lights and deep Q-learning for determining the order of the light phases. In this method, a state matrix which divides the arms of the traffic intersection into cells is used. A varying cell size in the determination of the state matrix is used in DQ-FLSI. A comparison between using constant (equal) cell sizes and varying cell sizes is also provided to demonstrate the efficiency of this adaptation. Theoretical stability analysis is also developed for the proposed method, the robustness of which is demonstrated by simulations.

As other main contributions of this thesis, a couple of reservation-based methods are examined for intersection management. In particular, a platoon algorithm is proposed to increase the efficiency of reservation-based traffic intersection management. Excessive simulation results demonstrated the efficiency of the proposed approach. In addition, an agent-based route planning method for vehicles has also been developed.

A taxi agent with deep Q-learning algorithm, which helps with dynamic route planning, is used. For the taxi agent to learn optimum route planning, a state vector including traffic light information, density information of neighbouring roads, location information of neighbouring intersections, and location information of the agent and destination is proposed to be used.

1.1.2 Impact

Traffic congestion is one of the leading causes of loss of productivity and reduced living standards in urban areas. Recent developments in artificial intelligence show that in the near future, vehicle navigation by autonomous agents is possible. With the development of technology and the increase in its application, it is seen that MAS can offer solutions to many traffic problems.

With the developing technology, carbon emission, sociological stress, loss of time and accidents can be reduced by manipulating traffic with a satisfactory modelling and control algorithm using artificial intelligence and IoT (Internet of Things) to solve these problems.

Recent developments in artificial intelligence show that in the near future, vehicle navigation by autonomous agents is possible. The efficiency of transport systems is a priority for modern society. Technological developments have made it possible for transport systems to collect large volumes of data on an unprecedented scale. Several researchers have extended the application of Multi-Agent Systems (MAS) to a wide range of different areas, including shared services for various purposes.

It is demonstrated in this thesis that MAS can provide efficient solutions to many problems caused by traffic. With the help of agent-based traffic light control methods proposed in this thesis, energy-efficient and environmental friendly solutions are provided. The proposed FLC-supported agent-based DQ FLSI method, in particular, is committed to both choosing the appropriate traffic light sequence and the duration of the light. With the DQ FLSI, which is the recommended traffic light control method in the thesis, unnecessary waiting at traffic lights is reduced and harmful emissions are reduced.

Reservation-based multi-agent traffic intersection management systems such as the platoon method proposed promise a further increase in efficiency of traffic intersections. With agent-based route planning, target points can be made accessible with less energy and time. It can be said that the efficiency of the route planning method has increased, especially with the use of the dynamic route planning proposed.

1.2 Literature Review

Multi-Agent Systems (MAS) have been widely applied in recent years [37–40]. Control and coordination of MAS is an important and challenging problem besides many application areas are available. Some of them are; mobile robotics [41], vehicle formation [42], flocking [43], consensus [44], unmanned aerial vehicles [45] and Traffic Simulation [46]. The most common applications of MAS are cooperative control [47–50]. The aim of cooperative control is that multiple autonomous agents work together efficiently to achieve collective group behaviour through local interaction. The most discussed topics in agent systems are consensus and formation control [51]. This study [52] discusses cluster consensus problem for generic linear heterogeneous MAS. The main purpose of this study is to demonstrate how agents are confronted with the effects of in-group couplings and couplings between clusters, and to reach clustering consensus. Actuator monitoring of non-linear MAS, and cooperative monitoring control with actuator hysteresis on diazors are discussed [53]. Each agent is modelled with a higher-order non-linear system in the form of a solid feedback with generalized Prandtl-Ishlinskii hysteresis input and unknown time-varying virtual control coefficients. In another study, The consensus issue for transition topologies and time delays and second-order MAS are discussed [54]. Switching topologies and time delays in communication are declared by Markoc chains. The problem of tracking is also a subject of many issues in MAS, This study investigates the cooperative monitoring problem for high order nonlinear MAS under a directed communication topology [55]. Discrete-time double-integrated consensus problem is addressed for MAS with directed switching proximity topologies and input constraints [56]. Model Predictive Control (MPC) approach was applied to the problem of entry constraints. In a similar study, [57] the Consensus problem with single and double integral coefficients was discussed by using the Model Predictive Control approach. The problem of Flocking in MAS has been processed by using the model predictive control method [58]. A model predictive flocking control scheme for second-order MAS with access restrictions is proposed. The cooperative regulation problem of linear Simultaneous Localization and Mapping switching between communication topologies is discussed [59]. An event-triggered control scheme has been proposed to solve the problem of cooperative regulation only through intermittent communication. The communication topology does not always have to be connected under the common assumption. With the proposed trigger mechanism, each agent transmits the information only to its neighbours at their trigger times or at switching times. The effectiveness of the proposed control scheme is illustrated by an example. And in this study, a gradient-based algorithm is focused on using an event-triggered algorithm [60]. A new gradient-based optimization consensus algorithm has been proposed to solve the optimization consensus problem and an event-triggered control strategy based on sample data was used. In this study [61], finite time consensus and monitoring problems for nonlinear MAS with directed topology are discussed. The consensus problem of MAS is studied under communication constraints [62]. Especially funnel control is proposed as a new control method to achieve consensus. Funnel Control is a high-gain adaptive control method that can guarantee monitoring with a predetermined degree of accuracy. The formation-containment problem of general linear homogeneous and heterogeneous (MAS) has been discussed [63]. In this problem, each output of the followers of reference changes in time, that is, the leaders of the output of multiple leaders to reach an agreement on the centroid, and thus aims to keep a time-changing offset. The controllability problem in MAS is also studied [64]. It focuses on group controllability problems of continuous-time MAS and provides a general definition of group controllability, and thus, group controllability criteria are generated. Another MAS applications are about mobile robots. As a matter of fact, the problem of mapping and localization in the problems of mobile robot applications has been applied with MAS [65]. To overcome this problem, Robot Agents (RA) fulfil the task of Simultaneous Localization and Mapping (SLAM) to find the mediator in the environment while simultaneously creating the geometric or topological map. MAS are also widely used in transportation [46, 66–69]. Amount this, the problem of intersection is also studied with MAS.

Traffic lights are signal devices commonly used at traffic intersections to control traffic flows around the world. A well-designed traffic light controller can increase traffic flow and reduce waiting for both vehicles and pedestrians. A traffic light controller with an inefficient controller can cause traffic congestion and therefore increase the waiting times at traffic intersections. Therefore, many traffic light control methods have been proposed to solve this problem.

The efficiency of urban traffic control systems is undeniably affected by intersections. Studies have shown that intersections have a significant effect on traffic accidents and traffic delays in urban areas because they are nodes of traffic flow, have too many stop-and-go and conflict zones, and human behaviour cannot be predicted at intersections [31, 70]. For this reason, intelligent traffic control methods are widely researched. One of the most researched smart traffic control methods in traffic intersection control is traffic light control [71, 72].

Because it is easy to use among the traffic light control methods, the most common is fixed-time traffic light control. In this method, traffic light control is performed by predetermining different green light duration for certain days and hours by using observation and statistical data. This method may give a good result with proper data processing, but with the result of any change in traffic, this control method can be quite inefficient. However, it is still highly preferred due to its cost-effectiveness. Another commonly used method is dynamic traffic light control. This method requires detectors such as sensors or cameras at traffic intersections to detect the number of vehicles [73]. Using the information from the detectors, the traffic light controller can adjust the signal phase and timing. In addition, Adaptive traffic light control is an effective solution method proposed in recent years. Adaptive control solutions try to adapt the traffic signal timing according to the road information at one or more traffic intersections. As expected, adaptive traffic light control is a more effective method than fixed-time traffic light control. However, the use of adaptive traffic light control systems is very low due to the higher investment cost. However, the adaptive traffic light control method can easily meet the investment cost by saving energy, especially in areas with heavy traffic. In their study [74] used a genetic algorithm to contemplate pedestrian crossing in traffic light control. The pedestrian metric was used in the fitness function to assess the efficacy of candidate chromosomes.

In order to do traffic control studies, the processing power was low, and the simulation environment was limited. So fuzzy logic [75] and linear programming [76] were first used to solve the problem.

With the increasing interest in the use of artificial intelligence, deep learning has been successful for many problems. Many transportation problems form an important application area for deep learning, which is a method used in many traffic control applications, including route planning. Supervised learning, unsupervised learning, and reinforcement learning are all types of deep learning [77]. Given the difficulties in modelling the variability of pedestrian and vehicle behaviour due to its unpredictable and variable nature, researchers have recently applied machine learning to traffic light control and demonstrated proven performance [78]. El-Tantawy et al. summarized methods for controlling traffic light timing with reinforcement learning used from 1997 to 2010 [26]. However, the use of reinforcement learning methods was limited at that time. Therefore to estimate the value of Q, table Q learning and a linear function is commonly utilized. As a result of technical limitations in reinforcement learning, a small-sized state area is used. The number of vehicles waiting [79] and traffic flow statistics [80] can be given as examples of commonly used ones. Since the second half of the 1990s, the use of learning algorithms for the control of traffic lights has increased. The agent or agents optimize traffic using an RL (Reinforcement Learning) algorithm. Reinforcement Learning, also known as Q learning, is one of the successful approaches to learning algorithms used in traffic light control applications. Many studies have adopted it due to the benefits of making decisions without the need for a model, and it is suitable for online use. SARSA algorithm is used for RL-based traffic light control, and it is one of the first effective approaches in the literature [81].In SARSA, traffic light control was carried out at the traffic intersection with a 4x4 grid connection by excluding the yellow light phase. In another study [82], the phase cycle has been changed for the first time by using simple binary action. Queue length was used as the input that corresponds to the current state of the system at a given point in time, and the total waiting time between two actions was used as a reward. Araghi et al. used a Q-learning approach to calculate signal timing times based on traffic data for traffic intersections [83]. Each intersection, on the other hand, only calculates using local data and attempts to maximize local performance.

In [79], 24 different probabilities that emerged by comparing the row lengths at the traffic intersection legs were used as the state. Traffic light control, based on queue length and green light duration, is proposed [84]. The authors used the clustering amount of cars and a linear function to calculate the Q value and used only the queue length for the state. When a large amount of useful related information is ignored in the constrained states, it appears incapable of acting optimally in traffic light control.

deep reinforcement learning is formed by the use of reinforcement learning, which is used to estimate the Q value, and deep learning together.

In [22], RL-based traffic light control is simulated, giving priority to high-density roads. For multiple intersections, the traffic light control is done using a fixed time, taking into account all phase configurations, where a mathematical model is used to carry out simulations.

In these studies, Deep Reinforcement Learning (DRL) has been used to control wireless communication [85]. Nevertheless, the timing of traffic signals throughout the duration of a cycle is not specified in any of the prior studies [86].

In this study [4], SOA (Simple Object Access) based multi agent intelligent transportation system model is presented. The Model consists of four main sections: infrastructure, services, element agents and coordination agents. The agents in this model are divided into different levels and groups such as organization agent, regional control agent, road intersection agent, road segment control agent, and vehicle to achieve different functions and targets.

In another Traffic intersection problem [87], it offers a highly agented architecture for the artificial transportation system. In this architecture, the Petri network is used as a basic model for representing agents. At an intersection, agents are divided into two groups: one is for traffic signals, and the other one is for vehicle intersections. It is integrated to represent the intersection behaviour. In addition, these agents can be used as modularity to scale the urban network more. To coordinate different intersection agents, game theory is used to design the coordination strategy between agents. In another study [86] involving the application of MAS in traffic, learning control policies for traffic lights were investigated. For a scalable approach to control coordinated traffic lights, it is proposed to combine the popular Deep Q- Network learning (DQN) algorithm with a coordination algorithm. However, the DQN algorithm can oscillate. Further research is needed to reinforce the situations in which DQN is not stable and to find approaches that make it more reliable. In this study, [80] using the learning algorithm, a MAS and a new use of the Reinforcement Learning (RL) framework are introduced to obtain an effective traffic signal control policy. It aims to minimize the possibility of crossing the average delay, congestion and intersection. Five intersecting traffic networks have been studied where each intersection is managed by an autonomous intelligent agent. The new methodology proposed here uses the Q-Learning algorithm with a forward neural network for the value function approach. LQF (Longest-Queue-First) algorithm is used. And in this study [67], Cooperative Vehicle Intersection Control (CVIC) system is formed. And in this study. The CVIC algorithm is designed to control the manoeuvres of vehicles so that the vehicles can safely cross the intersection without colliding with other vehicles. An additional algorithm has been designed to deal with system failures at the intersection. However, this study should consider expanding to include multiple intersections along the corridor or network and simulations based on simulation should be made. A reservation-based system has been proposed, especially at traffic intersections, under the assumption that cars are controlled by agents [33]. An exact measurement has been determined to assess the quality of traffic control at an intersection. There are restrictions that vehicles cannot turn and vehicles cannot change their speed at the intersection. In this study [27], a traffic micro simulator in SUMO is applied to modern deep reinforcement learning methods to generate a real-adaptive traffic signal control agent. A new state space with information density and separately coded traffic state are recommended. Discrete traffic situation coding is used as an input to a deep convolutional neural network, trained by Q-learning with experience repetition. In this study [88], a MAS based intersection management algorithm has been developed considering fully autonomous vehicles. The intersection is divided into three regions; communication, deceleration and acceleration zones.

Reservation-based traffic intersection control, which will also be explained in this study and which enables the communication between VAs and IAs, and regulates the intersection transition periods of autonomous vehicles without traffic lights, is also one of the most frequently studied methods in intelligent traffic intersection control.

At crowded intersections, autonomous vehicles connecting with reservation-based intersection managers can reduce delays and make significantly better use of limited road capacity [35]. According to Stone and Dresner, it is suggested the proposed reservation-based method is more efficient than conventional methods [33]. In reservation-based traffic intersection control, vehicles that will enter the intersection inform the intersection agent the area they want to occupy while passing through the intersection and the arrival time. In line with the transmitted information, the reservation information generated by the intersection agent will be transmitted to the vehicles and the vehicles will be able to pass without stopping by adjusting their speed in the most effective way. In this way, the flow will be ensured without waiting at intersections and queues will be avoided [89]. In order to apply the multi-agent method at traffic intersections, studies have been carried out with realistic flow models and Shared Autonomous Vehicle (SAV) approach to show an approach for research in the future to use realistic flow models to obtain more accurate estimates of SAV solutions [90]. In another study, Vasirani and Ossowski indicate that varying policies can be evaluated empirically to regulate an intersection controlling with reservation-based method [91].

In recent years, the subject of platooning, which reduces the cost in terms of time at intersections by arranging the order of the vehicles approaching the intersection according to the direction to turn, has been frequently researched in terms of the advantages it offers. In this direction, studies have been carried out on the efficiency, formation, dispersion and routing of the platoons and it is seen that ordering vehicles according to their turning directions can be an effective method [92–95]. In addition, studies have been conducted to further improve the reservation-based method by ordering the VAs according to the turning directions thanks to the platooning method at intersections managed with the reservation-based method [96]. Bashiri and Fleming proposed a platoon-based approach to deal with cooperative intersection management problems and they also developed a new approach which guarantees the safety of platoons in conflict zone [97]. Jin et al, proposed a platoon-based multi-agent intersection management system which can reduce fuel consumption and carbon dioxide emissions by almost 23% and average travel time by up to 30% when compared to the current traffic signal control system [98]. Thus, by using both intersection traffic regulation methods together, it is seen that improvements can be made for both the platooning method and the reservation-based method, and it can be presented as a valid alternative for the control of autonomous vehicles at intersections.

In this study [46], drivers and intersections are considered as autonomous agents in a multiple system. In this multi agent system, intersections use a new reservation-based approach built around a detailed communication protocol. This article also attributes to two aspects of the mechanism. The first one allows the system to control human-driven vehicles in addition to autonomous vehicles. And the second one gives priority to emergency vehicles without imposing significant costs on civilian vehicles. However, in this study, a more detailed study of the security features of the system- how it responds to various errors and whether the effects of these errors will be mitigated or not- is studied. Another area to be improved is the intersection manager agent. A manager who can switch between a variety of policies and learn from booking dates, which is the policy that best fits certain policy requirements, can significantly improve performance. Furthermore, a traffic light model that can react not only to traffic conditions but also to the presence of individual vehicles will be able to better utilize the capabilities of autonomous vehicles without adversely affecting human drivers. The driver himself may be able to take advantage of some Machine Learning techniques, perhaps to learn to make more accurate reservations and thus to cancel less often.

Route planning applications are also among the application areas of multi-agent systems. In this study, MAS are used as a support system for route planning [99]. A Decision Support System (DSS) has been proposed for co-modal passenger transport based on MAS architecture to respond to multi-criteria user demands. The DSS has been developed to respond to multi-path planning demands in common mode, such as vehicle preference and conflicting criteria, such as minimizing costs, time and gas emissions. The DSS architecture is based on a naturally distributed MAS framework, which allows the route planning problem to be split into more than one simple task. A genetic algorithm is used to obtain optimal user-vehicle-route combinations according to user preferences. MAS has also been used in the problems of logistics in transportation. For example, [100] offers an ontology-based multi-tool automotive parts transportation system. The system is used by Dijkstra's algorithm and the

Ontology Concept to determine a transport route to find the shortest route. This system collects the traffic data and the vehicle's position for an appropriate road decision. Thus, the user can control and monitor the automotive parts, which are transported to the production lines, especially in traffic jams. Fleet control is one of the problems that can be applied to MAS. In this study [101], MADARP (Multi-Agent Dial-a-Ride Problem) agent architecture is dedicated to the implementation of passenger transport systems. A number of main tools that perform the basic interface provide service and support services using a heterogeneous fleet to manage different transport demands. Agent usage integrates tools and users widely and allows you to easily adapt architecture to different planning models. In this study [102], The dynamic orientation of a fleet of cyber vehicles have been discussed with a view to minimizing the combined system cost, which includes the total time spent and the total energy consumption of all cybers. A model of the dynamics and energy consumption of a cyber car fleet based on the definition of the dynamics of each cyber car and road network conditions is proposed. A number of traceable and scalable multi-agent control methods have been proposed, including the multi-agent model predictive control and parameterized control for the dynamic direction of cyber vehicles. MAS was also used with rail systems [103–106]. For example, in this study [106], The movement of a series of high-speed trains running on the railway line is modelled by a MAS in which each train communicates with adjacent trains to adjust its speed. This paper [104], considers the problem of passing the trains moving in the same direction and proposes a multi-agent-based solution to take immediate decisions by negotiation between train agents and reduce the overall delay of the system up to an acceptable limit. A method for real-time train conflict resolution by cooperative, multi-agent negotiation is presented in this paper [107] A dispatch agent, as the leader, builds the negotiation set of alternative solutions and transport operator agents negotiate using the Monotonic Concession Protocol In each dispatching area. Conflicts considering the objectives of authority and train companies can be solved considering the timetable modifications suggested by the negotiation. A multi-agent-based solution has been proposed that automates train passage and minimizes system latency. The communication and coordination between adjacent intersections are facilitated with MAS [108]. The aim is to attain the emerging effect of minimizing the time loss due to traffic congestion over time for a chosen area. The proposed perimeter gating control mechanism assesses the Simulation of Urban Mobility (SUMO) traffic simulation suite along with Java Agent Development Environment (JADE). In [109], Autonomous Intersection Management (AIM) problem is investigated using RL.

1.3 Hypothesis

This study proposes solutions to traffic problems through the utilization of MAS. The thesis demonstrates that the effectiveness of traffic light controls can be increased using the deep Q-learning method for determining the phase sequence and the fuzzy logic method for determining the timings for green and red phases. In addition to this, the state information based on vehicle position information can be used as input values for both deep Q learning and fuzzy logic. In the celling method used as state information, the efficiency of the intelligent traffic intersection controller can further be increased by keeping the cell lengths of the regions close to the intersection small and the cell lengths of the far areas larger.

The use of the platoon method in reservation-based traffic intersection management systems can also increase efficiency at the traffic intersection. In optimum route planning, the use of state vectors including information on the state of traffic lights, the density of neighbouring roads, the location of neighbouring intersections, the location of the agent and the destination increases efficiency.

2. TRAFFIC LIGHT CONTROL SYSTEMS

Traffic intersections are one of the most important places that directly affect traffic flow, as they are the intersection points of more than one road. The traffic light is an important solution to change the pass permission for vehicles and pedestrians. It is actually possible to have less traffic density with adaptive changes in lighting periods depending on changing traffic density situations at the intersections.

Traffic light control is an important way to reduce traffic congestion. There are basically two types of Traffic Light Control (TLC) methods. The first is the periodic change of traffic lights at predetermined times. The second method is to change traffic lights automatically according to the data from the sensors. Most traffic intersection signal controllers are of the fixed cycle type (traditional), meaning there are constant green/red phase times for each traffic signal cycle. This method is relatively easy to implement, but it usually results in poor performance. With the development of technology, intelligent controllers started to be used instead of fixed-time traffic light control systems [110–112]. Intelligent traffic lights offer undeniable benefits, especially in metropolitan areas. As a result, the fuzzy control technique has been widely used in many applications of traffic light control after Lotfi A. Zadeh described the theory of indefinite sets in 1965 [113]. Pappis and Mamdani presented a Fuzzy Logic Control (FLC) at the traffic intersection of two one-way streets [114]. The FLC obtains an output based on three inputs: the elapsed time of the current interval, the number of vehicles passing through the junction at the green light, and the number of vehicles waiting at the red light. The green phase duration is calculated using the FLC's fuzzy output, with five rules used for each ten-second interval. Favilla et al. proposed an FLC with adaptive strategies [115], which adjusts the membership functions according to traffic conditions to optimize the performance of the control function. There are both statistical and fuzzy adaptation strategies available for traffic light control.

This section discusses various traffic light controllers, including PI control, FLC, and state space model control. The studies are presented in order of increasing complexity, from simple to complicated structures. None of the studies conducted in this section utilized any learning algorithms. The design of the environments and their corresponding simulations were carried out in the SUMO program. The studies in this section include excerpts from these publications [116–120].

2.1 Fuzzy Logic and PI Control for Traffic Lights

Control of traffic lights at traffic intersections can have important consequences in reducing traffic density and shortening waiting time in traffic. Intelligent Intersection Management (IIM) technology has started to develop in traffic intersections as part of TLC systems. Fuzzy logic and Proportional Integral (PI) control methods are proposed to be used in the Intelligent Intersection method which is an alternative method to the classical traffic lights, which are the places where the most traffic is experienced. A traffic intersection and vehicles were made using the SUMO traffic simulation program. When planning the routes of the vehicles, the vehicle density from the east-west direction is thought to be higher. Simulations for the designed controllers and conventional traffic light controllers are performed, and the results are compared.

2.1.1 System overview

The general structure of the simulation environment of the traffic light control system controlled by the Traffic intersection agent method is shown as in Figure 2.1. There are two detectors placed on the road for each strip. The detectors in each lane are used to determine the number of vehicles in the lane. The traffic light controller is responsible for controlling the duration of the green or red status of traffic lights at the intersection according to the traffic conditions. The optimum cycle time was calculated by Webster's method. As the traffic density increases in the simulation, the optimum cycle time can exceed 120 seconds. Cycle time selected to 120 seconds to minimize delay and driver frustration

The traffic light control system at the intersection is designed according to the following assumptions and limitations:

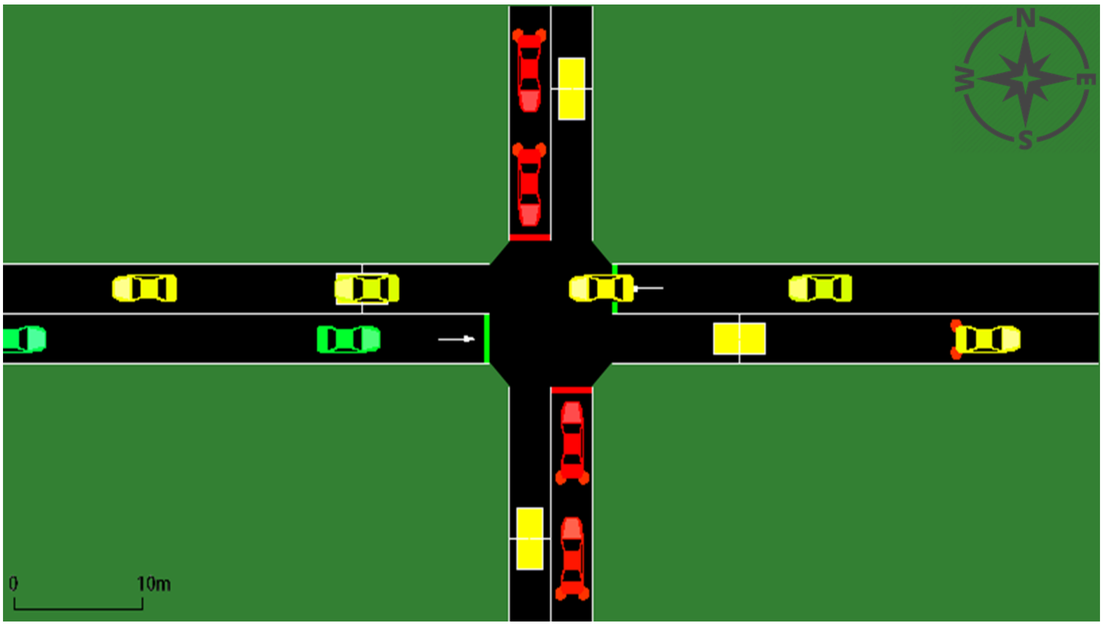


Figure 2.1 : Traffic intersection model

- Traffic moves from north to south, from west to east and vice versa.
- When the green light is on for the vehicles from the north and south, the red light is on for the vehicles from the east and west, and vice versa.
- Right and left turns are not allowed at the intersection.
- Number of lanes on the roadway is one.
- Cycle length of the signal program is 120 sec.
- The minimum and the maximum durations for the green light in both directions are 6 seconds and 60 seconds, respectively.
- No amber time
- Table 2.1 shows order of phases.

Table 2.1 : Order of phases

	North	South	East	West
Phase 1	-	-	Straight	Straight
Phase 2	Straight	Straight	-	-

2.1.2 Traffic light design with fuzzy logic controller

Fuzzy logic technology allows the implementation of real-life rules similar to the way humans would think. For example, humans would think in the following way to control traffic situation at a certain junction: “If the traffic is heavier on the north or south lanes and the traffic on the west or east lanes is less, then the traffic lights should stay green longer for the north and south lanes”. Such rules can be easily accommodated in the fuzzy logic controller. The beauty of fuzzy logic is that it allows fuzzy terms and conditions such as “heavy”, “less”, and “longer” to be quantized and understood by a computer. It is possible to show that fuzzy logic-based TLC systems can achieve better results in comparison to conventional ones. In intersection management based on fuzzy logic, the intersection controller changes the traffic lights depending on the number of vehicles at the intersection. The fuzzy logic controller is designed for a 4-way traffic junction: north, south, east and west as shown in Figure 2.1.

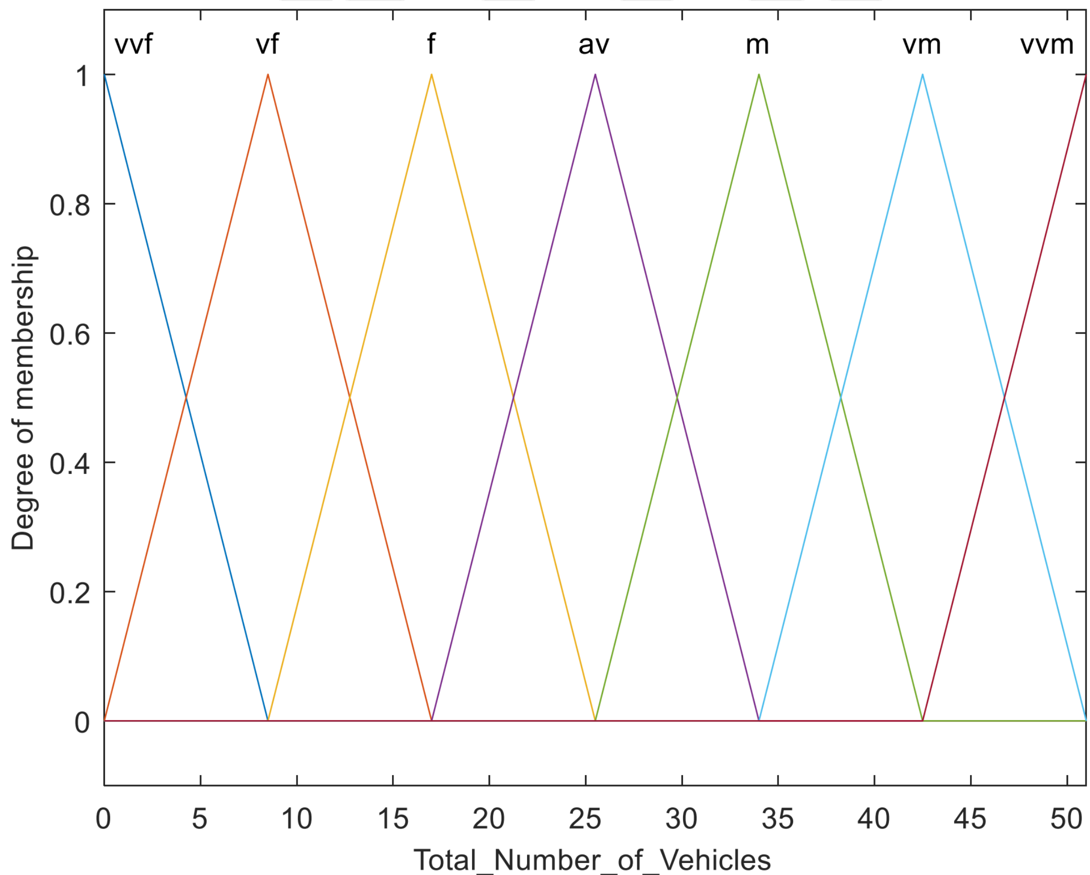


Figure 2.2 : Membership function of the total number of vehicles from input value for FLC.

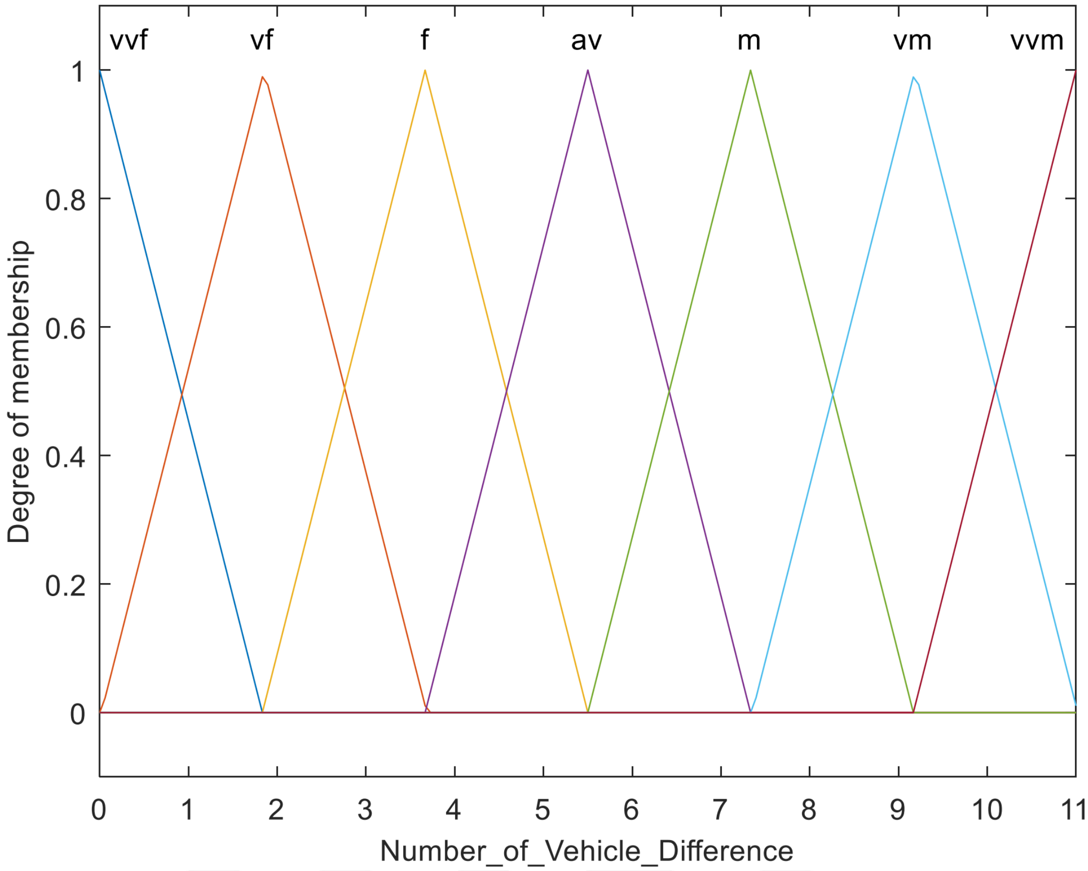


Figure 2.3 : Membership function of the number of vehicle differences from input value for FLC.

Two fuzzy input variables have been selected in the traffic lights controller. The first is the total number of vehicles (TNV) at the intersection. The other variable is the difference between the number of vehicles (VND) coming from the east and west and the total number of vehicles coming from the north and south. It includes 7 membership functions which are very very few (vvf), very few (vf), few (f), average (av), much (m), very much (vm) and very very much (vvm). Based on the fuzzy rules as given in Table 2.2, the fuzzy controller produces an output according to current traffic conditions to determine the green light duration. The direction in which the green phase will be active is determined according to the difference in the number of vehicles. For example, if the difference in the number of vehicles (the difference between total vehicles from east and west and the total vehicles from north and south) is negative, the green phase is effective for vehicles from north and south. If the difference is positive or zero, the green phase is active for vehicles from the east and west. The green phase duration, which is the fuzzy logic output value, is then calculated according to the fuzzy logic input values. The green phase is recalculated

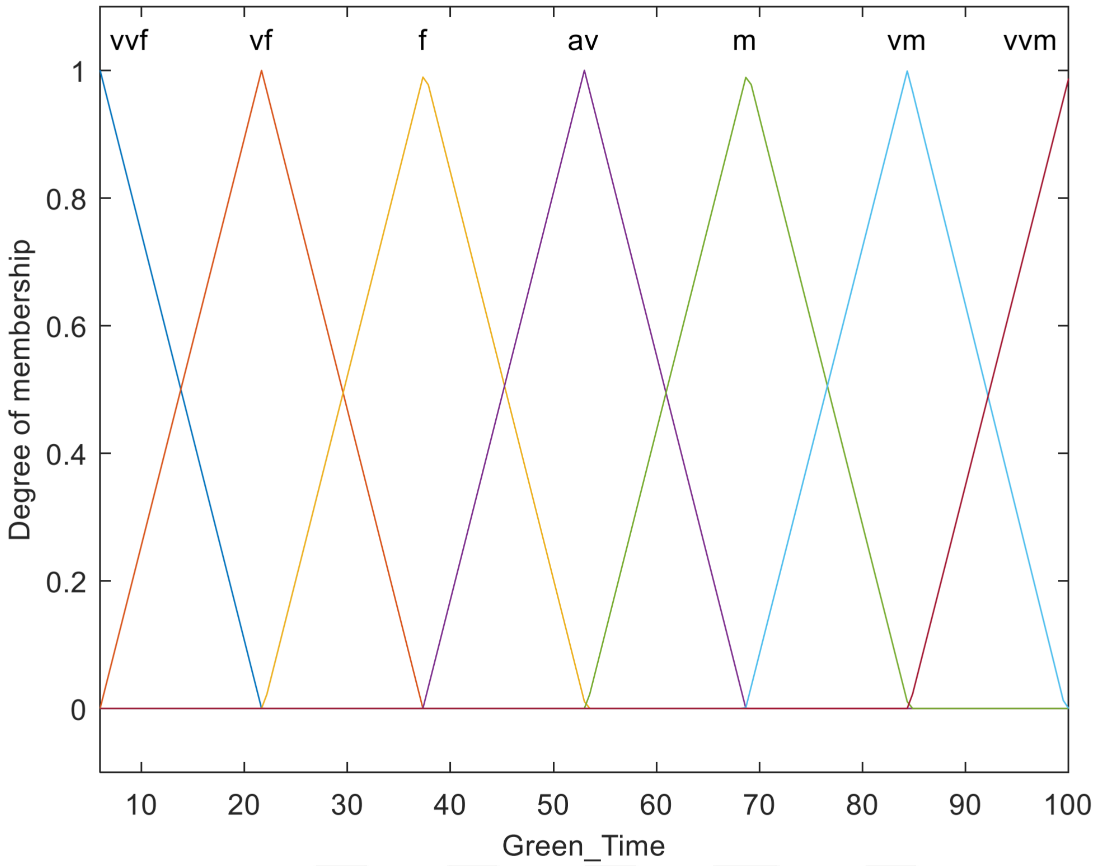


Figure 2.4 : Traffic light green phase time member function.

in every second, so the green phase times change dynamically. However, it is expected that the time will be complete when the sign of the difference in the number of vehicles changes (from positive to negative or vice versa). In this case, the green phase will remain active for the last value before the signal change, then the green phase will be active for the other direction. The cycle continues in this way. While there are five members for the graph to control the traffic lights, there are nine member functions for the output. A graphical representation of the membership functions of the output value is given in Figure 2.4 and input values are given in Figure 2.2 and Figure 2.3.

2.1.3 Traffic light design with PI controller

When performing PI-type traffic light control, the difference in the number of vehicles in both directions is considered as the error of the traffic intersection system. The green phase times in the PI type Traffic Light Controller are determined by the multiplication of the K_p coefficient by error and the multiplication of the K_i coefficient by the total error. Here, the green phase times can be negative or positive. This helps the determination of the direction of the green phase. For example, when the green

Table 2.2 : Rule table for fuzzy logic

TNV \ VND	vvf	vf	f	av	m	vm	vvm
vvf	vvf	vvf	vf	vf	f	f	av
vf	vvf	vf	vf	f	f	av	m
f	vf	vf	f	f	av	m	m
av	vf	f	f	av	m	m	vm
m	f	f	av	m	m	vm	vm
vm	f	av	m	m	vm	vm	vvm
vvm	av	m	m	vm	vm	vvm	vvm

phase time is positive, there will be a green phase for vehicles from the east and west directions. When the green phase time is negative, the green phase will be for vehicles from the north and south. The total error of the system does not increase much because the error values can be positive or negative. The duration of the green phase is recalculated every second, so the green phase times change dynamically. However, when the sign of the green phase time changes (from positive to negative, or vice versa), the time is expected to complete. In this case, the green phase will remain active for the last value before the signal change, then the green phase will be active for the other direction. The cycle continues in this way. PI parameters K_p and K_i values were determined by considering the effects of proportional and integral coefficients on the system.

2.1.4 Simulation results

A simulation environment is designed and implemented using SUMO. The CO_2 emission outputs and average speed values of the vehicles were taken directly from the SUMO program. Simultaneous vehicles are produced for 300 seconds during simulation. In the simulation, the ratio of the number of vehicles coming from the east-west direction to the number of vehicles coming from the north-south direction is 1,5. In the simulation, half, one, one and a half, two and two and a half vehicles are produced per second to determine the vehicle density. Therefore, during simulation, 150, 300, 450, 600 and 750 vehicles were produced for vehicle densities of 0.5, 1, 1.5, 2 and 2.5, respectively.

Figure 2.5 shows the average speed values for different traffic light control techniques relative to the change in vehicle density. Figure 2.6 shows the total CO_2 emission

Table 2.3 : Simulation results

	Vehicle Density	Fixed Time	Fuzzy Control	PI Control
CO_2 *Emission* (kg/s)	0.5	32,54	25,86	25,28
	1	80,23	67,52	80,68
	1.5	177,18	156,34	172,99
	2	270,45	235,14	260,63
	2.5	356,88	304,60	347,79
Average Speed (km/h)	0.5	35,73	43,38	44,09
	1	29,22	32,94	29,58
	1.5	19,92	21,89	21,15
	2	17,05	18,77	18,38
	2.5	16,15	18,51	17,44

values according to vehicle density for different control techniques. Table 2.3 shows the results of CO_2 emission and the average speed of vehicles. It can be stated that the fuzzy logic type traffic light controller and the PI type traffic light controller give much better results than the traditional traffic light controllers shown in Figure 2.5 and Figure 2.6. In the methods we propose, efficiency is seen more clearly in average speed values. However, as shown in Figure 2.6 The sum of CO_2 emission was less changed for either FLC or PI. The reason for this is that vehicles do not emit CO_2 emissions when they wait at the traffic junction, i.e. when their speed is 0. An important advantage of the PI type controller over the fuzzy logic type controller is that the processing load is less. Indeed, this was also observed during the simulation. In addition, it is seen that the fuzzy Logic TLC system gives slightly better results than PI-type TLC system as the density of vehicles increases.

Each vehicle crossed the intersection once. In the traditional method, the green light is steadily lit for 60 seconds for each phase. Traffic lights are calculated dynamically according to fuzzy logic input values with constraints of minimum 6 seconds and a maximum of 100 seconds for both proposed methods. Also, there is no amber time in the traffic junction system.

As can be seen from the simulation results, the proposed methods give better results than the traditional methods.

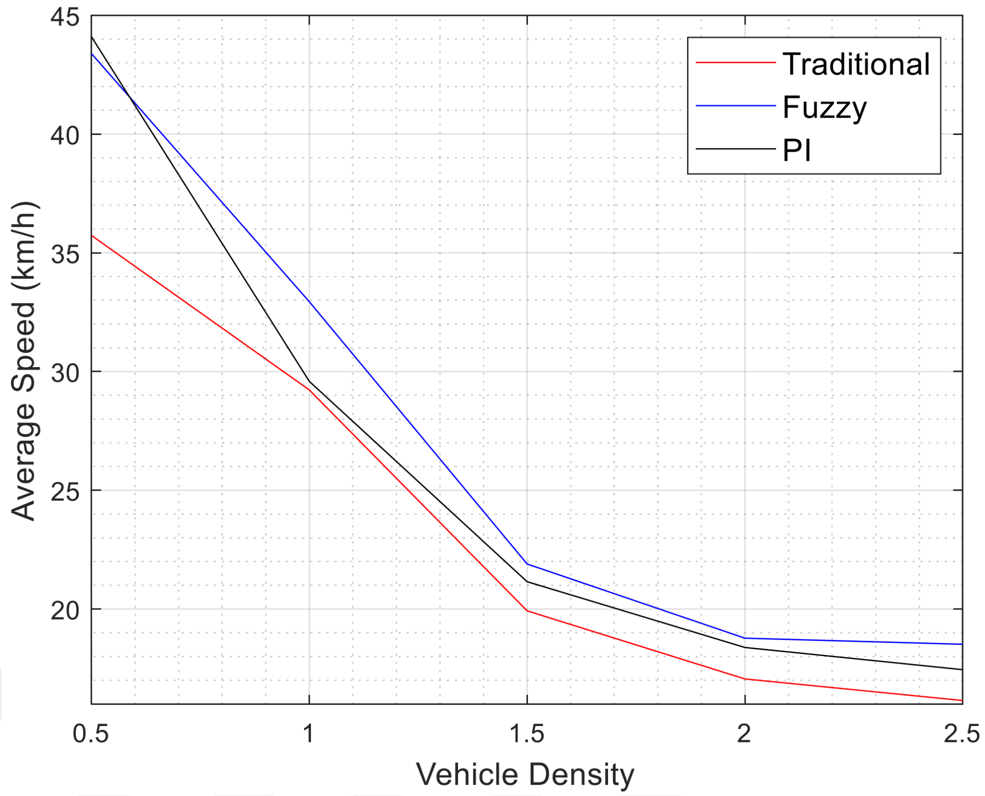


Figure 2.5 : Average speed values according to changes in the number of vehicle density.

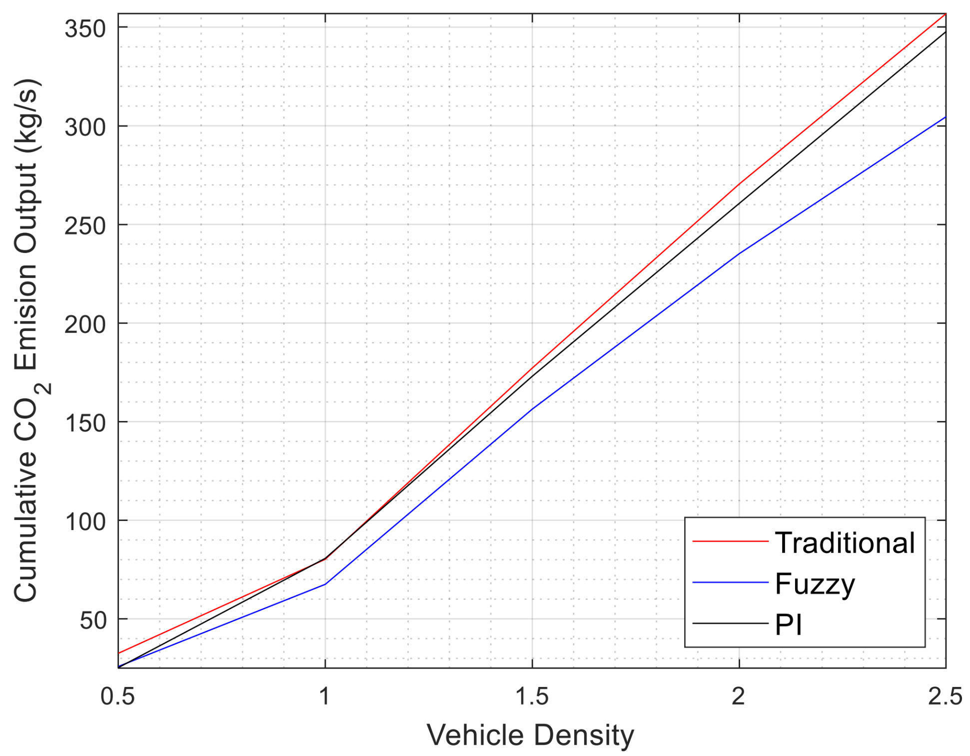


Figure 2.6 : Total emission values according to changes in the number of vehicle density.

2.2 State Feedback Control for Traffic Light Systems

In this section, the State Feedback Controller and FLC with fixed controller parameters are used to control the period of green light time with consideration of traffic light cycle time, maximum and minimum green time and are compared with each other. Ackermann's Formula was used while designing the State-Feedback control [121]. Simulation of this control system is made and handled using Simulation of Urban Mobility (SUMO). Results are compared for the proposed types of Traffic Light Control Systems. The traffic flow scenario is simulated so that the number of vehicles coming from the east-west direction is higher than the number of vehicles coming from the north-south direction.

2.2.1 System overview

In this study, the traffic light control on the 4-way intersection was carried out. As shown in Figure 2.7, 4 roads are named after 4 main directions (east, west, north and south). The traffic light at the intersection has two different phases. Table 2.4 shows the order of phases. The 4-way intersection has 4 different entrances and 4 different exits. All roads have two lanes (one in each direction). Detectors were placed in the entry and exit areas of each road to determine the number of vehicles on each road. Figure 2.7 shows the traffic intersection simulation model.

The TLC at the intersection is designed according to the following assumptions and limitations:

- Right and left turns are not allowed at the intersection.
- The minimum time for the green light in both directions is 6 seconds.
- Maximum green light duration for FLC and State Feedback Traffic Light Controllers is 100 seconds.
- No amber time.

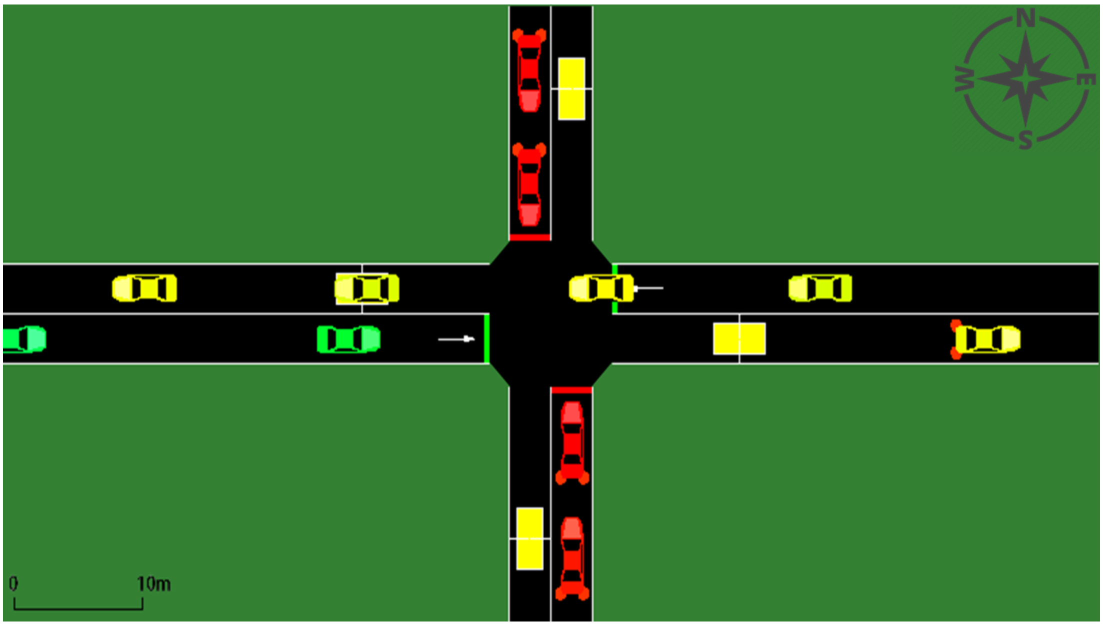


Figure 2.7 : Traffic intersection simulation model.

Table 2.4 : Order of phases

	North	South	East	West
Phase 1	-	-	Straight	Straight
Phase 2	Straight	Straight	-	-

2.2.2 Traffic light design with fuzzy logic controller

FLC works similarly to what people think. With the rules used, FLC can be used close to human thought. This method is also very useful for traffic light control. If you can accurately express the amount of traffic flow with the rules, you can get proper results.

Table 2.5 : Rule table for fuzzy logic

TNV \ VND	vvf	vf	f	av	m	vm	vvm
vvf	vvf	vvf	vf	vf	f	f	av
vf	vvf	vf	vf	f	f	av	m
f	vf	vf	f	f	av	m	m
av	vf	f	f	av	m	m	vm
m	f	f	av	m	m	vm	vm
vm	f	av	m	m	vm	vm	vvm
vvm	av	m	m	vm	vm	vvm	vvm

Inputs for FLC are chosen as the Total Number of Vehicles (TNV) at the intersection and the NVD at the intersection. TNV input represents the actual vehicle number in the

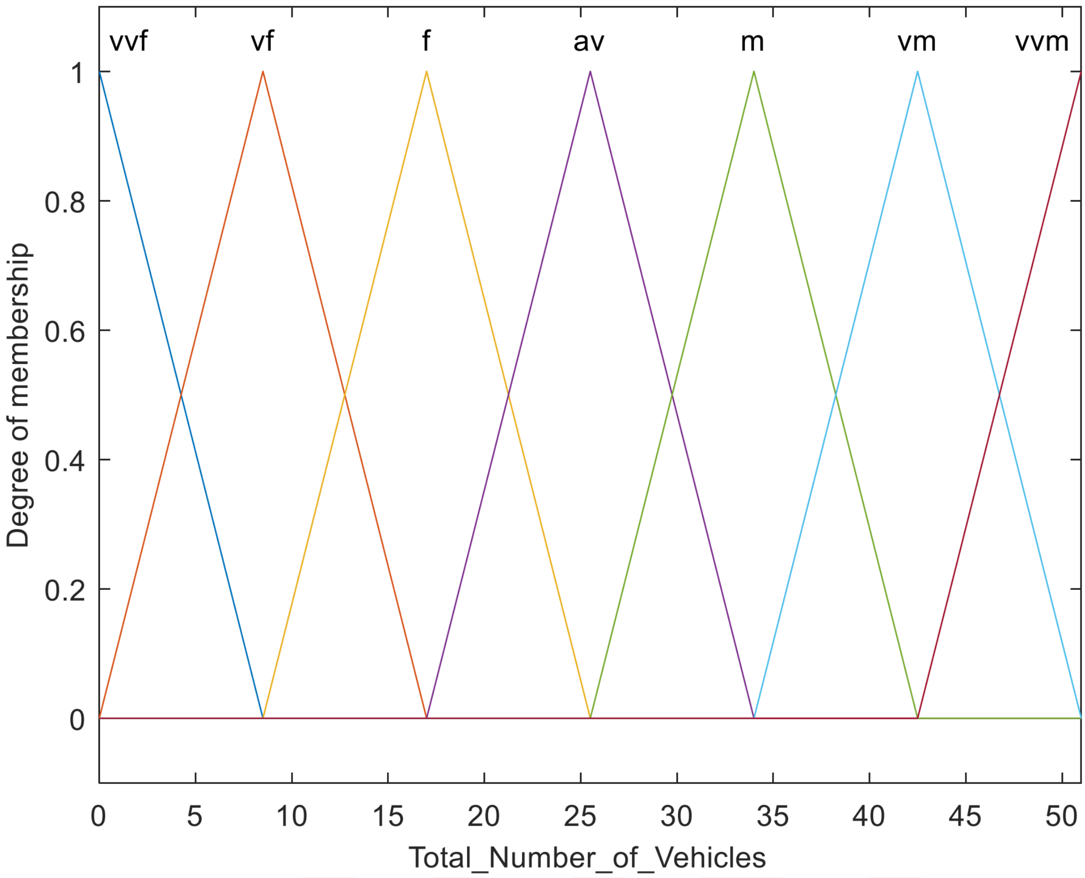


Figure 2.8 : Membership function of the total number of vehicles.

simulation. NVD is the difference between the sum of the number of vehicles coming from the east and west and the number of vehicles coming from the north and south.

Two fuzzy input variables have been selected in the traffic lights controller. The first is the total number of vehicles (TNV) at the intersection. The other variable is the difference between the number of vehicles (VND) coming from the east and west and the total number of vehicles coming from the north and south. The fuzzy controller has 7 membership functions which are very very few (vvf), very few (vf), few (f), average (av), much (m), very much (vm) and very very much (vvm). Based on the fuzzy rules as given in Table 2.5, the fuzzy controller produces an output according to current traffic conditions to determine the green light duration.

The green phase is recalculated every second, so the green phase times change dynamically. The green phase duration, which is the FLC output value, is calculated according to the FLC input values. Membership functions of NVD and TNV inputs for FLC are shown in Figure 2.8 and Figure 2.9. A graphical representation of the membership functions of the output value is given in Figure 2.10.

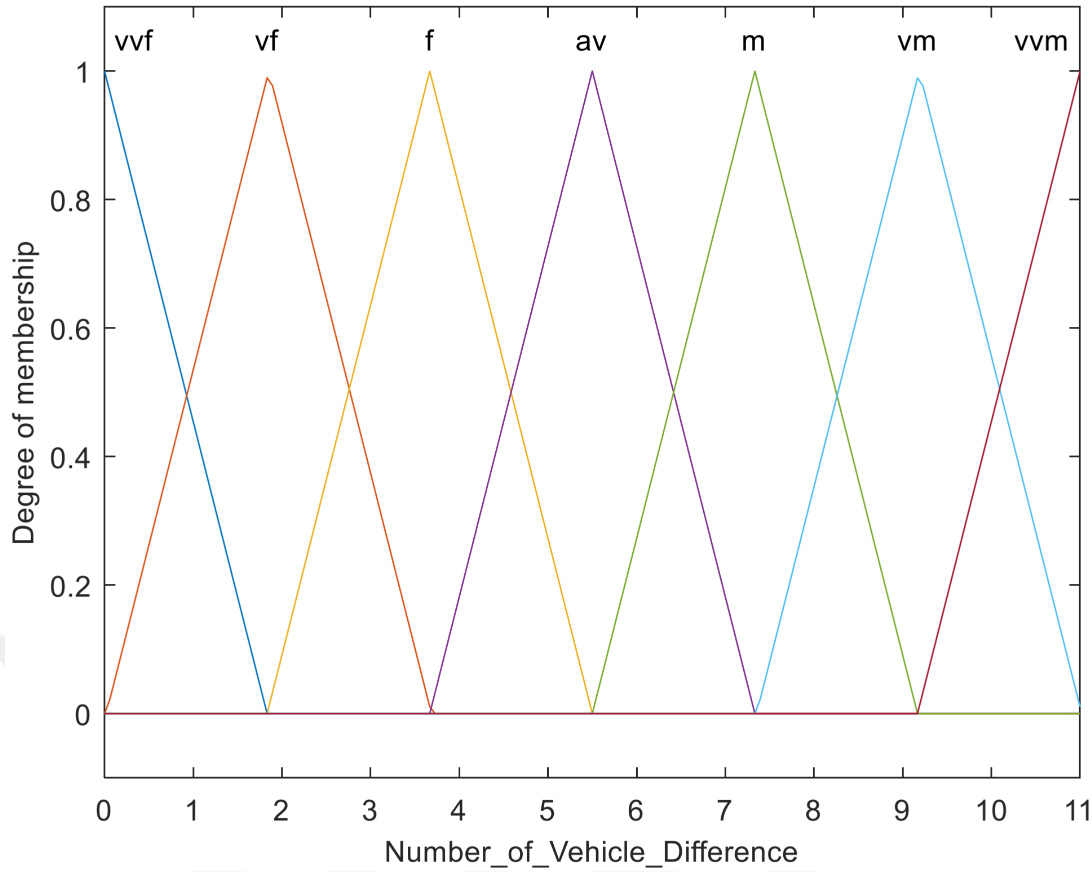


Figure 2.9 : Membership function of the number of vehicle differences.

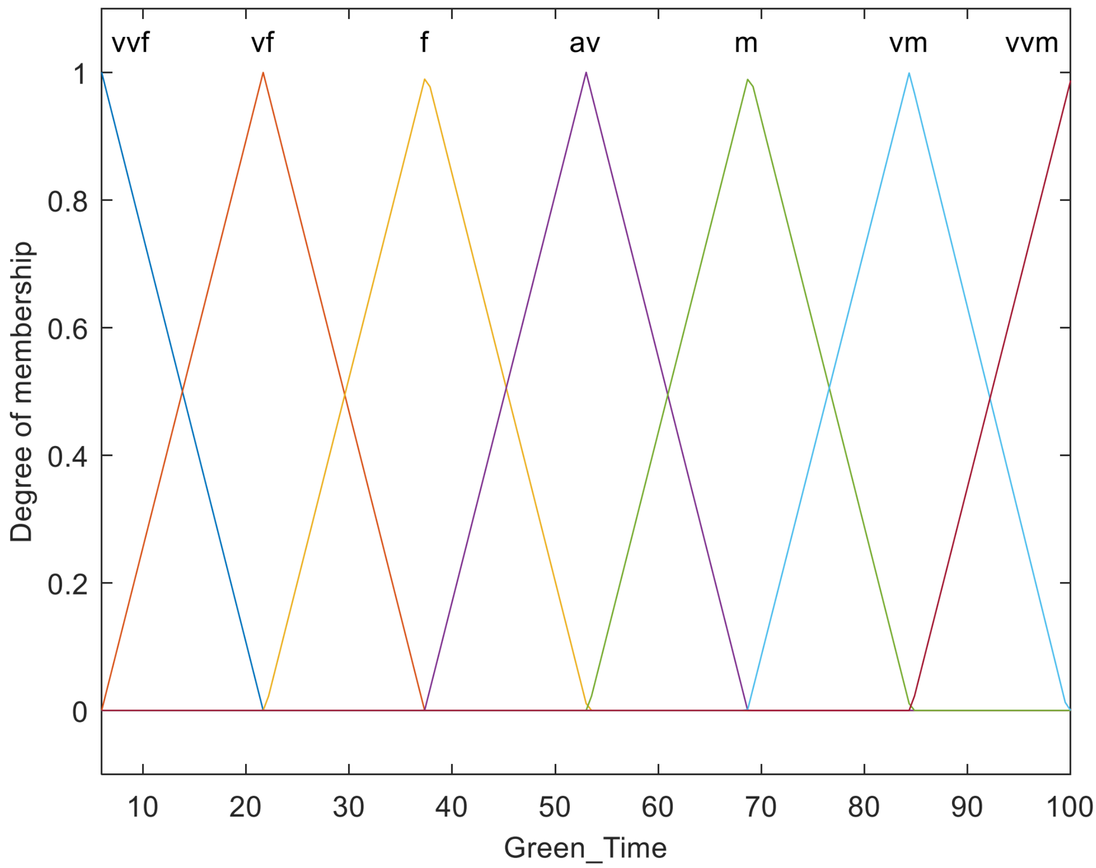


Figure 2.10 : Traffic light green phase time member function.

2.2.3 State feedback control for intelligent traffic light systems

2.2.3.1 State-space equations

Dynamic equations of the system are derived in discrete time. The system equations are given using two state variables named as Q_i and W_i. Q_1 is the sum of vehicle numbers coming from the east. Q_2 is the sum of vehicle numbers coming from the west. Similarly, Q_3 and Q_4 are the sum of vehicles from the north and the south, respectively. T is the sampling time. The state equations are given as follows:

$$Q_i(n+1) = Q_i(n) + q_i(n) - d_i(n)S_i(n) \tag{2.1}$$

$$W_i(n+1) = W_i(n) + TQ_i(n) + \frac{1}{2}Tq_i(n) - \frac{1}{2}Td_i(n)S_i(n) \tag{2.2}$$

The other state variable W represents the waiting times of the vehicles for each direction. That is W_1 and W_2 are the sum of the waiting times of vehicles coming from the east and the west, respectively. On the other hand, W_3 and W_4 are the sum of waiting times of vehicles coming from the north and the south, respectively. In equation 2.1 and equation 2.2, d_i is the number of vehicles leaving the intersection, q_i is the number of vehicles entering the intersection.

$$X(n+1) = AX(n) + B(n)S(n) + C(n) \tag{2.3}$$

$$X(n) = \begin{bmatrix} Q_{1,2}(n) - Q_{3,4}(n) & W_{1,2}(n) - W_{3,4}(n) \end{bmatrix} \tag{2.4}$$

$$\begin{aligned} Q_{1,2}(n) &= Q_1(n) + Q_2(n) \\ Q_{3,4}(n) &= Q_3(n) + Q_4(n) \end{aligned} \tag{2.5}$$

$$\begin{aligned} W_{1,2}(n) &= W_1(n) + W_2(n) \\ W_{3,4}(n) &= W_3(n) + W_4(n) \end{aligned} \tag{2.6}$$

$$S(n) = \begin{bmatrix} S_1(n) & S_2(n) \end{bmatrix}^T \tag{2.7}$$

When the green light is on in the east and west directions, the input signal S_1 is applied. The S_2 input signal is applied for the north and south directions.

$$\begin{aligned} S_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix}^T \\ S_2 &= \begin{bmatrix} 0 & 1 \end{bmatrix}^T \end{aligned} \tag{2.8}$$

The state equations of the traffic intersection model are seen in equation 2.9, equation 2.10, equation 2.11 and equation 2.12.

$$A = \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix} \tag{2.9}$$

$$B = \begin{bmatrix} -d_1 - d_2 & -d_3 - d_4 \\ -\frac{1}{2}Td_1 - \frac{1}{2}Td_2 & -\frac{1}{2}Td_3 - \frac{1}{2}Td_4 \end{bmatrix} \tag{2.10}$$

$$C = \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{2.11}$$

$$C(n) = \begin{bmatrix} Tq_{1,2} & Tq_{3,4} \end{bmatrix} \tag{2.12}$$

$$\begin{aligned} q_{1,2} &= q_1 + q_2 \\ q_{3,4} &= q_3 + q_4 \end{aligned} \tag{2.13}$$

2.2.3.2 Ackermann's formula

The Ackermann formula is a very useful method for controlling systems with state space models, especially in high-grade systems. When the desired poles are known, $z = \lambda_1$, $z = \lambda_2$,... $z = \lambda_n$ for an n^{th} order system, the characteristic equation,

$$\alpha_c(z) = (z - \lambda_1)(z - \lambda_2)...(z - \lambda_n) \tag{2.14}$$

$$\alpha_c(z) = z^n + \alpha_{n-1}z^{n-1} + \alpha_{n-2}z^{n-2}...\alpha_1(z) + \alpha_0 = 0 \tag{2.15}$$

the characteristic equation of state feedback systems is;

$$det[zI - A + BK] = 0$$
$$det[zI - A + BK] = \alpha_c(z) \tag{2.16}$$

Ackermann's formula for the gain matrix K is given by

$$K = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} B & AB \end{bmatrix}^{-1} \alpha(A) \tag{2.17}$$

K is a row vector of n elements. In equation 2.17, $\alpha(A)$ is a matrix polynomial with coefficients determined by the desired closed loop system characteristic polynomial as follows.

$$\alpha_c(A) = A^n + \alpha_{n-1}A^{n-1} + \alpha_{n-2}A^{n-2}...\alpha_1(A) + \alpha_0 = 0 \tag{2.18}$$

Considering the fact that the desired reference signal is zero (equal distribution of vehicles and waiting times in the junction), the green light time for the east-west bound is calculated by multiplying the gain vector obtained using equation 2.17 by system states.

$$S_1(n) = KX(n) \tag{2.19}$$

2.2.4 Simulation results

A simulation environment is designed and implemented using SUMO. The CO_2 emission values, the average speed values of the vehicles and the total waiting time are taken directly from the SUMO. Simultaneous vehicles are produced for 1 hour during the simulation. In the simulation, the ratio of the number of vehicles coming from the east-west direction to the number of vehicles coming from the north-south direction is 1.5. Moreover, each vehicle crosses the intersection once. In the simulation, 0.1, 0.2, 0.3,0.4, 0.5 and 0.6 vehicles are produced per second to determine the vehicle density. Therefore, during simulation, 360, 720,1080, 1440, 1800 and 2160 vehicles are produced for vehicle densities of 0.1, 0.2, 0.3,0.4, 0.5 and 0.6, respectively.

In fixed-time traffic light control, the first 50 seconds green light is on for vehicles coming from an east-west direction and then 50 seconds of green light is on for vehicles from the north-south direction. Then, for vehicles coming from the east-west direction,

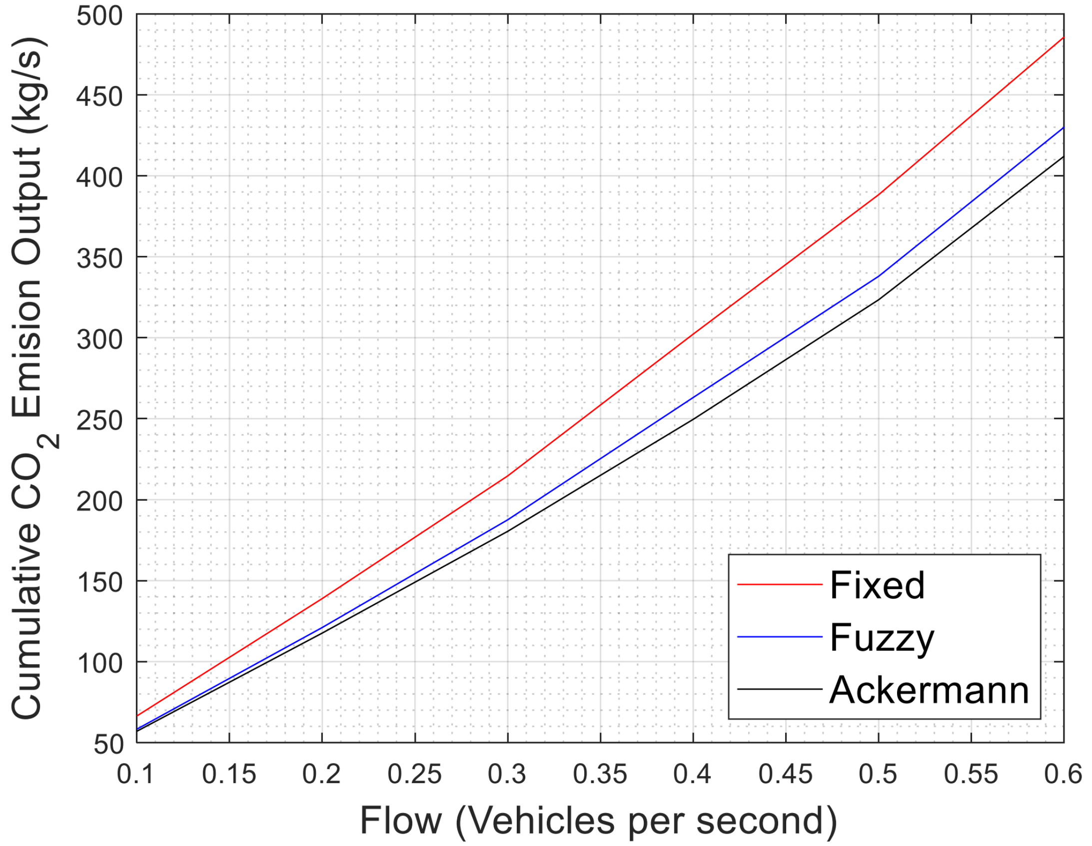


Figure 2.11 : Simulation results based on cumulative CO_2 emission.

the green light is on for 50 seconds, and the cycle continues. For FLC, two fuzzy input variables and one output have been selected in the traffic lights controller. Membership functions of these variables are shown in Figure 2.8, Figure 2.9 and Figure 2.10. The characteristic polynomial selected for State Feedback TLC can be seen in equation 2.20. Poles for the selected characteristic polynomial are z = 0.9480 and z = 0.9737. The performance criteria of the selected characteristic polynomial are overshoot = 0.1 and settling time = 100 seconds. And sampling time T=1

$$\alpha_c(z) = z^2 - 1.92175z + 0.92311 \tag{2.20}$$

Figure 2.11 shows the total CO_2 emission values according to traffic flow (vehicles per second) for different control techniques. Figure 2.12 shows the average speed values for different traffic light control techniques relative to the change in traffic flow. In addition to this, in Figure 2.13, the total waiting times of all vehicles waiting at the traffic intersection are shown in minutes according to the change in the traffic flow. It can be seen from the simulation results that the state feedback traffic light controller performs better than both the fuzzy logic traffic light controller and the fixed-time traffic light controller. In Figure 2.12 and Figure 2.13, where the average speed and total waiting times are shown, it is clear that the proposed method gives better results. It is seen that the traffic light controller with FLC gives better results than the fixed-time traffic light controller. However, as shown in Figure 2.11, the sum of CO_2 emission values is closer to each other in all methods. The reason for this is that the vehicles are assumed not to emit CO_2 when they wait at the traffic junction, i.e. when their speed is 0. An important advantage of the State Feedback Traffic Light Controller over the FLC is that the processing load is less. This situation was observed during the simulation process.

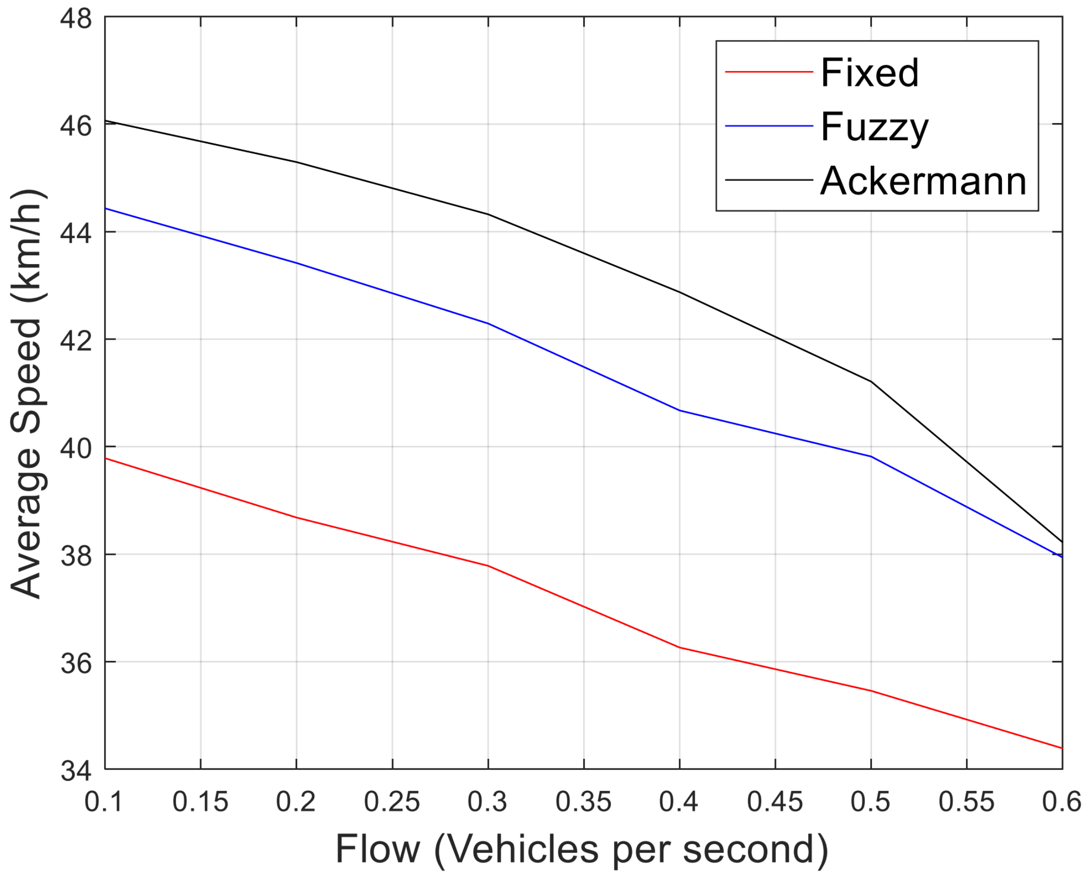


Figure 2.12 : Simulation results based on average speed.

In addition, as the traffic flow at the traffic junction increases, it is seen that the average speed values approach each other for FLC and State feedback TLC. However, this event was not observed in the total waiting time.

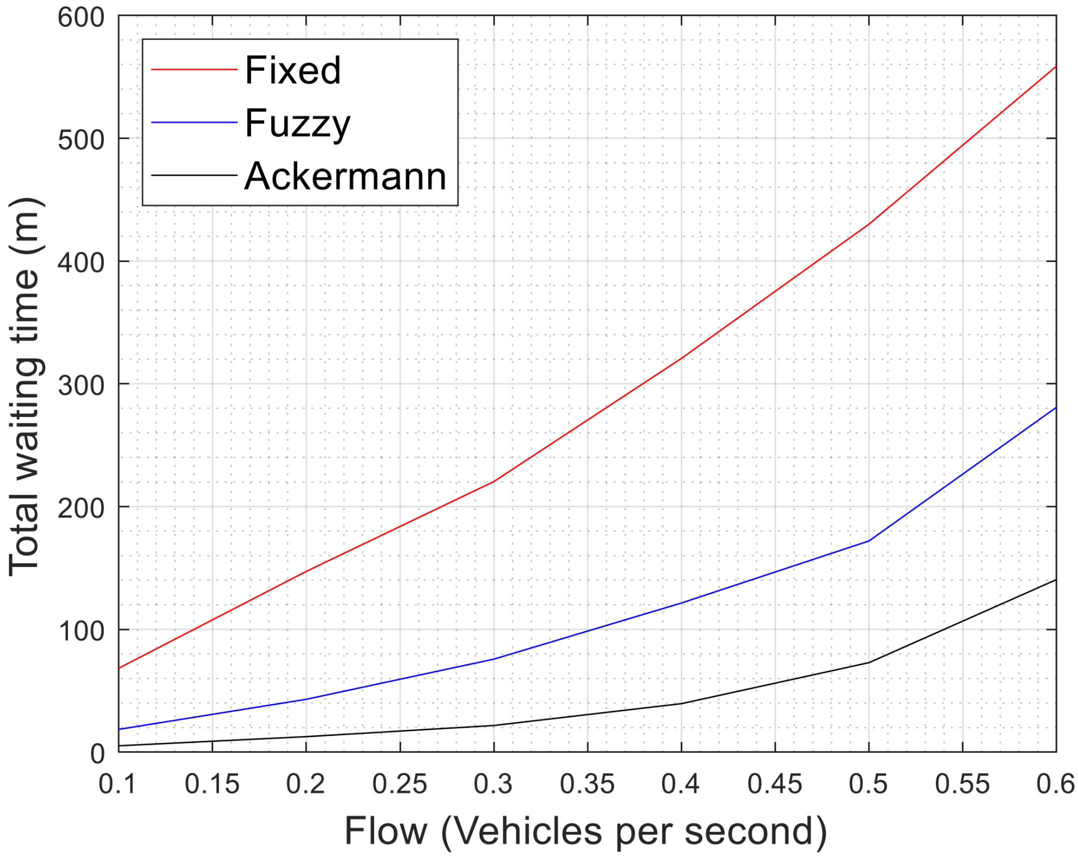


Figure 2.13 : Simulation results based on total waiting time.

As can be seen from the simulation results, the proposed method gives better results than both FLC and fixed-time traffic light controllers. This result is seen more clearly in total waiting time values, which are among the most important performance criteria for traffic light control. It is observed that the proposed method gives better results also in other performance criteria, average speed and total CO_2 emission output values.

2.3 Traffic Light Control System Simulation for Different Strategies with Fuzzy Logic Controller

In this study, the FLC simulation is performed for two different traffic light strategies to control the timing (green/red) of traffic light phases. Simulation of these two strategies is performed by using the Simulation of Urban MObility (SUMO) program, and the results are compared. Traffic light control simulation is performed using a FLC on a four-leg intersection. The simulation environment is made via using the Simulation of Urban MObility (SUMO) program. Developed controllers are simulated for two different strategies. When the routes of the vehicles are planned, the vehicle densities are planned to be equal. Besides, the conventional Traffic Light Controllers are also simulated, and the results are compared with each other.

2.3.1 System overview

TLC includes a 4-way traffic intersection to simulate the control of traffic lights in this study. The simulation includes the intersection which contains four different entrances and four different exits in four-leg intersection ways. Besides, all roads in the simulation consist of three lanes. Two detectors are placed in the path for each strip. The detectors in each lane are used to determine the number of vehicles in the lane. For each leg, traffic lights are added. Figure 2.14 shows the traffic intersection environment.

The TLC is responsible for checking the light (green or red lights) status of the traffic lights at the junction according to traffic density controlled by the FLC method. Figure 2.15 presents the traffic intersection model in the SUMO program environment. The TLC at the intersection is designed according to the following assumptions and limitations:

- Right and left turns are allowed at the intersection.
- The minimum time for the green light in both directions is 6 seconds.
- The yellow light is on for 3 seconds during the transition from red light to green light.

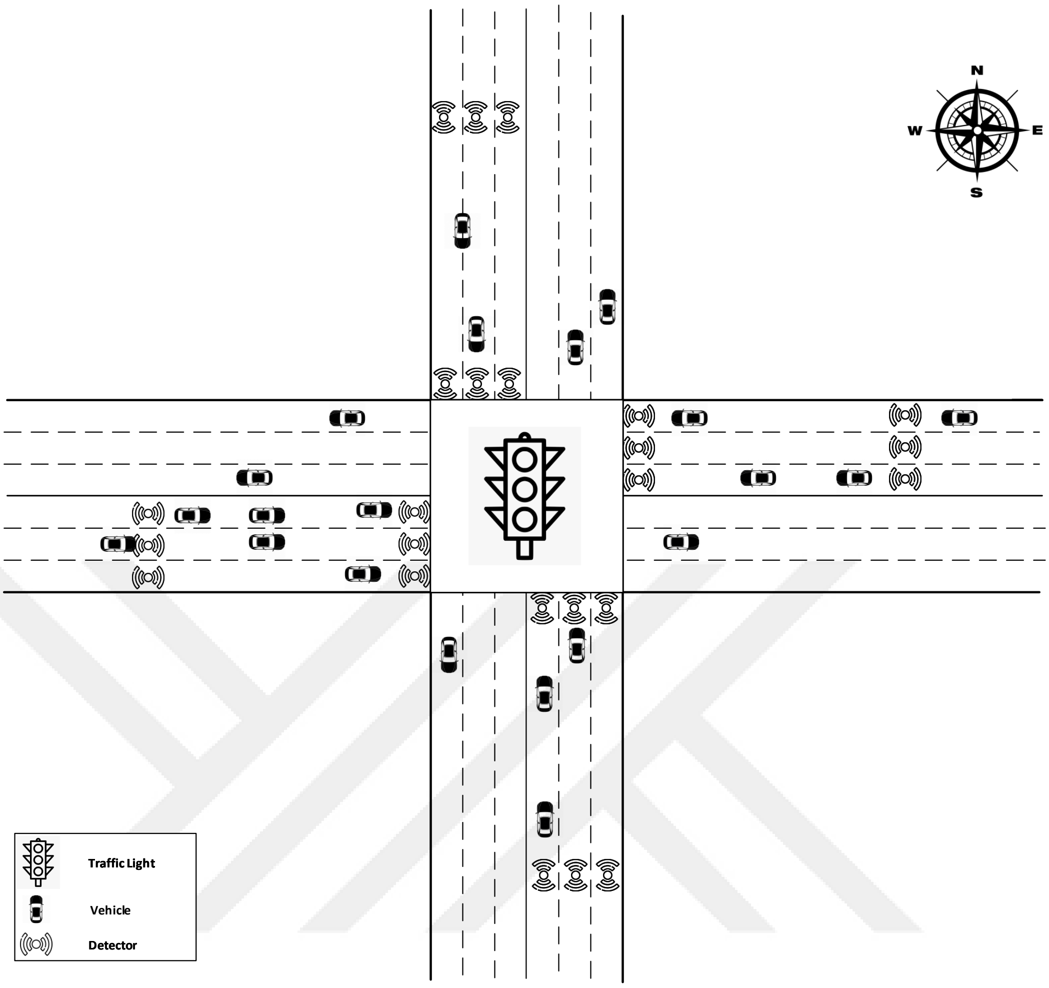


Figure 2.14 : Traffic intersection model general architecture.

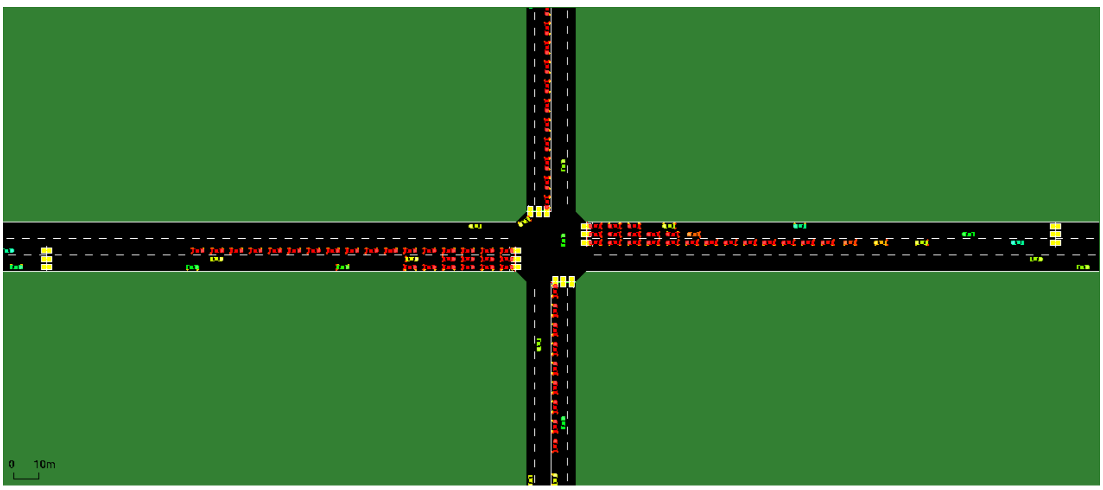


Figure 2.15 : Traffic intersection simulation model.

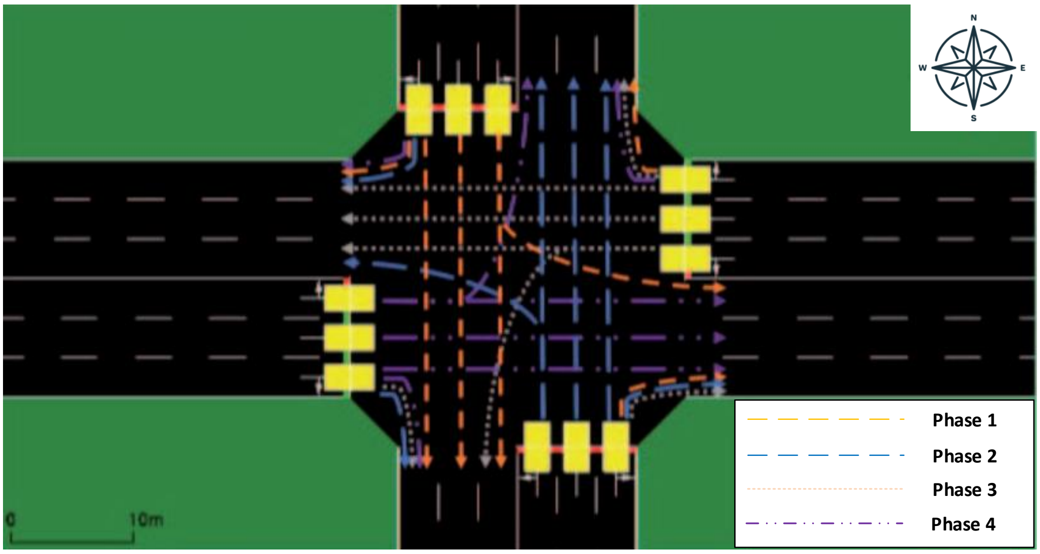


Figure 2.16 : Strategy 1 pass permission topology

In this study, simulation and control strategy of traffic intersection model is performed for two different strategies which are called as Strategy 1 and Strategy 2. Start time of green lights for both strategies are found by the FLC. In the case of Strategy 1, pass permission is allowed for one direction only and not for the other three directions. However, drivers can turn right for vehicles from the opposite direction. For example, when the permission is given for vehicles coming from the east, vehicles coming from the east have the permission to turn or pass west, north or south. In addition, there is a permit for vehicles coming from the west to the south. The strategy consists of four different light phases. Pass permission for directions represented for each lighting phases at Table 2.6 and in Figure 2.16.

Table 2.6 : Strategy 1 pass permission topology

	North	South	East	West
Phase 1	Left, Right Straight	Right	Right	-
Phase 2	Right	Left,Right Straight	-	Right
Phase 3	-	Right	Left,Right Straight	Right
Phase 4	Right	-	Right	Left,Right Straight

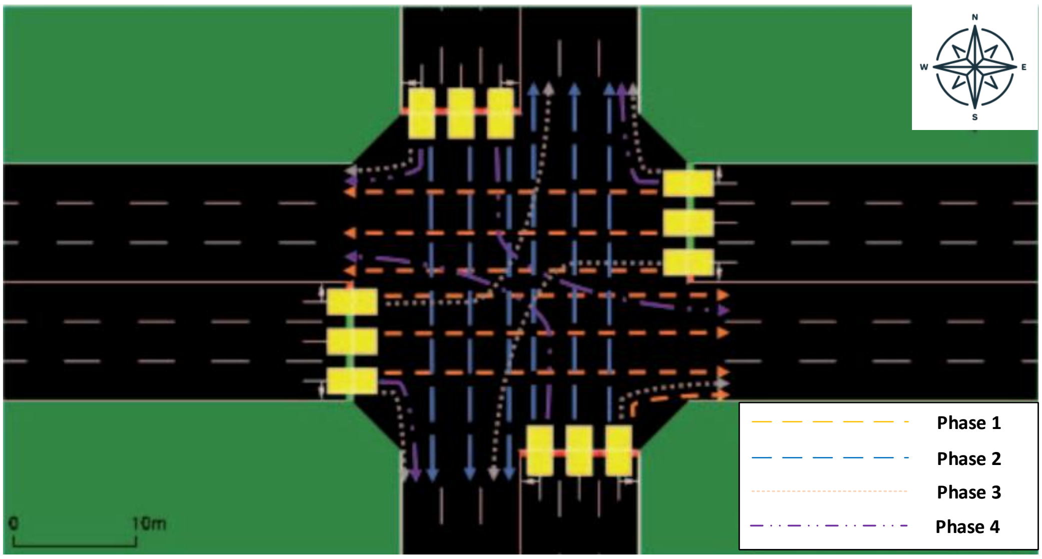


Figure 2.17 : Strategy 2 pass permission topology.

In the case of Strategy 2, left and right turn permissions are given in different phases. This strategy also consists of four different light phases. Pass permission for directions represented for each lighting phases Table 2.7 and in Figure 2.17.

Table 2.7 : Strategy 2 pass permission topology

	North	South	East	West
Phase 1	-	-	Straight	Straight
Phase 2	Straight	Straight	-	-
Phase 3	Right	Right	Left, Right	Left, Right
Phase 4	Left,Right	Left,Right	Right	Right

2.3.2 Traffic light design with fuzzy logic controller

Fuzzy logic technology allows real-life rules to be implemented as human thinks. For example, people think of the traffic situation at a particular intersection as follows: "If the traffic density in the north and south lanes is higher than the traffic density in the west and east lanes, the traffic lights should remain green longer for the north and south". Such rules can be easily placed in the FLC. The advantage of FLC is that it allows a computer to measure and understand fuzzy terms and conditions such as "few", "much" and "longer". It is possible to show that fuzzy logic-based TLCs can achieve better results than traditional ones. The FLC is designed for a 4-way traffic intersection: north, south, east and west, as shown in Figure 2.14.

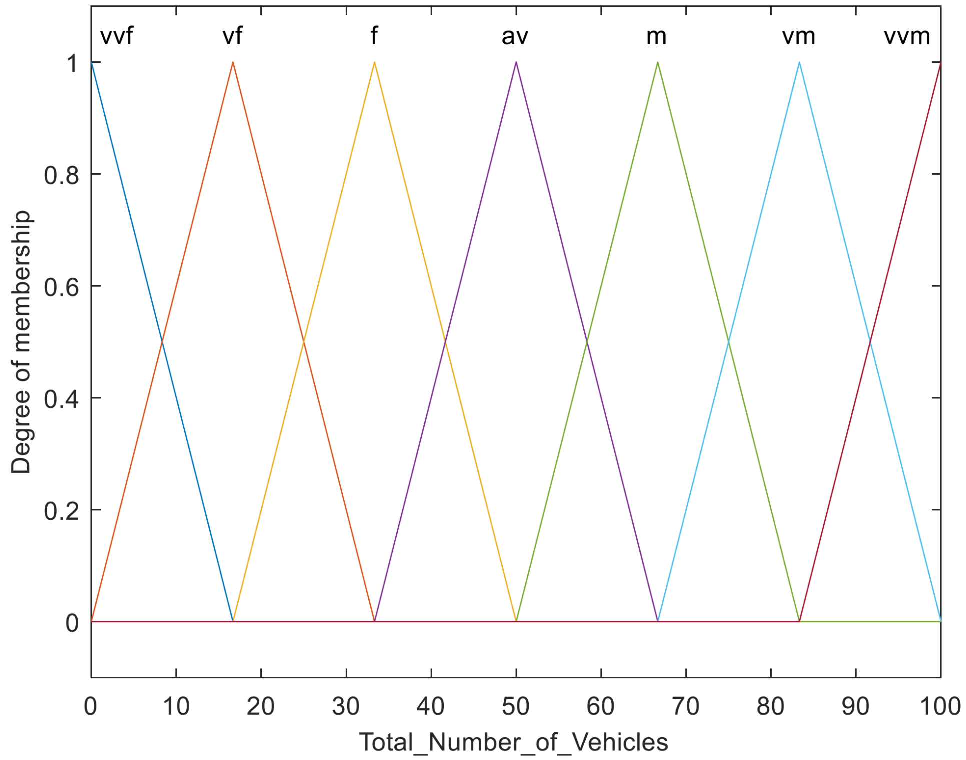


Figure 2.18 : Membership function of the total number of vehicles from input values for Strategy 1 and Strategy 2.

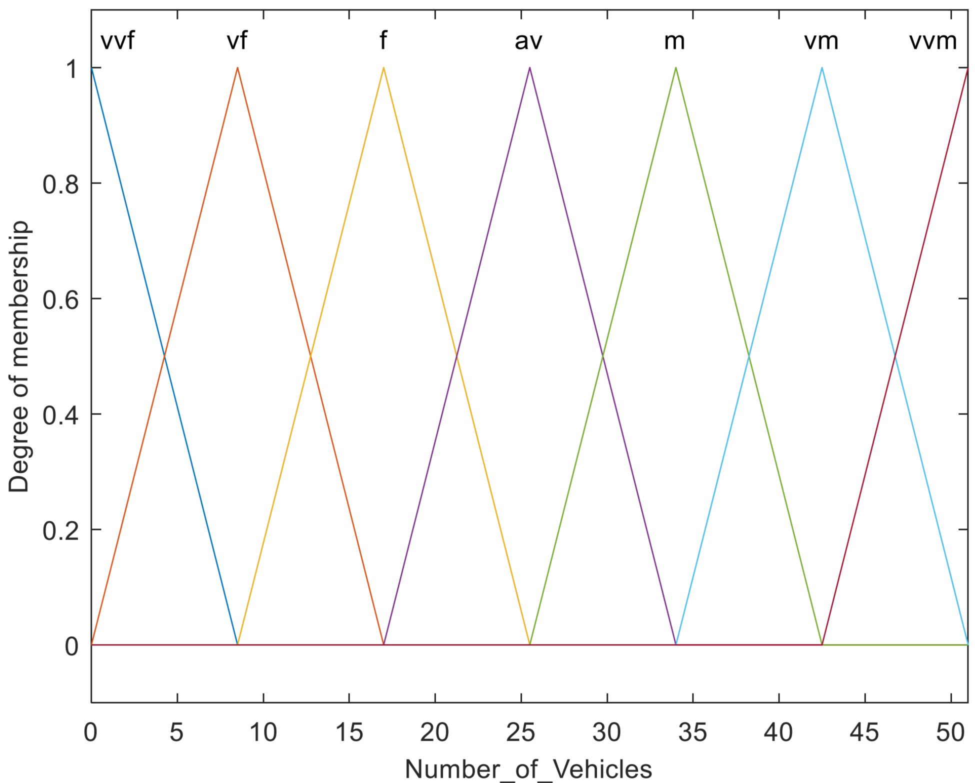


Figure 2.19 : Membership function of the number of vehicles on the busiest road from input values for Strategy 1.

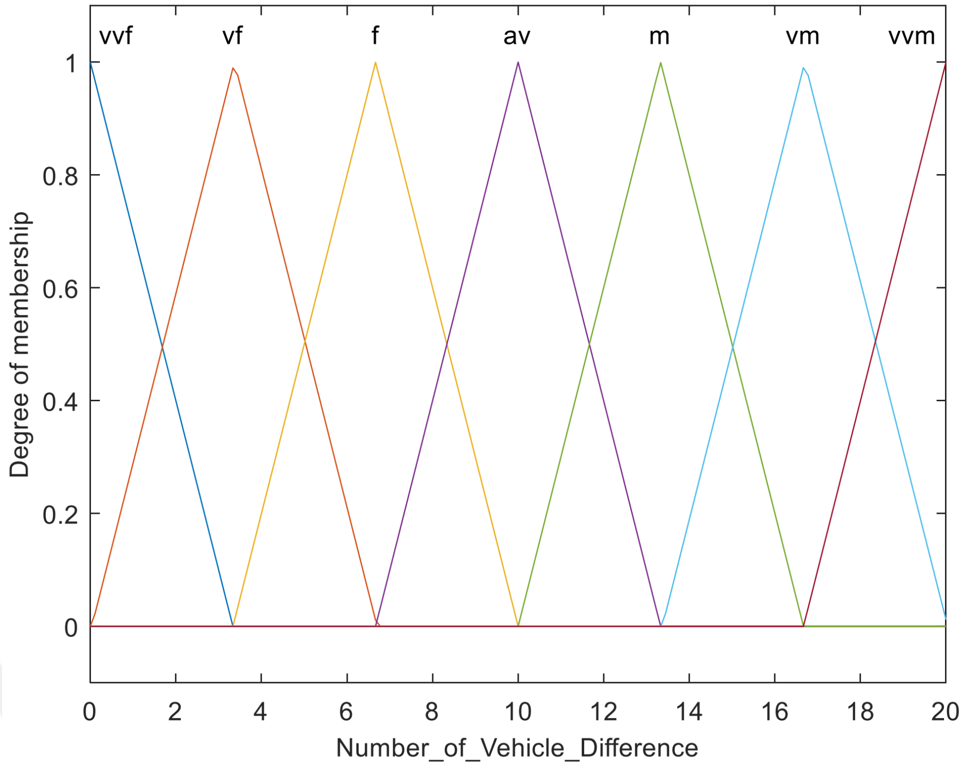


Figure 2.20 : Membership function of the number of vehicles difference from input values for strategy 2.

Similar FLCs are used for two different traffic light strategies for Strategy 1. Two inputs are used for the fuzzy controller of Strategy 1. Input variables for Strategy 1 are selected as The Total Number of Vehicles (TNV) at the intersection and the Number of Vehicles (NV) on the busiest road. The Total Number of Vehicles (TNV) is input for Strategy 2. Figure 2.18 shows the Total Number of Vehicles (TNV) input membership function graph for Strategy 1 and Strategy 2 because the same input membership function is used for both strategies. It includes 7 membership functions which are very very few (vvf), very few (vf), few (f), average (av), much (m), very much (vm) and very very much (vvm). Input variables for Strategy 2 are selected as The Total Number of Vehicles (TNV) at the intersection and the Number of Vehicles Difference (NVD) at the intersection. TNV input is the same as Strategy 1. NVD is the difference between the sum of the number of vehicles coming from the east and west and the number of vehicles coming from the north and south.

In the Mamdani-type fuzzy inference system, 49 rules are described and the weighted average defuzzification method is used. Based on the 49 fuzzy rules given in Table 2.8 and Table 2.9, an output is generated according to the current traffic conditions to determine the green light duration. The output of the FLC has also the same naming convention with the inputs of both strategies.

Table 2.8 : Rule table for fuzzy logic

TNV \ NV	vvf	vf	f	av	m	vm	vvm
vvf	vvf	vvf	vf	vf	f	f	av
vf	vvf	vf	vf	f	f	av	m
f	vf	vf	f	f	av	m	m
av	vf	f	f	av	m	m	vm
m	f	f	av	m	m	vm	vm
vm	f	av	m	m	vm	vm	vvm
vvm	av	m	m	vm	vm	vvm	vvm

The green phase is recalculated every second, so the green phase times change dynamically. For Strategy 1, when the busiest intersection changes and for Strategy 2, when the sign of the difference in the number of vehicles changes (from positive to negative or vice versa), the period is expected to be completed. In this case, the green phase will remain active for the last value before the signal change, then the green phase will be active for the other direction. The cycle continues in this way. A graphical representation of the membership functions of the output value is given in Figure 2.21.

Table 2.9 : Rule table for fuzzy logic

TNV \ VND	vvf	vf	f	av	m	vm	vvm
vvf	vvf	vvf	vf	vf	f	f	av
vf	vvf	vf	vf	f	f	av	m
f	vf	vf	f	f	av	m	m
av	vf	f	f	av	m	m	vm
m	f	f	av	m	m	vm	vm
vm	f	av	m	m	vm	vm	vvm
vvm	av	m	m	vm	vm	vvm	vvm

2.3.3 Simulation results

Simulation results are obtained using the SUMO program. CO_2 emission output and average velocity values are directly taken from the SUMO program. The amount of CO_2 emission is obtained from the CO_2 emission map which is given by SUMO program. This map depends on the vehicle velocity and vehicle acceleration. When

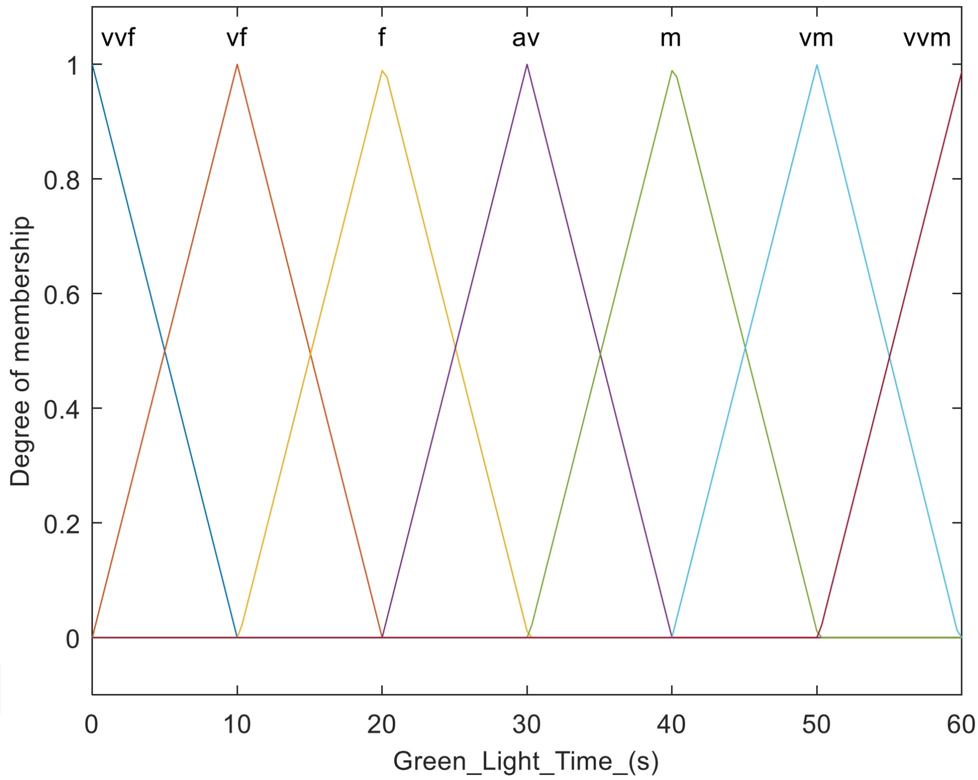


Figure 2.21 : Traffic light green time member function.

defining the problem for the intersection, the number of vehicles that turn right at the traffic intersection has less difficulty in traffic and their number is kept small. Simultaneous vehicles are produced during the simulation for 180 seconds. The numbers of vehicles coming from all directions are assumed to be equal to each other in the simulations. Besides, the number of vehicles directly crossing the junction and turning to the left are assumed to be equal to each other, while the number of vehicles turning to the right is about half. In the simulation, in order to determine vehicle density, half, one, one and a half and two vehicles per second are produced, and the results of the methods are compared according to the change in the vehicle density. Therefore, for 0.5, 1, 1.5 and 2 vehicle densities respectively, 90, 180, 270 and 360 vehicles are created during the simulation. Each vehicle crossed the intersection once. In the conventional method, the green light is steadily lit for 40 seconds for each phase. In fuzzy logic methods, traffic lights are calculated dynamically according to fuzzy logic input values with constraints of minimum of 6 seconds and a maximum of 60 seconds. Besides, the yellow light phase for 3 seconds after the green light phase is applied for both conventional (fixed time) and fuzzy Logic methods.

Table 2.10 : Simulation results

		Strategy 1		Strategy 2	
	Vehicle Density	Fixed Time	Fuzzy Control	Fixed Time	Fuzzy Control
CO_2 Emission (kg/s)	0.5	24.580	16.377	27.339	15.879
	1	55.036	40.378	58.076	40.942
	1.5	91.348	82.993	92.800	73.190
	2	136.567	133.786	134.797	125.136
Average Speed (m/s)	0.5	7.931	11.301	7.211	11.483
	1	6.750	9.405	6.752	8.590
	1.5	6.108	7.306	6.524	7.539
	2	6.108	6.232	6.003	6.052

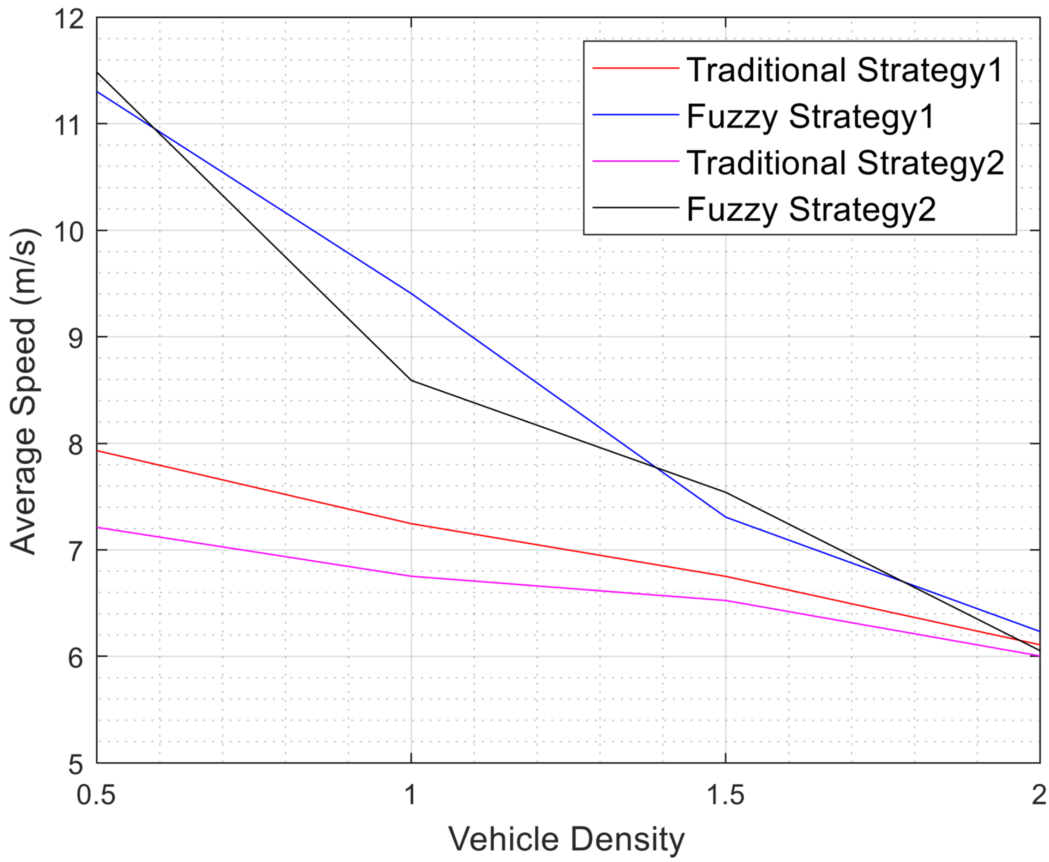


Figure 2.22 : Simulation results based on average speed.

Figure 2.22 shows the change in the average speed values of the vehicles in the simulation according to the number of vehicles produced per second. Figure 2.23 represents the change of total CO_2 emissions according to the number of vehicles produced per second up to the end of the simulation for a given condition. Table 2.10 shows the results of CO_2 emission and the average speed of vehicles. As it can be observed from the results, Strategy 2 gives better results in comparison to

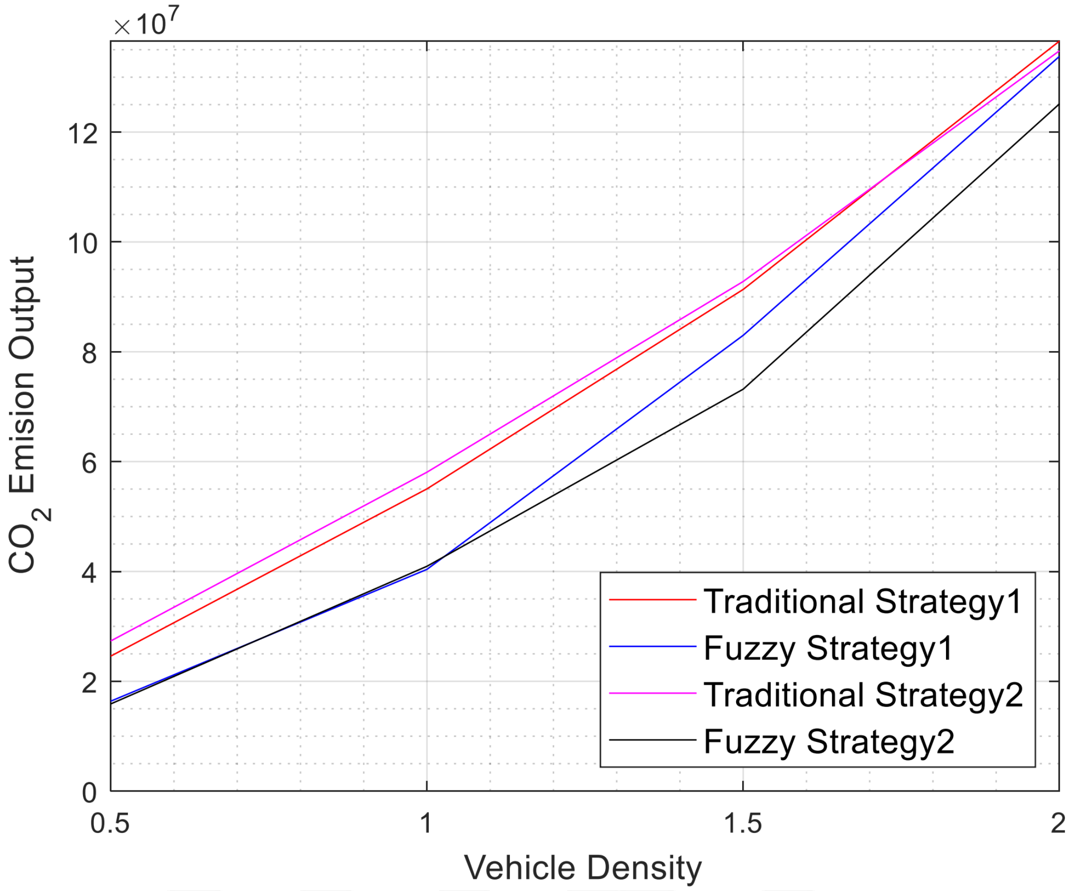


Figure 2.23 : Simulation results based on cumulative CO_2 emission.

Strategy 1. In Strategy 1, although the right turn passes are allowed for oncoming vehicles in a single phase, only the vehicles coming from one direction are allowed. In Strategy 2, however, mutual paths are permitted in a single phase. This enables Strategy 2 to give better results. The fuzzy logic method provides less CO_2 emission for each vehicle density and strategy against traditional control methods. The average speed is considerably high on fuzzy logic control methods, especially for lower vehicle densities. When the vehicle density increase, the average speed decrease but the fuzzy control method still provides better results.

As can be seen from the simulation results, the proposed method gives better results than the traditional methods. As can be seen in Figure 2.22 and Figure 2.23, Strategy 1 is better when the vehicle density is low, and Strategy 2 gives preferable results when vehicle density increases. Nevertheless, TLCs controlled by FLC give better results for both strategies in comparison to traditional light control.

2.4 Fuzzy Logic Control Strategies for Traffic Signal Timing Control with State Inputs

In this section, the Fuzzy Logic State Input (FLSI) controller and Fuzzy Logic Queue Length (FLQL) are used to determine the traffic light duration. The simulation was carried out using the Simulation of Urban Mobility (SUMO) software, an open-source, highly portable, microscopic road traffic simulation kit. Depending on waiting time and queue length, the results for the proposed types of Traffic Light Control Systems are compared. A traffic light system at a four-legged junction is controlled by a FLC with different input values which are queue length and state input. The recommended method is FLC with state input based on vehicle location. Results are compared for the proposed types of Traffic Light Control Systems depending on waiting time and queue length. The density of the vehicles coming from the east-west direction and the density of the vehicles coming from the north-south direction are assumed to be equal to each other. In addition, according to the traffic scenario considered, the number of vehicles going straight is higher than the number of vehicles turning left and right. Simulation has been carried out for the scenario in which the number of vehicles turning left, and right is almost equal to each other.

2.4.1 System overview

The simulation environment includes a 4-way intersection controlled by TLCs. For each leg, traffic lights are used. There are 4 different entrances and 4 different exits into the 4-way intersection considered. Moreover, all of the roads have 4 lanes. For each road, lanes closest to the turning direction were created for right and left turns as shown in Figure 2.24. Position information of the vehicles is obtained using SUMO. Lane and road which vehicles are on can be obtained via using this position information. Figure 2.24 shows the traffic intersection environment representation scheme. There are three main assumptions and limitations in TLCs simulation.

- The green light period should be bigger than 6 seconds for both directions.
- The yellow light period is only 4 seconds during transitions from red light to green light and from green light to red light.

- Turn is only possible for road with using lanes closest to the turning direction. Also, the vehicle travelling from west to east only uses the middle two lanes.

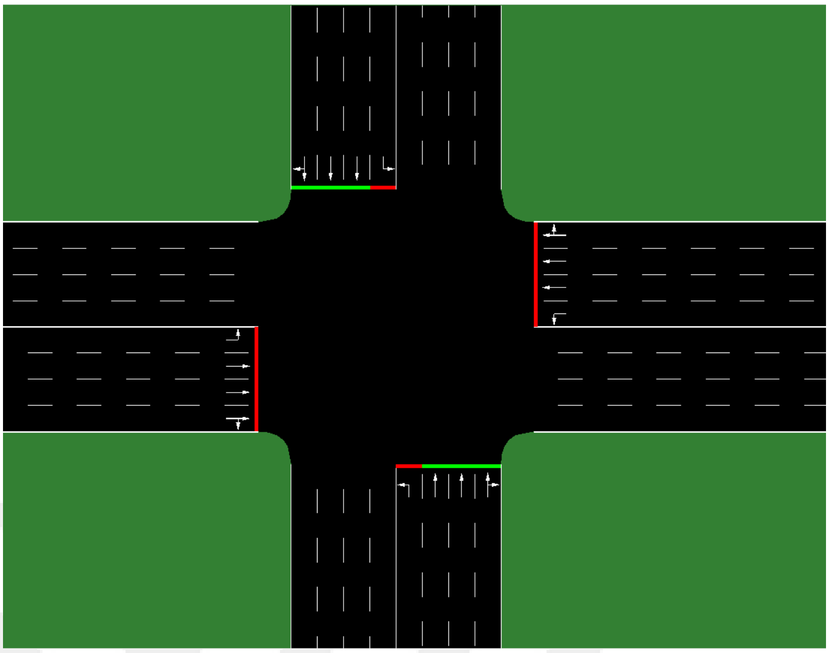


Figure 2.24 : TLC simulation environment representation scheme.

Table 2.11 : Pass permission table

	North	South	East	West
Phase 1	Straight, Right	Straight,Right	-	-
Phase 2	Left	Left	-	-
Phase 3	-	-	Straight,Right	Straight,Right
Phase 4	-	-	Left	Left

2.4.2 Traffic light controller design

FLQL and FLSI methods are used to determine traffic light cycle time. The aim of the FLQL and FLSI is to decrease the waiting time and queue length. The basis of the FLQL and FLSI is the FLC. FLC makes it possible to implement rules similar to the way people think in real life. The FLC is capable of applying these laws efficiently. The advantage of FLC is that a machine can comprehend and use fuzzy concepts such as "small," "average" or "big". FLC is very useful for traffic light control when traffic data is used appropriately. For example, the effect of determining the traffic lights of 3 vehicles in the same phase and lined up in 3 lanes is the same as for a single vehicle. As a matter of fact, the time for 3 vehicles and a single vehicle to pass through the traffic junction is equal to each other. Considering this, the number of vehicles at the traffic intersection is not selected as the fuzzy logic input value. The number of vehicles may be appropriate as the fuzzy logic input value for a single-lane road but not for a multi-lane road. Instead, the traffic queue length and traffic junction status values were selected as FLC input values.

2.4.2.1 Fuzzy logic controller with queue length input

The input values of the FLQL traffic light controller consist of the queue length values of the vehicles at the traffic intersection. In this method, vehicle queue length values are taken as FLC input. FLC has two antecedents and one consequent. Antecedents for FLC of FLQL are chosen as the queue length in green phase and the queue length in red phase at the intersection. The queue length input in the green phase represents the length of the queue in lanes that can cross the traffic junction in the simulation. For instance, when the green phase is for vehicles coming from the north and south directions, the queue length input in the green phase value is the sum of the longest queue lengths in the north and south directions (excluding the leftmost lane). The queue length input at the red phase input is the sum of the longest queue lengths in the red phase at the traffic junction in the simulation.

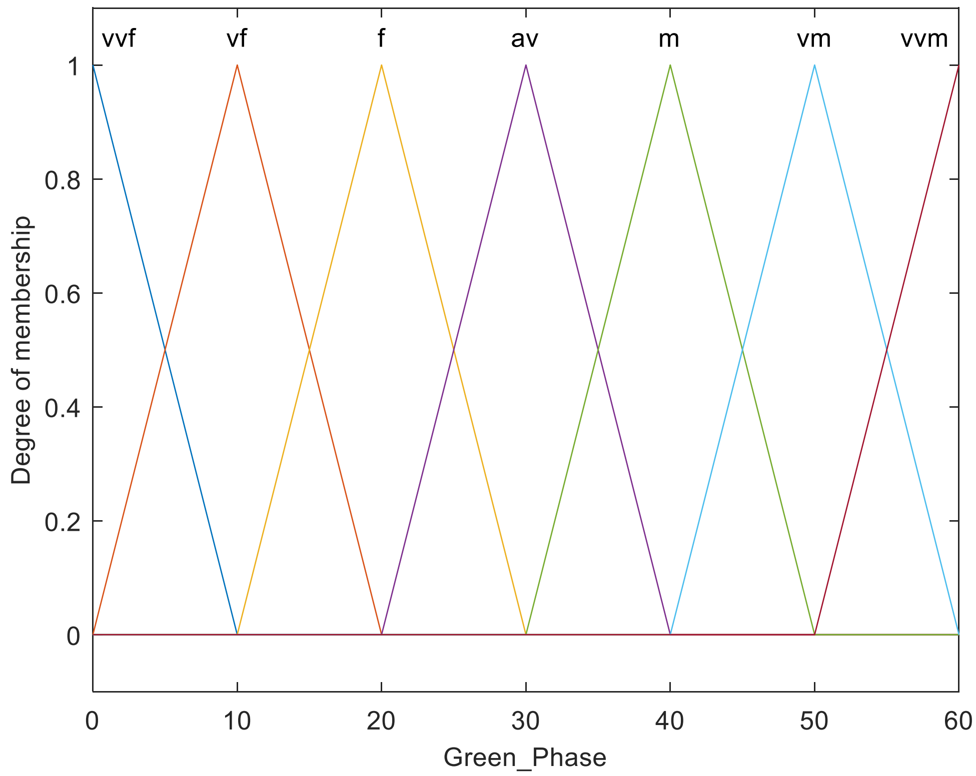


Figure 2.25 : Membership function of queue length in GP input for FLQL .

The green phase duration, which is the FLC output value, is calculated according to the FLC input values. Membership functions of the number of vehicles in green phase (GP) and number of vehicles in red phase (RP) inputs for FLQL are shown in Figure 2.25 and Figure 2.26. A graphical representation of the membership functions of the output value is given in Figure 2.27.

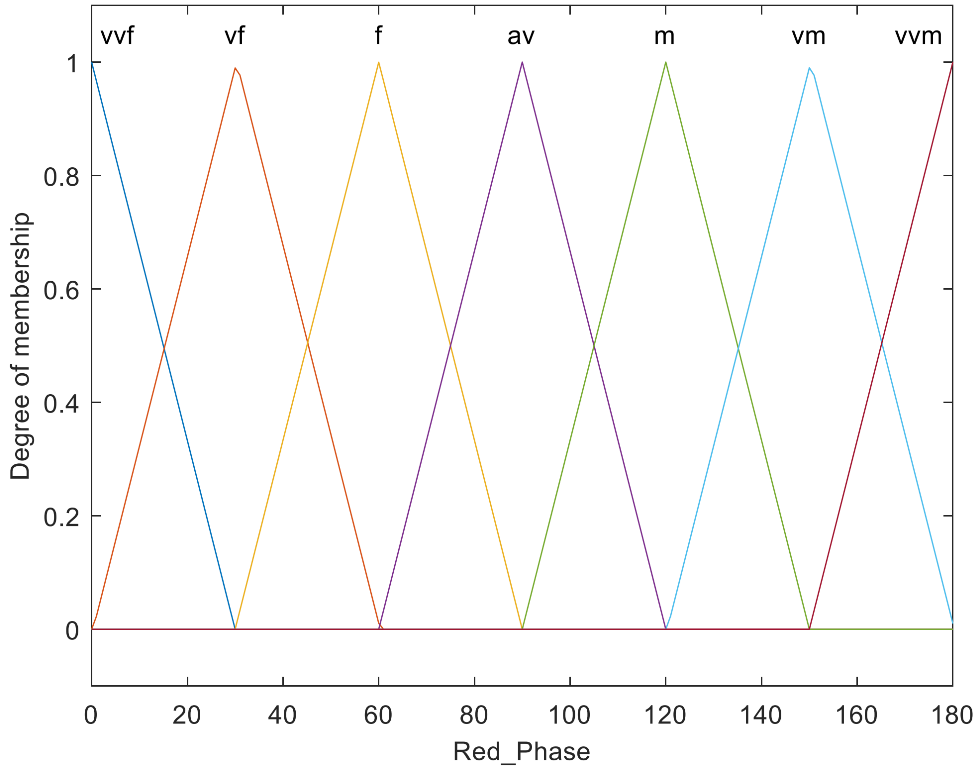


Figure 2.26 : Membership function of queue length in RP input for FLQL.

There are seven membership functions in the fuzzy controller. They are very very few (vvf), very few (vf), few (f), average (av), much (m), very much (vm) and very very much (vvm). The fuzzy controller generates an output based on current traffic conditions to decide the green light length, using the fuzzy rules mentioned in Table 2.12. In the Mamdani-type fuzzy inference scheme, 49 rules are specified, and the weighted average method of defuzzification is used. To determine the duration of the green light, an output is produced based on the current traffic conditions using the 49 fuzzy rules mentioned in Table 2.12.

2.4.2.2 Fuzzy Logic traffic signal timing with states input

The input values of the FLSI traffic light controller consist of status values based on the position of the vehicles. The state of the traffic intersection defines a representation

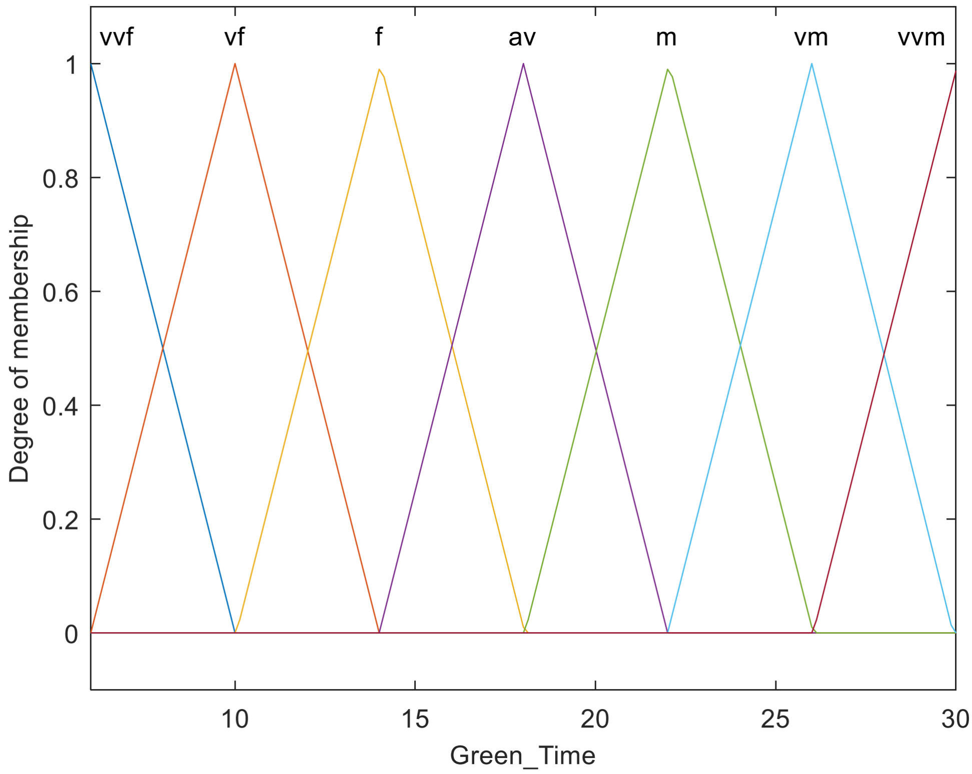


Figure 2.27 : Traffic light green time member function for FLC.

Table 2.12 : Rule table for FLQL and FLSI

GP \ RP	vvf	vf	f	av	m	vm	vvm
vvf	av	f	f	vf	vf	vvf	vvf
vf	m	av	f	f	vf	vf	vvf
f	m	m	av	f	f	vf	vf
av	vm	m	m	av	f	f	vf
m	vm	vm	m	m	av	f	f
vm	vvm	vm	vm	m	m	av	f
vvm	vvm	vvm	vm	vm	m	m	av

of the state of the environment in a given time period t and is denoted by st. In order to optimize traffic, the state must provide sufficient information on the distribution of vehicles on each road. The purpose of this presentation is to enable the controller to know the position of the vehicles in the environment in a timely manner.

Each arm of the intersection, the incoming lanes, was parsed into cells with specific dimensions that could describe the presence or absence of a vehicle in them. The three lanes dedicated to going straight and turning right have the same traffic phase. Thus, there is no need to separate them. However, there is a separate group of cells in the lane devoted to the left turn. As seen in Figure 2.28, there are 10 cells along each

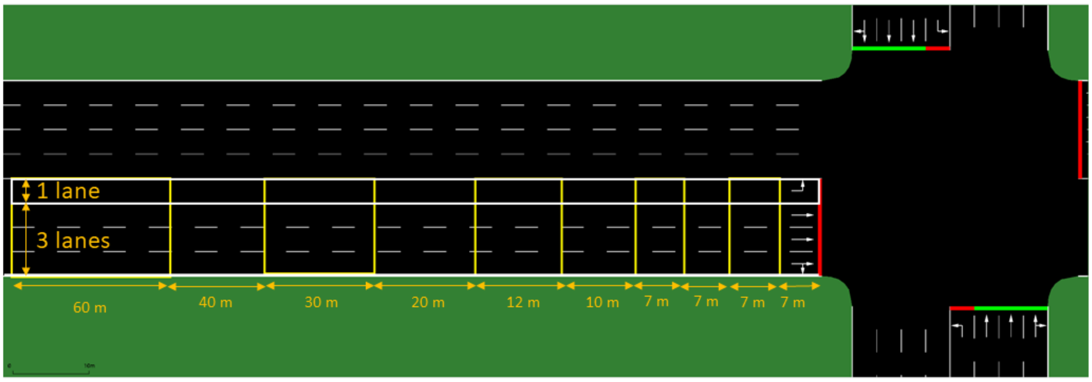


Figure 2.28 : Separation of a leg of a traffic junction into cells.

lane. Since 3 lanes using the same phase are not separated, only the leftmost lane is separated, so there are a total of 20 cells in one arm. There are 80 cells in total at our 4-leg traffic junction.

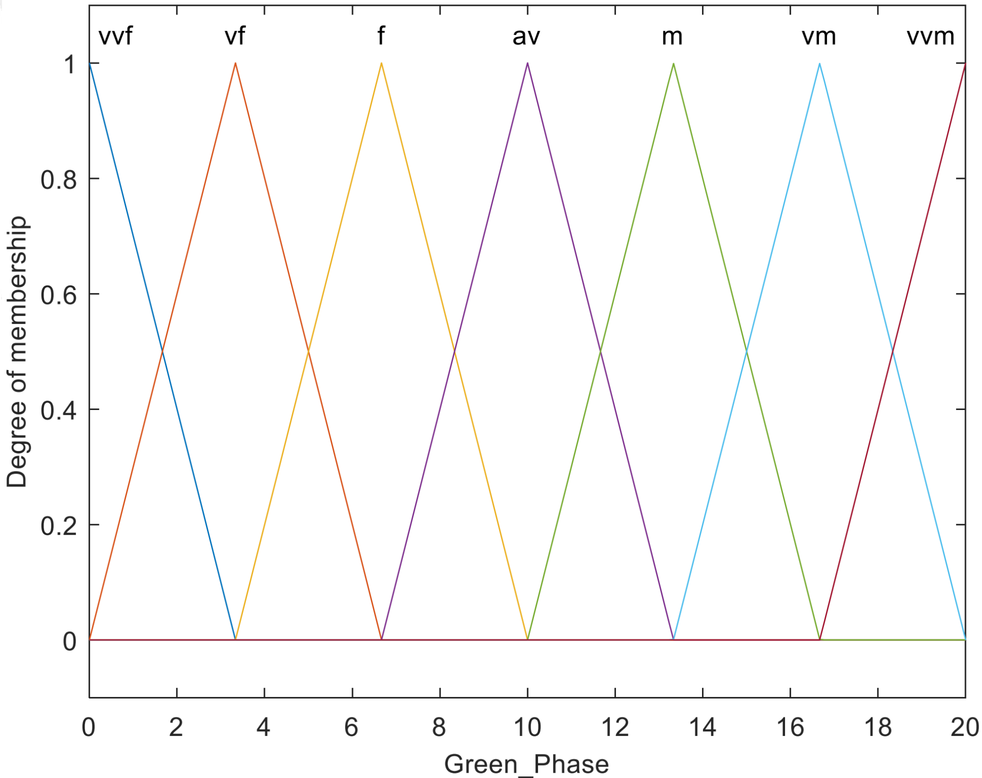


Figure 2.29 : Membership function of state input in GP input for FLSI.

The mathematical model of state space, *LSD* (Lane Space Discretization), is calculated according to equation 2.21.

$$LSD_{l,k} = c_{l,k} \tag{2.21}$$

c_{lk} is the k^{th} Cell of the l^{th} lane. Vector LSD, "$LSD_{k,l} = 1$ if there is more than one vehicle in ck, otherwise $LSD_{k,l} = 0$." It depends on the rule. then the states in each lane are summed up separately and used as the FLC input value.

$$\sum_{k=1}^{10} = c_{l,k} \tag{2.22}$$

There are two fuzzy logic input values. first, the sum of the state values of the roads that are allowed to pass is GP (Gren Phase). The second is RP (Red Phase), the sum of the state values of the roads that are not allowed to pass. Membership functions of GP and RP inputs for FLSI are shown in Figure 2.29 and Figure 2.30. A graphical representation of the membership functions of the output value is given in Figure 2.27. Based on the fuzzy rules as given in Table 2.12, the fuzzy controller produces an output according to current traffic conditions to determine the green light duration.

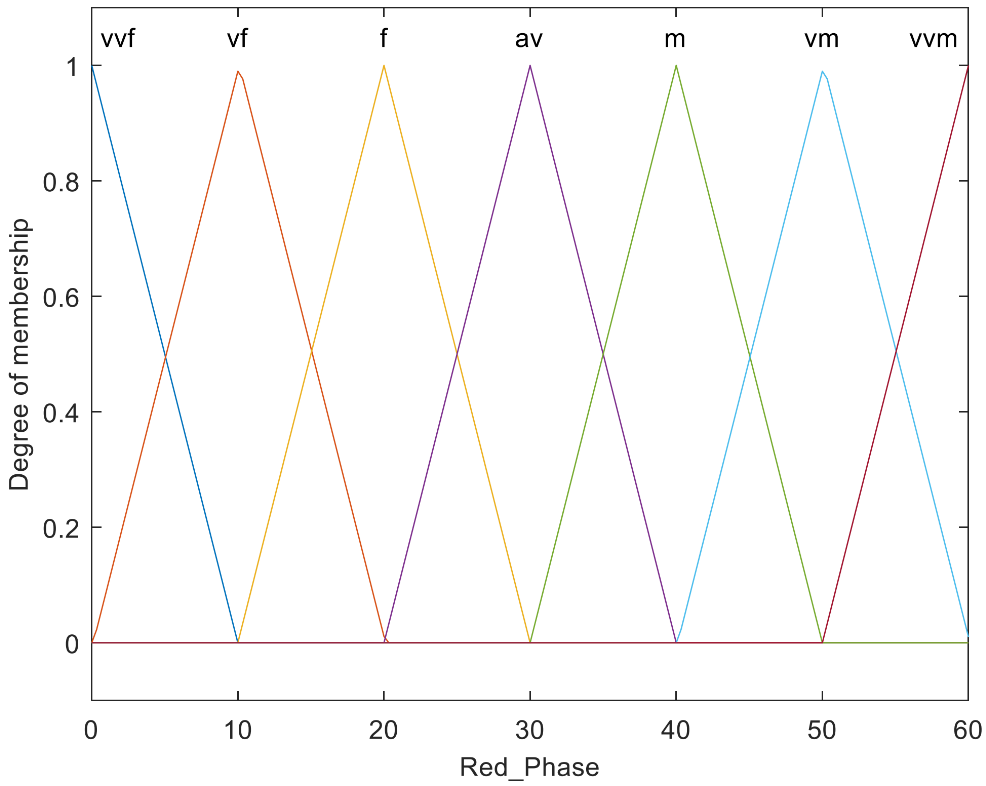


Figure 2.30 : Membership function of state input in RP input for FLSI.

2.4.3 Simulation results

A simulation environment is made using the SUMO program, and the simulation is run through this program. Vehicles are produced simultaneously for 5400 seconds during the simulation. Different scenarios are simulated according to the number of 2000 and

3000 vehicles produced. Moreover, Each vehicle passes through the traffic intersection once. In addition, according to the scenario designed, the number of vehicles coming from the east-west direction is almost equal to the number of vehicles coming from the north-south direction. Also, in the simulation scenario, the number of vehicles driving straight is higher than the number of vehicles turning left or right.

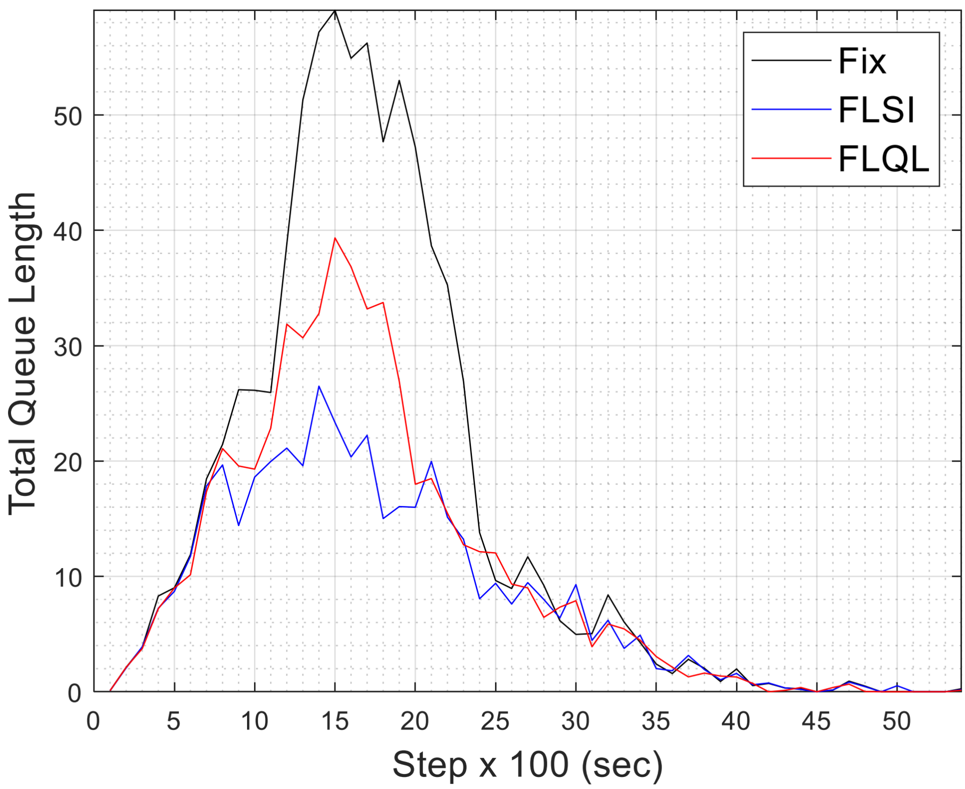


Figure 2.31 : Simulation results based on total queue length (2000 vehicles).

Each vehicle produced during the simulation is five meters, and its entry speed into the simulation is 36 km/h. Acceleration and deceleration rates of all vehicles are $1m/s^2$ and $4.5m/s^2$, respectively. And the vehicle's maximum speed value is 90 km/h.

Figure 2.31 and Figure 2.33 show the total queue length variation values for different control techniques. Figure 2.32 and Figure 2.34 show the total waiting time variation values for different control techniques. It can be said that the proposed FLSI traffic light controller gives better results than conventional traffic light controllers.

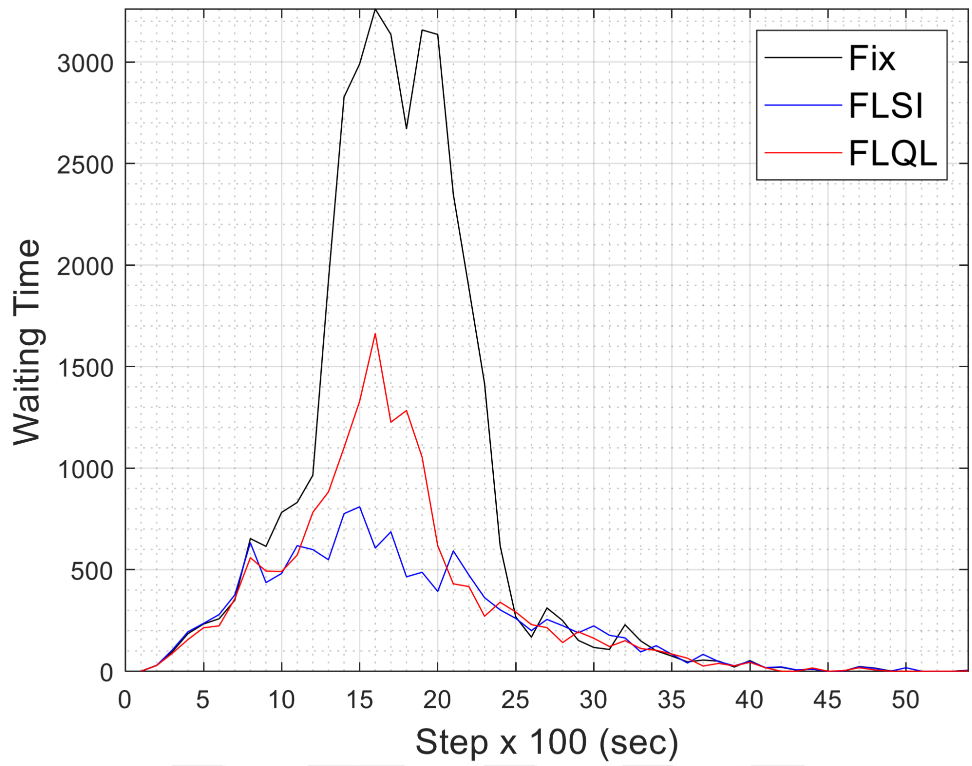


Figure 2.32 : Simulation results based on waiting time (2000 vehicles).

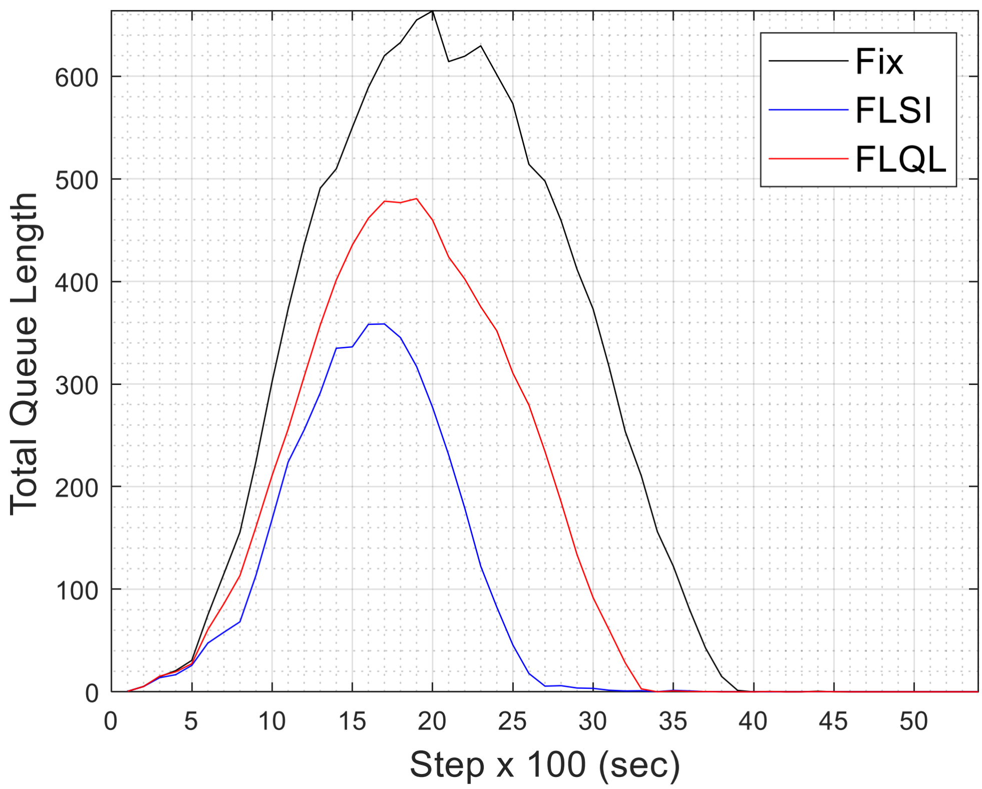


Figure 2.33 : Simulation results based on total queue length (3000 vehicles).

The effectiveness of the proposed method is seen in both the queue length and the total waiting times change. It is seen that the proposed method gives much better results, especially in the time periods (1000-2000 seconds intervals) when the vehicle density increases at the traffic intersection. In Figure 2.31 to Figure 2.34, the x-axis represents the average value of each 100 samples or steps. There are 5400 steps because of that x-axes are split into 54 points.

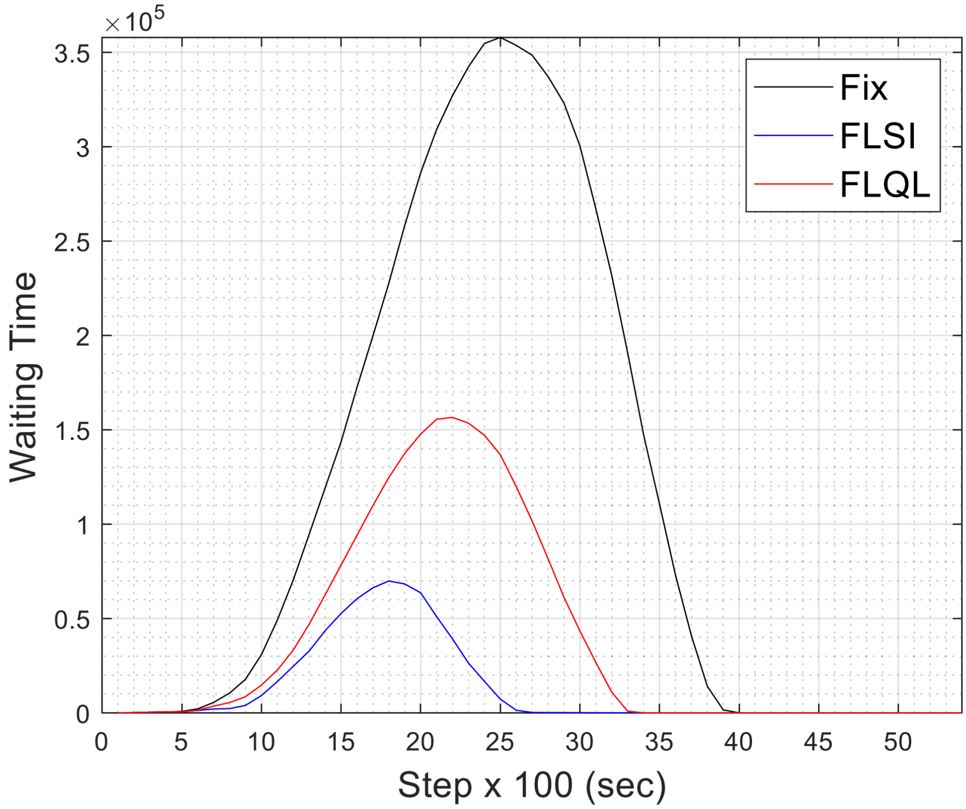


Figure 2.34 : Simulation results based on waiting time (3000 vehicles).

The proposed approach FLSI gives the best results as shown in the simulation results. It was also observed that the fuzzy logic controller, which has the input value according to the queue length at the traffic intersection, which is the other proposed method, gives better results than the fixed-time traffic light control. It is predicted that much better results can be obtained when these methods are used together with various learning algorithms. As a matter of fact, no control or learning algorithm was used and the phase sequence changes in a fixed cycle. A method can be developed to determine the phase direction, and better results can be obtained.

2.5 Traffic Light Control for Multi Intersection Model in Istanbul/Altunizade

The growing population and limited road capacities of the metropolises, Istanbul, New York and Hong Kong, lead to increased traffic congestion leading to traffic queues and accidents, a serious urban management problem. In this section, using the Simulation of Urban MObility (SUMO) program, a multi-traffic intersection simulation environment is designed by gathering data from a crucial real-life region in Istanbul. While obtaining data on real-world regions, OpenStreetMap is used. The 4-leg traffic intersection model and vehicles are built using the SUMO traffic simulation program. it is suggested to use the traffic control method in two different control methods apart from the traditional method for multiple intersections in a real-world region. Using FLC and actuated control methods, traffic light controllers are designed for multiple intersections. The results are compared using the data obtained from the designed traffic light control methods and traditional traffic light control methods. In this study, control of multiple traffic lights in an area is considered. Although a fuzzy control for each phase is proposed by considering the number of vehicles waiting at the red light and the number of vehicles passing through at the green light, the control of the sequences of the phases is not considered.

2.5.1 System overview

Altunizade is one of the most important transportation points of the Anatolian side, between the roads leading to Kadıköy and Üsküdar. Altunizade region, which is shown in Figure 2.35, is a central, multi-junction and traffic control area that is affected during rush hours. As can be seen in Figure 2.35, this region is exposed to intense and unresolved traffic problems, especially during rush hour. Figure 2.35 shows the traffic density of the region at 6:35 PM on a typical Friday, using Google Maps data.

In this study, the Altunizade region is simulated. Since a case study from the real world is used, many problems need to be overcome. First of all, getting roads and traffic light data correctly is a problem. This problem is overcome by scanning multiple sources with applications such as Google Maps, Yandex Maps and OpenStreetMap(OSM). Applying a real-world control method in simulation takes it one step further in terms of applicability. For this reason, the applicability of the Altunizade region, which has a multi-dimensional structure and has a critical position for Istanbul traffic flow, is essential. Especially with the online shopping brought about by the pandemic period, the increase in motor and vehicle courier services and the increase in the use of individual vehicles caused by avoiding public transportation negatively affected Istanbul traffic. Therefore, this study is also relevant to today's global problem, the pandemic. Simulations are done with the assumption that the vehicle's location information is received correctly via GPS.

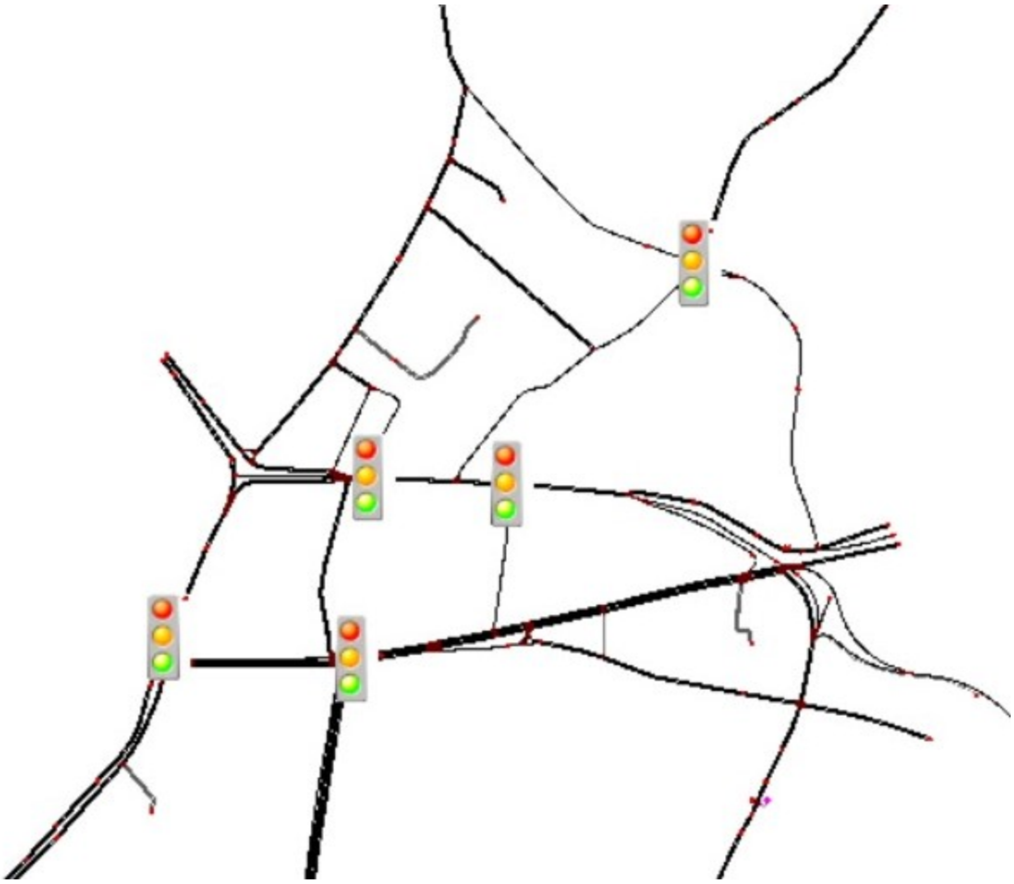


Figure 2.35 : Traffic density map and display of traffic lights.

The region implemented is taken as a file from OpenStreetMap(OSM) application. While the data about vulnerable users (pedestrians, bicycles, etc.) are transferred with the OSM file, the side roads such as pedestrians and bicycles are removed and the vehicle simulation is focused. The simulations are performed for 7200 seconds (2 hours) by generating 6000 vehicles. A total of 5 traffic lights, which are named as TL_i for each intersection as shown in Figure 2.36, are considered in the area used in the simulation. In Table 1, five traffic lights and phase numbers from TL_1 to TL_5 and their views for each traffic light are shown. For example, there are 3 phases for TL_1 and these are expressed as $TL_{1,1}$ for the first phase, $TL_{1,2}$for the second phase and $TL_{1,3}$ for the third phase. One phase duration is 18 seconds. Yellow light duration for a phase is 3 seconds. The total cycle time according to this connector is also shown in Table I. Accordingly, total cycle times are shown in Table 2.12. These values are used to simulate the fixed-time method.

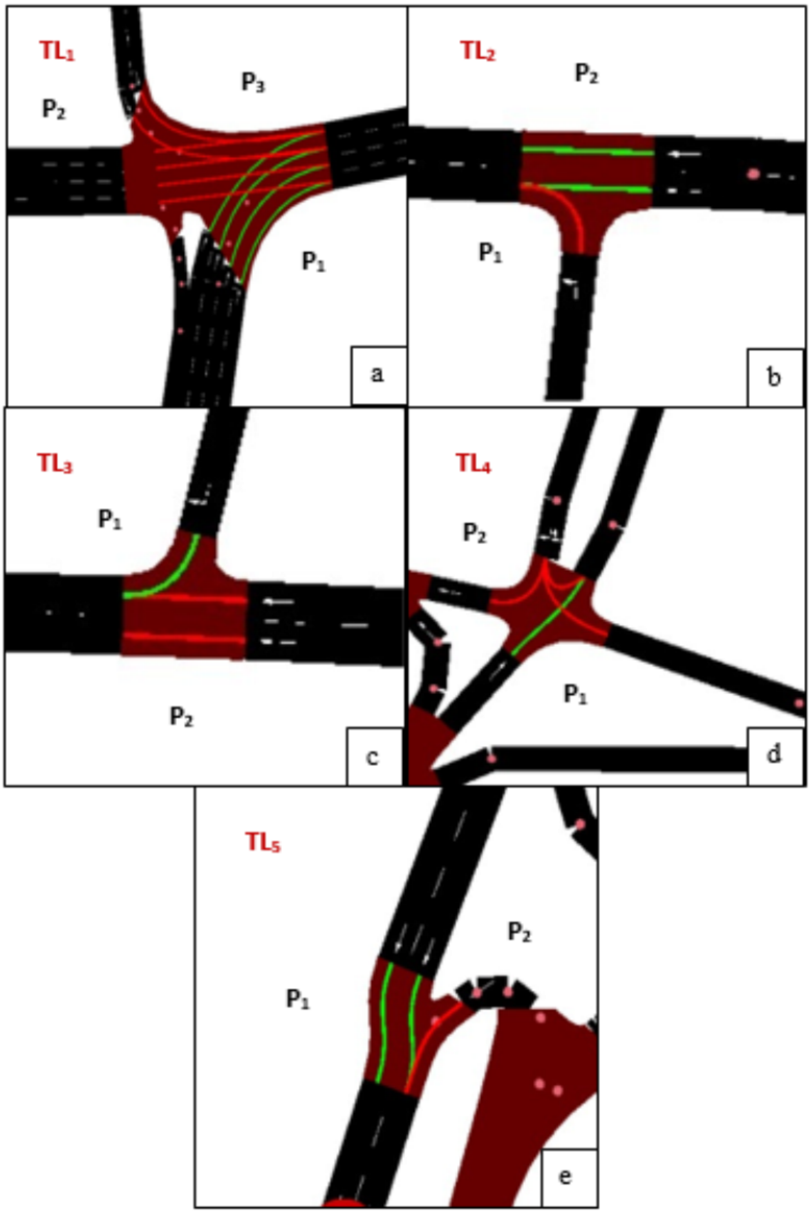


Figure 2.36 : Detailed presentation of traffic lights, intersections and phases.

2.5.2 Traffic light control system

For both traffic control methods, the minimum green light duration for each traffic light from TL_1 to TL_5 was determined as 6 seconds and the maximum green light

Table 2.13 : Phase, cycle, minimum and maximum green light times

Traffic Lights	Phases	Cycle	Min Green Light	Max Green Light
	$TL_{1,1} = 18sec$	63 sec	6 sec	30 sec
TL_1	$TL_{1,2} = 18sec$	63 sec	6 sec	30 sec
	$TL_{1,3} = 18sec$	63 sec	6 sec	30 sec
	$TL_{2,1} = 18sec$	42 sec	6 sec	30 sec
TL_2	$TL_{2,2} = 18sec$	42 sec	6 sec	30 sec
	$TL_{2,3} = 18sec$	42 sec	6 sec	30 sec
	$TL_{3,1} = 18sec$	42 sec	6 sec	30 sec
TL_3	$TL_{3,2} = 18sec$	42 sec	6 sec	30 sec
	$TL_{3,3} = 18sec$	42 sec	6 sec	30 sec
	$TL_{4,1} = 18sec$	42 sec	6 sec	30 sec
TL_4	$TL_{4,2} = 18sec$	42 sec	6 sec	30 sec
	$TL_{4,3} = 18sec$	42 sec	6 sec	30 sec
	$TL_{5,1} = 18sec$	42 sec	6 sec	30 sec
TL_5	$TL_{5,2} = 18sec$	42 sec	6 sec	30 sec
	$TL_{5,3} = 18sec$	42 sec	6 sec	30 sec

duration as 30 seconds, and this is shown in Table 2.12. For the implementation of the two traffic control methods, the input values are the number of vehicles waiting on different roads at intersections. Data is taken from SUMO's TRACI library to find the number of vehicles on the roads. TRACI allows the values of simulated objects to be retrieved and their behaviour changed instantly. For TL_1, TL_2 and TL_3 intersections, vehicles from 70 meters away from all roads connected to the intersections began to be listed. Since the actual path lengths for TL_4 and TL_5 do not allow detection from such a long distance; The number of vehicles is found by detecting the vehicles from a distance of 30 meters to the TL_4 intersection and from TL_5 to 58 meters.

2.5.2.1 Actuated traffic light control

The actuated method is a primitive method like a fixed-phase traffic light and does not contain any control method or learning algorithm. However, there are some improvements over fixed-phase traffic light signalization. A traffic light managed with the actuated method receives vehicle information data from the roads approaching to the intersection at a certain distance from the intersection and may decide to extend the green time for the stage related to certain periods. The actuated method algorithm works as follows. If the minimum green phase time is over and there is no information about an approaching vehicle, the green phase is finished. If the minimum green light time is about to finish and there is information about a vehicle approaching to the intersection and the arrival time is less than the end of the green phase, add additional time to the green light duration. This algorithm ensures that vehicles travelling along the road do not reduce their speed when they are not needed. Although the Actuated method is a better method than fixed-time traffic light signalling, it is not a sufficient method for complex traffic system dynamics. In the actuated traffic control system, some disadvantages in real applications can be seen because of the structure of the system and increased delays of vehicles. Since this method makes phase regulation with instantaneous triggers, drivers cannot be given green light duration information in advance. Another problem may occur due to hardware malfunctions. Since the actuated traffic control method is not a robust control method, minor problems such as communication delays and measurement errors may cause big problems for signalling regulation.

Algorithm 1 Actuated Traffic Light Control

```
Initialization: min green light, max green light,
                additional green light, traffic light = phase1
Variables: count, add, step, traffic light,is there vehicle
Set: count=0, add=0, step=1 second,
 while (count < Green Light + add) do
    traffic light = green
     count = count + step time
    if there is vehicle approaching
     intersection is there vehicle = True then
    if count < max green light then
        if (count - min green light) % add = 0
         and is there vehicle then
            add = add + additional green light
    end
traffic light = phase2
```

2.5.2.2 Fuzzy logic traffic light control

Two values are used as fuzzy Logic input parameters in the traffic control process with FLC. Both of these consist of the number of vehicles within a certain distance from the intersection. The first parameter is the number of vehicles close to the intersection in the phase that is lit green at the current step; The second is the number of vehicles that are lit in the red phase at the current step at the same intersection. Vehicle numbers are taken as inputs, with the input values being 0 minimum and 20 vehicles maximum.

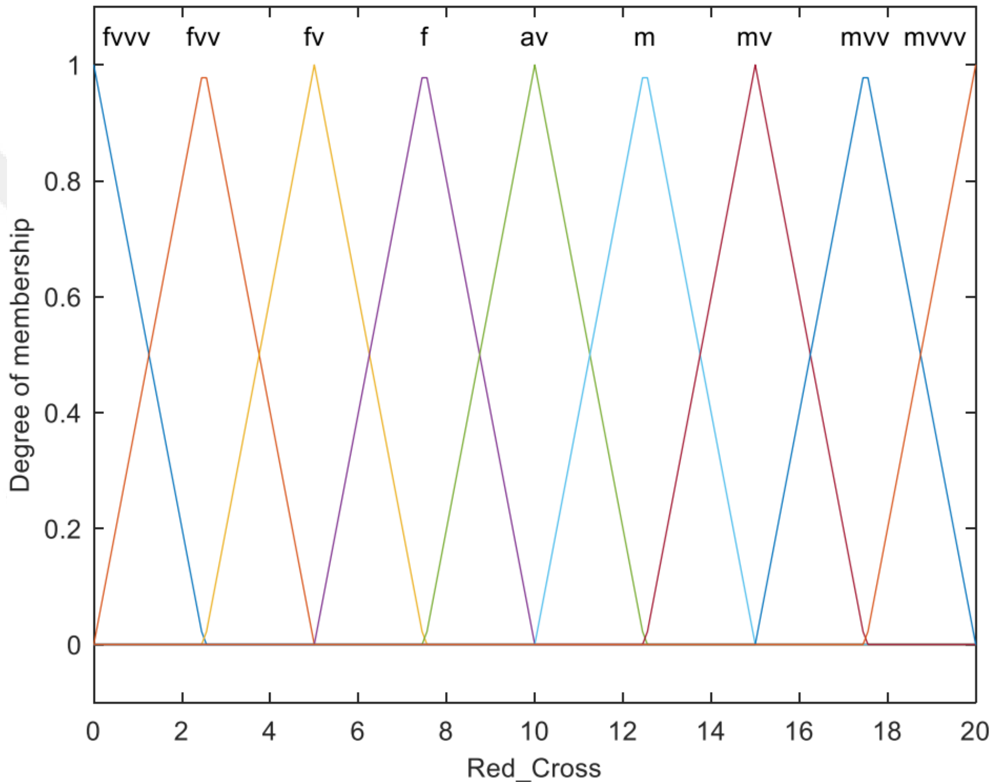


Figure 2.37 : Membership function of the total number of vehicles in red phase.

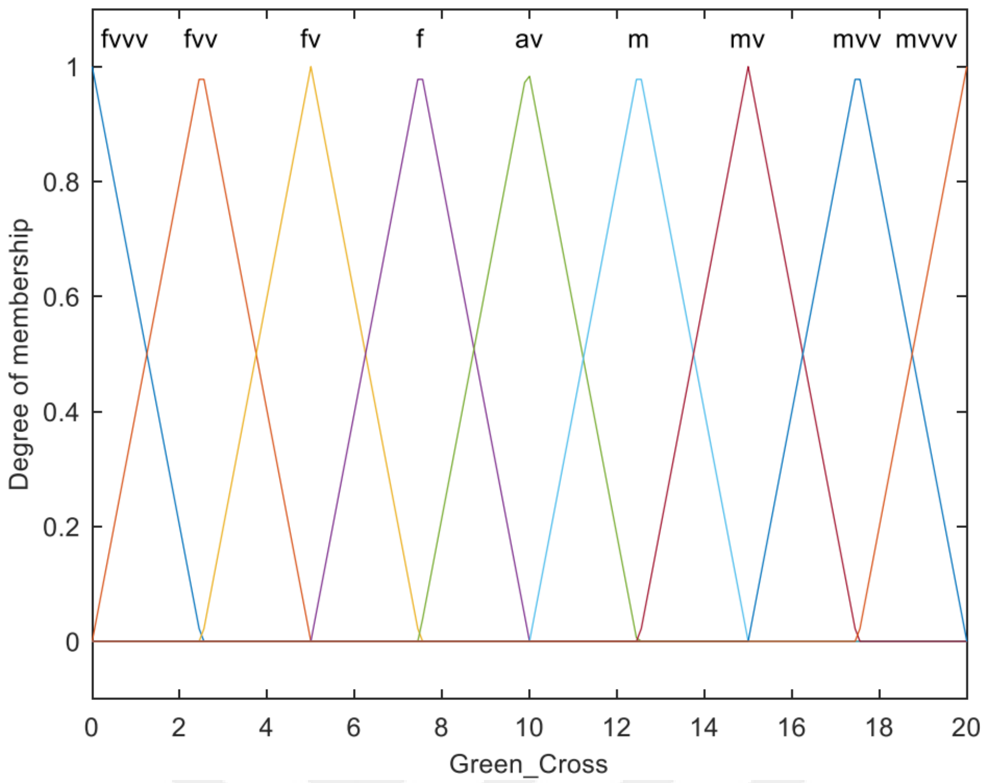


Figure 2.38 : Membership function of total number of vehicles in green phase.

Input values from 0 to 20 are divided into 100 parts with linear space, with fuzzy logic these parts are transferred to 9 membership functions according to the number of vehicles, these are: ('vvvf', 'vvf', 'vf', 'f', 'av', 'm', 'vm', 'vvm', 'vvvm'). While the "vvvf" function shows that the number of vehicles is the least for that phase; the highest function is 'vvvm'. These membership functions are shown in Figure 2.37 - Figure 2.39 as inputs and output. FLC rules are shown in Table 2.14. Moreover, the duration of the green light should be determined according to these input values, and the green light should be applied dynamically to each road of each intersection, respectively. The output value (green light duration) is transferred to the 9 membership function as a linear space for a minimum of 6 seconds and a maximum of 30 seconds. For the 9 membership function, 9 fuzzy rules are determined, as seen in Table 2.14, the membership function of the number of vehicles in the green phase (GP) and the membership function of the number of vehicles in the red phase (RP) with the rules determine an output value membership function. The membership function of the output value is converted into a value between 6-30 given in linear space. Thus, according to the number of vehicles passing in the green phase and the vehicles waiting in red, it is determined how much green light time will be given to the current phase. Figure 2.40 shows the detailed representation of the fuzzy logic TLC method diagram.

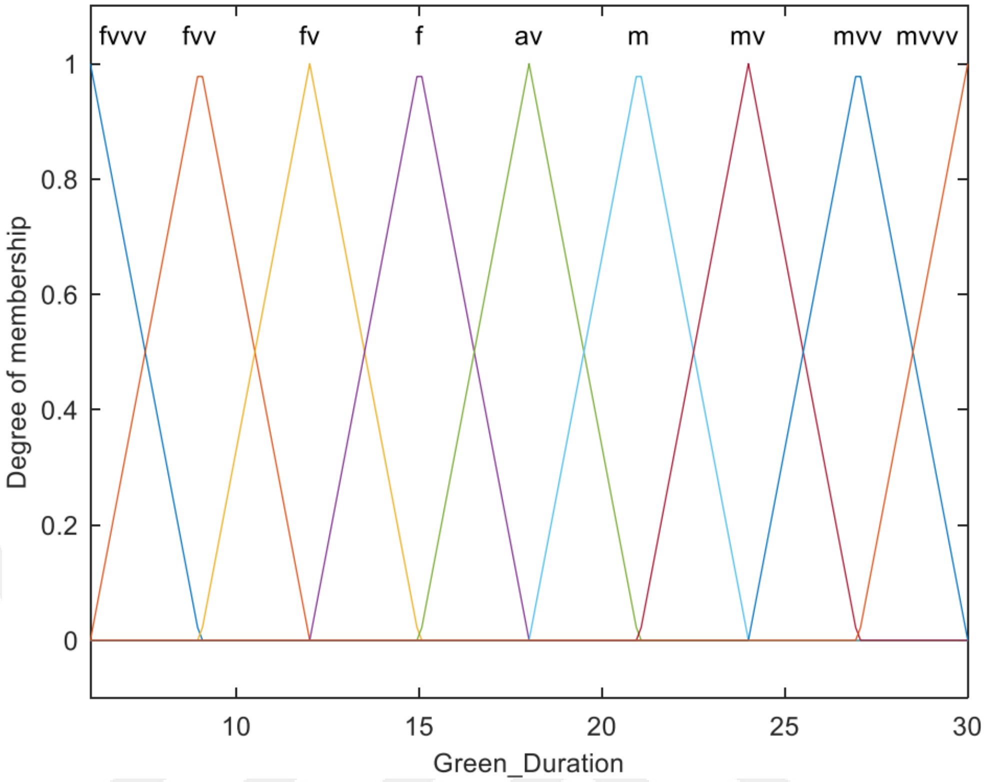


Figure 2.39 : Membership function of green time duration.

For all five intersections, green light durations are applied to the intersections by taking the fuzzy output dynamically. Furthermore, When the number of lanes at intersections is different, for example, if there is a single lane on one road and double lanes in the other, calculations were made by reducing or increasing the number of lanes to two lanes on all roads. A single fuzzy set is used in the study, and vehicle input information at all 5 intersections is processed from the same fuzzy set. However, since the roads at the intersections have different lane numbers, lane-based manipulations are to be made when determining the fuzzy inputs. When vehicles come within 70 meters of the intersection, they start to be included in the fuzzy input list, so there could be a maximum of 10 vehicle input values for a single lane. The number of input vehicles is set as min 0 and max 20 in the fuzzy set, so if there are 10 vehicles each in two lanes, the input value reaches the maximum. However, the number of vehicles on a 4-lane road may reach up to 40, and it becomes meaningless in the fuzzy set; Moreover, the number of vehicles on a single-lane road would be a maximum of 10, since the intersections on one road at the intersection could not reach a maximum of 20 inputs.

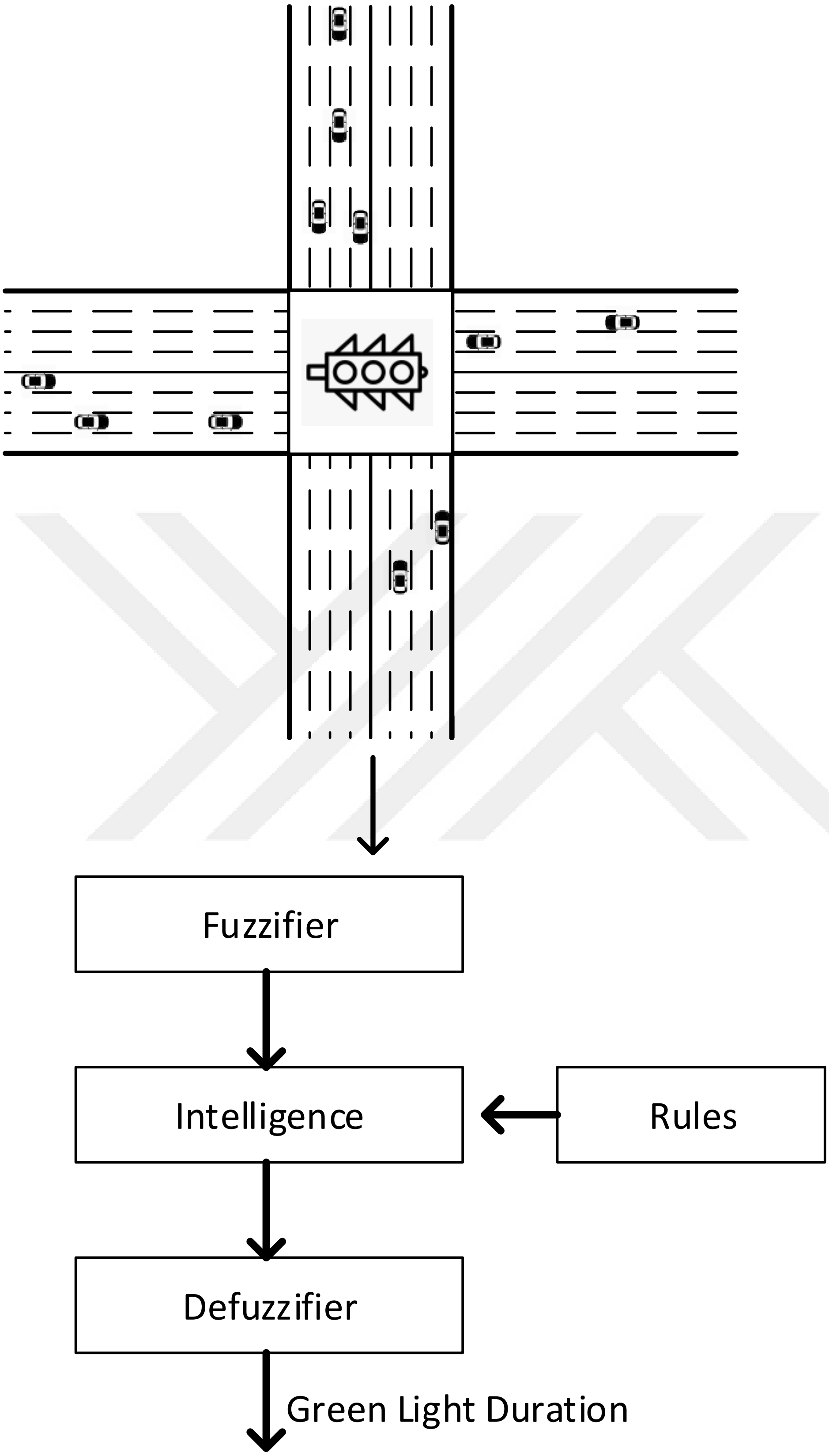


Figure 2.40 : Detailed representation of fuzzy logic traffic light control method diagram.

In order to prevent long queues on such roads, the number of vehicles on the roads is converted/normalized to 2-lane and then used as input to the fuzzy set. (For example: If there are 32 vehicles on a 4-lane road, 16 vehicles are used as the input; if there are 3 vehicles on a single-lane road, 6 vehicles are used as the input).

Table 2.14 : Rule table for fuzzy logic

GP \ RP	vvvf	vvf	vf	f	av	m	vm	vvm	vvvm
vvvf	vvvf	vvvf	vvvf	vvvf	vvvf	vvvf	vvvf	vvvf	vvvf
vvf	vvf	vvf	vvvf	vvvf	vvvf	vvvf	vvvf	vvvf	vvvf
vf	vf	vvf	vvf	vvf	vvf	vvf	vvf	vvvf	vvvf
f	av	f	f	vf	vf	vf	vf	vf	vvf
av	m	m	av	av	av	f	f	f	vf
m	vm	vm	vm	m	m	m	av	av	f
vm	vvm	vvm	vm	vm	m	m	m	av	av
vvm	vvvm	vvvm	vvm	vvm	vm	vm	vm	vm	m
vvvm	vvvm	vvvm	vvvm	vvm	vvm	vvm	vm	vm	vm

2.5.3 Simulation results

Simulation results are obtained using SUMO. CO_2 emission outputs and average velocity values are taken using SUMO. In the simulation, 6000 vehicles are produced during 7200 seconds. While the average speed for the fixed time control method is $41.46km/h$, the average speed for the actuated control method is $42.07km/h$ and for the FLC method, it is $43.81km/h$. The CO_2 emission values are $2.57x10^9$ mg/s for the fixed control method, $2.55x10^9 mg/s$ for the actuated control method and $2.5x10^9 mg/s$ for the FLC method.

Figure 2.42 and Figure 2.43 show the average velocity graph of all vehicles measured per second. As seen in Figure 2.42 and Figure 2.43 the method with the highest average speed is FLC, the method with the second highest average speed is seen as the actuated control method, and the method with the lowest average speed is the fixed time control method. In Figure 2.41, CO_2 emissions are calculated every second for all vehicles. It is assumed that CO_2 emission is zero when the vehicle is halting. In a real scenario, there will be much greater differences in CO_2 emissions between the stationary phase method and the fuzzy control method. CO_2 emission values are also different for each method depending on the average speed.

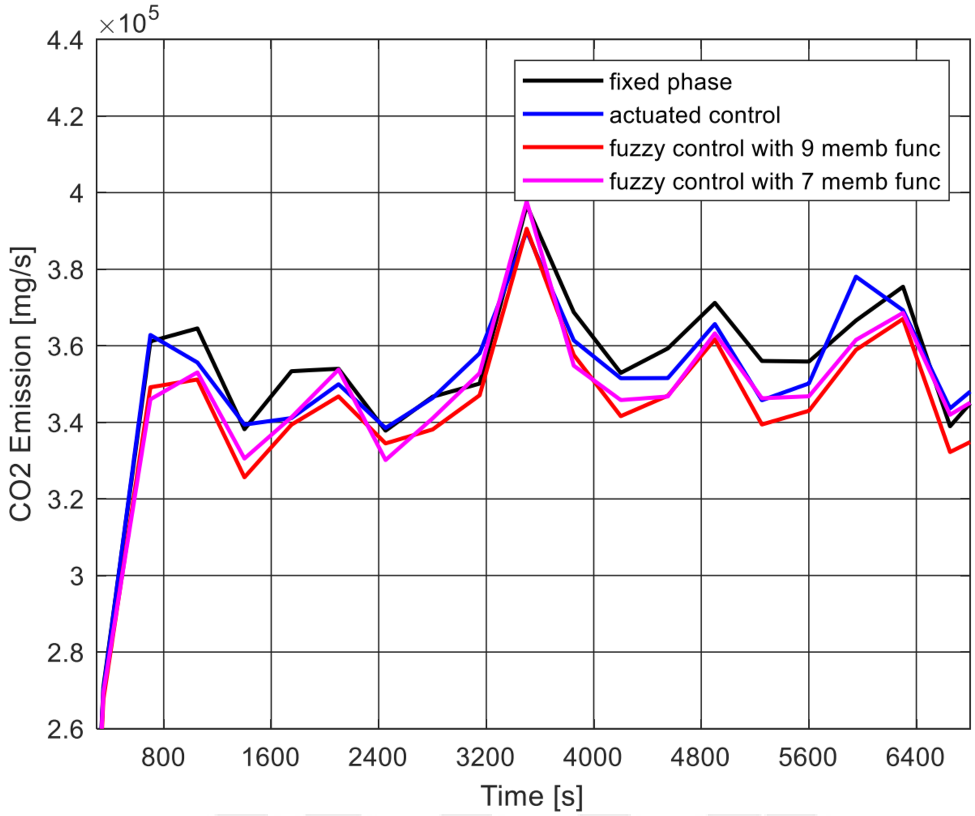


Figure 2.41 : Simulation results based on CO_2 emission for each methods.

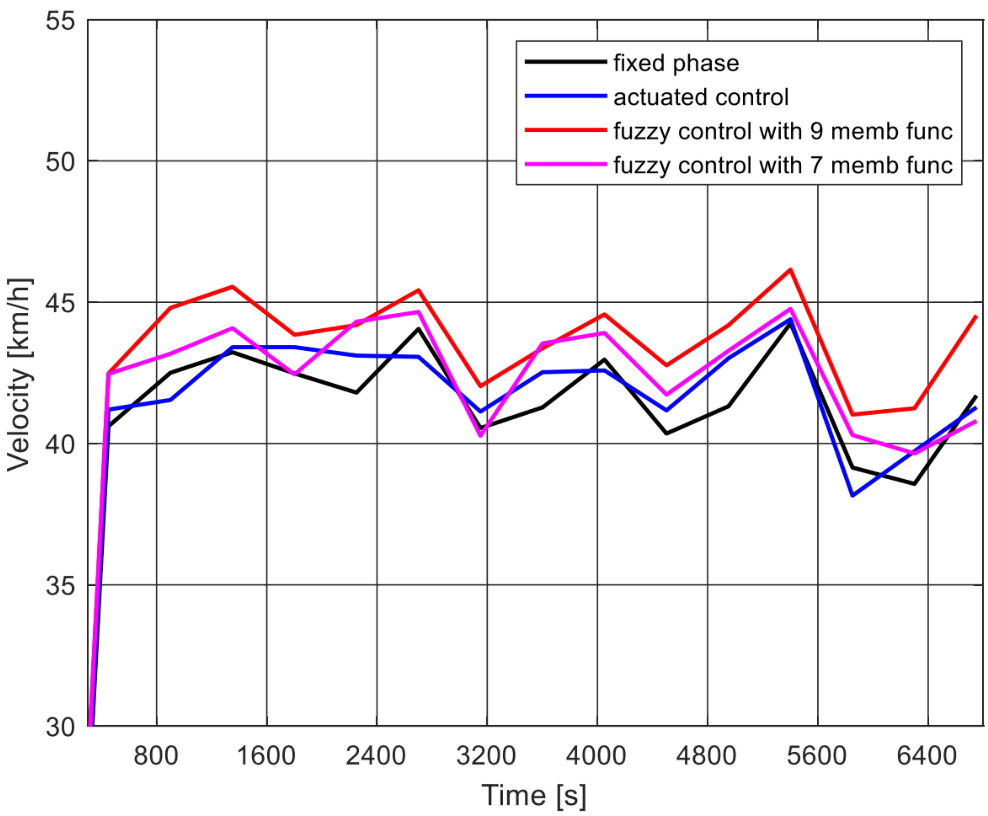


Figure 2.42 : Simulation results based on average speed for each method.

Greater average speed also means less CO_2 emission as shown in Figure 2.41. The method with the least CO_2 emission value is the FLC method. With these parameters, it is clear that the most efficient method is fuzzy logic TLC. The graphs show that traffic density and carbon footprint are reduced. In addition, a fuzzy method with 7 membership functions is also used in simulations for comparison. However, as expected the fuzzy method with nine membership functions outperforms the one with seven membership functions.

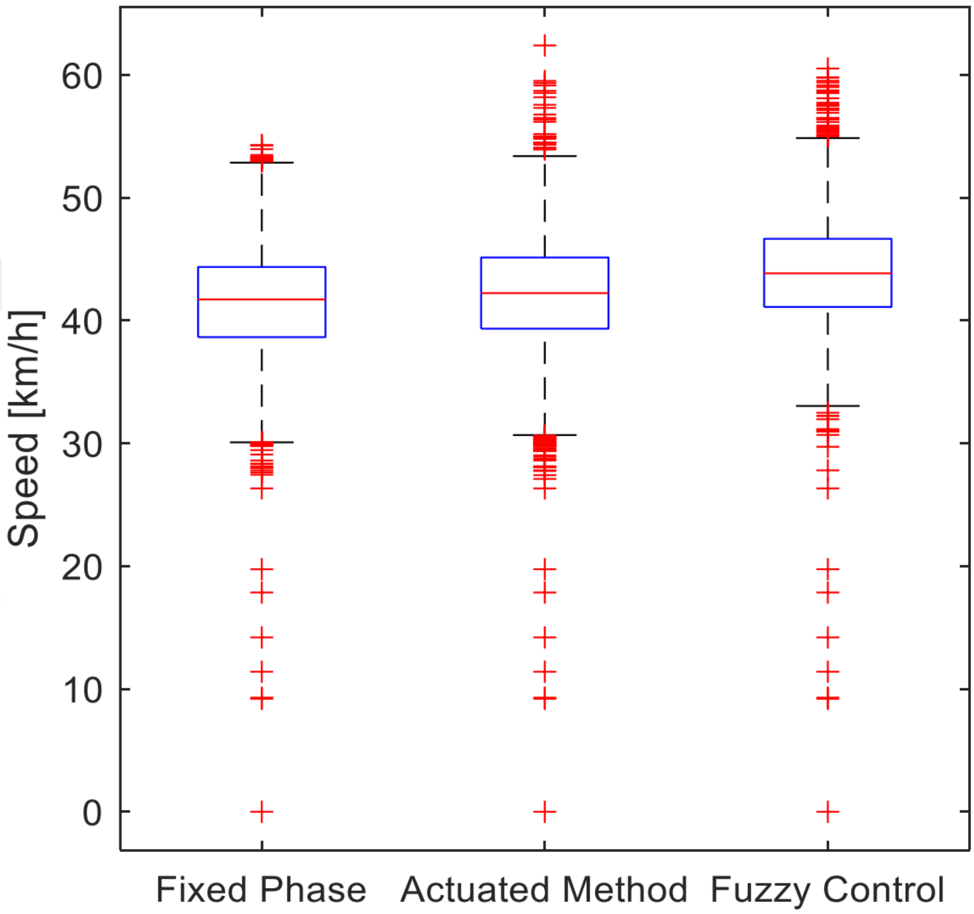


Figure 2.43 : Simulation results based on average speed for each methods with median values.

Actuated and FLC methods are compared to classical traffic light control, as shown in Figure 2.41, Figure 2.42 and Figure 2.43. It is shown that both methods are better than the traditional method, but the fuzzy logic method gives the most efficient results. Since it sends predetermined signals for dynamically changing traffic density, the fixed time control method is the most unfavourable compared to the others. Although the actuated method is better than the traditional method, it is considered not as robust as fuzzy control as the feedback information is not used fully.

3. FUZZY LOGIC AND DEEP Q LEARNING BASED CONTROL FOR TRAFFIC LIGHTS

In this section, we propose a new agent-based fuzzy logic assisted traffic light signal timing for traffic intersections. Deep Q-learning algorithms and Fuzzy Logic Control (FLC) are used together in the proposed method. The proposed method and many traffic light control methods in the literature were simulated. In order to demonstrate the effectiveness of the proposed method, some of the important metrics of evaluation such as traffic congestion, air pollution, and waiting time were used in the assessment of the simulation results. In the method proposed in this section, the phase sequence is determined by using the deep Q-learning algorithm, and the green light duration is determined according to the traffic intersection state. In addition, with the proposed method, it has been shown that the stability and robustness of the system are increased.

Several recent studies have suggested the application of deep reinforcement in the traffic light control problem [122, 123]. However, in these studies, the traffic light durations are divided into fixed time intervals and are increased only by multiples of these fixed time intervals. This is not an efficient method. In addition to this, the application of such a method without further precautions is not safe for drivers as the green light period can change at any time. The main motivation of this study is to determine the phase sequence and duration of the green light in an optimal way while enabling the system to deliver accurate information to the drivers. Major contributions of this study can be listed as follows:

- To the best knowledge of the author, either the duration of the green light is constant, or the phase sequence of lights is predetermined in the current literature. In this section, deep Q learning and FLC are used in combination for the first time such that the phase sequence is controlled by the deep Q learning algorithm and the duration of the green light is determined by the fuzzy logic controller. It is shown by the help of simulation that this combination results in a better option for determining the green light duration in comparison to the methods available in the literature.

- The idea of using a varying cell size in the determination of the state matrix used in the deep Q learning algorithm, which was used in [124], is also adopted to the proposed method. A comparison between using constant (equal) cell sizes and varying cell sizes is also provided to demonstrate the efficiency of this adaptation. It is shown that using varying cell sizes in the determination of the state matrix such that shorter cells are used as the distance to the intersection gets smaller provides a better solution.

- A theoretical stability analysis is developed and made. Test simulations are observed to confirm this analysis.

- It has also been observed with the help of simulations that the proposed method is more robust in comparison to some other methods available in the literature.

3.1 Reinforcement Learning

Reinforcement Learning (RL) [74] is a machine learning approach that is widely used in applications where online learning is required. It helps the agents to take the best action in order to maximize cumulative utility over time. As a result, the learning process reinforces the agent to learn to take the best possible action in the environment of interest. The reinforcement learning method is used in many areas. Mobile robot applications, traffic control and decision-making methods can be given as examples. In the reinforcement learning method, the agent in an environment performs an action depending on the activity of the other agents and the current state of the environment, Then, the environment responds with a numerical reward.

Q-Learning [125] is a form of model-free reinforcement learning and is very popular in the Markov decision process (MDP) as no information is required for transition possibilities [20]. Q-Learning is one of the most commonly used RL methods for TLC [126]. It entails providing a numerical value, known as the Q-value, which involves an action done in response to a certain state of the environment. For the control of an agent, the Q-learning rule is expressed as follows [125]:

$$Q_{t+1}(s_t,a_t) = Q(s_t,a_t) + \alpha(r_{t+1}) + \gamma.max_A Q(s_{t+1},a_t) - Q(s_t,a_t) \qquad \textbf{(3.1)}$$

The value of $Q(s_t, a_t)$ in equation 3.1 is updated decreasingly depending on the learning rate α and the action's value in state s_t is $Q(s_t, a_t)$. The r_{t+1} in equation 3.1 is the reward value obtained after performing an action in the s_t state. $Q(s_{t+1}, a_t)$ represents the next value of Q and s_{t+1} is the state that occurs after the action while in the s_t state. The term max_A denotes that the highest valued action among the potential actions in the s_{t+1} state is chosen. γ is a discount factor, and it's used to make the future reward less important than the immediate effectiveness and its value ranges from 0 to 1. A significantly modified version of equation 3.1 is used in this study, and it is presented in the form of equation [127].

$$Q(s_t, a_t) = (r_{t+1}) + \gamma.max_A Q'(s_{t+1}, a_{t+1}) \quad \textbf{(3.2)}$$

The term $Q'(s_{t+1}, a_{t+1})$ denotes the Q-value associated with performing action a_{t+1} in state s_{t+1}, i.e. represents the state after the action.

In equation 3.2, there is a rule that updates the Q-value of the current action taken in state st with the immediate reward as well as the discounted Q-value of future actions. As a result, the term $Q'(s_{t+1}, a_{t+1})$, which represents the value of future actions, holds the maximum discounted reward of the state after s_{t+1} implicitly. $Q''(s_{t+2}, a_{t+2})$ and $Q'''(s_{t+3}, a_{t+3})$ similarly hold the maximum reward for the next state, and the next Q values are calculated accordingly. This is how the agent will select an action based not only on the immediate reward but also on the anticipated future discounted rewards. The rule can be unpacked as follows for the sake of simplicity,

$$Q(s_t, a_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + ... + \gamma^{y-1} r_{t+y} \quad \textbf{(3.3)}$$

In equation 3.3, y is a random value indicating only the last time step before the end of the episode; since no more actions are possible, the value of r_{t+y} is 0. Dynamic programming may be used to solve this equation, but it must be finite in order for the computational complexity to be manageable.

A traffic light controller with Q learning tool placed at the traffic intersection can perform the Q calculation. Agent-based traffic light control for transportation systems is highly efficient as it can adapt to different scenarios.

3.2 System Architecture

In this article, in the simulation, a 4-legged traffic intersection with four entrances and four exits is controlled. As seen in Figure 3.1, Each road at the traffic intersection has four lanes. Vehicles that will turn right use the rightmost lane at the traffic intersection, and vehicles that will turn left use the leftmost lane. The left turn lane is solely used for turning left, whereas the right turn lane is used for both turning right and moving straight forward. To put it another way, if a vehicle is going to turn left, it must be in the far-left lane, and if it is going to turn right, it must be in the far-right lane.

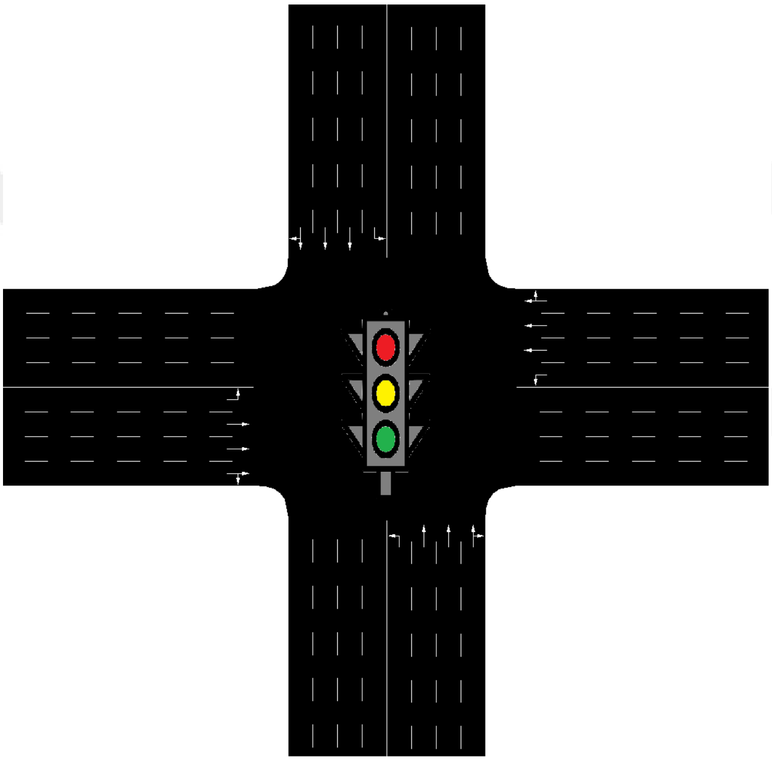


Figure 3.1 : TLC simulation environment

Passage permits for opposite directions are also given. Left turns, on the other hand, are permitted in a separate phase. The yellow light is lit for four seconds when changing from red to green and vice versa. Traffic light pass directions for each action are shown in Figure 3.2. In addition, the traffic light passing directions shown in Figure 3.2 are deep Q learning output actions.

3.2.1 State

In the proposed Q learning algorithm, the state variables are made up of vehicle position information. The traffic intersection's state is described by st, and at a given

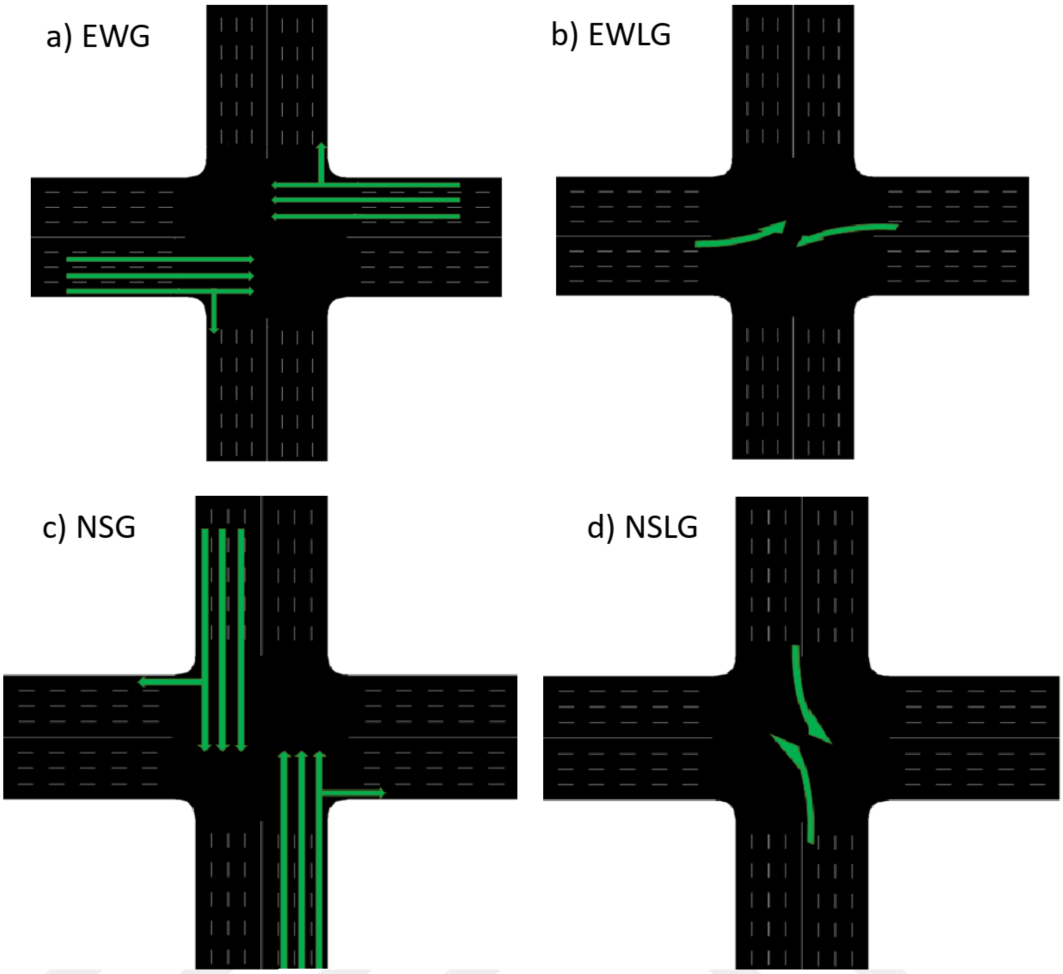


Figure 3.2 : Actions for traffic light agent

time step t it represents a description of the state of the environment. The state must supply adequate knowledge of the distribution of cars on intersections to enable the learning algorithm to learn how to optimize traffic effectively. The purpose of this state information is to provide instant information to the agency about the location of the vehicles in the environment.

However, unlike studies in general [128], cell sizes are not equal. This specific state design, in particular, just includes positional details about the vehicles housed in the medium, and the cell sizes used to separate the continuous medium are not equal [124]. Cell sizes are small in areas close to the traffic intersection, and they increase as they move away from the traffic intersection. This method has been applied to ensure that the effect of vehicles near the traffic intersection in the state vector is greater. As a matter of fact, simulations are made for both cases where the cell sizes are taken equal to and not equal to each other. The results are discussed in the simulation results section. The chosen state representation design is based on realism: Information-rich states have been proposed in recent research on traffic signal controllers, but they are difficult to implement in practice because the information needed for such representations is difficult to obtain.

At the 4-legged traffic intersection, each incoming road was discretized into cells. The values of these cells are 1 when there are one or more vehicles in the cell, and 0 when there is no vehicle in the cell. There are 10 cells along the lane. At the traffic junction, there are 20 cells on each incoming road. Therefore, there are 20 cells in each incoming road and 80 cells in total. The three lanes on the right are part of the same cell since they share a traffic signal, but the track on the left has independent cell lines.

The Lane Space Discretization method has been developed to detect the presence and absence of cars in each branch of the traffic intersection. A Lane Space Discretization (L_i) vector is a mathematical representation of the state space, in which every $L_{i,k}$ unit shall be calculated according to equation 3.4. A state example of the multidirectional traffic intersection generated by the L_i equation is shown in Figure 3.3 (c).

$$L_{i,k}(n) = sgn(C_{i,k}(n) + q_{i,k}(n) - d_{i,k}(n)sgn(q_{i,k-1})) \tag{3.4}$$

in equation 3.4 $i = 1,2\ldots m$ is the index of the traffic streams; $n = 0,1,2\ldots,n-1$ is the index of the discretized time intervals. The value k represents the number of cells in the paths. As seen in Figure 3.6, there are 10 cells for each L_i, and a total of 20 cells in each length. The value of each cell is found as seen in equation 3.4. Where, C is the number of vehicles in the cell. q and d are the numbers of vehicles entering the cell and the number of vehicles leaving the cell, respectively. The corresponding element of the L_i vector is 1 if there are one or more cars in $L_{i,k}$; otherwise, it is 0.

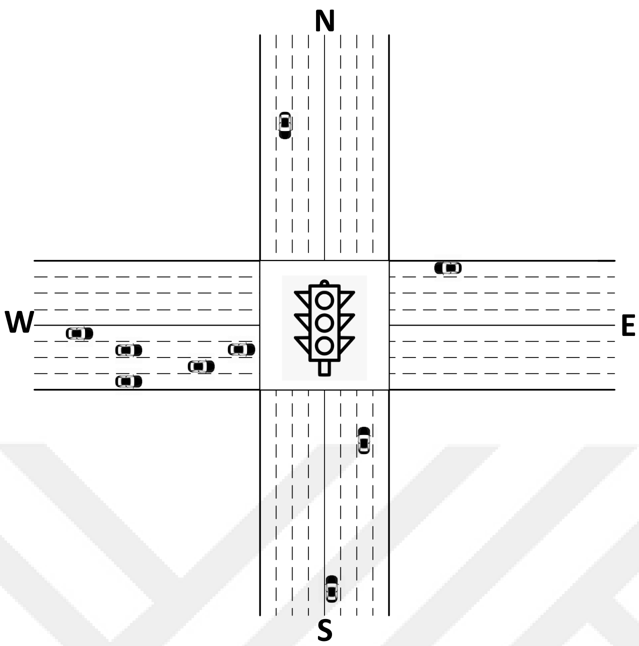
In equation 3.4, $sgn(q_{i,k})) = sgn(q_{i,0}))$ for k=1. $q_{i,0}$ represents the state of vehicle entry to the intersection area from lane i. Therefore, as seen in the equation 3.5, $sgn(q_{i,0}))$ becomes equal to the traffic light phase. if Z_i is 0 stop, if 1 then go. Z_i is the actions of the deep Q-learning algorithm according to the phases in section 4.1.3.

$$sgn(q_{i,0})) = Z_i \tag{3.5}$$

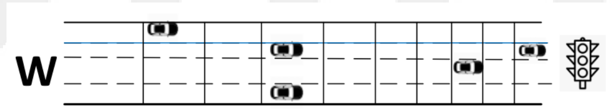
Figure 3 shows (a) a snapshot of traffic at the traffic intersection, (b) a snapshot of traffic at the western leg of the traffic intersection divided into cells, and (c) the corresponding position matrix in this traffic intersection. The labels given on the left of the matrix in Figure 3 (c) indicate the corresponding paths for each row. For instance, WES (West to East or South) represents vehicles coming from the west direction and heading toward the east or south direction. These vehicles are assumed to be in the rightmost three lanes of the western leg of the intersection. WN (West to North) represents vehicles coming from the west direction and heading towards the north direction. Therefore, the corresponding row in the state matrix is found by looking at the locations of the vehicles in the leftmost lane of the western leg. Similarly, NSW (North to South or West), SNE (South to North or East), and EWN (East to West or North) represent vehicles in the three lanes to the right for vehicles approaching from the north, south, and east directions, respectively. Additionally, NE (North to East), SW (South to West), and ES (East to South) represent vehicles in the left lane approaching from the north, south, and east directions, respectively. The distances of the cell spacings in Figure 3 (c) are shown in Figure 3.5.

When a vector L_i is shown by the agent in time, the environment in that time is represented by a vector L_i This is the main environmental information the agent receives, and so it is structured to be as accurate as possible but not too specific so as not to increase the computational complexity of the neural network's training. The length of the agent's exploration of the state space in reinforcement learning is critical to the agent's own performance: unless it explores a large part of the state space, it will not be able to estimate the best action in each case correctly. After the preparation, also in an unseen state, the agent should be able to pick the best action as it looks like a similar condition within its expertise, in which it knows the resulting performance of any action. This means that the proposed state-space architecture should be appropriate for the agent's anticipated learning time.

There are 80 boolean cells in the proposed state space. The agent must explore only the most important subset of the state space in order to learn the best actions, so the choice of boolean cells for the environment representation is also essential. The critical situation for the environment is that at least one vehicle stops at the traffic intersection and waits for the green phase. Therefore, cells closest to the stop line are more critical than the cells that are farther away. This suggests that the combinations of states with active cells closer to the stop line contribute more to the agent's successful results, where the training time is within expectations.



(a) Snapshot of traffic at traffic intersection



(b) Snapshot of vehicle positions at the western leg of the traffic intersection divided into cells

$$
\begin{array}{r}
WDR \\ WL \\ EDR \\ EL \\ NDR \\ NL \\ SDR \\ SL
\end{array}
\begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

(c) The corresponding position matrix in this traffic intersection

Figure 3.3 : The process of obtaining the state matrix

This study assumes that monitoring cameras (or suitable sensors) are installed at intersections to identify vehicles and pedestrians.

3.2.2 Action

Steps in accordance with the current traffic light rules are designed for the Q-learning algorithm. Just one operation can be performed in any time slot. This operation is measured and chosen from the Q learning agent action sets that can optimize the benefits. The possible different actions of the agent are defined as a set of actions. The agent is the traffic light system, which means that certain traffic lights are turned to green for a certain number of routes and kept in that state for a specified period of time. The time for the green light is 10 seconds, while the time for the yellow light is 4 seconds. The duty of the traffic intersection agent is to start the green light selection process. The action set is shown in equation 3.6.

$$A = \{EWG, EWLG, NSG, NSLG\} \quad \textbf{(3.6)}$$

Note that the meanings of EWG, $EWLG$, NSG, and $NSLG$ in equation 3.6 are illustrated in Figure 3.2. For the state where successive actions are equal to each other (the selected traffic phase has not changed), the yellow phase is not used, and the current green phase continues. For the state where the green light and yellow light durations are constant 10 seconds and 4 seconds, respectively, the other operation will not start for at least 14 seconds until a different action is performed. Therefore, a total of fourteen simulation steps pass.

3.2.3 Reward

The reward is that after the agent chooses the action in advanced learning, the result of the action is taken from the environment. The choice of the parameter is very important, as the agent interprets the reward value to evaluate the outcome of the action taken and improve the pattern of future action decisions. The reward is, therefore, an essential element in the process of learning. Two potential values are typically present: positive or negative. The purpose of this application is to optimize traffic flow over time through the intersection. Different control or optimization goals can be achieved using various Q-learning rewards. For instance, it can be a negative value, such as the number of cars and vehicle queue length. The goal used in this study is to reduce the total waiting time, which is an important parameter that represents the traffic flow situation.

In order for the traffic intersection agent to make the right choices, the reward value must be obtained from the traffic performance criteria. The most effective intersection is one that does not require cars to wait in the traffic lights. As a result, the principle of waiting time is critical for selecting the incentive measure. The overall waiting period is considered as the most reliable of the suggested measures. The entire waiting period has been selected for the calculation of the value of the reward. The total waiting time is the cumulative value of the waiting times of the vehicles waiting at the intersection at time t [128]. The waiting times are determined from vehicles moving at a speed of less than 0.1 m/s. The total waiting time is shown in equation 3.7.

$$Twt_t = \sum_{i=1}^{n} wt_{(i,t)} \tag{3.7}$$

In equation 3.7, $wt_{(i;t)}$ is the time in seconds at which a vehicle has a velocity less than 0.1 m/s in time step t. Twt_t is the total waiting time in t time step. At t time step, the total number of vehicles is n. The reward function can be seen in equation 3.8.

$$r_t = Twt_{t-1} - Twt_t \tag{3.8}$$

3.3 Deep Q Learning with Fuzzy Logic

Deep reinforcement learning, which combines deep learning and reinforcement learning, is an extensively used technique in TLC [129]. Through experience replay, the agent stores the experience again in memory and trains itself again with randomly selected experiences from memory. The agent obtains a copy of the main network, uses its weights to compute the target Q-value and computes the minimized loss function using gradient descent increase. The target network weights are fixed to improve training stability [130]. The main network allows the agent to choose action after observing its state from the environment and updating the Q value in the main network.

3.3.1 Deep neural network

A deep neural network is designed to map states in the system to Q values, which represent values associated with a behaviour. The vector $L_{i,t}$ is the input of the network at time t. The Q values of the potential actions from state st are the network's outputs.

The input of the neural network is defined as seen in the equation 3.9.

$$n_{i,t}^{in} = L_{i,t} \tag{3.9}$$

The input dimension of the neural network is $|nin|$ which is equal to $|L| = 80$. The neural network output is shown in equation 3.10.

$$n_{j,t}^{out} = Q(s_t, a_{j,t}) \tag{3.10}$$

In the equation 3.10, is the $j-th$ output of the neural network at timestep t and j,t. The Q-value of the $j-th$ action taken from state st at timestep t is $Q(s_t, a_{j,t})$. Algorithm 1 shows the process of the deep Q-learning traffic light control method.

The L vector represents the input to the network. Then the hidden layers and finally the output layer with 4 different potential outputs. Hidden layers are used to make distinguishable intermediate representations between the inputs and outputs. Hidden layers also provide space and tools for the transformations that are needed in order to have more meaningful output representations.

Using the waiting times at time t and time t-1, the agent calculates online the reward value for the selected action at $t-1$. Then the agent saves this information as a packet to the memory along with the environment state. It then chooses a new action based on the available information and applies it to the traffic intersection. In this study, as explained earlier, the actions chosen by the agent are traffic lights. It will perform agent learning with 500 different episodes where it can encounter many traffic states. Vehicles are produced for 90 minutes in each section.

In the form of a set of random samples called Batch, the information needed for learning is gradually presented to the network that is expected to create more recognizable representations of the data. The Batch receives the information from the memory that stores each set of samples throughout the training. A memory instance (M) consists of 4 elements.

Algorithm 2 Deep Q learning Traffic Light Control

Input: memory size M, batch size , learning rate, discount factor , number of state, number of action, discount factor γ.

Output: Traffic Light Phase

Notations:

m: the replay memory.

i: step number.

while *there exists a state s* **do**

    Choose an action based on the ε greedy.
    Carry out action a and observe state s and reward r.

if *the memory size* $m > M$ **then**

    Remove the oldest experience in the memory.
    end

Initialization: s, green light duration, traffic light = phase1

Variables: count, add, step, traffic light

for *obtain environment state* **do**

    calculate reward
    save sample to memory
    train
    choose new action
    set: count=0, step=1 second
    while *count < Green Light Duration* **do**
        traffic light = green
        count = count + step time
    end
end

$$M = \{s_t, a_t, r_{t+1}, a_{t+1}\} \tag{3.11}$$

The r_{t+1} in equation 3.11 is the reward value obtained after performing a_t in the s_t state. In the experience replay technique used, the memory size, which determines how many samples the memory can store, is set to 50000. A batch is the number of data retrieved from memory in a training sample at each epoch. However, when the memory is full after a certain training, the oldest data in the memory is deleted. In this way memory crash that would be inevitable by the constant accumulation of data is prevented.

An agent tries to find the right action during the training phase. However, in the first chosen actions, the agent knows little or no which action might be right. To overcome this problem, the agent must be able to make more discoveries in the initial steps.

Once the agent has a significant amount of information, exploring and taking different actions may not yield good results.

Therefore, the agent needs to do less exploration after getting to know the environment. Thus, the exploration value is high at the beginning of the training process and should decrease towards the end [27]. The exploration rate is seen in equation 3.12.

$$E_h = 1 - \frac{n}{N} \tag{3.12}$$

where n is the current episode in the equation 3.12 and N is the total number of episodes.

3.3.2 Deep Q learning with FLSI

In classical logical thinking, the results are precisely given, such as true and false, and there are no grey areas.

In contrast, indefinite values such as nearly true or nearly false can be used in fuzzy logic [131]. An important application of fuzzy logic is in the control of nonlinear systems such as traffic control systems. Actually, it has been shown that it is very efficient in the control of traffic lights [72, 75]. FLC consists of 4 main parts. These are rules, fuzzification, defuzzification, and intelligence. Usually, rules and definitions are changed to achieve better results.

A block diagram for the Deep Q-Learning Fuzzy Logic with State Inputs (DQ FLSI) is shown in Figure 3.4. As can be seen from this Figure, the duration of green light is controlled by the FLC, which is continuously interacting with the environment and the deep Q-Learning module, which decides on the phases of traffic lights. In this method, the deep Q-learning algorithm and the FLC work together. While green phase actions are determined with deep Q-learning, the duration of green light is determined by the FLC. the input values of the traffic light controller denoted the vehicle position information. The state of the traffic intersection is denoted by s_t, representing the vehicle positions information at the intersection at time t.

In this method, the traffic intersection officer accesses the location information of the vehicles on each road and tries to decrease the total of waiting time for vehicles at the traffic intersection. The aim of this method is to allow the controller to quickly

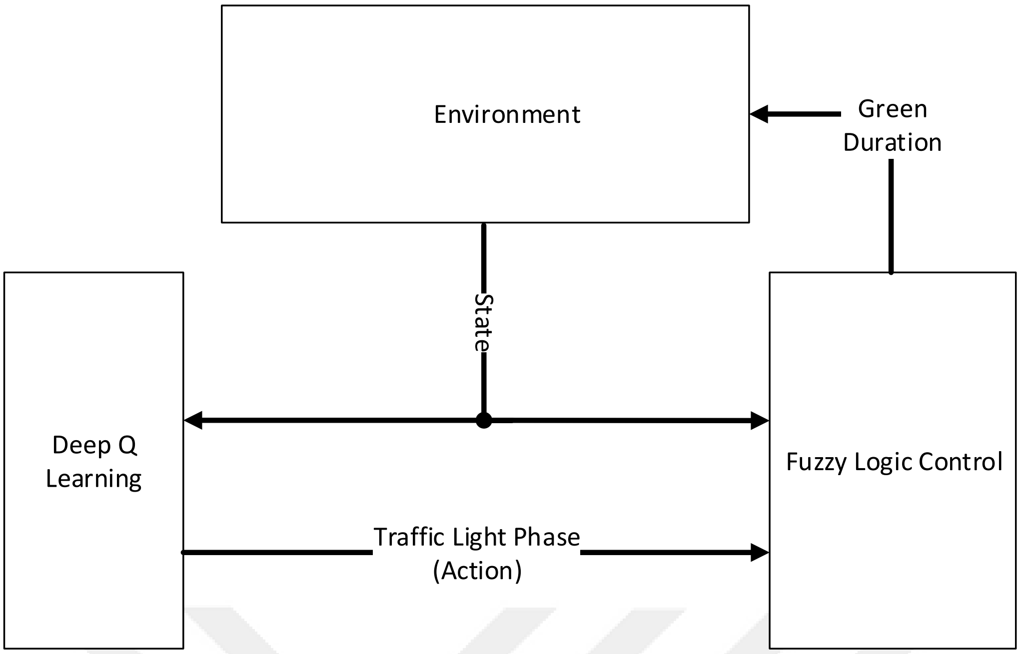


Figure 3.4 : Block diagram of DQ FLSI

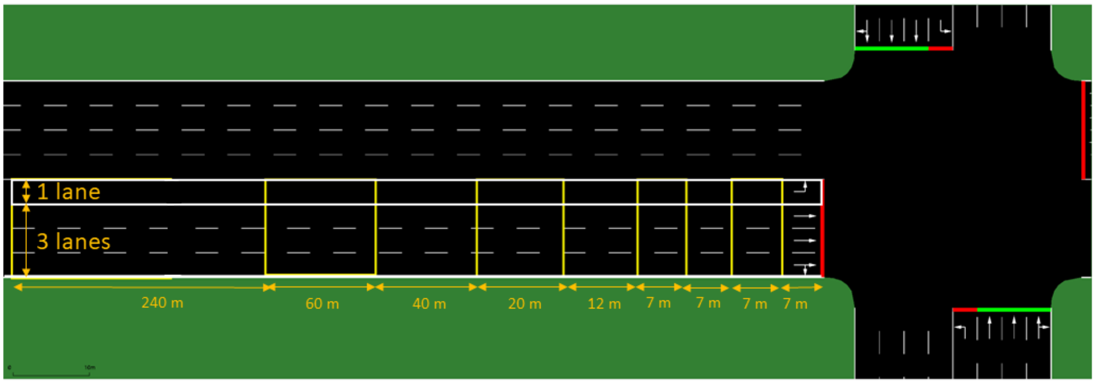


Figure 3.5 : Division of an arm at a traffic intersection into cells

determine the locations of the vehicles in the area. While green phase actions are taken with the help of deep Q learning mechanism the duration of green light is determined by FLC. This method is related to the positions of the vehicles. The same cell can be used for vehicles moving for the same phase in different lanes, so we can consider their strips to be the same cell. However, there is a different phase for a left turn, there is a different group of cells. Each lane has ten cells, as shown in Figure 3.5. The leftmost lane is the only one that is separated because three lanes using the same step are not separated, resulting in a total of 16 cells in one arm. For the four-leg traffic intersection considered in this study, there are a total of 64 cells. Algorithm 3 describes the learning process of deep Q learning with the FLSI traffic light control method.

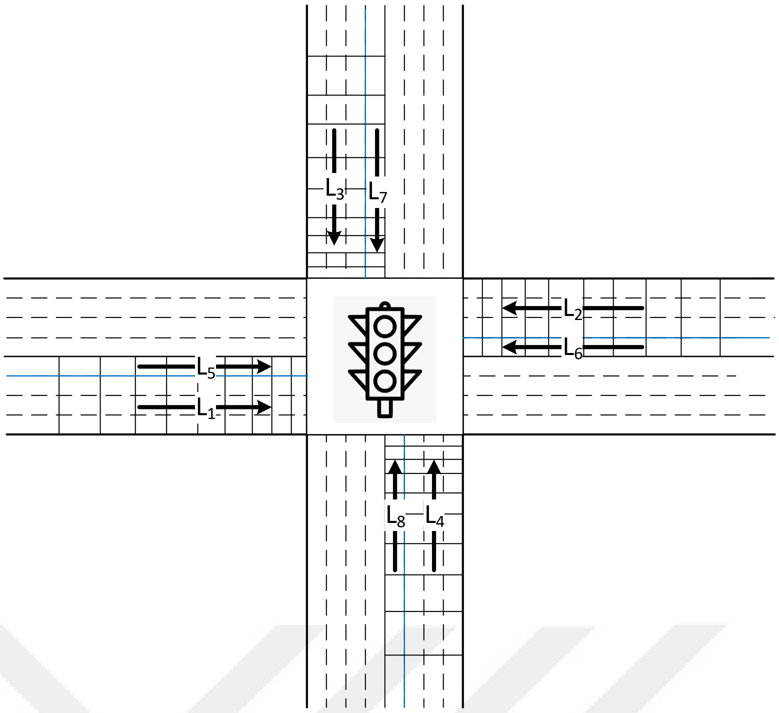


Figure 3.6 : Trafik stream L_i values and cell compartments at the traffic intersection.

Algorithm 3 Deep Q learning with FLSI Traffic Light Control

Input: memory size M, batch size , learning rate, discount factor , number of state, number of action, discount factor γ., fuzzy logic 2 input (Green Phase and Red Phase)

Output: Traffic Light Phase, Green Duration

Notations:

m: the replay memory.

i: step number.

```
while there exists a state s do
    Choose an action based on the ε greedy.
    Carry out action a and observe state s and reward r.
if the memory size m > M then
    Remove the oldest experiences in the memory.
    end
```

Initialization: s, min green light duration, max green light duration, traffic light = phase1

Variables: count, add, step, traffic light

```
for obtain environment state do
    calculate reward
    save sample to memory
    train
    choose new action
    calculate: Green Light Duration in linear relation to fuzzy parameter.
    set: count=0, step=1 second
    while count < Green Light Duration do
        traffic light = green
        count = count + step time
        end
    end
```

$$FL_i(n+1) = L_{i,1}(n) + L_{i,1}(n) + ... + L_{i,k}(n) \qquad \textbf{(3.13)}$$

Fuzzy logic input values are found using equation 3.13 and equation 3.14. In equation 3.13 $i = 1, 2 \ldots m$ is the index of the traffic streams; $n = 0, 1, 2 \ldots, n-1$ is the index of the discretized time intervals. The value k represents the number of cells in the paths. As seen in Figure 3.6, there are 8 cells for each L_i, and a total of 16 cells in each length. The value of each cell is found as seen in equation 3.13, where C is the number of vehicles in the cell. q and d are the numbers of vehicles entering the cell and the number of vehicles leaving the cell, respectively.

$$FL_{i,k}(n) = sgn(C_{i,k}(n) + q_{i,k}(n) - d_{i,k}(n) sgn(q_{i,k-1})) \qquad \textbf{(3.14)}$$

There are two input values for FLC. To begin, GP is the sum of the FL_i values of the roads in the green phase, that is, the sum of the FL_i values of the roads allowed to pass. The second one is RP , which represents the number of the sum of FL_i values of the no-passing paths. In other words, the sum of the FL_i values of the roads waiting for the red light.

Table 3.1 shows the fuzzy logic rule table, and Table 5.2 contains the explanations of the abbreviations in the rule table. Input values of the designed FLC correspond to the triangular membership function. The interval for GP is [0 16] while it is [0 48] for RP. The green light duration, which is the output value of the fuzzy controller, is [0 35]. Figure 3.7 demonstrates the learning process of deep Q learning with FLSI.

Table 3.1 : Rule table for FLSI

GP \ RP	vvf	vf	f	av	m	vm	vvm
vvf	av	f	f	vf	vf	vvf	vvf
vf	m	av	f	f	vf	vf	vvf
f	m	m	av	f	f	vf	vf
av	vm	m	m	av	f	f	vf
m	vm	vm	m	m	av	f	f
vm	vvm	vm	vm	m	m	av	f
vvm	vvm	vvm	vm	vm	m	m	av

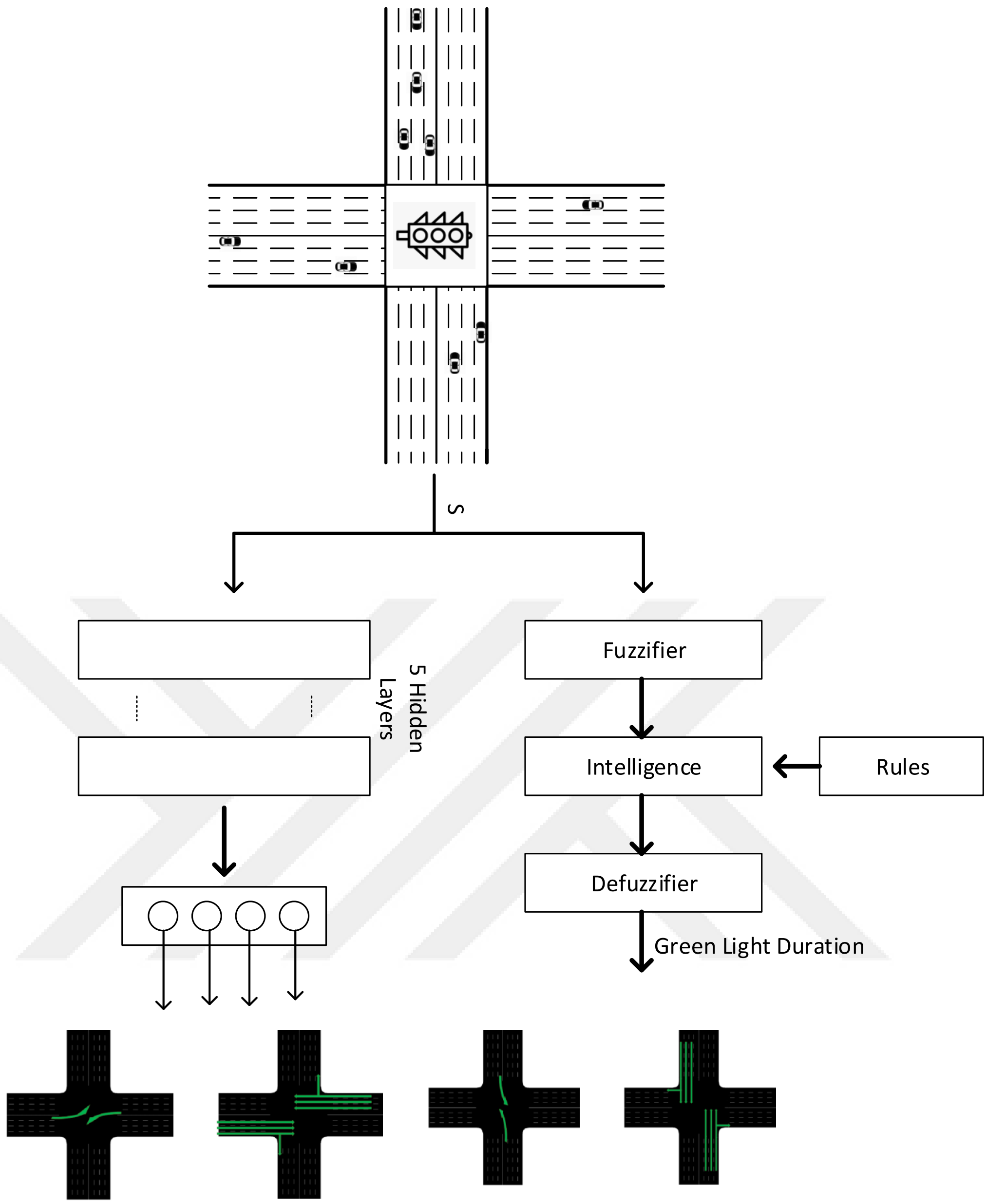


Figure 3.7 : Traffic light control process with deep Q learning with FLSI method.

3.4 Simulation results

SUMO software is used to make a simulation environment, and the simulation is run through it. During one episode, vehicles are generated at the same time for 5400 seconds. 3000 vehicles are produced for each episode, and the total number of episodes is 500. Furthermore, vehicles produced during the scenario pass the intersection only once, and the number of vehicles coming from all directions is almost equal. In

Table 3.2 : Abbreviations of rule table

vvf	very very few
vf	very few
f	few
av	average
m	much
vm	very much
vvm	very very much

addition, in all the simulation cases, it is assumed that approximately 75% of the vehicles are going straight and 25% are diverging to left or right.

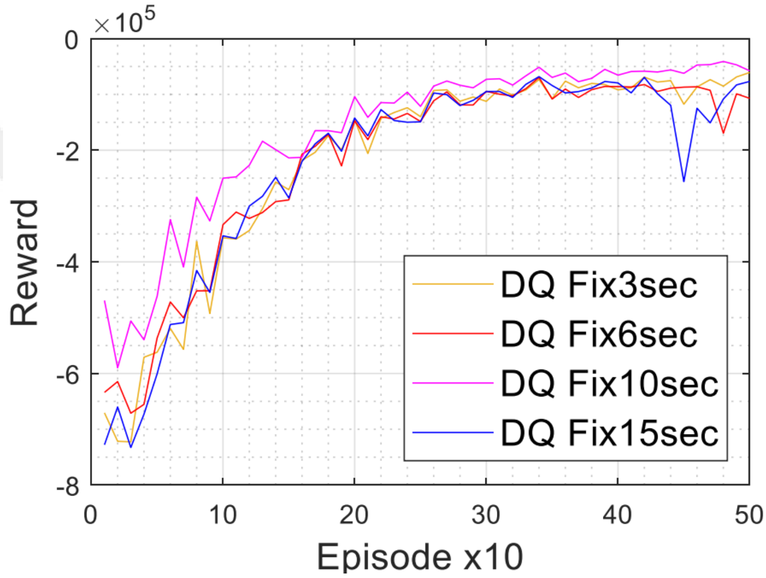


Figure 3.8 : Simulation results based on reward values according to fixed green light time values.

Table 5.3 shows all the methods used in the simulation environment. In the methods in which deep Q learning methods are used, learning takes place in the first 450 episodes. The values shown in Table 5.3 are the average values of the 50 episodes after the learning takes place. In Table 5.3, simulation results are based on total queue length, CO_2 emission output and cumulative delay values. DQ Fix3sec in Table 5.3 are the results obtained with the deep Q learning method for a fixed 3 seconds green light duration. Likewise, DQ Fix6sec, DQ Fix10sec, and DQ Fix15sec are the simulation results of the deep Q learning method for 6, 10, and 15 seconds, respectively. DQ Fix10sec eqst is a deep Q learning method based on states calculated according to equal cell lengths for 10 seconds of fix light duration. Likewise, DQ Fix6sec eqst is the simulation result of the deep Q learning method for 6 seconds according to equal cell lengths. The Queue length indicates the number of vehicles waiting for the green light at a given time.

Table 3.3 : Simulation results

Method	Queue	$CO_2(kg)$	Delay(h)
DQ Fix3sec	48.7250	1020.25	73.08
DQ Fix6sec	59.0745	1174.75	88.61
DQ Fix10sec	33.7422	802.28	50.61
DQ Fix15sec	59.5796	1183.24	89.37
DQ Fix10sec eqst	54.2325	1104.16	81.34
DQ FLSI eqst	15.9657	341.49	23.94
DQ FLSI	8.8838	228.70	13.32
DQ Fix6sec eqst	86.5387	1586.19	129.80
Fix10sec	348.367	5127.55	522.55
Fix12DR-8Lsec	191.254	3034.33	286.88
Fix12DR-6Lsec	151.77	2481.21	227.65
FL Q Length	29.02	556.17	43.52
FLSI eqst	17.359	375.70	26.03
FLSI	17.781	368.84	26.67

'DQ FLSI' seen in the simulation results is the Intelligent Traffic Light controller proposed in this study. In this method, deep Q learning and fuzzy logic are used together. In addition, a state matrix is designed to be sensitive to the distance to the traffic intersection, as explained in Section 4.2. The difference of the DQ FLSI eqst method from the DQ FLSI method is that equal cell lengths are used in the former. It should be remarked that Fix10sec, Fix12DR-8Lsec, Fix12DR-8Lsec, FL Q Length, FLSI eqst, and FLSI methods do not implement learning algorithms. Since there is no learning process in these methods, only values for one episode are shown in the Table. In these methods, the traffic light action phases are sequential. Fix10sec fixed 10-second green light times are predetermined, and there is no feedback control. For Fix12DR- 8Lsec method, the fixed 12 seconds for the straight and right-turn phases and 8 seconds for the left-turn phase is predetermined, and there is no feedback control. Likewise, in the Fix12DR-6Lsec method, a fixed green light duration of 12 seconds for straight and right turns and 6 seconds for left turns is predetermined. The fuzzy logic traffic light control method is used in FL Q Length, FLSI eqst, and FLSI traffic light control methods, and the output is the green light duration in all of them. The input values in the FL Q Length method are the queue length, and their ranges are 0-60 and 0-180. The difference between the FLSI and FLSI eqst methods is in the determination of cell sizes used in the fuzzy logic inputs. While the cell sizes are equal in the FLSI eqst method, they are changing with respect to proximity to the traffic intersection.

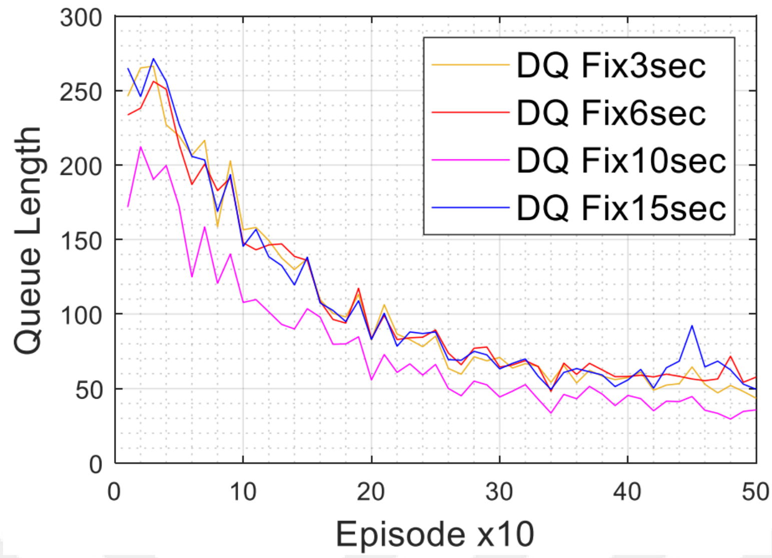


Figure 3.9 : Simulation results based on total queue length values according to fixed green light time values.

All vehicles generated during the simulation are identical and have a first speed of 36 km/h and a length of five meters. The top speed of the vehicles is 90 km/h. Vehicles accelerate and decelerate at 1 and 4.5 m/s^2, respectively.

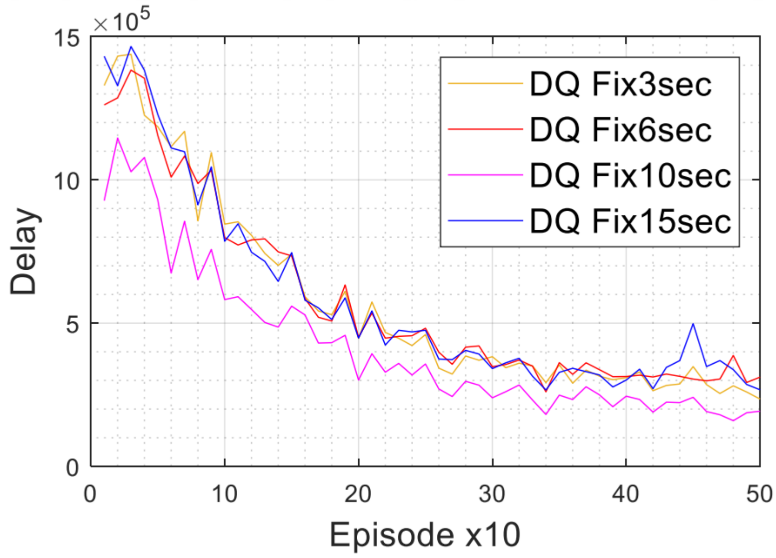


Figure 3.10 : Simulation results based on cumulative delay values according to fixed green light time values.

The simulation results are shown in Figure 3.8 to Figure 3.19. The x-axes in Figure 3.8 to Figure 3.19 represent the average value of each 10 samples or measures. Since there are 500 phases, the x-axes are divided into 50 points.

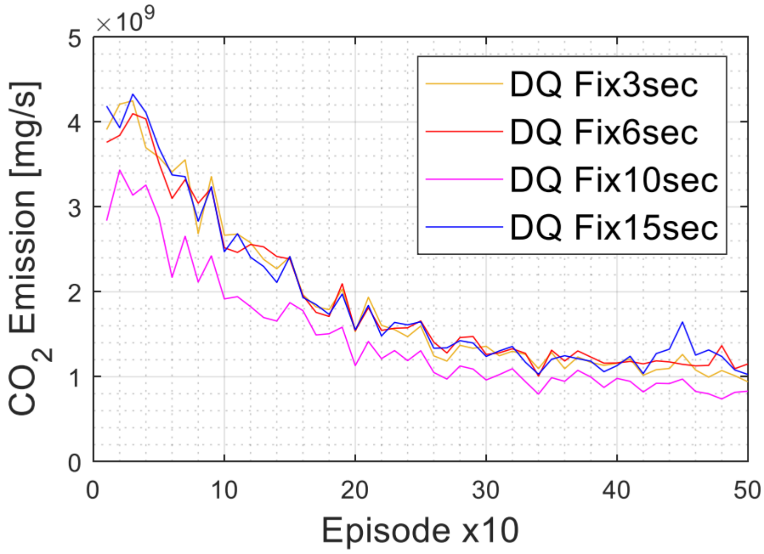


Figure 3.11 : Simulation results based on CO_2 emission output values according to fixed green light time values.

According to the fixed green light duration values, the simulation results are shown in Figure 3.8 to Figure 3.11. The values of the reward for various fixed time values are shown in Figure 3.8. The variation of queue lengths, delay values and CO_2 emission output values are shown in Figure 3.9, Figure 3.10 and Figure 3.11, respectively. As shown in Figure 3.8 to Figure 3.11, the results of DQ Fix 10sec are better compared to other fixed time green time values.

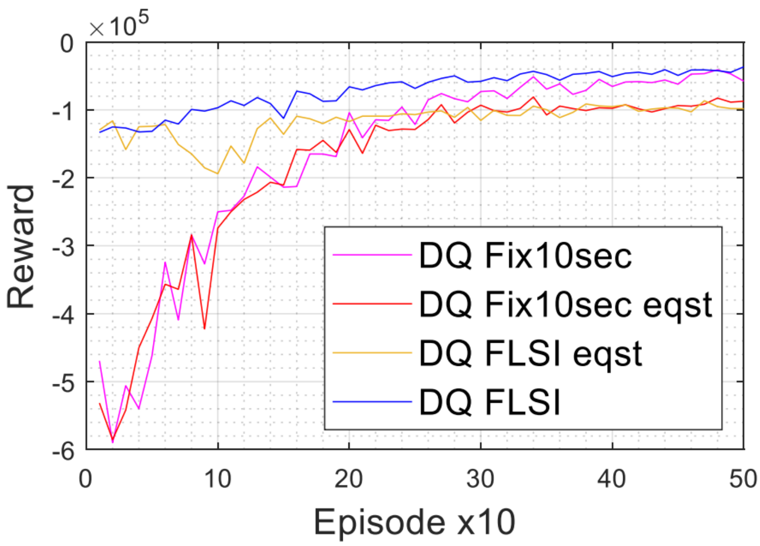


Figure 3.12 : Simulation results based on reward values according to different states.

According to different state values, the simulation results are shown in Figure 3.12 to Figure 3.15. The values of the reward for various fixed time values are shown in Figure 3.12. The variation of queue lengths, delay values and CO_2 emission output values are shown in Figure 3.13, Figure 3.14 and Figure 3.15, respectively. It is seen in Figure 3.12 to Figure 3.15 the method that is divided into changing length cells, depending on the proximity of the traffic intersection, is better than the method that is divided equally.

The simulation results obtained showed that, the simulation results of DQ FLSI and DQ FLSI eqst are better than all other methods. As seen in Table 5.3, although there is no learning algorithm, FLSI and FLSI eqst methods have better simulation results than fixed time direct deep Q learning methods.

Figure 3.13 : Simulation results based on total queue length values according to different states.

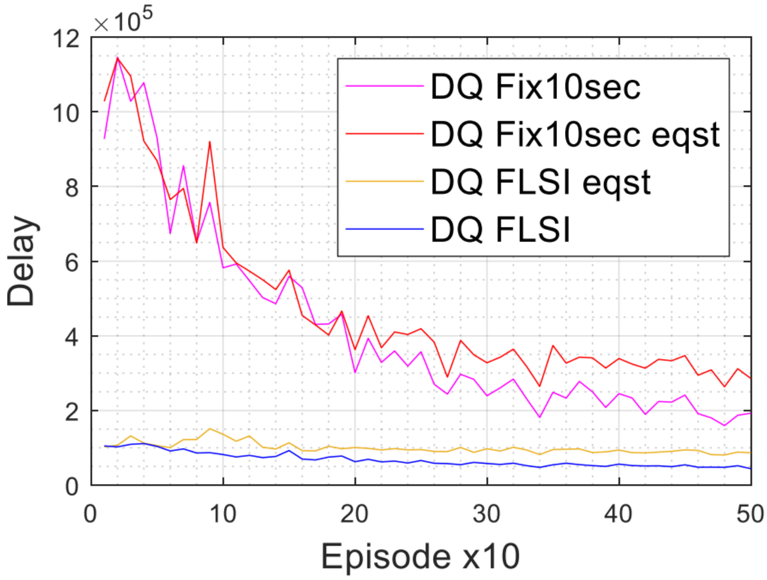


Figure 3.14 : Simulation results based on cumulative delay values according to different states.

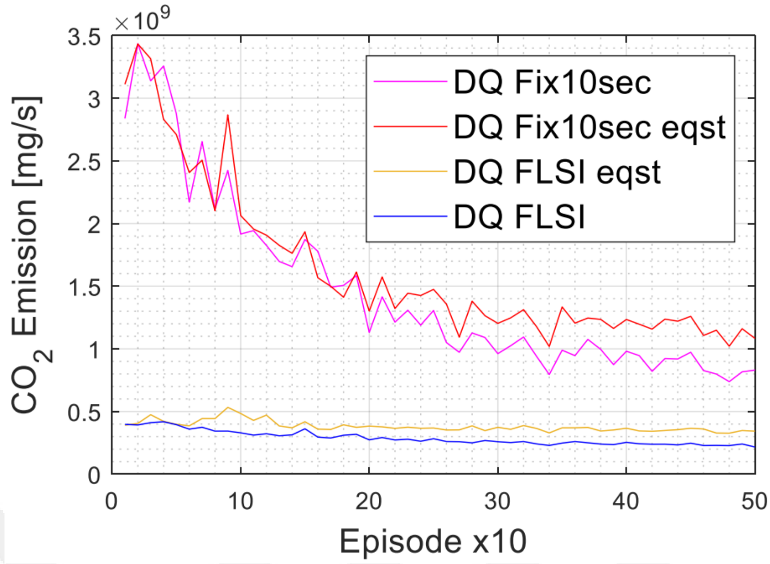


Figure 3.15 : Simulation results based on CO_2 emission output values according to different states.

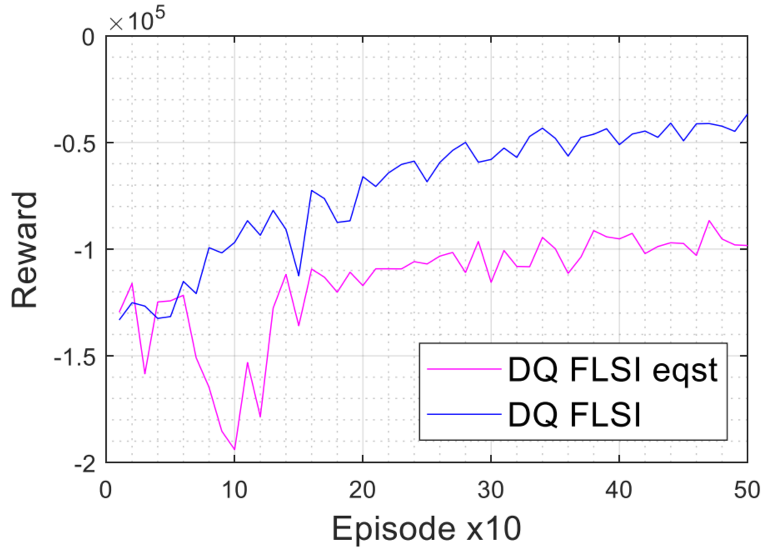


Figure 3.16 : Simulation results based on reward values for DQ FLSI and DQ FLSI eqst.

Figure 3.16 to Figure 3.19 show comparative simulation results for the DQ FLSI and DQ FLSI eqst methods. Also, using a flexible cell length instead of a fixed cell length gives better results.

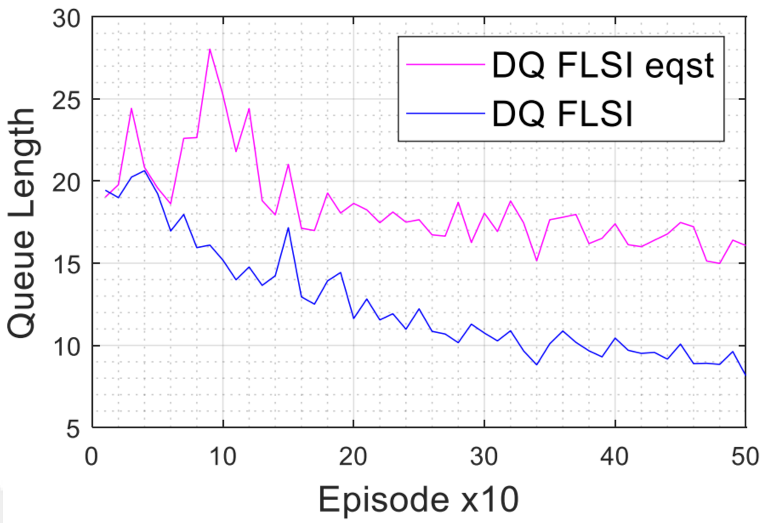


Figure 3.17 : Simulation results based on total queue length values for DQ FLSI and DQ FLSI eqst.

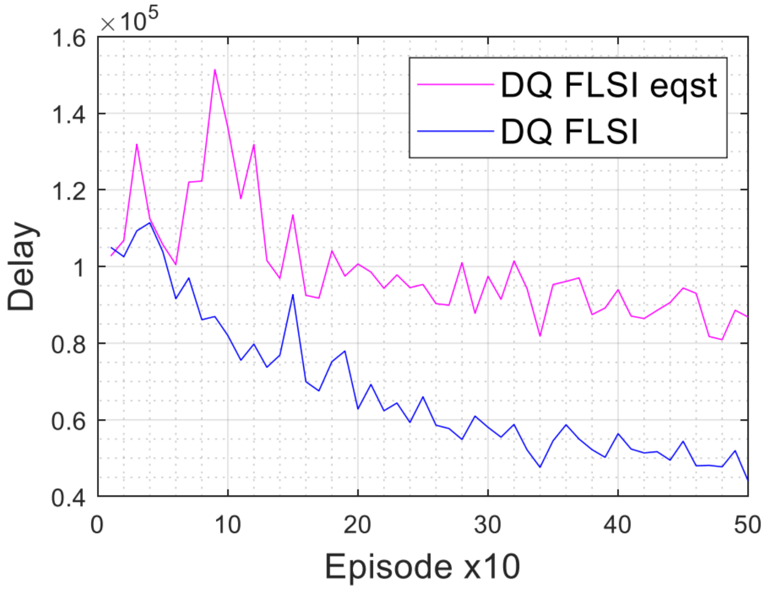


Figure 3.18 : Simulation results based on cumulative delay values for DQ FLSI and DQ FLSI eqst.

The stability analysis of the proposed traffic control systems will be handled in this section.

Let X(t) be the number of vehicles in the direction of arrival at the traffic light at time t.

$$P_n(n) = \lim_{t \to +\infty} P\{X(t) = n\} \tag{3.15}$$

In equation 3.15, P_n is the steady-state probability of exactly n vehicles in the intersection. The balance of the rates of vehicles entering and exiting the intersection for each state is shown in equation 3.16 (balance equation) [132].

$$\begin{gathered} \lambda P_0 = \mu P_1 \\ \lambda + \mu P_n = \lambda P_{n-1} \mu P_{n+1} \end{gathered} \tag{3.16}$$

In equation 3.16, λ and μ are the steady-state arrival rate and service rate of the traffic light, respectively.It is defined as seen in equation 3.17 and equation 3.18.

$$\lambda = \lim_{t \to +\infty} (\lambda)_t \tag{3.17}$$

$$\mu = \lim_{t \to +\infty} (\mu)_t \tag{3.18}$$

$$P_1 = \frac{\lambda}{\mu} P_0 \tag{3.19}$$

When we express P_n in terms of P_0;

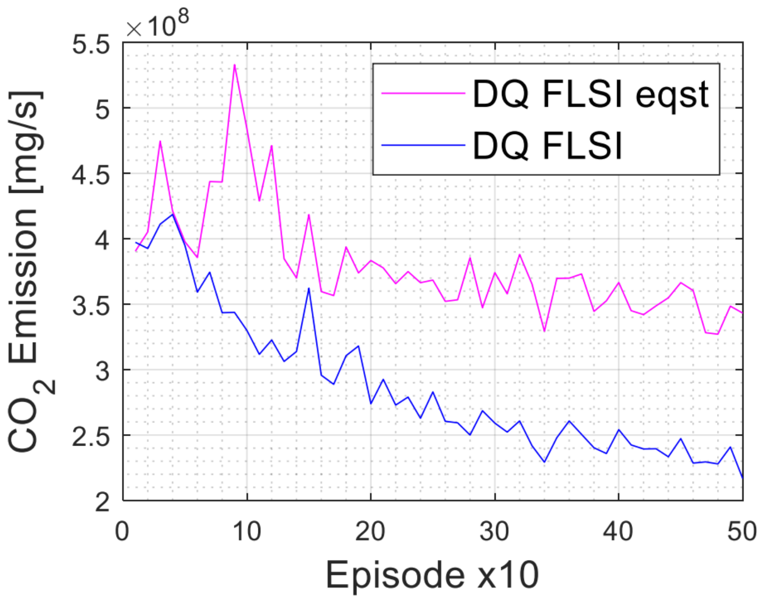


Figure 3.19 : Simulation results based on CO_2 emission output values for DQ FLSI and DQ FLSI eqst.

$$P_{n+1} = \frac{\lambda}{\mu}P_n + (P_n - \frac{\lambda}{\mu}P_{n-1}) = \frac{\lambda}{\mu}P_n = (\frac{\lambda}{\mu})^{n+1}P_0 \quad \textbf{(3.20)}$$

When we use the requirement that the sum of P_n be 1 when determining the expression P_0,

$$1 = \sum_{n=0}^{\infty} P_n = \sum_{n=0}^{\infty} (\frac{\lambda}{\mu})^n P_0 = \frac{P_0}{1 - \frac{\lambda}{\mu}} \quad \textbf{(3.21)}$$

or If we define it another way

$$\begin{aligned} P_0 &= 1 - \frac{\lambda}{\mu} \\ P_n &= (\frac{\lambda}{\mu})^n (1 - \frac{\lambda}{\mu}), \qquad n \geq 1 \end{aligned} \quad \textbf{(3.22)}$$

For the equation 3.22, λ/μ must be less than 1. The number of vehicles in the arrival direction of the traffic intersection at time t is calculated as in equation 3.23. Here c_1 is the number of vehicles at the beginning.

$$c_1 + \lambda t - \mu t = (\lambda - \mu)t \quad \textbf{(3.23)}$$

The ρ in equation 3.24 is the Utilization factor and as shown, in the equation it is equal to λ/μ [133].

$$\rho = \frac{\lambda}{\mu} \quad \textbf{(3.24)}$$

If ρ is greater than 1, the number of vehicles at the traffic intersection will constantly increase, and the number will go to infinity. So the stability of the system can be tested according to the value of ρ. The stability test simulation is performed for the method proposed for various arrival rate values. Simulation results are shown in Figure 3.20 to Figure 3.24.Figure 3.20, Figure 3.21, Figure 3.22, Figure 3.23, and Figure 3.24 show the simulation results for lambda values 937, 1887, 2831, and 4713, respectively. As seen in the simulation results for stability analysis, the proposed method, DQ FLSI, works much more efficiently. In Figure 3.24, despite the heavy traffic conditions, the traffic flow did not change to instability, although its performance decreased compared to other conditions. The simulation result for the behaviour of the proposed method for the case with very heavy traffic is shown in Figure 3.25. The system does not go into instability, even in the case of a relatively large arrival rate, thanks to the proposed control method.

Table 3.4 : Stability analysis

Method	Arrival Rate (λ) per hour	Utilization Factor (ρ)
DQ Fix10sec	937	0.975
	1887	0.986
	2831	1.342
	3762	1.775
	4713	2.043
DQ FLSI eqst	937	0,972
	1887	0.975
	2831	0.981
	3762	0.989
	4713	1.229
DQ FLSI	937	0.969
	1887	0,971
	2831	0.975
	3762	0.976
	4713	0.979

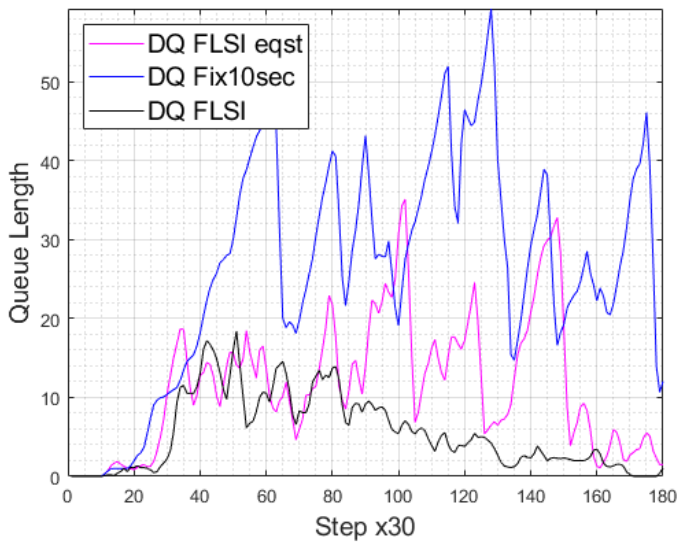


Figure 3.20 : For arrival rate of 937 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst

As seen in the Figure, the proposed method outperforms DQ FLSI results for all queue lengths. In addition, the changes in the ρ values of the methods against the λ values of equal amounts are seen in Table 5.4. For lambda range values used, the system is not unstable with the DQ FLSI method, but the system becomes unstable after $\lambda = 2831$ and $\lambda = 4713$ with the methods DQ Fix10sec and DQ FLSI eqst, respectively. In

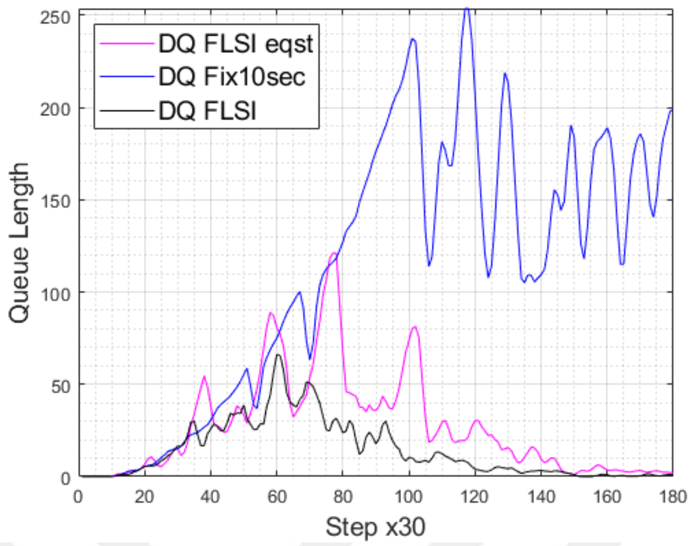


Figure 3.21 : For arrival rate of 1887 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst

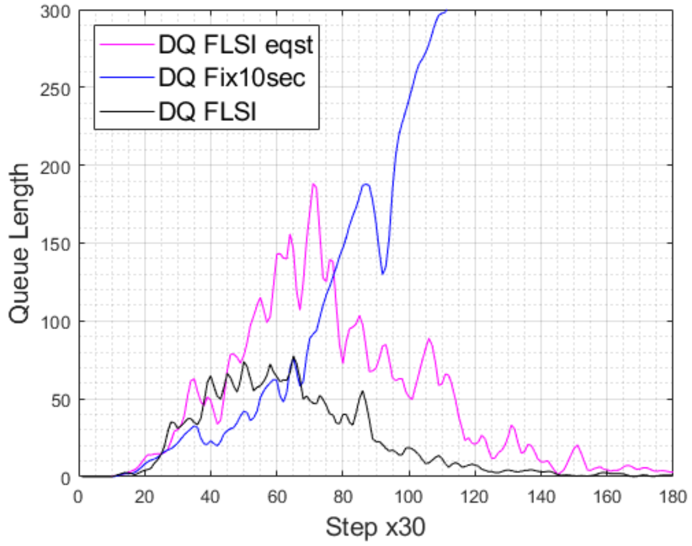


Figure 3.22 : For arrival rate of 2831 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst

addition, the Total queue length change according to different arrival rates is shown for the DQ FLSI method in Figure 3.25. As seen in the Figure 3.25 DQ FLSI method, the traffic light controller becomes unstable for $\lambda = 6608$.

Control methods trained for the two best methods are shown in Figure 3.26. Figure 3.26 demonstrates the distribution of the number of vehicles for the scenario. In the test scenario, vehicles are generated within a 17-hour period. Figure 3.27 showed the queue length result according to the DQ FLSI and DQ FLSI eqst methods for the test scenario. It is seen that the proposed method for a possible daily traffic scenario gives similar results in some traffic conditions, but mostly gives better results.

The simulation results show that the proposed DQ FLSI method has considerably better results compared to other methods existing in the literature. As a result of the effectiveness of the proposed method, there have been significant improvements in queue length, CO_2 emission output, and cumulative delay values. In addition, the advantages of the state matrix formed to be sensitive to the distance to the traffic intersection are also seen in the simulation results. The proposed method clearly demonstrates the advantage of using deep Q-learning in addition to fuzzy logic in the control of complex systems. In particular, it is shown that using the deep Q-learning method in the decision-making process for the selection of the traffic light phases and

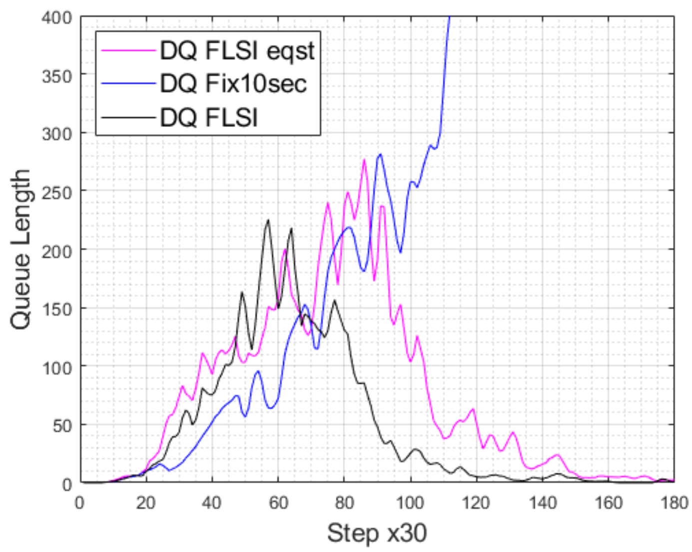


Figure 3.23 : For arrival rate 3762 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst

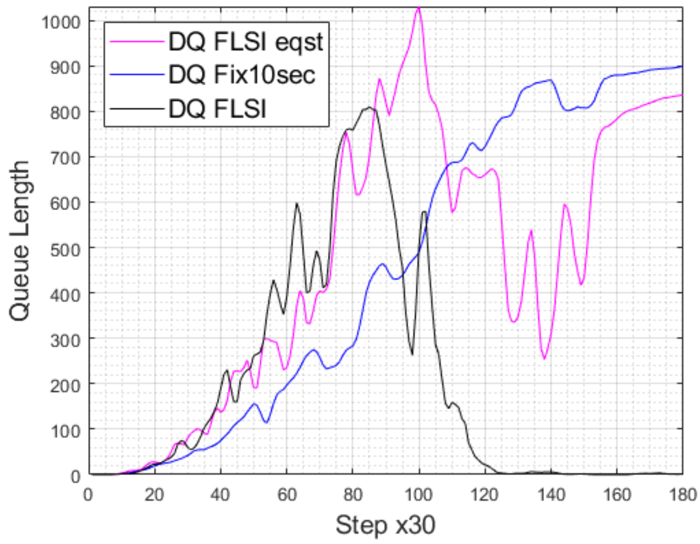


Figure 3.24 : For arrival rate 4713 per hour, total queue length values for DQ, DQ FLSI and DQ FLSI eqst

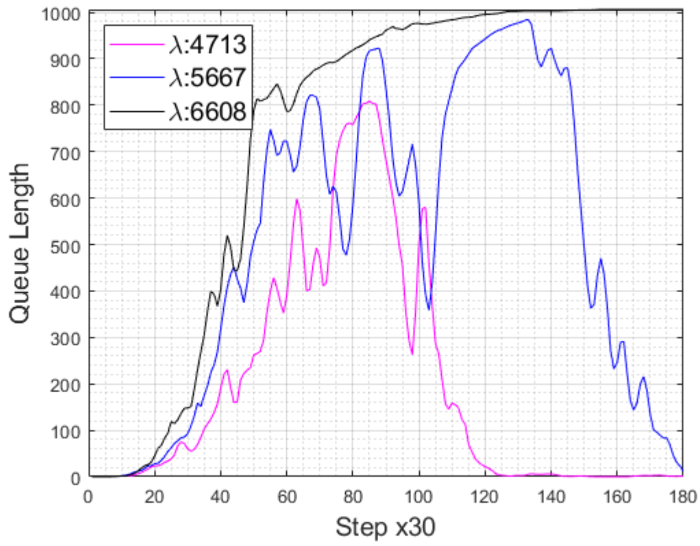


Figure 3.25 : Total queue length change according to different arrival rates for the DQ FLSI method.

using FLC in the determination of the duration of each phase is very effective. The findings of the study are supported by a theoretical stability analysis as well.

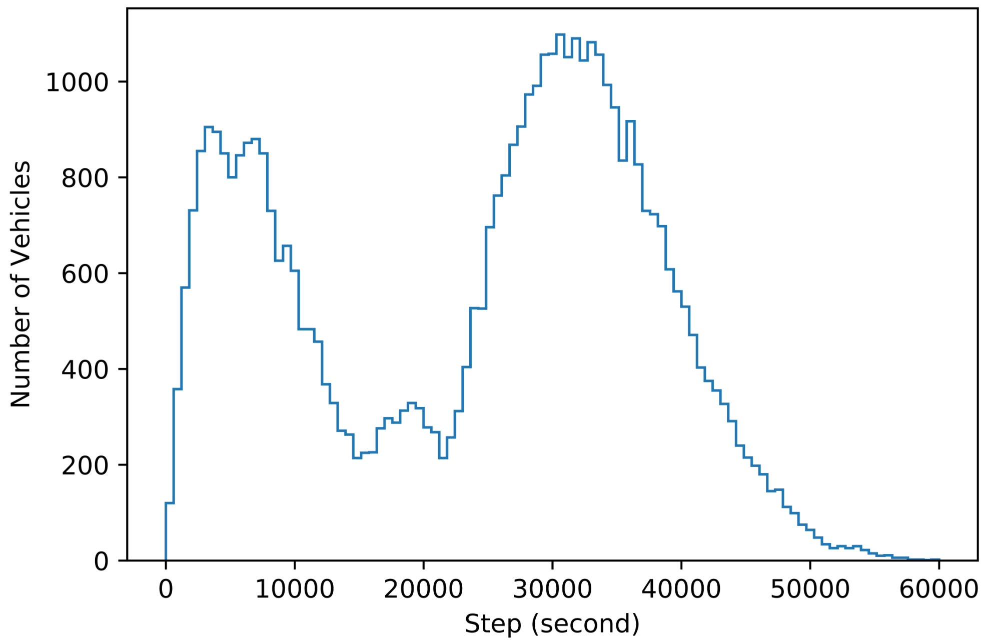


Figure 3.26 : Vehicle generation distribution for testing scenario.

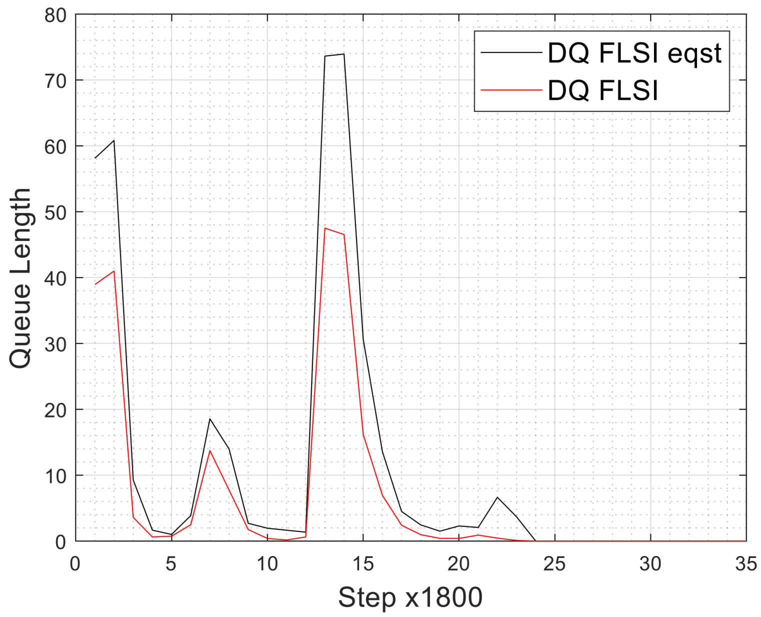


Figure 3.27 : Simulation results based on total queue length values for DQ FLSI and DQ FLSI eqst for test scenario.

4. AGENT-BASED ROUTE PLANNING WITH DEEP Q LEARNING

This section proposes a deep Q learning method to ensure optimum route planning of a fully autonomous taxi agent in an active traffic scenario. This section aims to reduce time in traffic by using an agent-based route planning method with deep Q learning. An agent which acts as a taxi in the generated traffic flow is also used to demonstrate the efficiency of the proposed method in taxi service. It is aimed to be able to comprehend the actions to be implemented in order to complete the given task in an effective way with deep Q learning, considering criteria such as travel time and waiting time for passengers as performance criteria in different scenarios. The study in this section includes excerpts from this publication [134].

4.1 Introduction

Since traffic control is a problem of sequential decision-making, it is one of the best suited to the reinforcement learning framework, in which agents learn through trial and error as they interact with their environment [135]. At this point, considering the complexity of traffic control, it would be appropriate to use the deep Q learning method, which solves more complex problems than reinforcement learning. It is known that agent-based studies have been applied to traffic lights before [136, 137]. Unlike other studies, route planning with deep Q learning is presented as a solution in this study. In particular, a taxi agent makes decisions to leave passengers most effectively with the feedback it receives from the environment. It is aimed to be able to comprehend the actions to be implemented to complete the given task most cost-effectively, with deep Q learning, considering criteria such as travel time and waiting time for passengers as performance criteria in different scenarios. For this purpose, a scenario is carried out in the region with active traffic flow, consisting of 12 roads.

In this study, the agent is considered as a taxi in the traffic scenario, and its performance is tried to be observed with different parameters. The taxi must reach the passengers

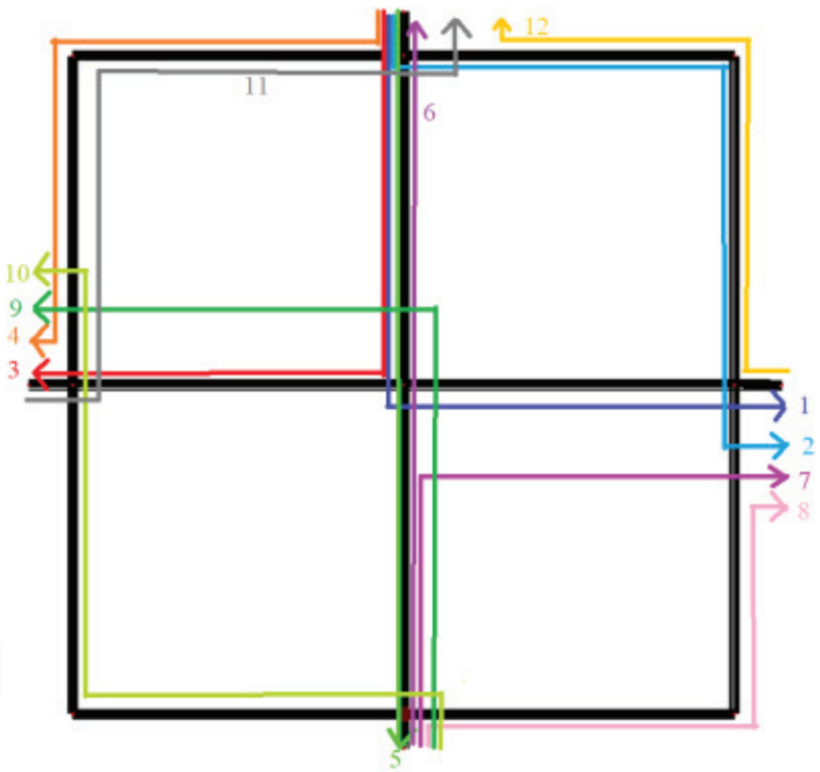


Figure 4.1 : Traffic flow scenario.

waiting to get on the vehicle from a certain location in the road network and take them to the point they want to get off. In the process of learning the optimum route, a performance change is observed when different state information is trained.

4.2 System Overview

In the SUMO simulation environment, 12 interconnected roads are created. All roads in the system have 4 lanes, and there are traffic lights at 5 intersections. The length of each road is 450 meters. The traffic scenario is generated with a separate file and produces a different traffic scenario for each section where learning will take place. This is a challenge the requested algorithm to overcome while it learns. When creating uncontrolled vehicles, it is aimed not to create the same density on every road on the map.

Because with less traffic on some roads, it is aimed for the agent to learn that they can go to their destination faster on these roads.

The vehicles are produced during the maximum simulation period given in the algorithm. A vehicle is randomly generated in one of the routes in Figure 4.1 with different start times in producing these vehicles. Continually producing vehicles on these routes ensures that the roads where the arrows pass more often have more density than the ones that pass less. For example, the roads leading to the intersection in the middle are more crowded than the side roads. The agent who learns to prefer the side roads reaches its destination faster. Thus, some streets remain relatively open, and the learning agent prefers these.

4.2.1 Deep Q learning model

Considering the complexity of the system, a 4-layer artificial neural network with 400 neurons in each layer is used. This neural network model is created using python language with Keras and TensorFlow libraries. In the input layer of the artificial neural network model, there are as many neurons as the state data in the proposed state vector. In the output layer, there are 3 neurons since the agent has the right to perform 3 actions. The output of these neurons corresponds to the predicted Q value for the relevant action. In cases where the agent does not have the right to explore, it takes the action corresponding to the largest Q value.

The ReLU function is chosen as the activation function of neurons. Computing costs are considered in this selection. Each batch contains 100 states. A batch learning technique is used to accelerate the learning process. In the learning process, the reward data collected from the environment during the simulation will act as feedback and train the parameters of the deep learning network according to the action required to be taught.

4.2.1.1 States

In the route planning problem, the necessary information for the taxi agent to make sense of the environment and choose an action is expressed with state vectors. The environment information sent to the learning model at any time t should be information that can help the agent achieve its goal. For this reason, the information that will be obtained from the environment is very critical and should be carefully selected. In the route planning problem, the taxi agent is planned to reach the destination in the shortest time possible. Therefore, the status information to be used should include information such as the traffic density on the roads, the location of the target point, and the location of the agent. In this study, the effects of different situational information on the learning process are tried to be observed experimentally. In this context, 13 status data are used, including six different status information containing the position information of the neighbouring roads, three status information expressing the vehicle congestion on the neighbouring roads, two coordinate status information indicating the location information of the agent, and two coordinate data for the position information of the target point. A state vector is created. It is observed that the traffic light also influences learning. Therefore, the light sequence of the road network is tried to be taught to the agent, and its effects on learning are observed.

Position data

Each position information will be taken as state information in 2D coordinate system.

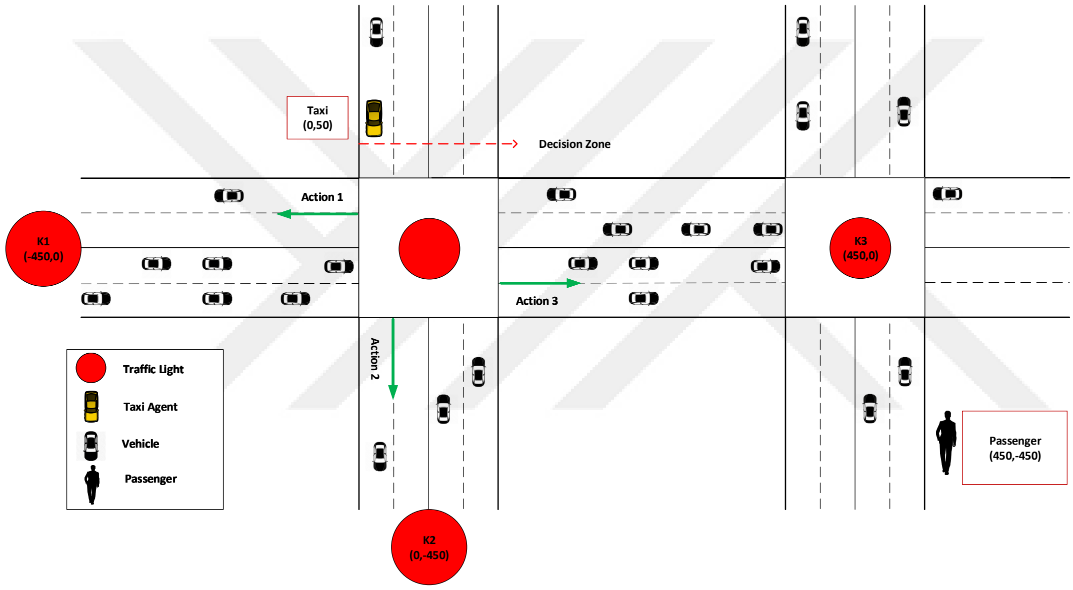


Figure 4.2 : Sample traffic scenario.

An example scenario is presented in Figure 4.2, which serves as an illustration of the system being analyzed. The associated state vector, which captures the state of the system at a given time, is provided in Table 4.1.

Traffic density of neighboring roads data

Traffic density information on neighbouring roads plays an important role in the agent's alternative route selections. In Figure 4.3, traffic density information from a sample scenario is shown. The length and shortness of the route are not the only factors affecting the time to reach the destination. It is affected by also how fast it is possible to travel on the route. In order to make sense of the relationship between the time to reach the destination and the traffic congestion on the neighbouring roads,

Table 4.1 : Sample state vector

	State Vector
1. neighbor x value	-450
1. neighbor y value	0
2. neighbor x value	0
2. neighbor y value	-450
3. neighbor x value	450
3. neighbor y value	0
Agent x value	0
Agent y value	50
Target destination x value	450
Target destination y value	-450

vehicle congestion information on the neighbouring roads should be added to the status vector.

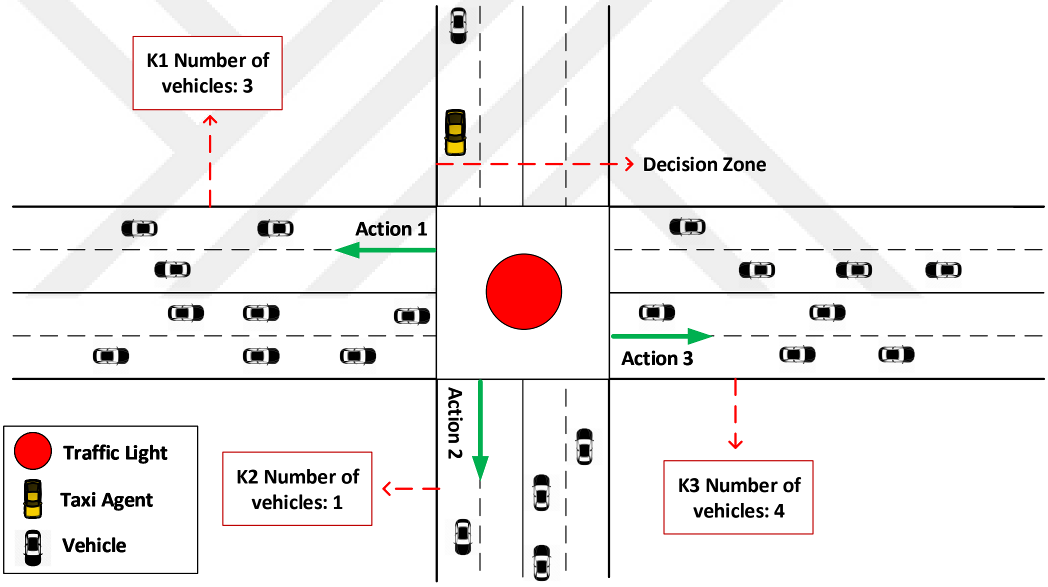


Figure 4.3 : Acquisition of traffic density information on neighbouring roads.

Traffic lights of neighboring intersections data

As can be seen from the example scenario in Figure 4.4, two pieces of data related to the traffic light come from each neighbouring intersection. These represent what the current traffic light is, and the time left to change. The taxi agent generates alternative routes by making sense of the traffic light sequences while going to the target point.

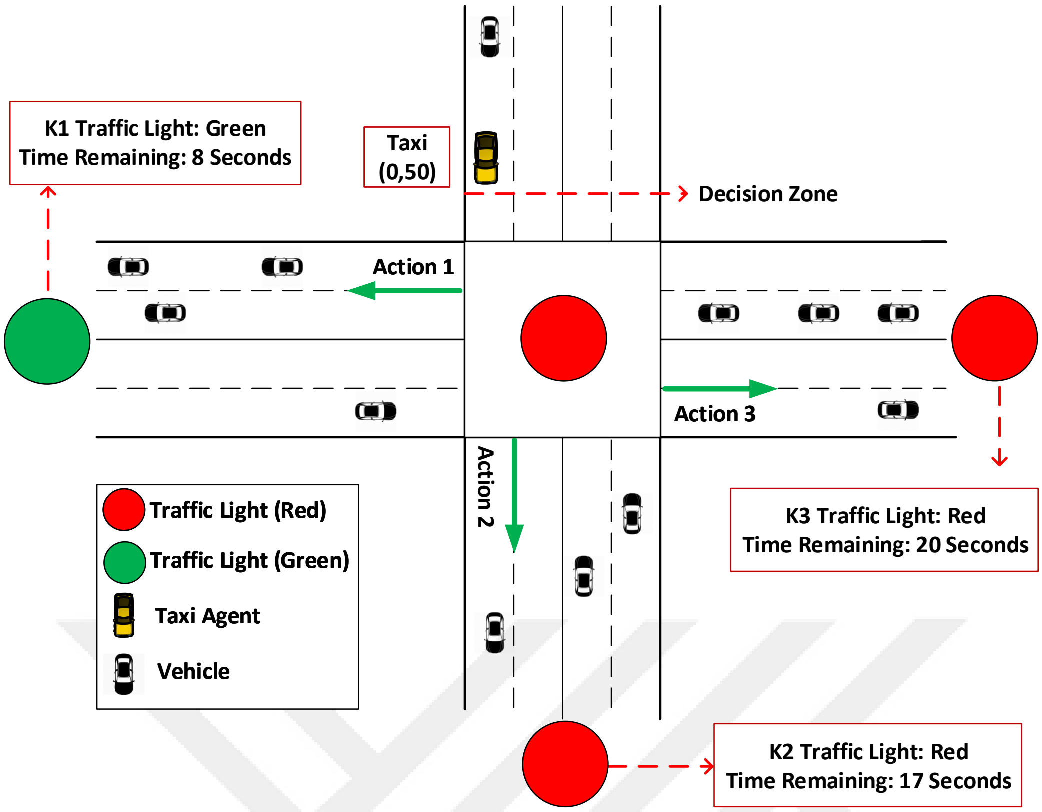


Figure 4.4 : Acquisition of traffic light information at neighbouring intersections.

4.2.1.2 Actions

There are three basic actions that the taxi agent must decide on at each intersection. These are respectively "turn right", “go straight” and “turn left”. These are the elements that make up the action set and these actions are predefined.

However, in the system shown in Figure 4.1, not all intersections are four-way. Therefore, the agent is not free to select all the elements of the action space when it comes to such intersections.

The Q value corresponding to this action from the deep learning network is effective for the agent choosing the action. However, since the environment that the agent is simulating is wide and open to exploration, it is aimed at the agent to discover ways that it has not tried before. Therefore, an adaptive exploration-exploitation method was used. A probabilistic variable ε is assigned, and its value is decreased in each simulation iteration. This means that the agent has the right to constantly discover new actions in the environment, up to the last simulation, high in the earlier simulations,

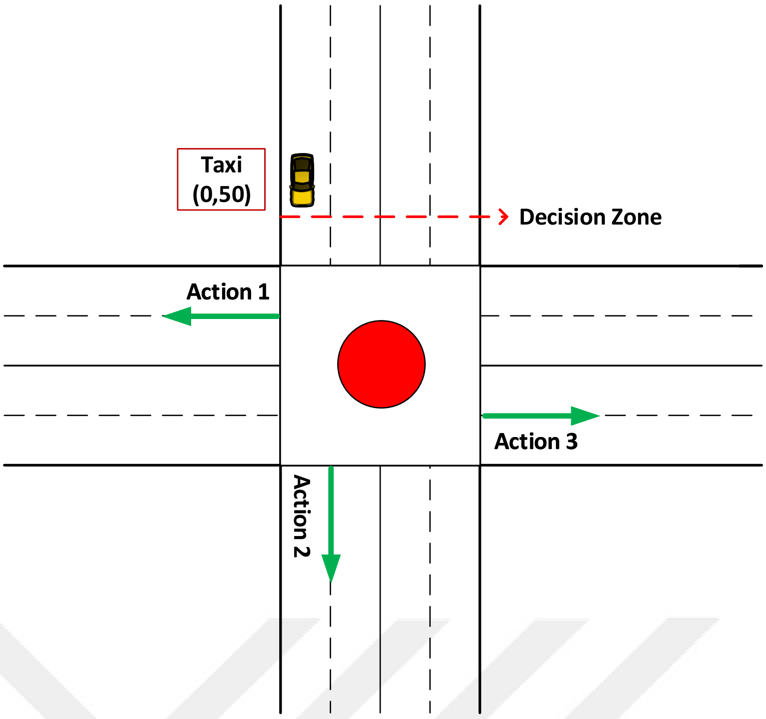


Figure 4.5 : Actions for a 4-way intersection.

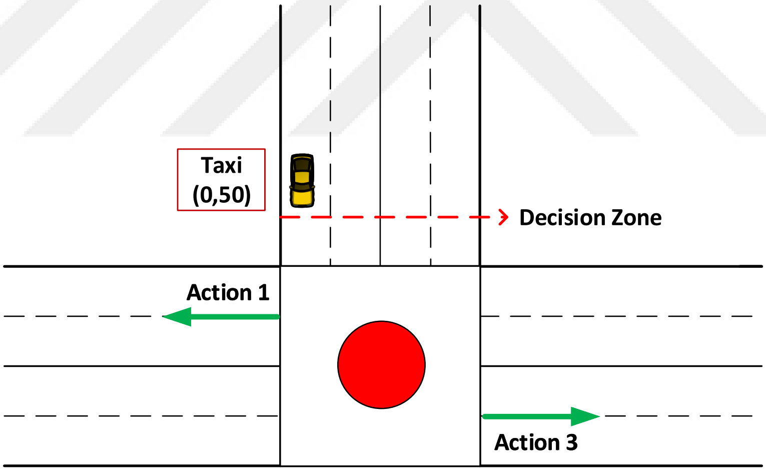


Figure 4.6 : Actions for a 3-way intersection.

and low in the later simulations. Thanks to the decreasing probability allocated for this exploration, it will continue to stay true to the parameters it learned in the last simulation iteration, and the final artificial neural network parameters will be found. Figure 4.5 and Figure 4.6 show the actions that the agent can choose for different intersection points.

4.2.1.3 Reward

The agent expects to take the waiting passenger to the desired location in the fastest way possible, and therefore, the elapsed time will be against the performance. Another performance expectation is to get as many passengers as possible to the desired location. Considering these performance criteria, a reward-punishment relation was proposed as in equation 4.1.

$$r(t_{sim}, t_{road}, a, \gamma) = -t - \gamma(\frac{t_{sim}}{T_{max}} p) \quad \textbf{(4.1)}$$

- t_{road}: Travel time of the agent on the current road
- t_{sim}:Simulation time
- p: Number of passengers waiting for a taxi
- γ: Waiting passenger weight coefficient
- T_{max}: Maximum simulation time

The reward relation given in equation 4.1 allows the vehicle to perceive the time it will spend on the road as a punishment in the learning process and enables the trained model to complete its task in the shortest time. Also, the reflection of waiting passengers on the reward correlation as a punishment provides feedback for the agent to reach the passenger as soon as possible.

The cumulative reward, on the other hand, proceeds by adding the reward value after each decision region and represents the reward-punishment value accumulated as a result of a certain sequence of actions. The cumulative reward value is updated after each decision-making step and stored in memory along with other state information to train the neural network model.

4.2.2 Simulation results

Figure 4.7 shows the starting position of the taxi and the passenger's getting on and off points.

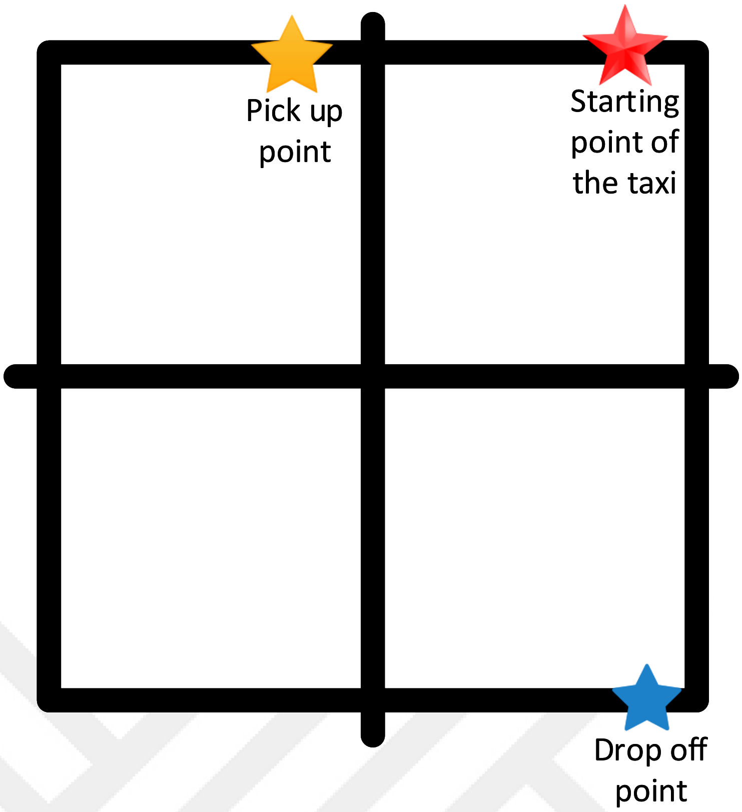


Figure 4.7 : The scenario of route planning.

Within the scenario, there are six passengers, and the taxi agent will learn to complete the routine of picking up all the passengers and dropping them off in the most optimal way.

As seen in Figure 4.9, with both techniques, the agent successfully completed the learning process and reached the maximum reward and minimum penalty level. As can be seen from the results in Figure 4.8 and Figure 4.9, learning is faster when traffic light information is included. With traffic light information, location and neighbour data become more meaningful.

Two different state vectors are used for the learning process of the agent. The optimum route is tried to be taught with the S_1 state vector given in equation 4.2 without using the traffic light information, and then the optimum route is taught using the traffic light information and the S_2 state vector given in equation 4.3.

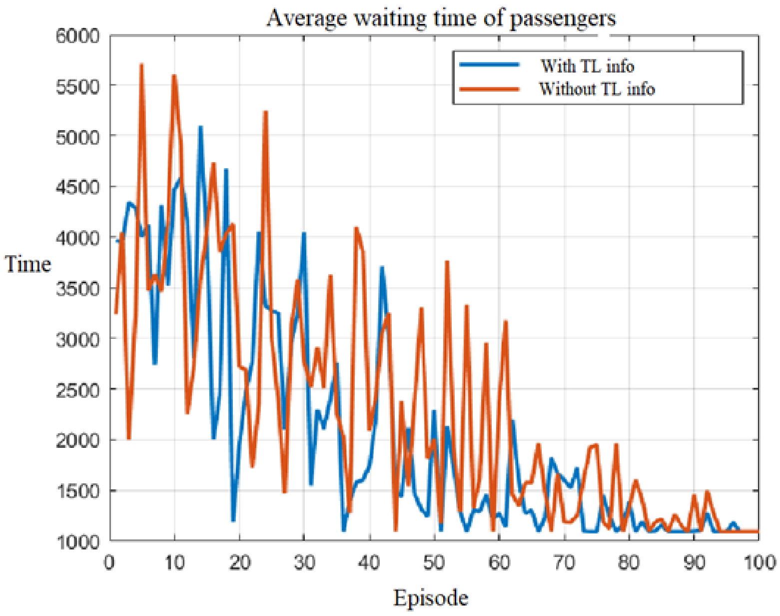


Figure 4.8 : Passenger average waiting times.

$$S_1 = \begin{bmatrix} K_1 \quad x\ position \\ K_1 \quad y\ position \\ K_2 \quad x\ position \\ K_2 \quad y\ position \\ K_3 \quad x\ position \\ K_3 \quad y\ position \\ Agent \quad x\ position \\ Agent \quad y\ position \\ Target \quad x\ position \\ Target \quad y\ position \\ K_1\ number\ of\ vehicle \\ K_2\ number\ of\ vehicle \\ K_3\ number\ of\ vehicle \end{bmatrix} \tag{4.2}$$

In the graph seen in Figure 4.8, the waiting time of the customers is minimized with both state vectors. However, as in the reward graph, learning is less oscillating with the state vectors containing traffic light information in the waiting time criterion.

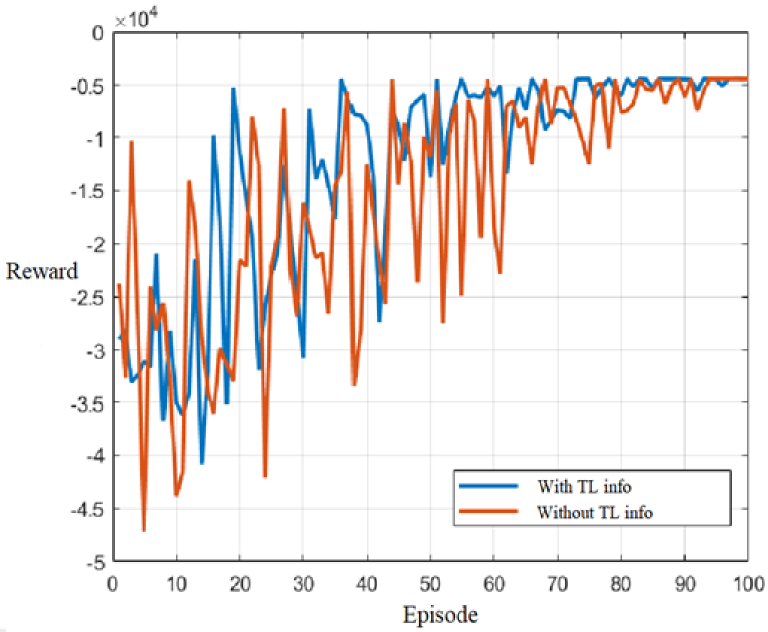


Figure 4.9 : Reward value of the agent for the number of episodes.

$$S_2 = \begin{bmatrix} K_1 \quad x\ position \\ K_1 \quad y\ position \\ K_2 \quad x\ position \\ K_2 \quad y\ position \\ K_3 \quad x\ position \\ K_3 \quad y\ position \\ Agent \quad x\ position \\ Agent \quad y\ position \\ Target \quad x\ position \\ Target \quad y\ position \\ K_1\ number\ of\ vehicle \\ K_2\ number\ of\ vehicle \\ K_3\ number\ of\ vehicle \\ K_1\ traffic\ light \\ K_1\ traffic\ light\ cycle \\ K_2\ traffic\ light \\ K_2\ traffic\ light\ cycle \\ K_3\ traffic\ light \\ K_3\ traffic\ light\ cycle \end{bmatrix} \quad \textbf{(4.3)}$$

One of the criteria considered as a performance criterion is the reward correlation, which is predefined. A balance between the waiting passenger and the elapsed time is also tried to be established. The taxi agent, which received feedback from the environment through this reward correlation, is put into a repetitive learning process in the simulation environment, and these simulations are repeated 100 times. Since another factor affecting the learning process is the situation information obtained from the environment, the learning performance of the agent was tested with different situation data. In this context, in addition to the common location and neighbouring road status information, the traffic light effect on the road network is tried to be taught to the agent. However, to look more closely, when traffic light information is added as state data in addition to location and neighbouring road information, it reaches the maximum reward value with a better graph and less oscillating characteristics. Another success criterion of the taxi agent is to plan the route in a way that minimizes the average waiting time of the customers. While repeating each route, the penalty value increases in the reward correlation of the waiting people with the elapsed time. The taxi agent that learns this situation ensures that the waiting people complete their route as quickly as possible. Considering all performance criteria; for the taxi agent's optimum route planning learning, it is recommended to use state vectors containing traffic light information as well as density information of neighbouring roads, location information of neighbouring intersections, location information of the agent and destination location information.

5. A PLATOON ORDERING ALGORITHM ON RESERVATION-BASED INTERSECTION MANAGEMENT SYSTEMS

Nowadays, the crowding of cities and the increase in traffic density affect human life negatively. To increase the quality of human life, it has become necessary to develop traffic control algorithms and perform traffic simulation studies. The efficiency of traffic control systems is highly dependent on traffic intersections as they are highly effective in the number of traffic accidents and traffic delays. Vehicles at the traffic intersection making right and left turns slow down while approaching the intersection and pass the intersection within the speed limits determined for the turn. Accordingly, the time spent by the vehicles at the intersection increases, and these vehicles cause the vehicles behind them to slow down as well. This study focuses on increasing the efficiency of the reservation-based intersection control of the platoons consisting of vehicles that will come to the intersection and turn in different directions. For this purpose, a reservation-based traffic intersection control algorithm was designed for a four-legged two-lane traffic intersection and a platoon ordering algorithm has been created to order the vehicles on the platoon approaching the traffic intersection as straight, right-turning and left-turning, respectively.

The reservation-based Multi-Agent Intersection Management (MSIM) and advantages of the platooning method with MSIM were investigated by considering parameters such as the time to cross the intersection and average speed. The main actors of the MSIM solution algorithm presented for intersection control are Vehicle Agents (VAs) and Intersection Agents (IAs). At each intersection, the IA oversees reservations in a period space premise to guarantee that two VAs do not be at the same intersection location spot simultaneously. This information, provided by IA, enables each VA to make the most efficient decision to accelerate, decelerate, cross the intersection or make a turn before they even reach the intersection area, and seamlessly implement it when it arrives at the intersection. Thus, unnecessary stop-start movements are avoided. In addition, it was desired to increase the effect of reservation-based intersection control by using the MSIM method and the platooning method together.

Vehicles coming to the traffic intersection making right and left turns slow down while approaching the intersection and pass the intersection within the speed limits determined for the turn. Accordingly, the time spent by the vehicles at the intersection increases and these vehicles cause the vehicles behind them to slow down as well. The grouping of vehicles that arrive at the traffic intersection in a platoon successively and mixed according to the direction of the turn, while coming to a reservation-based controlled traffic junction, in order according to the direction of the turn, affects the total time spent at the junction and the average speed of the vehicles. The vehicles entering the traffic intersection by arranging them as those going straight, turning right and turning left will reduce the total time spent by the vehicles at the traffic intersection and increase their average speed.

The simulation study was carried out in the SUMO environment, and the effect of the ordering algorithm according to different scenarios on the average speed of the vehicles in the platoon and the total time spent at the intersection was examined. By repeating the simulations without using the ordering algorithm, it has been observed that the ordering algorithm reduces the time spent by the vehicles at the traffic intersection and increases the average speed.

5.1 Reservation-Based Intersection Management

Reservation-based traffic intersection is designed with two lanes for each direction in order to allow the platoons to be sorted according to their turning directions before approaching the intersection. Reservation-based traffic intersection control is multi-agent, and the control system consists of an intersection agent (IA) and vehicle agents (VA). To implement reservation-based intersection control, the assumptions listed below are made:

- Vehicle agents are assumed to be SAE 4-5 level autonomous vehicles.
- Each of the vehicles has GPS to determine the location.
- Vehicles have the necessary network systems to communicate with the intersection agent.

In order to carry out reservation-based traffic intersection control, 800 meters of the roads that are connected to the intersection are included in the intersection area (Figure 5.1). The first 200 meters of the 800-meter parts of these roads are called the Communication Zone (CZ). Vehicles can communicate with the intersection agent within this region. The second 200 meters of the 800 meters parts of these roads are called Ordering Zone (OZ). In this section, the vehicles are ordering themselves among each other. The area 400 meters from the intersection is called the Action Zone (AZ). In this region, vehicles take appropriate actions to pass through the traffic intersection according to their reservations. In case of the possibility of the vehicles coming from different directions being in the same place at the same time, regions where vehicles are likely to have an accident occur at the traffic intersection.

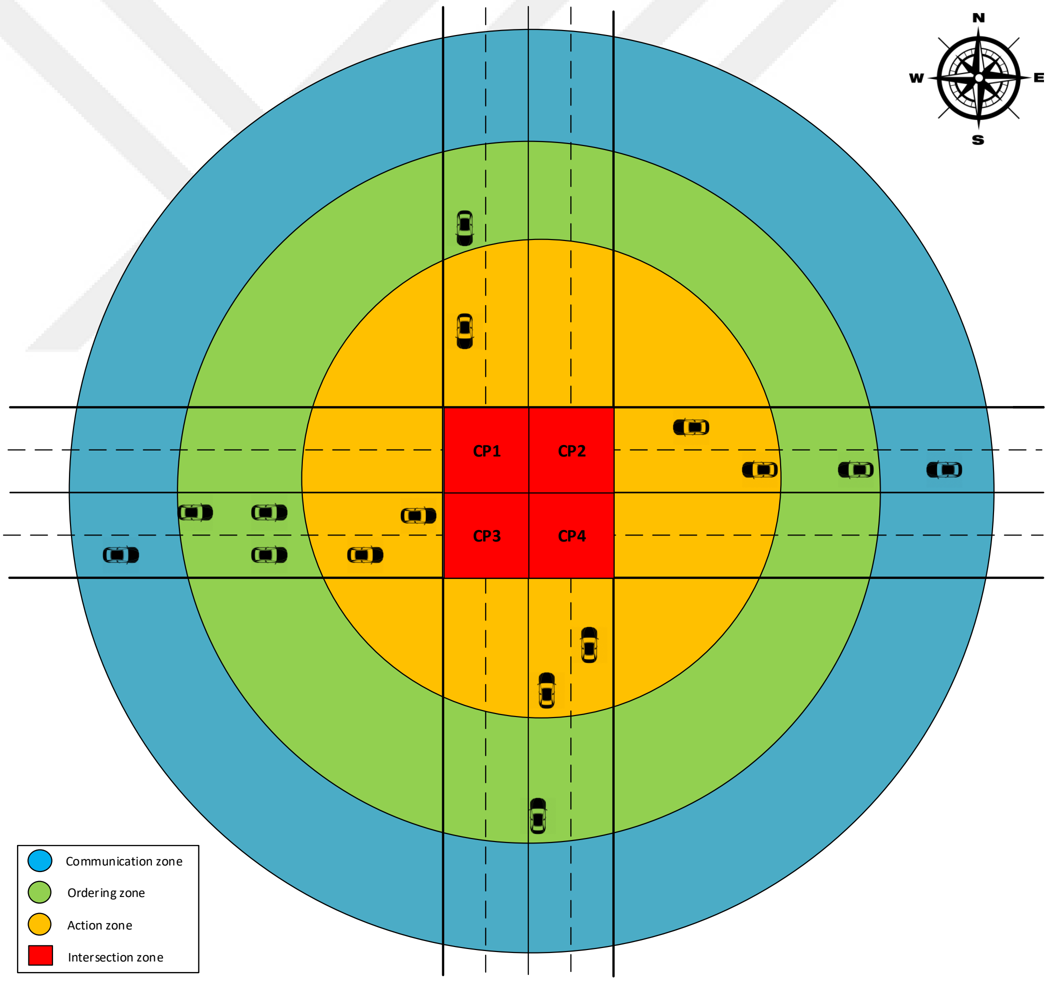


Figure 5.1 : Intersection zones.

The traffic intersection is divided into 4 regions for ease of calculation (Figure 5.2). Since the vehicles used in the study are autonomous and follow a certain route, there are 12 possible routes. For these routes, the regions that the vehicles will occupy while passing through the intersection have been determined and examined in a chart (Table 5.1). In this way, the regions where the vehicles entering the traffic intersection are at risk of collision with other vehicles can be observed clearly.

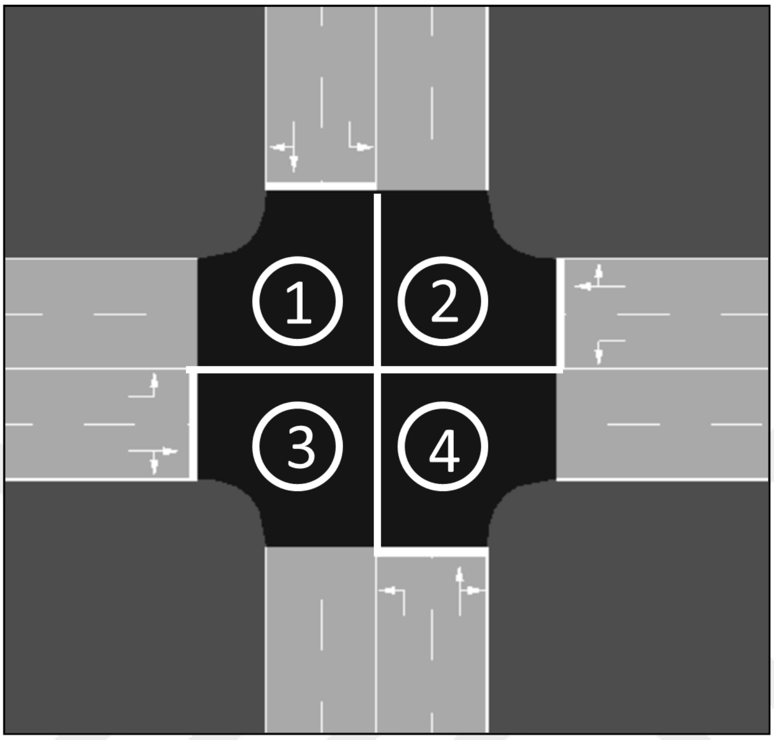


Figure 5.2 : Intersection regions.

Vehicle agents transmit information about vehicle ID, route information, vehicle cost, intersection regions to be occupied and the entry and exit time to the intersection regions to the intersection agent. Entry and exit times to the intersection regions are calculated by considering the length of the vehicle and the dimensions of the intersection regions. Using the intersection entry and exit points, the vehicle agent can calculate the times it will reach the intersection and occupy the intersection regions. The intersection agent, on the other hand, places the time that the vehicles will occupy in the reservation matrix, based on the entry and exit times, according to the information collected from the vehicle agents. After the reservation information is transferred to the vehicle agents, the vehicle agents pass through the intersection by adjusting their speed according to the reservation information. While making the reservation, the safe speed and acceleration limits determined for the vehicles are taken into consideration. In this way, it is aimed that the vehicles will pass through the intersection without stopping and without accidents.

Table 5.1 : Vehicle routes and the regions they occupy.

Routes	Region 1	Region 2	Region 3	Region 4
Route 1			x	x
Route 2			x	
Route 3	x	x	x	
Route 4	x	x		
Route 5		x		
Route 6		x	x	x
Route 7		x		x
Route 8				x
Route 9	x		x	
Route 10	x		x	
Route 11	x			
Route 12		x		x

5.1.1 Intersection agent

The reservation process is performed by writing the occupancy times of vehicles into the reservation matrix. The reservation matrix consists of a total of four rows representing the four intersection regions. In order to carry out the reservation process, information must be collected from the vehicle agents. At this stage, the distance between the vehicles is checked and whether the vehicles can pass in convoy and the cost of this situation is checked. In cases where it is possible to form a convoy, the vehicles are evaluated as a single vehicle by the reservation agent. After this stage, if there is no conflict between the vehicle agents, the reservation process is made by recording the entry and exit times to the intersection regions and the vehicle ID in the reservation matrix. According to the information from the vehicle agents, if there is a conflict in any of the intersection regions, a prioritization process should be carried out by the intersection agent, since conflicts will cause possible accidents. For this process, the costs of the vehicles are used. Priority is given to the high-cost vehicle. If the costs are equal, the sorting process is carried out according to the IDs of the vehicles. In case of a conflict, the reservation agent transmits the updated reservation information to the vehicle agents. The reservation matrix is constantly updated and it is aimed to renew the reservation process in case of any problems. An example of how the reservation matrix is filled is given in Table 5.2. According to the reservation matrix, Vehicle 1 (veh1) follows Route 12 and occupies Intersection Regions 1 and 2.

At time t1, the vehicle enters region 1. At time t2, the vehicle exits region 1 and enters region 2. At time t4, the vehicle exits Intersection Region 2 and leaves the intersection. The residence time of the vehicles in the intersection areas is calculated by the distance of the vehicles to the intersection area and their instantaneous speed.

Table 5.2 : Vehicle routes and the regions they occupy.

	t1	t2	t3	t4	t5	t6	t7	t8	t9
Region1	Veh1	Veh1		Veh2	Veh2				
Region2			Veh1	Veh1		Veh2	Veh2		
Region3								Veh2	Veh2
Region4	Veh3	Veh3	Veh3						

5.1.2 Vehicle agent

Vehicle Agent, communicates and works together with Intersection Agent. The first duty of VA is to sense the environment, this action returns the vehicles' ID's, intersection zones, costs, possible arrival and departure times. If there is an intersection on the route of the vehicle, this information is shared with that IA. The shared information includes reservation points to be occupied, present road conditions and leader VA's data. Figure 5.3 show the communication architecture of the Multi-Agent System.

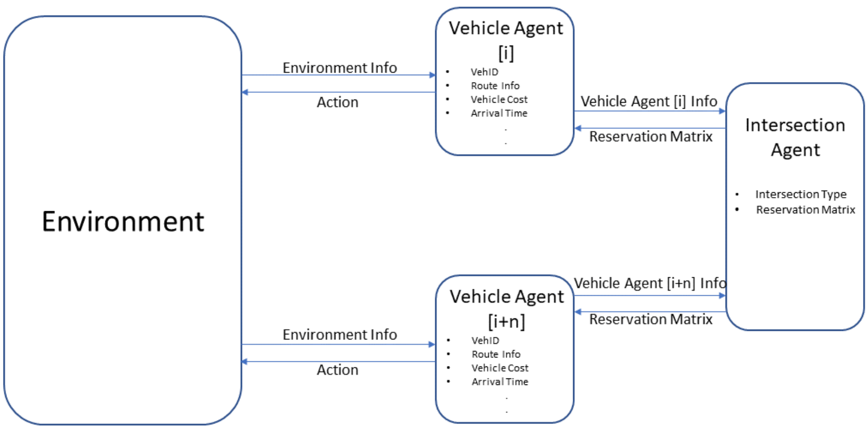


Figure 5.3 : Communication architecture of multi-agent system.

The vehicles that are outside of the Action Zone are controlled by the ACC method that SUMO provides. The method is the Krauss Model. After travelling into the AZ, the vehicles switch to ICACC method. Figure 5.4 shows the flowchart of the vehicle agent's operation.

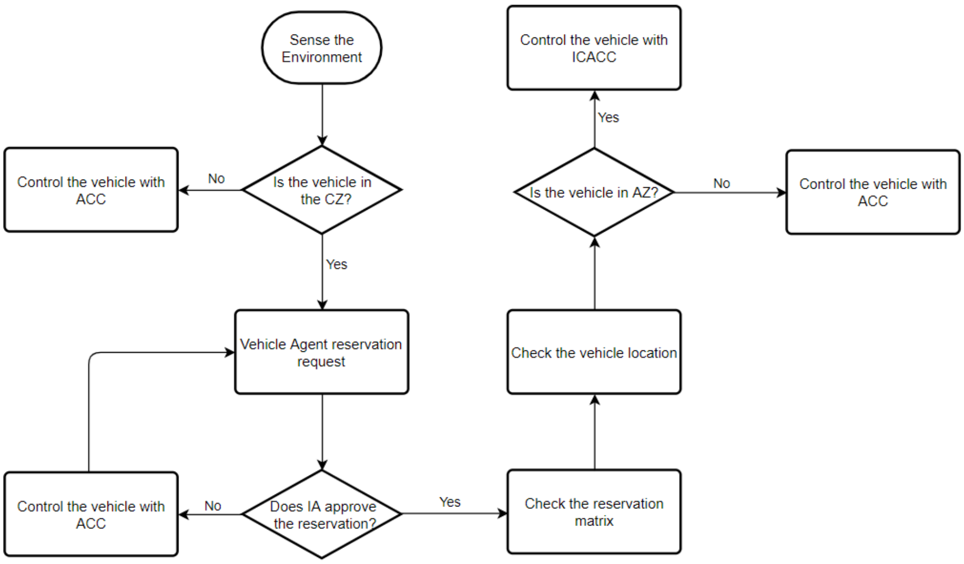


Figure 5.4 : Vehicle agent control algorithm.

With these definitions the duties of the VA can be divided into two. The first one is sending a reservation request to the IA and the second one is to manage and decide the control methods mentioned above.

5.1.2.1 Vehicle agent cost

SUMO has lots of vehicle types to choose from. These vehicles can be internal combustion vehicles or electric vehicles. In this paper the vehicles are chosen as electric powered vehicles. Because of this choice, the cost function is constructed with kinetic energy and waiting time minimization in mind. As seen below the cost function takes kinetic energy and waiting time as parameters. In equation 5.1 the calculation of the vehicle cost can be seen.

$$Cost = (\frac{1}{2}(v_{nrml}^2))(t_{dly} - t_{nrml}) \quad \textbf{(5.1)}$$

5.1.2.2 Vehicle agent reservation request

When the vehicles enter the CZ, the IA shares the vehicles' zones and reservation information with VAs. According to this info, the VA determines possible reservation points by taking possible arrival and departure times, vehicles' distances to the arrival and departure points, speeds and vehicles' lengths into account. If these points are not available, VAs make reservations to the nearest point that allows vehicles' speed control. The speed control is done by aiming for efficient braking and accelerating.

After determining the occupied reservation points, the VA creates a reservation request for these points. This request includes an arrival and a departure time info; this time information is calculated by taking duration, location, current speed and the location of the intersection data into account. In equation 5.2 and equation 5.3, the calculation of these arrival and departure time information can be seen.

$$t_{a[i][j]} = \frac{x_{a[i]} - d_{[j]}}{v_{cur[j]}} \tag{5.2}$$

$$t_{d[i][j]} = \frac{x_{d[i]} - d_{[j]} + l_{[j]}}{v_{cur[j]}} \tag{5.3}$$

5.1.2.3 Vehicle agent action

VA determines the actions to be taken by checking if the vehicle is in CZ or not. This knowledge is provided by IA. VAs, neither can make reservations nor receive info about the other reservations until travelling into the CZ. Vehicle speed is controlled by ACC between the AZ and the IZ. If there is no other vehicle in front of the vehicle, the speed of this vehicle is set to the maximum speed limit of the road. If there is any other vehicle in front, the ACC controls the speed of this vehicle. If the VA enters the AZ with a successful reservation, it determines the further actions with using the reservation information that the IA provides.

5.1.2.4 Intelligent connected adaptive cruise control

When the VA receives the reservation information from the IA, ICACC is initiated. This controller sets acceleration limits for the vehicles and the speed of the vehicle is changed according to these limits. Acceleration limits makes the speed change smoother, thus achieving maximum passenger comfort and minimum energy loss. After the vehicle is slowed down, it keeps its speed constant until it reaches the intersection.

5.2 Platoon Ordering Algorithm

Vehicles at the traffic intersection making right and left turns slow down while approaching the intersection and pass the intersection within the speed limits determined for the turn. Accordingly, the time spent by the vehicles at the intersection increases and these vehicles cause the vehicles behind them to slow down as well. Ordering the vehicles approaching the traffic intersection in a platoon form according to the direction of the turn, has an effect on the total time spent at the intersection and the average speed of the vehicles. As a result of the observations that were made, in order to facilitate the reservation of vehicles arriving at a reservation-based intersection, the optimal order of the vehicles was determined as straight, right turn and left turn. The working principle of the algorithm is briefly shown in Figure 6. The following assumptions are made in order to sort the vehicles approaching the intersection:

- Vehicles on the platoon enter the ordering zone from the right lane while approaching the intersection.
- To facilitate the reservation process, it is aimed that vehicles that turn left come to the intersection from the left lane, while vehicles that go straight and turn right stay in the right lane.
- After the vehicles are ordered according to their turning directions, they regroup, and the vehicles that turn in the same direction decrease their following distances again.

Under these assumptions, the following steps were followed to sort the vehicles:

1. When the vehicles in the platoon enter the ordering zone, their vehicle IDs are recorded through detectors.

2. By taking the route information of the vehicles, their turning directions are determined.

3. Vehicles that turn right and left switch to the left lane.

4. After the completion of the first lane change, the position information of the first of the vehicles that turn right and of all the vehicles that go straight is obtained and compared.

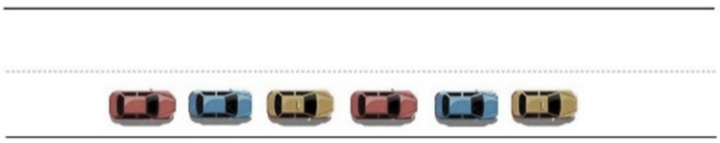
5. If the first vehicle that turns right is ahead of the last vehicle that goes straight, the vehicles that will turn right and those that will turn left slow down and the vehicles that go straight will move forward.
6. When the vehicles that go straight pass a certain distance ahead of all the vehicles that will go to the right, they pass back to the right lane and get behind the vehicles that go straight.
7. After ensuring that the vehicles that turn left are behind the vehicles that turn right, the vehicles that turn right and those that turn left accelerate again.
8. After the ordering process is completed, all vehicles catch the front vehicle that turns in the same direction, reduces the following distance and become platoon again.
9. In this way, vehicles that turn left are grouped in the left lane, vehicles that turn right and vehicles that go straight are grouped in the right lane sequentially.
10. The algorithm is applied to vehicles coming from all directions.

Algorithm 4 Platoon Ordering

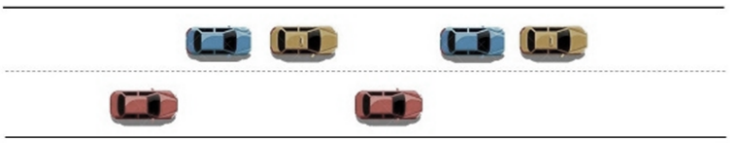
```
if platoon is already ordered then
    break
    end
for i in range (platoon size) do
    route = take vehicle's route (vehicle(i))
    if route==left || route==right then
        ChangeLane (vehicle(i))
        end
    if first of right is behind last of straight then
        ChangeLane (right)
    else
        SlowDown (right,left)
    end
    if platoon is ordered then
        CatchTheLeadingVehicle (all vehicles)
        end
    end
```

5.3 Simulation results

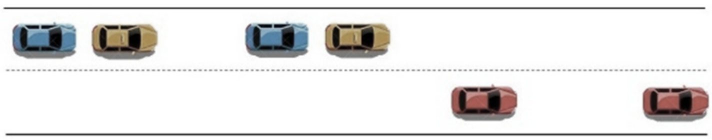
For establishing a simulation environment, a four-legged two-lane intersection is created using SUMO. Python is used to manipulate the traffic flow and to control the vehicle actions. The simulation's time step is set to 100 milliseconds. At every



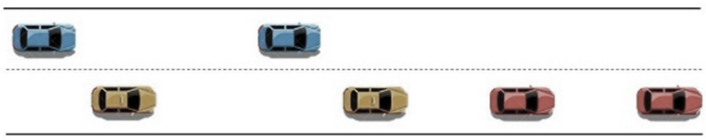
(a) Vehicles approach the traffic intersection and are mixed according to their turning directions.



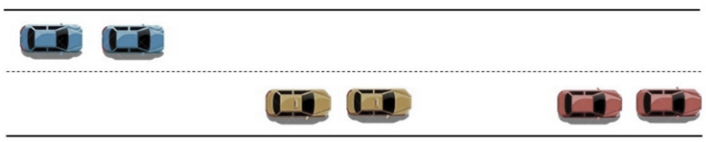
(b) Vehicles that make right or left turn pass to the left lane.



(c) Vehicles going straight pass ahead of other vehicles.



(d) Vehicles making a right turn pass back to the right lane.



(e) Vehicles are ordered according to their routes.

Figure 5.5 : Ordering of the vehicles in a platoon

timestep, VAs that are inside the CZ, send reservation requests, then IA takes these requests and fills the reservation matrix. The vehicle type used in this simulation is an electric car. All the roads that merge into the intersection have a speed limit of 20 m/s. To observe the effects of the platoon ordering algorithm on the reservation process and the vehicles, the simulations are run for two different cases with multiple scenarios. From these simulation results, the time that vehicles spent in the intersection zone and the average speeds of vehicles are taken. Intersection time data is acquired by measuring the time passed between the entry of the first vehicle in the platoon to CZ and the exit of the last vehicle in the platoon from the intersection regions.

5.3.1 Case 1

For Case 1, two platoons approaching the intersection from both east and west are created. These platoons consist of six vehicles each, and they include two-vehicle groups that have three different routes to take. Six different scenarios are created depending on where these two-vehicle groups are in the platoon. The simulations are run for all the scenarios with using the platoon ordering algorithm and without using the ordering algorithm. The following data are acquired from these simulations. In Table 5.3, the routes of the vehicles belonging to the platoons before being ordered by the ordering algorithm are given. The simulation results are listed in Table 5.4, in Figure 5.6 and Figure 5.7, there are column graphs regarding the time spent at the intersection and the average speed of the vehicles.

Table 5.3 : Routes of the vehicles in different scenarios

Scenario	Platoon Order
1	Straight(2) - Right(2) - Left(2)
2	Straight(2) - Left(2) - Right(2)
3	Right(2) - Straight(2) - Left(2)
4	Right(2) - Left(2) - Straight(2)
5	Left(2) - Straight(2) - Right(2)
6	Left(2) - Right(2) - Straight(2)

From the simulation results, it can be seen that the platoon ordering algorithm lowered the time passed in the intersection. Without using the ordering algorithm, the vehicles in front of the platoon that are going to take a left or right turn at the intersection slow down all the vehicles behind, especially the ones that have the straight route. This results in high intersection times.

It is obvious that the slow processing of the vehicles without the ordering algorithm also results in lower average speeds of the vehicles. It is shown in the average speed graph that using the ordering algorithm, it is possible to speed up the reservation-based intersection system.

Table 5.4 : Intersection time and average speed data in simulations with ordering and without ordering for case 1.

	With Ordering		Without Ordering	
Scenario	Intersection Time (s)	Average Speed (km/h)	Intersection Time (s)	Average Speed (km/h)
1	77.8	65.92	67.7	65.88
2	83.7	65.90	79.9	66.15
3	80.5	66.55	90.3	60.20
4	79.3	66.04	98.7	55.2
5	79.1	65.62	98.7	55.19
6	79.5	65.65	175.8	54.11

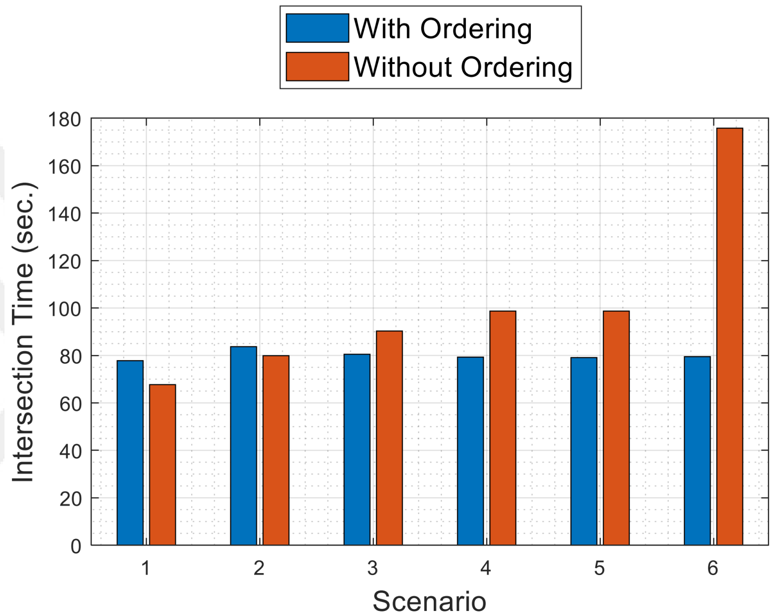


Figure 5.6 : Intersection times with ordering and without ordering for case 1.

5.3.2 Case 2

For the simulation of the second case, two six-vehicle platoons are created with random routes. These platoons are approaching from the east and west of the intersections like in the first case and random traffic is added from the north and the south of the intersection. With this configuration, six scenarios are picked for generating the results and graphs. In Table 5.5, the routes of the vehicles belonging to the platoons before being ordered by the ordering algorithm are given. The simulation results are listed in Table 5.6, in Figure 5.8 and Figure 5.9, there are column graphs regarding the time spent at the intersection and the average speed of the vehicles.

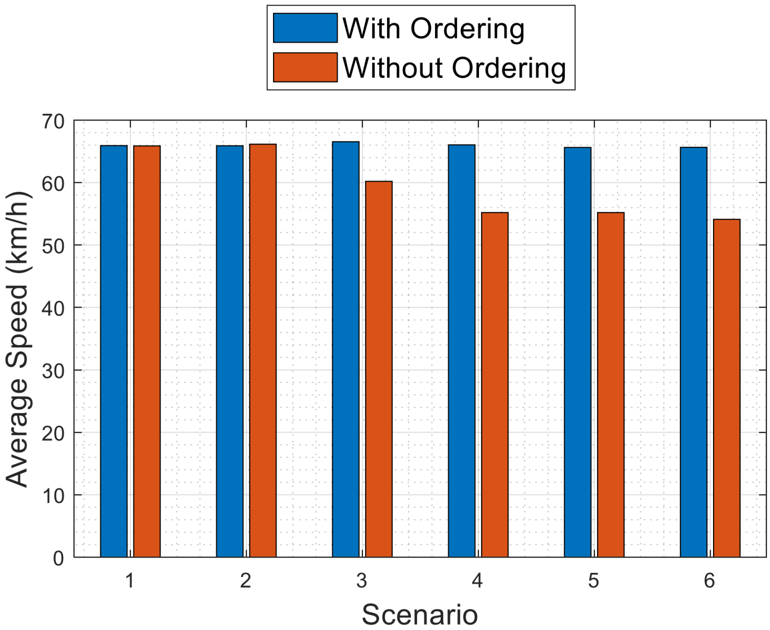


Figure 5.7 : Average speeds with ordering and without ordering for case 1.

Table 5.5 : Routes of the vehicles in different scenarios

Scenario	Platoon 1	Platoon 2
1	Left(2) - Straight(2) - Right(2)	Left(2) - Straight(2) - Right(2)
2	Straight(2) - Right(2) - Left(2)	Straight(2) - Right(2) - Left(2)
3	Straight-Left-Straight-Right-Left-Right	Right-Straight-Right-Straight-Left-Left
4	Left-Straight-Right-Right-Straight-Left	Left-Straight-Right-Straight-Right-Left
5	Straight-Left-Straight-Right-Left-Right	Right-Straight-Right-Straight-Left-Left
6	Right-Left-Straight-Right-Left-Straight	Straight-Right-Left-Straight-Right-Right

This case reveals the importance of using the ordering algorithm when there is traffic presence outside of platoons. In the 1^{st} and 2^{nd} scenarios, two platoons with symmetrical routes were created on two opposite roads. Apart from these platoons, 8 vehicles come from other roads. The routes and departure times of these vehicles are random. In the remaining scenarios, the routes of the platoons are mixed. The effect of using the ordering algorithm in these scenarios is clearly visible.

It has been observed that when ordering is not used, vehicles that go straight are slowed down due to vehicles that turn right and left, while random vehicles from other directions increase this deceleration even more. When the ordering algorithm is used, this slowdown is prevented, and the effect of random vehicles is minimized by the priority given to vehicle groups that will go in the same direction. Figure 5.3 shows

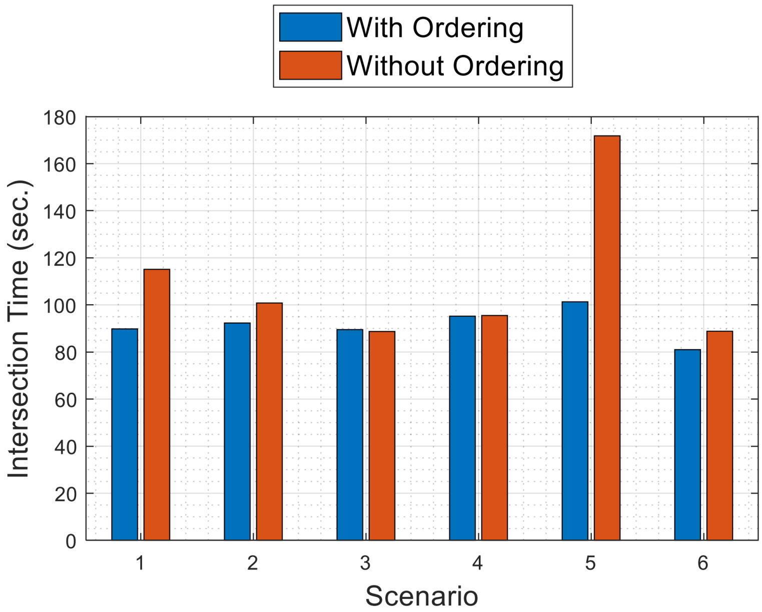


Figure 5.8 : Intersection times with ordering and without ordering for case 2.

Table 5.6 : Intersection time and average speed data in simulations with ordering and without ordering for case 2.

	With Ordering		Without Ordering	
Scenario	Intersection Time (s)	Average Speed (km/h)	Intersection Time (s)	Average Speed (km/h)
1	89.8	63.62	115.10	59.30
2	92.30	62.12	100.8	57.75
3	89.5	62.55	88.70	59.80
4	92.5	62.23	95.50	59.14
5	101.30	62.17	171.8	58.56
6	81.0	60.43	88.8	58.32

that the sorting algorithm reduces the time spent at the intersection. As seen in Figure 5.4, the sorting algorithm steadily increased the average speed of the vehicles.

The advantages of the proposed system are demonstrated by comparing the scenarios where the sorting algorithm is used and the scenarios where the sorting algorithm is not used. According to the results, in reservation-based intersection management, sorting the vehicles in the platoon according to their routes by using the proposed sorting algorithm is superior to the situation in which no sorting is applied in terms of the total time spent at the intersection and the average speed in the intersection area. The effect of the sorting algorithm is consistent with the objectives of increasing efficiency and safety and shortening travel time, which are the basis of the platoon application. Arriving at the intersection by sorting the vehicles usually results in being reserved as

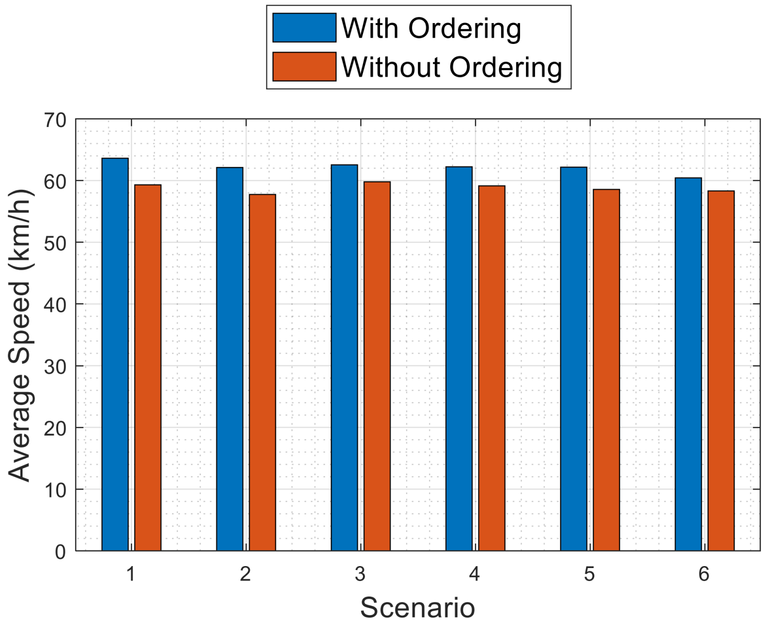


Figure 5.9 : Average speeds with ordering and without ordering for case 2.

a convoy, depending on the cost function. In this way, other vehicles that are not in the platoon are prevented from interfering, and vehicles turning in the same direction catch each other and continue on the road as a platoon easier. When the simulation results obtained using the sorting algorithm are examined, it is seen that the vehicles behave more stable in case of variable situations. Since the deceleration of the vehicle in front at the entrance to the intersection also affects the vehicles behind it, it is ensured that the vehicles that will turn stay behind the vehicles that will go straight so that the vehicles that go straight are not affected by this slowdown. With this study, it has been clearly seen that sorting the vehicles on the platoon is effective in reservation-based traffic intersection control. It is planned to develop various learning algorithms so that the sorting algorithm can adapt to every situation and increase its efficiency. In this way, it will be possible to ensure that the algorithm works more effectively in much more complex scenarios.

6. CONCLUSIONS AND RECOMMENDATIONS

Traffic congestion has been an important factor affecting the quality of human life continuously in the developing world. Increasing demand for individual vehicles decreases the quality of human life with increasing CO_2 emission, congestion and waiting time in traffic. The use of personal vehicles is increasing day by day. Especially these days, as a result of the Covid-19 pandemic, the rate of increase in the use of individual vehicles is increasing significantly. More lockdowns are normal in crowded urban communities as large numbers of vehicles can cause major transportation delays on existing transportation bases. There are many reasons for the occurrence of traffic, but the traffic happens most frequently at the complex road structures where vehicles intersect regularly because of the decision-making problem for drivers and traffic management systems. Intelligent Transportation Systems (ITSs) were developed to control the factors that cause traffic congestion and to improve the quality of life of people. Nowadays, increasing traffic monitoring units make a great contribution to developing ITS that can decrease traffic density. ITS has an extensive structure that performs information, communication, and control of traffic items. ITS systems may be used to build more secure systems in the traffic environment. Especially by reducing the load on drivers, accidents due to fatigue and carelessness can be avoided. In addition, with the optimum suggestions offered by ITS, traffic jams can be reduced, and accordingly, environmental pollution is reduced. This has been an important inspiration to many researchers, and various studies have been carried out recently.

In this thesis, the control of many traffic problems is considered. ITS-based multi-agent methods are proposed as a solution to traffic problems. Extensive simulations are carried out for both agent-based and multi-agent-based environments so as to demonstrate the validity and efficiency of the proposed methods.

First, a traffic light controller that does not deploy any learning algorithm is proposed on a single intersection, and its simulation is weighted. This proposed traffic light controller aims to increase the traffic flow and to reduce the overall waiting time of the cars and the emissions released by them. Various control methods have been proposed for a better traffic light controller architecture. Among these methods are the fuzzy logic controller, PI controller, and state space model controller. Firstly, a traffic light controller is developed, and simulations are performed for a single-lane traffic intersection with only two phases, no right and left turns and no yellow light duration. FLC, PI Control and state space control methods are proposed as traffic light controllers for this simple structured traffic intersection. Various simulations have been made to test the effectiveness of the proposed methods. From the simulation results, it has been seen that all proposed methods give better results than the traditional constant-time traffic light control method. With the proposed methods, an increase in average speed and a decrease in CO_2 emission values have been observed. Later, various control methods have been proposed for more complex traffic intersections with right and left turns and yellow light duration. The recommended control method for a 3-lane and 4-lane traffic intersection is basically the FLC method. However, for each proposed FLC, different input values have been proposed depending on the strategies. The recommended input values for FLC mainly include the change in the number of vehicles. However, in the next step, FLC input data depending on the vehicle position is proposed. In other words, state information based on vehicle position information is used as input values for fuzzy logic. Using this so-called "celling method", where more weight is given to the vehicles near the junction in comparison to further away vehicles by using smaller cell sizes near the traffic junction, it is demonstrated that more efficient traffic intersection controllers are obtained. The generality of the proposed methods is demonstrated by considering a case study as the control of traffic lights in the Istanbul Altunizade region. As a result, the effectiveness of the proposed methods is observed in a realistic region. The simulation results showed us that, particularly FLC and actuated TLC systems give noteworthy results, by increasing the traffic flow rate and reducing the amount of CO_2 emissions.

The effect of using a learning algorithm is examined in the second part of this thesis. An agent-based traffic light control that can learn and adapt to the environment has been proposed. A traffic light controller with a deep Q learning algorithm, which works more efficiently and increases the stability of the system, has been developed and the results are discussed. A deep Q-learning method used with FLSI (named DQ FLSI) is proposed as an intelligent traffic light controller. In this proposed method, a state matrix that divides the arms of the traffic intersection into cells is used. A varying cell size in the determination of the state matrix is used in the deep Q learning algorithm and FLC input. A comparison between using constant (equal) cell sizes and varying cell sizes is also provided to demonstrate the efficiency of this adaptation. The Reinforcement Learning tool is used for determining the actions of the traffic light phases, and the states are defined depending on the vehicle positions. The waiting times in the traffic intersection are used as reward values. The Q-learning paradigm is combined with a deep-learning model for training the agent. While the deep Q-learning model is used to determine the action set (the order of traffic lights), the green light duration is proposed to be handled by the FLC. In other words, the traffic light durations are determined using fuzzy logic, and traffic light actions are determined with the help of deep Q-learning. In addition, a stability analysis is done for the proposed method. An increase in robustness is shown when using the newly developed DQ FLSI method using the proposed stability analysis. The results demonstrate that the proposed method can adapt to many traffic conditions and outperform conventional methods in low and medium-density situations. Furthermore, it is observed that the learned method outperforms many traffic performance parameters in test scenarios. It is also demonstrated by extensive simulations that the developed system is more robust in terms of stability.

Another problem studied in this thesis is the route planning of vehicles in traffic. A deep Q-learning algorithm is used again in this proposed method, where an agent-based route planning method for vehicles has been developed. A taxi agent with a deep Q-learning algorithm, which makes dynamic route planning, is considered. For the taxi agency to learn the optimum route planning, a state vector including information on traffic lights, the density of neighbouring roads, the location of neighbouring intersections, the location of the agent and the destination is proposed. The simulation results demonstrate that the proposed method can be used for dynamic problems such as route planning.

Another work proposed in this thesis is based on reservation-based intelligent traffic intersection control. In this method, without the need for any traffic lights, vehicle agents communicate directly with the intersection agents and pass through the intersection by following the proposed actions of the intersection agents. The intersection agent provides passage permission to the vehicles by making a reservation for them. In addition, a platoon method is suggested to be combined with the reservation-based traffic intersection control method. Here, vehicles are requested to change their lanes with respect to their upcoming actions before they arrive at the traffic intersection and therefore unnecessary decelerations are prevented. It is seen by observing the simulation results that the proposed method performs better.

As a result, it is possible to claim that the ideas proposed in this study increase efficiency of traffic controllers. Especially for nonlinear multivariate structures such as traffic conditions, control methods that can adapt to the environment and even learn have the potential to overcome many problems. As the number of autonomous vehicles increases depending on the development of technology, vehicles with smart agent systems will form MAS. With this, it is expected that many problems in traffic can be solved with multi-agent system solutions. However, with the existing technological infrastructures, many innovative control methods can still be used. Route planning and smart traffic intersection management systems can be given as examples. Nevertheless, the studies proposed in this thesis are not final, and many improvements are possible without a doubt.

One of the possible future studies is to control more than one traffic intersection with the MAS theory. Especially, traffic control of a large realistic area needs to be considered to better prove the applicability of the proposed methods.

Another possible extension is to consider the fleet control problem with MAS theory. A multi-agent System for the control of an ambulance fleet, which involves extra complexity and contains many parameters, is currently being studied in this direction.

The reservation-based traffic intersection management problem is an important area of research in transportation engineering and has the potential to significantly improve traffic flow and reduce congestion. Combining the platoon algorithm with deep learning methods could indeed increase its effectiveness and enhance its capabilities. Other deep learning methods can be used to analyze traffic patterns and predict the behavior of individual vehicles, allowing for more accurate coordination of platoons and more efficient management of the intersection. Additionally, reinforcement learning algorithms can be used to continually adjust the timing and coordination of platoons based on real-time traffic conditions. Overall, the combination of the Platoon algorithm with deep learning methods has the potential to significantly improve traffic flow and reduce congestion in urban areas. There are many possible ways to approach this problem, and further research in this area could lead to important advancements in transportation engineering.

Agent-based and multi-Agent-based studies are a branch of computer science that focus on modeling complex systems as a collection of interacting agents, each with their own set of goals and behaviors. These agents can be anything from robots, autonomous vehicles, and humans to software agents that operate in virtual environments. One promising area of research within agent-based modeling is the use of reinforcement learning. Reinforcement learning is a type of machine learning that involves an agent interacting with an environment to learn through trial and error. The agent receives feedback in the form of rewards or punishments based on its actions, and it learns to make decisions that maximize its expected reward over time. In the context of agent-based modeling, reinforcement learning can be used to allow agents to learn from their experiences and make better decisions in real-time.

In the context of traffic management, agent-based and multi-agent-based studies can be used to model and simulate the behavior of drivers, vehicles, and other entities in a transportation network. By using reinforcement learning, these systems can learn to make optimal decisions in real time, such as determining the best route to take or adjusting traffic signals to reduce congestion. For instance, agents could be used to monitor traffic flow and adjust traffic lights or redirect traffic in real-time based on their learning. This would lead to more efficient and safer traffic flow, and could even help to reduce congestion and emissions. Similarly, in case of a catastrophic event, agents could be used to coordinate emergency response efforts and allocate resources more efficiently, potentially saving lives and minimizing the impact of the disaster. Moreover, in situations where composure is needed, such as during disasters or emergencies, agent-based and multi-agent-based systems can provide valuable decision-making assistance. These systems can help emergency responders to coordinate their efforts and optimize their resources to respond effectively to the situation.

With the increasing use of autonomous vehicles, the need for agent-based and multi-agent-based systems in traffic management is expected to increase significantly. These systems can help autonomous vehicles to coordinate their actions and communicate with other agents in the transportation network to avoid accidents and reduce traffic congestion. However, it's important to note that the use of autonomous decision-making agents also raises ethical considerations. As these agents make decisions based on their learning and programming, it's important to ensure that they prioritize human safety and well-being.

In conclusion, agent-based and multi-agent-based studies, coupled with reinforcement learning, have the potential to provide effective solutions for decision-making in various domains, including traffic management, disaster management, and many more. These systems can help to improve the safety, efficiency, and effectiveness of complex systems by enabling agents to learn and make optimal decisions in real time. Overall, while agent-based and multi-agent-based studies have great potential as decision-making tools in a variety of domains.

REFERENCES

[1] **Russell, S.J. and Norvig, P.**, (2016), Artificial intelligence: a modern approach. Malaysia.

[2] **Shoham, Y. and Leyton-Brown, K.** (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*, Cambridge University Press.

[3] **Deakin, E.** (2003). Mainstreaming Intelligent Transportation Systems: Findings from a Survey of California Leaders, *Transportation Research Board 82nd Annual MeetingTransportation Research Board*.

[4] **Wang, D.**, **Ren, L. and Li, J.** (2010). Modeling intelligent transportation systems with multi-agent on SOA, *2010 International Conference on Intelligent Computing and Integrated Systems*, IEEE, pp.717–720.

[5] **Kumar, M.**, **Albert, S. and Deeter, D.** (2005). A summary of rural intelligent transportation systems (ITS) benefits as applied to ODOT region 1, *Oregon Department of Transportation Region*, *1*.

[6] **Ezell, S.**, (2010), Intelligent transportation systems. the information technology & innovation foundation. january 2010.

[7] **Franco, J.**, **Aris, A.**, **Canberk, B. and Uluagac, A.S.** (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems, *IEEE Communications Surveys & Tutorials*, *23*(4), 2351–2383.

[8] **Lian, Y.**, **Zhang, G.**, **Lee, J. and Huang, H.** (2020). Review on big data applications in safety research of intelligent transportation systems and connected/automated vehicles, *Accident Analysis & Prevention*, *146*, 105711.

[9] **Choi, M.**, **Rubenecia, A. and Choi, H.H.** (2019). Reservation-based intersection crossing scheme for autonomous vehicles traveling in a speed range, *2019 International Conference on Information Networking (ICOIN)*, IEEE, pp.84–87.

[10] **Pasquale, C.**, **Sacone, S.**, **Siri, S. and Ferrara, A.** (2019). Traffic control for freeway networks with sustainability-related objectives: Review and future challenges, *Annual Reviews in Control*, *48*, 312–324.

[11] **Yan, J.**, **Liu, J. and Tseng, F.M.** (2020). An evaluation system based on the self-organizing system framework of smart cities: A case study of smart transportation systems in China, *Technological Forecasting and Social Change*, *153*, 119371.

[12] **Guo, Y., Tang, Z. and Guo, J.** (2020). Could a smart city ameliorate urban traffic congestion? A quasi-natural experiment based on a smart city pilot program in China, *Sustainability*, *12*(6), 2291.

[13] **Zhang, L., Zeng, G., Li, D., Huang, H.J., Stanley, H.E. and Havlin, S.** (2019). Scale-free resilience of real traffic jams, *Proceedings of the National Academy of Sciences*, *116*(18), 8673–8678.

[14] **Wu, J., Zou, C., He, S., Sun, X., Wang, X. and Yan, Q.** (2019). Traffic noise exposure of high-rise residential buildings in urban area, *Environmental Science and Pollution Research*, *26*(9), 8502–8515.

[15] **Ma, M. and Li, Z.** (2021). A time-independent trajectory optimization approach for connected and autonomous vehicles under reservation-based intersection control, *Transportation Research Interdisciplinary Perspectives*, *9*, 100312.

[16] **Li, W. and Ban, X.** (2020). Connected vehicle-based traffic signal coordination, *Engineering*, *6*(12), 1463–1472.

[17] **Aleko, D.R. and Djahel, S.** (2020). An efficient adaptive traffic light control system for urban road traffic congestion reduction in smart cities, *Information*, *11*(2), 119.

[18] **Cobo, M.J., Chiclana, F., Collop, A., de Ona, J. and Herrera-Viedma, E.** (2013). A bibliometric analysis of the intelligent transportation systems research based on science mapping, *IEEE transactions on intelligent transportation systems*, *15*(2), 901–908.

[19] **Firdous, M., Iqbal, F.U.D., Ghafoor, N., Qureshi, N.K. and Naseer, N.** (2019). Traffic light control system for four-way intersection and T-crossing using fuzzy logic, *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, IEEE, pp.178–182.

[20] **Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z.S.H., Bazzan, A.L. and Far, B.** (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review, *Expert Systems with Applications*, 116830.

[21] **Sutton, R.S. and Barto, A.G.** (1999). Reinforcement learning: An introduction, *Robotica*, *17*(2), 229–235.

[22] **Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., Li, Y., Xu, K. and Li, Z.** (2019). Learning phase competition for traffic signal control, *Proceedings of the 28th ACM international conference on information and knowledge management*, pp.1963–1972.

[23] **Wang, X., Ke, L., Qiao, Z. and Chai, X.** (2020). Large-scale traffic signal control using a novel multiagent reinforcement learning, *IEEE transactions on cybernetics*, *51*(1), 174–187.

[24] **Joo, H.**, **Ahmed, S.H. and Lim, Y.** (2020). Traffic signal control for smart cities using reinforcement learning, *Computer Communications*, *154*, 324–330.

[25] **Zhang, R.**, **Ishikawa, A.**, **Wang, W.**, **Striner, B. and Tonguz, O.K.** (2020). Using reinforcement learning with partial vehicle detection for intelligent traffic signal control, *IEEE Transactions on Intelligent Transportation Systems*, *22*(1), 404–415.

[26] **El-Tantawy, S.**, **Abdulhai, B. and Abdelgawad, H.** (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control, *Journal of Intelligent Transportation Systems*, *18*(3), 227–245.

[27] **Genders, W. and Razavi, S.N.**, (2016), Using a Deep Reinforcement Learning Agent for Traffic Signal Control, `http://arxiv.org/abs/1611.01142`, `1611.01142`.

[28] **Hartenstein, H. and Laberteaux, L.P.** (2008). A tutorial survey on vehicular ad hoc networks, *IEEE Communications magazine*, *46*(6), 164–171.

[29] **Mnih, V.**, **Kavukcuoglu, K.**, **Silver, D.**, **Rusu, A.A.**, **Veness, J.**, **Bellemare, M.G.**, **Graves, A.**, **Riedmiller, M.**, **Fidjeland, A.K. and Ostrovski, G.** (2015). Human-level control through deep reinforcement learning, *Nature*, *518*(7540), 529–533.

[30] **Li, L.**, **Lv, Y. and Wang, F.Y.** (2016). Traffic signal timing via deep reinforcement learning, *IEEE/CAA Journal of Automatica Sinica*, *3*(3), 247–254.

[31] **LU, C.** Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic.

[32] **Dey, K.C.**, **Yan, L.**, **Wang, X.**, **Wang, Y.**, **Shen, H.**, **Chowdhury, M.**, **Yu, L.**, **Qiu, C. and Soundararaj, V.** (2015). A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC), *IEEE Transactions on Intelligent Transportation Systems*, *17*(2), 491–509.

[33] **Dresner, K. and Stone, P.** (2004). Multiagent traffic management: A reservation-based intersection control mechanism, *Autonomous Agents and Multiagent Systems, International Joint Conference on*, volume 3, IEEE Computer Society, pp.530–537.

[34] **Kavathekar, P. and Chen, Y.** (2011). Vehicle platooning: A brief survey and categorization, *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume54808, pp.829–845.

[35] **Carlino, D.**, **Boyles, S.D. and Stone, P.** (2013). Auction-based autonomous intersection management, *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, IEEE, pp.529–534.

[36] **Lopez, P.A.**, **Behrisch, M.**, **Bieker-Walz, L.**, **Erdmann, J.**, **Flötteröd, Y.P.**, **Hilbrich, R.**, **Lücken, L.**, **Rummel, J.**, **Wagner, P. and Wießner, E.** (2018). Microscopic traffic simulation using sumo, *2018 21st international conference on intelligent transportation systems (ITSC)*, IEEE, pp.2575–2582.

[37] **Olfati-Saber, R. and Murray, R.M.** (2004). Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on automatic control*, *49*(9), 1520–1533.

[38] **Ren, W. and Atkins, E.** (2007). Distributed multi-vehicle coordinated control via local information exchange, *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, *17*(10-11), 1002–1033.

[39] **Lewis, F.L.**, **Zhang, H.**, **Hengster-Movric, K. and Das, A.** (2013). *Cooperative control of multi-agent systems: optimal and adaptive design approaches*, Springer Science & Business Media.

[40] **Yang, T.**, **Meng, Z.**, **Dimarogonas, D.V. and Johansson, K.H.** (2014). Global consensus for discrete-time multi-agent systems with input saturation constraints, *Automatica*, *50*(2), 499–506.

[41] **Lin, Z.**, **Broucke, M. and Francis, B.** (2004). Local control strategies for groups of mobile autonomous agents, *IEEE Transactions on automatic control*, *49*(4), 622–629.

[42] **Fax, J.A. and Murray, R.M.** (2004). Information flow and cooperative control of vehicle formations, *IEEE transactions on automatic control*, *49*(9), 1465–1476.

[43] **Perea, L.**, **Gómez, G. and Elosegui, P.** (2009). Extension of the Cucker-Smale control law to space flight formations, *Journal of guidance, control, and dynamics*, *32*(2), 527–537.

[44] **Ren, W.** (2007). Multi-vehicle consensus with a time-varying reference state, *Systems & Control Letters*, *56*(7-8), 474–483.

[45] **Ren, W. and Beard, R.W.** (2004). Decentralized scheme for spacecraft formation flying via the virtual structure approach, *Journal of Guidance, Control, and Dynamics*, *27*(1), 73–82.

[46] **Dresner, K. and Stone, P.** (2008). A multiagent approach to autonomous intersection management, *Journal of artificial intelligence research*, *31*, 591–656.

[47] **Bidram, A.**, **Lewis, F.L. and Davoudi, A.** (2014). Synchronization of non-linear heterogeneous cooperative systems using input–output feedback linearization, *Automatica*, *50*(10), 2578–2585.

[48] **Ge, X. and Han, Q.L.** (2017). Distributed formation control of networked multi-agent systems using a dynamic event-triggered communication mechanism, *IEEE Transactions on Industrial Electronics*, *64*(10), 8118–8127.

[49] **Han, S.I.** (2018). Prescribed consensus and formation error constrained finite-time sliding mode control for multi-agent mobile robot systems, *IET Control Theory & Applications*, *12*(2), 282–290.

[50] **Huang, Y. and Jia, Y.** (2018). Distributed finite-time output feedback synchronisation control for six DOF spacecraft formation subject to input saturation, *IET Control Theory & Applications*, *12*(4), 532–542.

[51] **Li, Z. and Duan, Z.** (2017). *Cooperative control of multi-agent systems: a consensus region approach*, CRC press.

[52] **Chen, K.**, **Wang, J.**, **Zhang, Y. and Lewis, F.L.** (2018). Cluster consensus of heterogeneous linear multi-agent systems, *IET Control Theory & Applications*, *12*(11), 1533–1542.

[53] **Yu, T.**, **Ma, L. and Qin, N.** (2018). Adaptive cooperative tracking control of multi-agent systems with unknown actuators hysteresis, *IEEE Access*, *6*, 33015–33028.

[54] **Chen, C.**, **Wang, Z.**, **Jin, L.**, **Yin, Y. and Liu, F.** (2018). Consensus of second-order multi-agent systems with time delays, *2018 3rd International Conference on Control and Robotics Engineering (ICCRE)*, IEEE, pp.109–113.

[55] **Wang, Q.**, **Fu, J. and Wang, J.** (2018). Cooperative tracking for high-order non-linear multi-agent systems via adaptive control, *IET Control Theory & Applications*, *12*(11), 1592–1600.

[56] **Cheng, Z.**, **Zhang, H.T.**, **Fan, M.C. and Chen, G.** (2014). Distributed consensus of multi-agent systems with input constraints: A model predictive control approach, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *62*(3), 825–834.

[57] **Ferrari-Trecate, G.**, **Galbusera, L.**, **Marciandi, M.P.E. and Scattolini, R.** (2009). Model predictive control schemes for consensus in multi-agent systems with single-and double-integrator dynamics, *IEEE Transactions on Automatic Control*, *54*(11), 2560–2572.

[58] **Zhang, H.T.**, **Cheng, Z.**, **Chen, G. and Li, C.** (2015). Model predictive flocking control for second-order multi-agent systems with input constraints, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *62*(6), 1599–1606.

[59] **Hu, W.**, **Liu, L. and Feng, G.** (2018). Event-triggered cooperative output regulation of linear multi-agent systems under jointly connected topologies, *IEEE Transactions on Automatic Control*, *64*(3), 1317–1322.

[60] **Wang, A.**, **Liao, X. and Dong, T.** (2018). Event-triggered gradient-based distributed optimisation for multi-agent systems with state consensus constraint, *IET Control Theory & Applications*, *12*(10), 1515–1519.

[61] **Li, Z. and Ji, H.** (2018). Finite-time consensus and tracking control of a class of nonlinear multiagent systems, *IEEE transactions on automatic control*, *63*(12), 4413–4420.

[62] **Kimura, H. and Okuyama, A.** (2018). Funnel control in multi-agent systems with communication limitations, *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, IEEE, pp.569–574.

[63] **Zuo, S.**, **Song, Y.**, **Lewis, F.L. and Davoudi, A.** (2018). Time-varying output formation containment of general linear homogeneous and heterogeneous multiagent systems, *IEEE Transactions on Control of Network Systems*, *6*(2), 537–548.

[64] **Liu, B.**, **Su, H.**, **Jiang, F.**, **Gao, Y.**, **Liu, L. and Qian, J.** (2018). Group controllability of continuous-time multi-agent systems, *IET Control Theory & Applications*, *12*(11), 1665–1671.

[65] **Jiménez, A.C.**, **Bolaños, S.J. and Anzola, J.P.** (2017). Autonomous multi-agent system for on-line topological mapping and location using wireless sensor network, *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, IEEE, pp.216–221.

[66] **Van der Pol, E. and Oliehoek, F.A.** (2016). Coordinated deep reinforcement learners for traffic light control, *Proceedings of learning, inference and control of multi-agent systems (at NIPS 2016)*, *1*.

[67] **Lee, J. and Park, B.** (2012). Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment, *IEEE transactions on intelligent transportation systems*, *13*(1), 81–90.

[68] **Bouarfa, S.**, **Müller, J. and Blom, H.** (2018). Evaluation of a Multi-Agent System approach to airline disruption management, *Journal of Air Transport Management*, *71*, 108–118.

[69] **Aksyonov, K.**, **Bykov, E.**, **Aksyonova, O.**, **Goncharova, N. and Nevolina, A.** (2017). Multi-agent simulation of the resource conversion processes with the sample application to the logistical problem, *2017 European Modelling Symposium (EMS)*, IEEE, pp.177–181.

[70] **Azimi, R.**, **Bhatia, G.**, **Rajkumar, R.R. and Mudalige, P.** (2014). STIP: Spatio-temporal intersection protocols for autonomous vehicles, *2014 ACM/IEEE international conference on cyber-physical systems (ICCPS)*, IEEE, pp.1–12.

[71] **Liu, Y.**, **Liu, L. and Chen, W.P.** (2017). Intelligent traffic light control using distributed multi-agent Q learning, *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, IEEE, pp.1–8.

[72] **Khalid, M.** (1996). Intelligent traffic lights control by fuzzy logic, *Malaysian Journal of Computer Science*, *9*(2), 29–35.

[73] **RAHMAT, R. and Jumari, K.** (2000). Vehicle detection using image processing for traffic control and surveillance system, *PROCEEDINGS OF THE 7TH WORLD CONGRESS ON INTELLIGENT SYSTEMS*.

[74] **Turky, A.M.**, **Ahmad, M.S. and Yusoff, M.Z.M.** (2009). The use of genetic algorithm for traffic light and pedestrian crossing control, *International Journal of Computer Science and Network Security*, *9*(2), 88–96.

[75] **Chiu, S.** (1992). Adaptive traffic signal control using fuzzy logic, *Proceedings of the Intelligent Vehicles92 Symposium*, IEEE, pp.98–107.

[76] **De Schutter, B. and De Moor, B.** (1998). Optimal traffic light control for a single intersection, *European Journal of Control*, *4*(3), 260–276.

[77] **Fadlullah, Z.M.**, **Tang, F.**, **Mao, B.**, **Kato, N.**, **Akashi, O.**, **Inoue, T. and Mizutani, K.** (2017). State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems, *IEEE Communications Surveys Tutorials*, *19*(4), 2432–2455.

[78] **Prabuchandran, K.J.**, **AN, H.K. and Bhatnagar, S.** (2014). Multi-agent reinforcement learning for traffic signal control, *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, pp.2529–2534.

[79] **Abdoos, M.**, **Mozayani, N. and Bazzan, A.L.C.** (2013). Holonic multi-agent system for traffic signals control, *Engineering Applications of Artificial Intelligence*, *26*(5-6), 1575–1587.

[80] **Arel, I.**, **Liu, C.**, **Urbanik, T. and Kohls, A.G.** (2010). Reinforcement learning-based multi-agent system for network traffic signal control, *IET Intelligent Transport Systems*, *4*(2), 128–135.

[81] **Thorpe, T.L. and Anderson, C.W.** (1996). Traffic light control using sarsa with three state representations, *Technical report, Citeseer*.

[82] **Camponogara, E. and Kraus, W.** (2003). Distributed learning agents in urban traffic control, *Portuguese conference on artificial intelligence*, Springer, pp.324–335.

[83] **Araghi, S.**, **Khosravi, A.**, **Johnstone, M. and Creighton, D.** (2013). Intelligent traffic light control of isolated intersections using machine learning methods, *2013 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, pp.3621–3626.

[84] **Liu, W.**, **Qin, G.**, **He, Y. and Jiang, F.** (2017). Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering, *IEEE transactions on vehicular technology*, *66*(10), 8667–8681.

[85] **Tang, F.**, **Mao, B.**, **Fadlullah, Z.M.**, **Kato, N.**, **Akashi, O.**, **Inoue, T. and Mizutani, K.** (2017). On removing routing protocol from future wireless

networks: A real-time deep learning approach for intelligent traffic control, *IEEE Wireless Communications*, *25*(1), 154–160.

[86] **Van der Pol, E. and Oliehoek, F.A.** (2016). Coordinated deep reinforcement learners for traffic light control, *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.

[87] **Miao, Q.**, **Li, Y.**, **Wang, F.Y.**, **Tang, S.** *et al.* (2006). Modeling and analysis of artificial transportation system based on multi-agent technology, *2006 IEEE Intelligent Transportation Systems Conference*, IEEE, pp.1120–1124.

[88] **Hacıoğlu, F. and Söylemez, M.T.** (2017). Power consumption based multi agent intersection management method, *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, IEEE, pp.954–958.

[89] **Choi, M.**, **Rubenecia, A. and Choi, H.H.** (2019). Reservation-based traffic management for autonomous intersection crossing, *International Journal of Distributed Sensor Networks*, *15*(12), 1550147719895956.

[90] **Levin, M.W.**, **Fritz, H. and Boyles, S.D.** (2016). On optimizing reservation-based intersection controls, *IEEE Transactions on Intelligent Transportation Systems*, *18*(3), 505–515.

[91] **Vasirani, M. and Ossowski, S.** (2009). Evaluating policies for reservation-based intersection control, *Proceedings of the 14th Portuguese conference on artificial intelligence*, volume 14, pp.39–50.

[92] **Liang, K.Y.**, **Mårtensson, J. and Johansson, K.H.** (2015). Heavy-duty vehicle platoon formation for fuel efficiency, *IEEE Transactions on Intelligent Transportation Systems*, *17*(4), 1051–1061.

[93] **Larson, J.**, **Munson, T. and Sokolov, V.** (2016). Coordinated platoon routing in a metropolitan network, *2016 proceedings of the seventh siam workshop on combinatorial scientific computing*, SIAM, pp.73–82.

[94] **Wu, D.**, **Wu, J. and Wang, R.** (2019). An energy-efficient and trust-based formation algorithm for cooperative vehicle platooning, *2019 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, pp.702–707.

[95] **Saha, A.**, **Chandra, S. and Ghosh, I.** (2019). Modeling platoon dispersion at signalized intersections in mixed traffic scenario, *Arabian Journal for Science and Engineering*, *44*(5), 4829–4838.

[96] **Timmerman, R. and Boon, M.A.** (2021). Platoon forming algorithms for intelligent street intersections, *Transportmetrica A: transport science*, *17*(3), 278–307.

[97] **Bashiri, M. and Fleming, C.H.** (2017). A platoon-based intersection management system for autonomous vehicles, *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, pp.667–672.

[98] **Jin, Q.**, **Wu, G.**, **Boriboonsomsin, K. and Barth, M.** (2013). Platoon-based multi-agent intersection management for connected vehicle, *16th international ieee conference on intelligent transportation systems (itsc 2013)*, IEEE, pp.1462–1467.

[99] **Dotoli, M.**, **Hammadi, S.**, **Jeribi, K.**, **Russo, C. and Zgaya, H.** (2013). A multi-agent decision support system for optimization of co-modal transportation route planning services, *52nd IEEE Conference on Decision and Control*, IEEE, pp.911–916.

[100] **Phutthisathian, A.**, **Maneerat, N.**, **Varakulsiripunth, R.**, **Takahashi, K. and Kato, Y.** (2013). An ontology-based multi agent autotnotive parts transportation management system, *2013 IEEE Region 10 Humanitarian Technology Conference*, IEEE, pp.96–99.

[101] **Cubillos, C.**, **Guidi-Polanco, F. and Demartini, C.** (2005). MADARP: Multi-agent architecture for passenger transportation systems, *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, IEEE, pp.437–442.

[102] **Luo, R.**, **van den Boom, T.J. and De Schutter, B.** (2017). Multi-agent dynamic routing of a fleet of cybercars, *IEEE Transactions on Intelligent Transportation Systems*, *19*(5), 1340–1352.

[103] **Wenjun, W. and Duowang, F.** (2011). Study on multi-agent railway signal control system, *2011 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, IEEE, pp.235–238.

[104] **Bhardwaj, A.**, **Singh, V.K. and Kumar, P.** (2014). Multi-agent based train passing in railway system with minimum system delay, *2014 IEEE International Advance Computing Conference (IACC)*, IEEE, pp.855–859.

[105] **Rajkumar, R.I.**, **Sankaranarayanan, P. and Sundari, G.** (2015). An approach to avoiding train collision in railway sectors using multi agent system, *Procedia Computer Science*, *57*, 1067–1073.

[106] **Li, S.**, **Yang, L. and Gao, Z.** (2015). Coordinated cruise control for high-speed train movements based on a multi-agent model, *Transportation Research Part C: Emerging Technologies*, *56*, 281–292.

[107] **D'Ariano, A. and Hemelrijk, R.** (2006). Designing a multi-agent system for cooperative train dispatching, *IFAC Proceedings Volumes*, *39*(3), 369–374.

[108] **Withanawasam, J. and Karunananda, A.** (2017). Multi-agent based road traffic control optimization, *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, pp.977–981.

[109] **Mirzaei, H. and Givargis, T.** (2017). Fine-grained acceleration control for autonomous intersection management using deep reinforcement learning, *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, IEEE, pp.1–8.

[110] **De Schutter, B. and De Moor, B.** (1998). Optimal traffic light control for a single intersection, *European Journal of Control*, *4*(3), 260–276.

[111] **Wiering, M.A.** *et al.* (2000). Multi-agent reinforcement learning for traffic light control, *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp.1151–1158.

[112] **Tubaishat, M.**, **Shang, Y. and Shi, H.** (2007). Adaptive traffic light control with wireless sensor networks, *2007 4th IEEE Consumer Communications and Networking Conference*, IEEE, pp.187–191.

[113] **Zadeh, L.A.** (1965). Fuzzy sets, *Information and control*, *8*(3), 338–353.

[114] **Pappis, C.P. and Mamdani, E.H.** (1977). A fuzzy logic controller for a trafc junction, *IEEE Transactions on Systems, Man, and Cybernetics*, *7*(10), 707–717.

[115] **Favilla, J.**, **Machion, A. and Gomide, F.** (1993). Fuzzy traffic control: adaptive strategies, *[Proceedings 1993] Second IEEE International Conference on Fuzzy Systems*, IEEE, pp.506–511.

[116] **Tunca, I. and Soylemeza, M.T.** (2020). Intelligent Intersection Management using Fuzzy Logic control and PI control, *Proceedings of 8th Transport Research Arena TRA*, Helsinki, Finland, pp.27–30.

[117] **Tunc, I. and Soylemez, M.T.** (2020). State Feedback Control for Intelligent Traffic Light Systems, *2020 28th Mediterranean Conference on Control and Automation (MED)*, IEEE, pp.446–451.

[118] **Tunc, I.**, **Yesilyurt, A.Y. and Soylemez, M.T.** (2019). Intelligent traffic light control system simulation for different strategies with fuzzy logic controller, *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, IEEE, pp.830–834.

[119] **Tunc, I.**, **Yesilyurt, A.Y. and Soylemez, M.T.** (2021). Different fuzzy logic control strategies for traffic signal timing control with state inputs, *IFAC-PapersOnLine*, *54*(2), 265–270.

[120] **Tunc, I.**, **Koroglu, A.O.**, **Edem, A.E.**, **Akmese, S.N.**, **Elmas, O. and Soylemez, M.T.** (2021). Traffic Light Control for Multi Intersection Model in Istanbul/Altunizade, *2021 29th Mediterranean Conference on Control and Automation (MED)*, IEEE, pp.548–553.

[121] **Ackermann, J.** (1972). Der entwurf linearer regelungssysteme im zustandsraum, *at-Automatisierungstechnik*, *20*(1-12), 297–300.

[122] **Zheng, G.**, **Zang, X.**, **Xu, N.**, **Wei, H.**, **Yu, Z.**, **Gayah, V.**, **Xu, K. and Li, Z.**, (2019), Diagnosing Reinforcement Learning for Traffic Signal Control, `https://arxiv.org/abs/1905.04716`.

[123] **Tan, T.**, **Bao, F.**, **Deng, Y.**, **Jin, A.**, **Dai, Q. and Wang, J.** (2019). Cooperative deep reinforcement learning for large-scale traffic grid signal control, *IEEE transactions on cybernetics*, *50*(6), 2687–2700.

[124] **Vidali, A.**, **Crociani, L.**, **Vizzari, G. and Bandini, S.** (2019). A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management., *WOA*, pp.42–50.

[125] **Watkins, C.J.C.H.** (1989). Learning from delayed rewards, (1989).

[126] **Pálos, P. and Huszák, Á.** (2020). Comparison of q-learning based traffic light control methods and objective functions, *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, IEEE, pp.1–6.

[127] **Bellman, R.** (1957). Dynamic programming, princeton univ, *Press Princeton, New Jersey, 1957*.

[128] **Liang, X.**, **Du, X.**, **Wang, G. and Han, Z.** (2019). A deep reinforcement learning network for traffic light cycle control, *IEEE Transactions on Vehicular Technology*, *68*(2), 1243–1253.

[129] **Liu, Junxiu and Qin, Sheng and Luo, Yuling and Wang, Yanhu and Yang, Su** (2022). Intelligent Traffic Light Control by Exploring Strategies in an Optimised Space of Deep Q-Learning, *IEEE Transactions on Vehicular Technology, 2022*.

[130] **Takano, Y.**, **Inoue, H.**, **Thawonmas, R. and Harada, T.** (2019). Self-play for training general fighting game AI, *2019 Nicograph International (NicoInt)*, IEEE, pp.120–120.

[131] **Park, I.S.**, **Park, C.e.**, **Kwon, N.K. and Park, P.** (2021). Dynamic output-feedback control for singular interval-valued fuzzy systems: Linear matrix inequality approach, *Information Sciences*, *576*, 393–406, `https://www.sciencedirect.com/science/article/pii/S0020025521006447`.

[132] **Ross, S.M.** (1981). Introduction to probability theory, *Introduction to Probability Models*, 1–21.

[133] **Bertsekas, D. and Gallager, R.** (1992). *Data networks*, volume 2, Englewood Cliffs, NJ, USA: Prentice-Hall.

[134] **Köroğlu, A.O.**, **Edem, A.E.**, **Akmeşe, S.N.**, **Elmas, Ö.**, **Tunc, I. and Soylemez, M.T.** (2021). Agent-Based Route Planning with Deep Q Learning, *2021 13th International Conference on Electrical and Electronics Engineering (ELECO)*, IEEE, pp.403–407.

[135] **Xu, Y.**, **Xi, Y.**, **Li, D. and Zhou, Z.** (2016). Traffic signal control based on Markov decision process, *IFAC-PapersOnLine*, *49*(3), 67–72.

[136] **Wiering, M.**, **Veenen, J.v.**, **Vreeken, J.**, **Koopman, A.** *et al.* (2004). Intelligent traffic light control.

[137] **Hirankitti, V. and Krohkaew, J.** (2007). An agent approach for intelligent traffic-light control, *First Asia International Conference on Modelling & Simulation (AMS'07)*, IEEE, pp.496–501.

CURRICULUM VITAE

Name Surname : İlhan TUNÇ

EDUCATION :

- **B.Sc.** : 2011, Kocaeli University, Faculty of Engineering, Department of Mechatronics Engineering
- **M.Sc.** : 2016, Bursa Technical University, Faculty of Engineering and Natural Sciences, Department of Mechatronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- Research Assistant, Istanbul Technical University (2016 - now)
- Research Assistant, Bursa Technical University (2014 - 2016)

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Tunc I., Soylemez M.T.** 2022. Fuzzy Logic and Deep Q Learning Based Control for Traffic Lights. *Alexandria Engineering Journal,* 48(17), 5195–5220.

- **Köroğlu, A. O., Edem, A. E., Akmeşe, S. N., Elmas, Ö., Tunc, I., Soylemez, M. T.** Agent-Based Route Planning with Deep Q Learning, *13th International Conference on Electrical and Electronics Engineering (ELECO)*, November 2021, Bursa, Turkey.

- **Tunc, I., Elmas, Ö., Köroğlu, A. O., Edem, A. E., Akmeşe, S. N., Soylemez, M. T.** Derin Q Öğrenme Tekniği ile Ajan Bazlı Trafik Işık Kontrolü, *Otomatik Kontrol Ulusal Kongresi TOK 2021*, September 2021, Van, Turkey.

- **Tunc, I., Köroğlu, A. O., Edem, A. E., Akmeşe, S. N., Elmas, Ö., Soylemez, M. T.** Traffic Light Control for Multi Intersection Model in Istanbul/Altunizade, *29th Mediterranean Conference on Control and Automation (MED)*, 22-25 June 2021, Puglia, Italy.

- **Tunc I., Yesilyurt A.Y., Soylemez M.T.** Different fuzzy logic control strategies for traffic signal timing control with state inputs, *16th IFAC Symposium on Control in Transportation Systems CTS 2021*, 8-10 June 2021, Lille, France.

- **Yesilyurt A.Y., Tunc I., Soylemez M.T.** A Reservation Method for Multi-agent System Intersection Management with Energy Consumption Considerations, *16th IFAC Symposium on Control in Transportation Systems CTS 2021*, 8-10 June 2021, Lille, France.

- **Tunc I., Soylemez M.T.** State Feedback Control for Intelligent Traffic Light Systems, *28th Mediterranean Conference on Control and Automation (MED)*, 15-18 September 2020, Saint-Raphaël, FRANCE.

- **Tunc I., Soylemez M.T.** Intelligent Intersection Management using Fuzzy Logic control and PI control, *8th Transport Research Arena TRA 2020*, June 2020, Helsinki, Finland.

- **Tunc I., Yesilyurt A.Y., Soylemez M.T.** Intelligent traffic light control system simulation for different strategies with fuzzy logic controller, *11th International Conference on Electrical and Electronics Engineering (ELECO)*, November 2019, Bursa, Turkey.

- **Tunc I., Soylemez M.T.** Kavşak Yönetiminde Bulanık Mantık ve PI Kontrol Yöntemlerinin Benzetimi, *Otomatik Kontrol Ulusal Kongresi TOK 2019*, September 2019, Muğla, Turkey.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Aydın, H., Tunc, I., Soylemez, M. T.** Fuzzy Logic Controller for Leader-Follower Based Navigation of Mobile Robots in a ROS-enabled Environment, *13th International Conference on Electrical and Electronics Engineering (ELECO)*, November 2021, Bursa, Turkey.

- **Tunc I., Yesilyurt A.Y., Soylemez M.T.** Çok Ajanlı Sistemlerde İki Boyutlu Benzetim Programı: MUBEESIM, *Otomatik Kontrol Ulusal Kongresi TOK 2017*, September 2017, İstanbul, Turkey.