Multi-modal Simulation and Calibration for OSU Campus Mobility

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree
Master of Science in the Graduate School of The Ohio State University

By

Vikhyat Kalra, B.E(Hons.)Mechanical Engineering

Graduate Program in Department of Mechanical Engineering

The Ohio State University

2021

Master's Examination Committee:

Dr Punit Tulpule, Advisor

Dr Shawn Midlam Mohler, Co-Advisor

Dr Qadeer Ahmed

Abstract

With ongoing research in intelligent transport systems and connected and automated vehicles, enabled by advancements in artificial intelligence, the large-scale advanced simulation has become an important part of product/software development for the automotive industry. Nowadays, traffic simulations are used to mimic real-world environment scenarios for connected vehicle technologies. The focus of this thesis lies in the development of microscopic traffic simulation calibration and enhance traffic signal control systems

This thesis makes the following major contributions. First, a calibration framework is proposed which harnesses the exiting data set of OSU campus shuttles (CABS) to determine the traffic state and create a microscopic traffic simulation. The traffic simulation is implemented for a section of the OSU campus("Woody Hayes Drive") which can be extended to the entire OSU campus.

The second contribution is an investigation of an intelligent traffic signal control system. The signal control operation is formulated as a decision-making process where each controller or control component is modeled as an intelligent agent. The agents make decisions based on traffic conditions and their past knowledge of the environment. A state estimation method and an adaptive control scheme by reinforcement learning (RL) are introduced to implement such an intelligent system. Simulation experiments

have been performed to verify the improvements of intelligent traffic control systems and compare them with the existing control policy.

The third contribution summarises the initial integration work for the co-simulation framework completed by dSpace ASM and SUMO to create a complete real-time simulation of urban environments for ADAS testing. The demo scenario is the OSU campus with traffic demand generated using the calibrated model from the first part of the thesis.

Dedicated to the safe and sustainable future of earth.

Acknowledgments

During my Bachelor's in Mechanical Engineering from BITS Pilani, I developed an interest in the automotive and mobility sector. To pursue research in the field of advanced mobility systems, I decided to pursue MS at OSU, and this journey has been a pleasant one. Having now completed my thesis, I would like to take this opportunity to thank the people who have been vital to me in this journey.

First of all, I want to express my deep gratitude to my academic advisor, Prof. Punit Tulpule, for his consistent guidance and support. I am very grateful for the freedom he gave me in exploring different research topics in this study.

I am indebted to Prof. Shawn Midlam Mohler and Prof. Punit for allowing me to work in SIMCenter OSU on this research project. I am thankful to Prof. Qadeer Ahmed for being a part of the thesis committee.

I owe a big word of thanks to my past and present colleagues at SIMCenter for their assistance and encouragement. A special thanks to Jacob A. Isaman for his assistance in the Hardware in Loop Co-simulation setup part of this study.

I am wholeheartedly appreciative of the Honda R&D North America(HRA) division and CAR consortium at OSU to support the study under this thesis. I want to thank Mr. Duane Detwiler, Mr. Allen Sheldon, and Mr. Rajeev Chhajer from HRA for their great inputs in this work.

Finally, thanks to my family for their constant support and love. Their unconditional support inspired me to achieve my dreams and follow my passion.

Vikhyat Kalra

Columbus, July 2021

Vita

2016	B.E (Hons.) Mechanical Engineering, Birla Institute of Technology and Science,Pilani, India
2016-2017	Graduate Engineer Trainee, Faiveley Transport India Pvt Ltd. Hosur, India
2017-2019	Executive Engineer, Honda Cars India Pvt Ltd. India
2020-present	Graduate Research Associate, The Ohio State University Columbus, Ohio

Publications

Research Publications

'Heat Transfer Simulation Of An Active Electronic Module with Thermoelectric Generators 'IHMTC-2015 Conference'. Dec. 2015.

Fields of Study

Major Field: Mechanical Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In the 21st century, urbanization has increased at a rapid pace. City planners and government needs to understand the impact of their decision on society in a controlled environment. With ever-increasing digital data acquisitions, the concept of smart cities has risen. A smart city harnesses data to build a digital model for its various subsystems, which interact in a community The components of1.1 the different subsystems in a city. The data acquired from various sources by the government agencies are used to develop multiple simulation models to plan the decision-making.

Digital Twin : The term digital twin exists since the early 2000s. According to **IBM**, a digital twin "is a virtual representation of a physical object or system across its life-cycle, using real-time data to enable understanding, learning and reasoning". **Siemens** adds to that the ability of a digital twin "to simulate, predict, and optimize the product and production system before investing in physical prototypes and assets." For an Urban planning concept, the digital twin is focused on modeling subsystems of a city environment. According to **Arup**, "the promise of the city digital twin is to help provide a simulation environment, test policy options, bring out dependencies and allow for collaboration across policy areas, while improving engagement with citizens and communities."

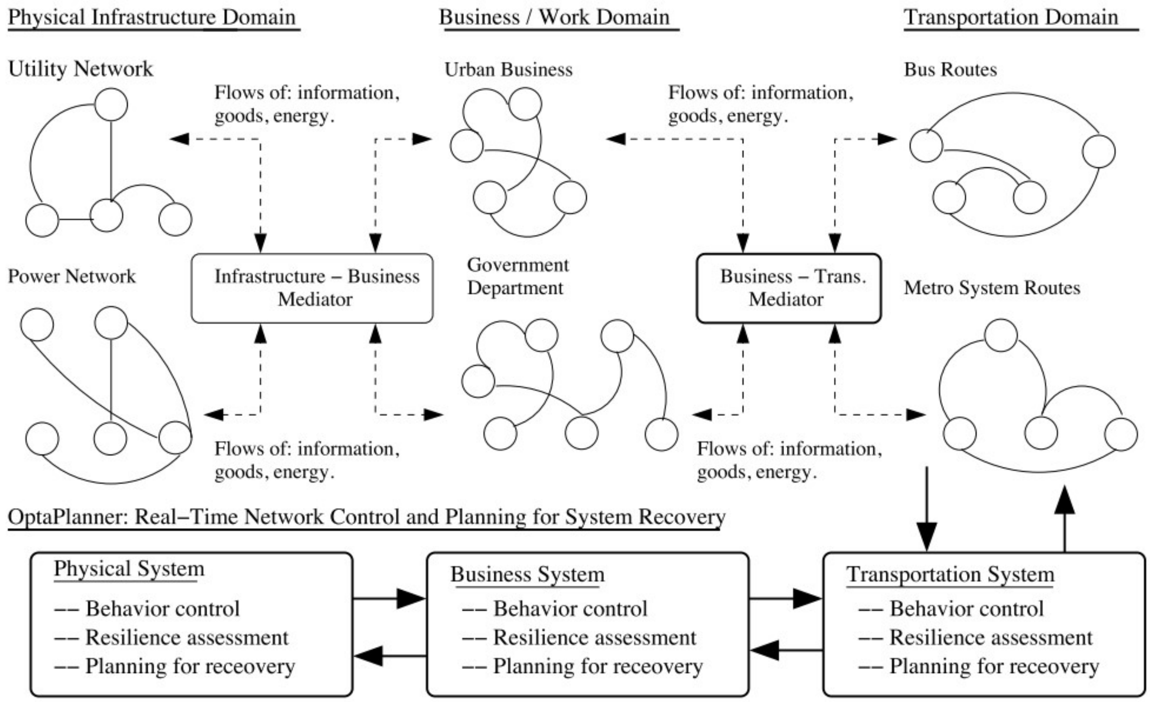


Figure 1.1: Illustration of relationship between city subsystems [1]

Traffic Challenge : As per the US Department of Transportation, there are 212 million licensed drivers in the USA. Americans own 252 million light-duty vehicles. Americans use approximately 180 billion gallons of fuel, driving 3.2 trillion miles a year[2]. The increasing number of vehicles overload the current road infrastructure, which creates congestion, leading to more fuel burning. More fuel consumption leads to irreversible environmental impacts.

In the last two decades, researchers and governments have focused a lot on solving the traffic congestion problems using multiple methods. In the initial stages of the 21st century, the traffic congestion problem was tackled by expanding the road infrastructure and adding more options for public transport. Expanding current road infrastructure is very costly, and restricted by available land space. The research focus to tackle the congestion problem has shifted to optimizing the current road infrastructure like traffic signals, dynamic lane assignments, dynamics speed limits. An accurate and reliable simulation model is needed to improve existing traffic systems. In a simulation modifications to the traffic lights and other parameters can improve traffic congestion and enhance safety on the road.

Figure 1.2: A typical traffic congestion scenario at intersection , Mumbai, India(Image Courtesy: Google Images: asurza.ca)

The way people are getting around is changing. Humankind is in the midst of a significant transformation in automobile transportation. With fast-paced research in connected and automated vehicles and intelligent infrastructure, soon there are many potentials to optimize the current traffic.

The next section asks the various research questions which are answered in this thesis.

1.1 Research Questions

Q1: What infrastructure characteristics are needed to depict the actual environment in the virtual environment accurately?

Q2: How can one define the parameters which characterize the traffic in an urban environment with high density?

Q1 is answered in the chapter 3that explores the road infrastructure: the number of lanes, speed limits, and traffic light phases directly impacting how the traffic flows through a network. Q2 is answered in the chapter 4. In this chapter, the travel time and calibration strategy are decided.

Q3: Can a dynamic traffic light control be implemented to reduce the traffic?

Q3 is answered in the chapter 5 which implements multi-agent reinforcement-based learning. The individual traffic light is characterized as a learning agent. The traffic simulation acts as an environment in which the algorithm learns and updates its control strategy to reach optimal phase timings.

Q4: How can a Co-simulation platform be developed to link high fidelity powertrain and vehicle dynamics models with the traffic model(low fidelity)?

Chapter 6 implements the Q4 in which the dSPACE ASM simulator is used as a high fidelity model platform and linked with the SUMO traffic simulator. The benefit of the interface between the high fidelity and low fidelity traffic model gives a platform for developing and testing various energy management and autonomous vehicle algorithms.

1.2 Motivation

In partnership with other departments and industry partners, the Center for Engineering at the Ohio State University is trying to create a digital replica for the campus. The digital twin("replica") would encompass various aspects within the campus like the building's energy and water requirements, waste resource management, communication, and mobility of people and goods around the campus 1.3. This thesis works contribute towards the mobility aspect of the digital twin model. The focus is to create a traffic simulation for road sections with the highest traffic density so that the same methodology can be expanded to the entire campus. The simulation model harnesses data sets from Campus Area Bus Service(CABS) from the period between 2011-2019. (Covid-19 pandemic drastically changed mobility during 2020 and 2021 academic year, thus data before 2020 is used.)

1.3 Objective

As explained in the section, 1.2 the motivation is to create a mobility simulation that can be expanded to the entire campus. This thesis aims to create a traffic simulation environment and calibrate it with real-world data-sets to capture the Ohio State University campus traffic scenario ("Woody Hayes Drive Section"). In 6a Co-simulation environment, with dSPACE Automotive Simulation Model(ASM) is developed, which can be used to create autonomous and energy management algorithms for vehicles. A5 multi-agent Reinforcement learning-based approach is

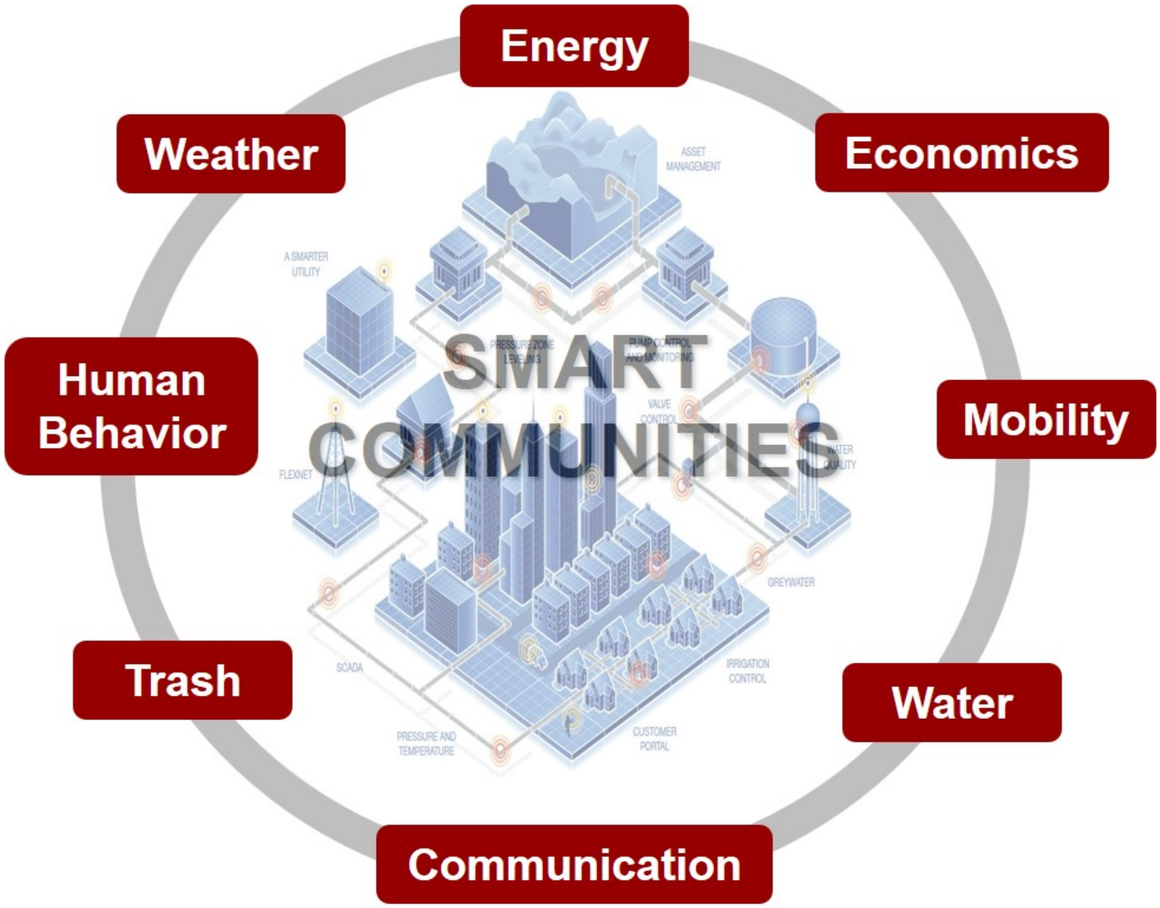


Figure 1.3: Smart Community Subsystems Interactions

used to optimize the traffic light green phase timings in the chapter.The objective for developing such a simulation platform is following:

- Use the simulation platform for testing emerging technologies like connected and automated CABS, smart traffic lights, dynamic speed limit, etc.
- Create an improved traffic light control strategy which can be implemented to handle dynamic traffic conditions
- Use simulation platform for testing out various powertrain energy management strategies for electric CABS.

- Run "what-if scenarios" to understand their impact on the current mobility conditions around campus.

- Explore what more data collection is required to enhance the traffic model.

Some of these scenario changes cannot be safely executed in the real environment. Thus a simulation platform is required.

1.4 Definition of Problem Statement

In this section, the work scope of this thesis is defined, and some terminology is introduced:

The notion of “calibration" in the thesis refers to the following issues:

- Analyze the existing traffic data sources to determine the traffic state.

- Estimation of vehicle counts from entry points in the network to estimate traffic flow at intersections.

- Create a structured framework for parameters tuning in model based on existing data to reduce the uncertainty of inputs to traffic simulation. [3]

The notion of “RL traffic light agents" in the thesis refers to implementing an Artificial Agent-based traffic light that gets a reward based on reducing the traffic congestion to the served lanes. The target of the agent is to maximize the reward by updating the green phase timings.

1.5 Contribution and Outline

The figure 1.4 gives an illustration of the structure of the thesis and the contribution. In the subsequent sections, readers are referred to relevant content in the specific chapters.

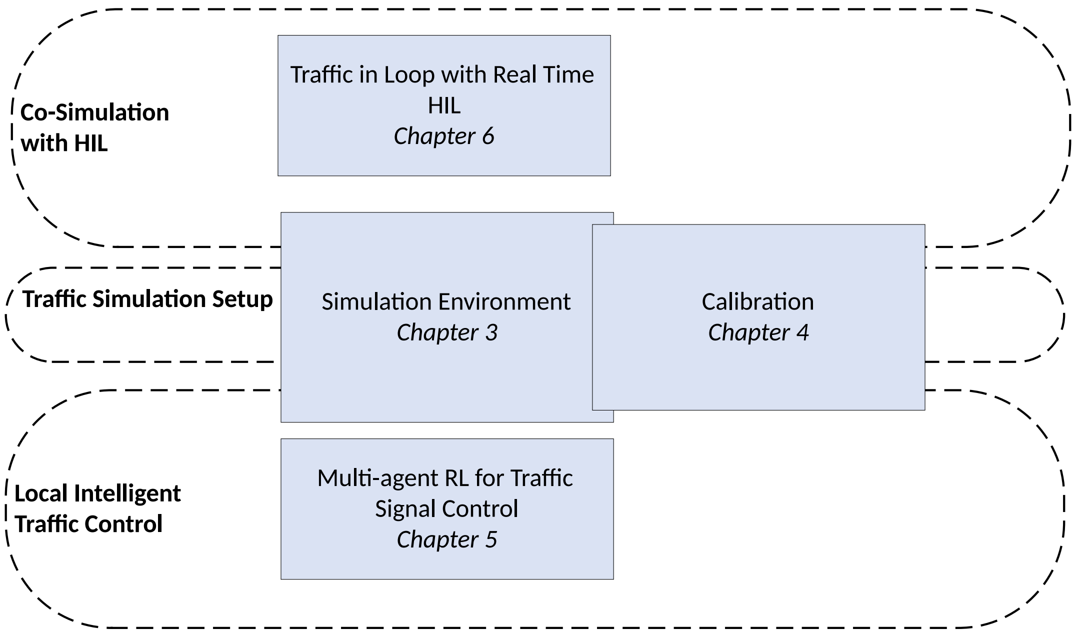


Figure 1.4: Illustration of the Contribution and Structure of the thesis.

Every chapter in the thesis describes a unique contribution. The below list presents brief insights into the contribution and contents of each chapter.

- Chapter 2 presents a high-level literature survey of various components of the project. The chapter starts with state of the art in traffic simulation. What efforts have been made in the real environment to map the traffic? Then chapter moves to explore which researchers use data types to calibrate a traffic simulation environment? The various traffic light control systems implemented are explained. Multi-agent Reinforcement Learning(RL) and its various implementation are explored. Then literature review for RL application in the field of traffic control is surveyed. In the last section of this chapter, Hardware in Loop modeling combined with traffic modeling is investigated.

- Chapter 3 outlines the SUMO traffic simulator. This chapter explains each component for a traffic simulation, starting from the network, vehicle demand modeling, to the car-following model. This chapter is very critical in setup for the chapter 4.

- Chapter 4 introduce the approach used to analyze the data sets received from the Traffic and Transpiration department(TTM). The floating car data (GPS timestamps) for CABS is analyzed in detail, and important ground truth parameters are extracted from the data. This chapter also describes the calibration approach and the objective function. The target for the calibration algorithm is to match the parameters extracted from the real environment data set to the simulation environment. Important simulation assumptions are mentioned in this chapter.A detailed analysis of the calibrated simulation is done. The shortcoming of the calibration approach is discussed. This chapter also examines what other real-world data might be needed to improve the calibration efforts.

- Chapter 5 investigates the implementation of the Multi-Agent RL approach to optimize the traffic light control algorithm. Local intelligent control is compared to the static phases approach.

- Chapter 6 integrate the calibrated simulation environment with a high fidelity vehicle dynamics & powertrain model. The high-fidelity model is running in an embedded hardware environment. A communication algorithm is introduced between the traffic simulator and the high fidelity model, which shares the data in real-time.
- Chapter 7 summarizes the contribution of this thesis and identifies the area for further research.

1.6 Definition of Common Terms

- **SUMO**: "Simulation for Urban Mobility". The open-source microscopic traffic simulator.
- **dSPACE SCALEXIO**: Hardware in Loop system provided by dSPACE used to simulate the Ego vehicle high fidelity models(high fidelity models are built-in Simulink.)
- **dSPACE ASM**: Automotive Simulation Model(ASM) is the commercial software package used to visualize the real-time running model.
- **Network or map**: Road infrastructure including lanes, traffic signals junctions, bus stops, pedestrian lanes, etc. This virtual representation of the connected roads, connections(defining movement directions) in the SUMO environment.
- **Edge** : A road is defined as edge in SUMO.
- **Host/Ego Vehicle**: Refers to a specialized vehicle whose powertrain and vehicle dynamics high fidelity model is running the dSPACE SCALEXIO hardware.

- **Leader Vehicle**: The vehicle which is immediately in front of ego vehicle in the same lanes.

- **Fellow/Neighbour Vehicles**: Vehicles from traffic that are in the nearby vicinity of the ego vehicle. These vehicles directly or indirectly affect the motion of the ego vehicle.

- **Route**: This is the defined path from origin to destination for a vehicle. The user specifies the path route for ego vehicle.

- **Flow**: Defined the number of vehicles entering from a particular direction in the network

Chapter 2: Literature Review

Simulation, according to Shannon (1975)[4], is “the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within limits imposed by a criterion or set of criteria) for the operation of the system."

In this chapter, current research and state of the art in the following fields of simulation is discussed:

- Traffic Flow Modeling
- Traffic Simulation Calibration
- Traffic Signal Optimization
- Co-Simulation for ADAS and powertrain in the traffic environment.

2.1 Traffic Flow Modeling

Traffic flow theory originated in the 1930s. Traffic flow theory was pioneered by American Bruce D Greenshields (fig 2.1). After the 1990s, the traffic demand has increased a lot, and more data can be collected; thus, traffic flow modeling has gained a lot of research interest. In the early 2000s, the research in autonomous cars and driver assisting systems has also helped advance understanding traffic flow dynamics at the microscopic level. The field of traffic research is mostly divided into two subcategories. The first one is *traffic flow modeling*, and the second is *transportation planning*. There is some difference between traffic flow modeling and transportation planning:

- Temporal aspect: Transportation planning time-span is very large, covering periods from several hours to days or even years. The traffic flow dynamics timescale is very small in the order of minutes to few hours.

- Objective aspect: Infrastructure and traffic demand is not variable in traffic flow dynamics, while transportation planning considers the dynamics of traffic demand and effects of infrastructure changes.

- Subjective aspect: The higher-level decisions like activity choice (number and type of trips), destination choice, mode choice, and route choice belong to transportation planning, whereas traffic flow dynamics analyses human/automated driving behavior (acceleration, braking, lane-changing, turning) [5]

There are different applications for traffic flow dynamics and transportation planning. Both the modeling types can approach a problem, but the solution would be quite different. Traffic flow dynamics can help in optimizing the current infrastructure to reduce traffic jams. Transportation planning is usually used to add infrastructure(new roads, traffic signal junctions, bridges, roundabouts, etc.) to alleviate traffic jams. A detailed understanding of the changes suggested by transportation planning is verified by using traffic flow models.

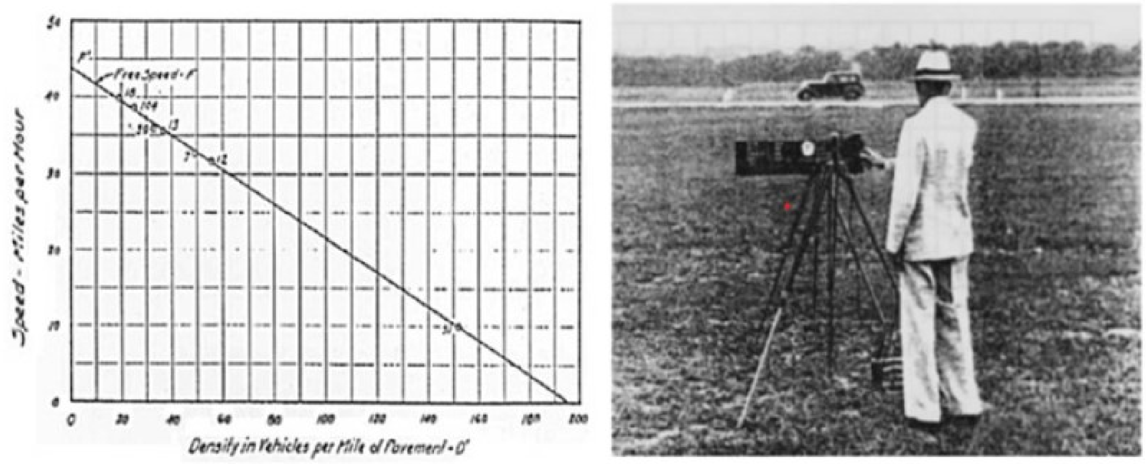


Figure 2.1: Traffic theory in the 1930s: Historical speed-density diagram and the experiment carried out by Bruce D. Greenshields. [From Greenshields, B.D., A study of traffic capacity. In: Proceedings of the Highway Research Board, Vol. 14. Highway Research Board, Washington, D.C. (1935)]

2.1.1 Model Classification

Traffic flow models can be classified by many aspects: Aggregation Level, mathematical structure, and conceptual aspects.

Aggregation Level

There are multiple ways to represent the real-world traffic events in the simulation(mathematical representation) (fig 2.2)

Macroscopic Models describe traffic flow similar to gas flow dynamics. These models are referred to as *hydrodynamics models*. The variables are traffic density $\rho(x,t)$, flow $Q(x,t)$, mean speed $V(x,t)$ or the speed variance $\sigma_V^2(x,t)$. These variables aggregate in a local environment; thus, the variables vary across space and time, i.e., they correspond to *dynamic fields*. Macroscopic models are useful in the below cases:

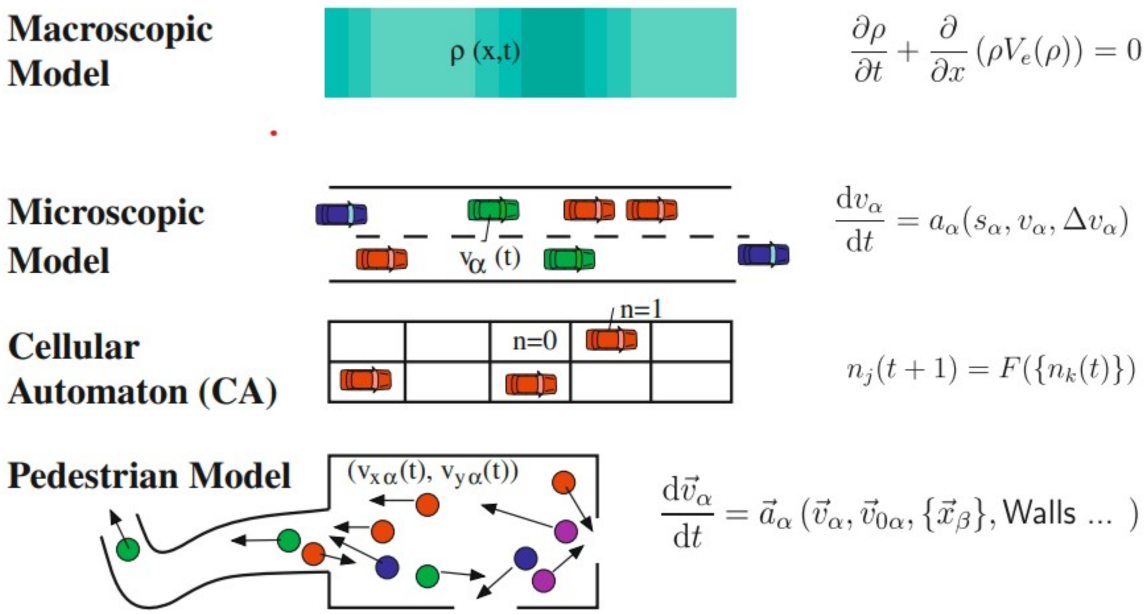


Figure 2.2: Comparison of various model categories (concerning the way they represent reality), including typical model equations [5]

- Effects that are computationally intensive for a macroscopic simulation(e.g., lane changes, several driver-vehicle types).
- Input data available is from heterogeneous sources or is inconsistent, thus making data fusion necessary.

Microscopic Models use car-following models and are focused on individual vehicle dynamics. The car-following models describe the reaction of each driver (acceleration, braking lane changes) based on the surrounding traffic. Microscopic traffic flow models are examples of driven multi-particle models. The variables which are dynamics are: $x_\alpha(t)$, speeds $v_\alpha(t)$, and accelerations $\dot{v}_\alpha(t)$. Microscopic models are used for the following applications :

- Modeling how single vehicle movement affects traffic: This field is becoming very important for the *advanced driver-assistance systems* (ADAS) such as adaptive cruise control (ACC) or Vehicle to Infrastructure (V2X) and other applications of Intelligent Transportation Systems.

- Modeling heterogeneous vehicle types and their interaction on the road. These models help understand the effects of speed limits or vehicle-specific lanes in urban/highway scenarios with heterogeneous vehicles.

- Predicting human driving behavior, including estimation errors, reaction times. Microscopic models allow accessing how various driving parameters affect traffic capacity and stability. [5]

Mesoscopic Models are an intermediary between macroscopic and macroscopic traffic flow models. They are similar to gas kinetic traffic models, which use idealized "collisions" to describe the dynamics of the quantity called phase-space density $\rho(x, t, v)$ which includes the traffic density and the local probability distribution of vehicle speed. Mesoscopic models can be called hybrid traffic flow models. Hybrid models can be built which describe the critical parts of the traffic network (intersections and traffic lights) microscopically, and the rest of the network is modeled using macroscopic dynamics [5].

2.2 Traffic Simulation Calibration

Microscopic simulations are widely used in transportation planning, design, and analysis due to their cost-effectiveness, risk-free and high-speed benefits. IN traffic simulations models, there are multiple independent parameters (e.g., car-following model, flow rate) that determine the movement of vehicles in a network. The usefulness of a microscopic simulation is increasing if it can accurately depict the real work traffic. To match the traffic from a real environment, many sensors can be installed, which help determine the state of the traffic. In this section, the data types collected by various sensors, their application, and the optimization algorithms used to calibrate the simulation parameters to match the simulation data with the observed data from the real-world sensors.

2.2.1 Traffic Data

Traffic simulations represent a crucial element for today's mobility. For determining the state of traffic, many sensor technologies can be used. The parameters which determine the state of traffic in a network are :

- Origin-Destination Pairs
- Vehicle Counts
- Vehicle Flow
- Vehicle Queue Density on Lanes
- Travel Time
- Average Speed

The sensors used to determine the above-mentioned traffic states are:

Vehicle Detector Loops(VDL)

Vehicle Detector Loops are permanent counting stations that are the most common data source for traffic state estimation(2.3). The data provided by the VDL is limited in area coverage and vulnerable to errors and malfunction [6]. The data provided by the VDL is vehicles counts and flow for a particular lane in the network. The data is aggregated over a period of time. The usual time period for data collection is 15 minutes. Every 15 minutes, the data is aggregated and then stored in a central server. Inductor loops are the most cost-efficient method for estimating traffic used by government departments. The Ohio Department of Transportation provides this data to the public on their website[7]. The fig 2.4 represents the location of VDL marked in blue squares.

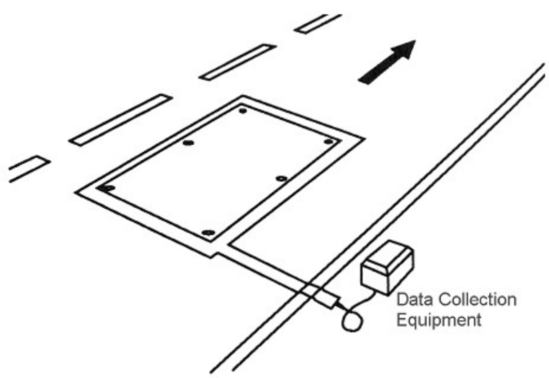


Figure 2.3: Mat Type Inductor Loop(image courtesy: www.fhwa.dot.gov)

2.2.2 Calibration Algorithms

A calibration process must include: the definition of an objective/fitness function to evaluate the performance, the parameter that will be optimized and calibrated, and the algorithm developed to optimize the calibration process and minimize the objective function[8].

Figure 2.4: Snapshot of Inductor Locations in Columbus, Ohio

Parameters

In traffic calibrations, most of the research is done to optimize the car following model parameters. Even in the car following model, every researcher focus on different parameters. For example, Ciuffo et al. calibrated only reaction time and speed acceptance[9]. In Cheu et al.'s research, the mainline free-flow speeds upstream and downstream of North Buena Vista Road off-ramp and the free-flow speeds at on-ramps and off-ramps, as well as the parameters that control the movement of vehicles (e.g., minimum car-following distance and sensitivity factor) were calibrated. A total of 12 parameters were calibrated in their research[10]. In Paz et al.'s study, 11 parameters.

Generally, less number of parameters can help the researcher on focusing better on the calibration when its value is changed. However, some parameters can have a much more significant impact on calibration compared to others when combined. Thus usually, the parameter optimization for calibration is not global optima for an objective function. Usually, algorithms find local optima. In this thesis, the focus is on flow probability parameters that decide the vehicle generation at the entry points to the network. For freeways, and 15 parameters for surface streets were calibrated[11].

Optimization algorithms

To obtain a match between simulation and observed(calibration data) traffic measurements a proper calibration of traffic simulation model parameters needs to be done. As there are large number of unknown parameters involved in microscopic simulation , the calibration task is a time-consuming complex task. The task of the optimization algorithm is to minmize the difference between calibration data and the simulation data by varying the parameters. Optimization algorithms can be broadly classified into two categories:

- Gradient dependent algorithms.
- Heuristic search algorithms.

The traffic simulation and calibration problem is a non-convex optimization problem [12]. The gradient dependent algorithms are not efficient to solve non-convex problems because they are dependent on the initial parameters provided to it. In a non-convex optimization problem the search based algorithms like, Particle Swarm Optimization (PSO) and GA are much more widely used because of there independence on initial parameters. The gradient based algorithm cannot find the global optimum solution because of multiple local optimum(minimum) in the search space of objective function. The search based algorithm can be adapted by using large population size in the search space of objective function which helps to find the global optimum of the calibration objective function. In heuristic search algorithms,Genetic algorithm and Simultaneous Perturbation Stochastic Approximation (SPSA) are widely used in calibration studies([13], [14], [15], [16], [17], [18], [11], [19], [20]). Some other algorithms used are OptQuest/Multi start. Algorithm (OQMA), non-linear programming techniques, Particle Swarm (PS), and Trial-and-Error Method (IA). GA is used as a popular calibration method for micro-simulation models, and it has given near-global optima for the objective function. Cheu et al. has used the GA approach to calibrate FRESIM parameters(average speed and average volume), and the objective function was to match the detector outputs from simulation to Singapore case expressway data[21]. Ma and Abdul have used the GA-based optimization approach to calibrate the PARAMICS model[22]. In this thesis, GA is used to optimize the flow probability for vehicle generation.

2.3 Traffic Signal Optimization

The primary objective for a traffic signal control policy is the safe and efficient flow of traffic across the intersection. Due to advancements in AI and data availability, a new algorithm can be developed which adapts the traffic light control policy based on the traffic congestion in its vicinity. Simulation-based optimization is a fast alternative to real-world testing of the control algorithms. Therefore, traffic simulators are often applied to evaluate optimized signal control systems (e.g.[23]). Many studies have been done to integrate traffic signal control with microscopic simulations.

Part et al.[24] were the first to integrate a genetic algorithm (GA) to minimize the average travel delay obtained in microscopic simulations with a local signal controller. The study was focused on over-saturated signalized intersections. In 2000 and 2009, Park et al. extended their work to maximize the traffic flow and reduce the overall emissions [25],[26].

Similar studies have been conducted by Stevanovic et al.([27],[28],[29]). Similar work has been done by Zhang et al.[30],they applied simulation-based optimization to arterial control, and various traffic simulators were used.

Li et al.[31] proposed a two-level optimization framework for optimal traffic signal control problems. The higher-level controller decided the traffic signal states to reduce the driver's average travel time, while the lower level aimed for achieving network equilibrium using the settings at a higher level.

2.3.1 Signal control by reinforcement learning

Some studies have simulated the signal timing decision as a sequential process. The green phase duration is adapted to the traffic determined by sensors. The sequential decision-making process is used in emerging AI fields like reinforcement learning (RL). Thorpe et al.[32] first implemented RL based agent approach to control an intersection. The simulation results show that RL based agent (adaptive controller) outperformed the fixed time controller by reducing the average wait time of vehicles crossing the intersection. For further details, a comprehensive work by Yau et al.[33] on a survey of various RL models dedicated to signal control systems can be referred.

In some RL studies for traffic control vehicles, states have been used to combine traffic signal states. Vehicle states can include information like vehicle position and destination. Khakis et al.[34] used vehicle states in their RL agent-based control. Implementing a vehicle state-based controller for traffic optimization may not be cost-effective since it requires a broad network of connected vehicles to infrastructure (V2I) so that vehicle information can be passed to the nearest local traffic controller.

2.4 Co-Simulation for ADAS and powertrain in the traffic environment

Modern vehicles systems are becoming complex with every new generation. Hardware in the Loop (HIL) simulation is used to support the development of various embedded controls in vehicle subsystems. With the new generation of ADAS and autonomous algorithms, researchers need real-time simulation platforms to test their models, which can mimic real work driving. The co-simulation platform consisting of HIL and traffic simulators can test V2X and V2I based algorithms. In recent years wireless communication has enabled cars to become part of smart city interacting systems[35]. To accurately simulate interacting systems in the context of a car- the Vehicle Ad-Hoc Network (VANET) - must therefore be simulated. For VANET simulation, the system under test(ego vehicle) and context (traffic and communication) are modeled in a city-scale simulation. The research work in the field of co-simulation is in the very nascent stage.

A simulation company **rFpro** has released its work on integrating a 3-D rendered city environment with a SUMO traffic simulator, and the end-user can decide the ego vehicle physics model. The fig 2.5 shows the important sub-component in the rFpro co-simulation framework [36].

In this thesis, the co-simulation between ego vehicle physics(modeled in Simulink) and SUMO is developed using dSpace ASM simulation environment.

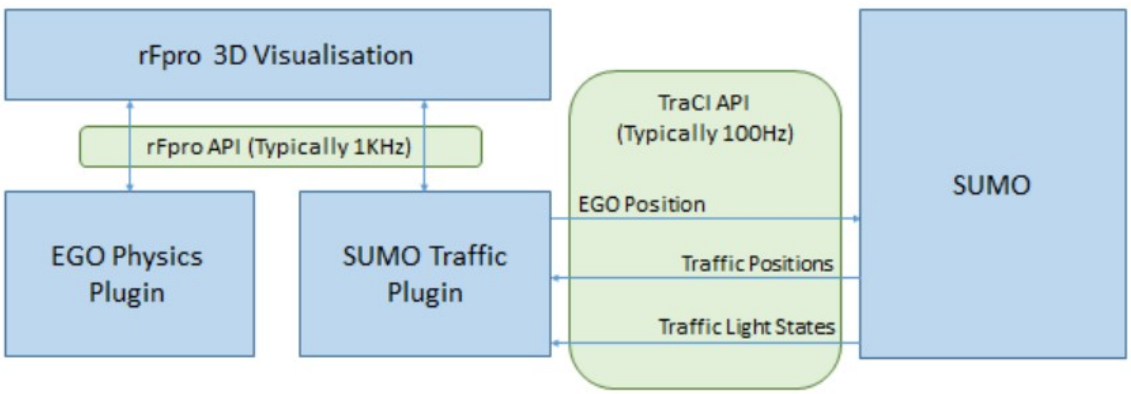


Figure 2.5: Data flows between rFpro, SUMO, and the end user's vehicle model vehicle

Chapter 3: Traffic Simulation Software

This chapter details the simulation software used for traffic generation. The software used for the virtual section of the OSU campus is SUMO(Simulation for Urban Mobility) [37]. The table 3.1 presents a preview of various traffic simulation software solutions. The objective of the thesis is to develop a simulation for traffic and a calibration algorithm. The simulation is calibrated with real-world data sets to represent traffic mobility accurately. For this purpose, a microscopic simulator is required. SUMO has both capabilities to simulate a dynamic traffic model and traffic light systems. The JAVA-based Matlab API called TRACI is utilized to integrate the heuristic (Genetic Algorithm) calibration algorithm.

3.1 Traffic Simulation

Traffic modeling is a mathematical representation of the component in the mobility systems. Traffic simulations involve the network elements like road infrastructure, traffic lights, junctions, roundabouts, and different modes of transportation. The elements like road infrastructure are static, but other model components like traffic lights, number of vehicles on road segments are dynamic. Traffic simulators employ different kinds of sub-models to simulate the interaction between various traffic modes. The traffic simulation technique can be classified into three types:

Modelling	**Software**	**Objective**
Microscopic	Vissim	1) Model urban transit traffic operations 2) Framework for animation of vehicle movements, traffic signal timings and travel time simulation results
Microsopic	AIMSUN	1) To produce real traffic conditions in urban network such as expressway and arterial routes 2) Simulation of traffic detectors like counts, occupancy etc.
Microscopic	SUMO	1) Comparison and evaluation of different car following and routing algorithms 2) Evaluation of traffic light systems
Microscopic	FRESIM(FREway)	1) Enhancement of geometric and operational capabilities in a road network and study effect of various car following and lane changing models
Microscopic	CORSIM	1) Simulation of traffic control systems(Traffic lights)
Microscopic	Paramics	1) Simulation of congested traffic networks
Macroscopic	Visum	1) Model traffic flows, forecast traffic congestions 2) Develop public transport routes
Macroscopic	Saturn	1) Evaluate impacts of one way streets & traffic control measures

Table 3.1: Traffic Simulation Softwares

1. Microscopic

2. Mesoscopic

3. Macroscopic

Microscopic

Microscopic traffic simulation is based on the emulation of individual motion of vehicles.SUMO falls under this category of traffic simulators. This simulation technique involves modeling vehicle dynamics, e.g., acceleration, deceleration, lane change of individual driver response to its surroundings. SUMO implements various car following and lane changing models to simulate the individual driver behavior, which are explained in section 3.2.5

Mesoscopic

Mesoscopic simulations create an aggregate level for vehicle groups. Vehicle groups or platoons are made from the general flow direction. The vehicles inside a group are assumed to be homogeneous. This model approach reduces the computational load compared to microscopic simulation as individual vehicle behavior is not modeled. Mesoscopic models are used in medium to large-scale networks.

Macroscopic

Macroscopic simulations are based on continuum traffic flow theory. The continuum flow theory variables characterize the flows: volume $q(x,t)$, speed $v(x,t)$, and density defined every instant in time t and every point in space x. The equation describing continuum flow theory is conservation equation ([38])

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = 0 \tag{3.1}$$

The equation 3.1 is completed by the following relationship:

$$q(x,t) = k(x,t)v(x,t) \tag{3.2}$$

3.2 SUMO Traffic Simulator

SUMO("Simulation for Urban MObility") is a microscopic simulator chosen for creating a virtual section of OSU campus mobility by the author. It is an open-source traffic simulator supporting multi-modal agents. SUMO is available under a public license(GPL). SUMO is developed in collaboration with the Center for Applied Informatics Cologne (ZAIK) and the Institute of Transportation Systems (ITS) at the German Aerospace Center (DLR). Figure 3.1shows how the traffic is assigned in the simulation within SUMO. SUMO is a discrete time-space continuous traffic simulator. In this thesis, SUMO is linked with a calibration algorithm developed in MATLAB using TraCI.

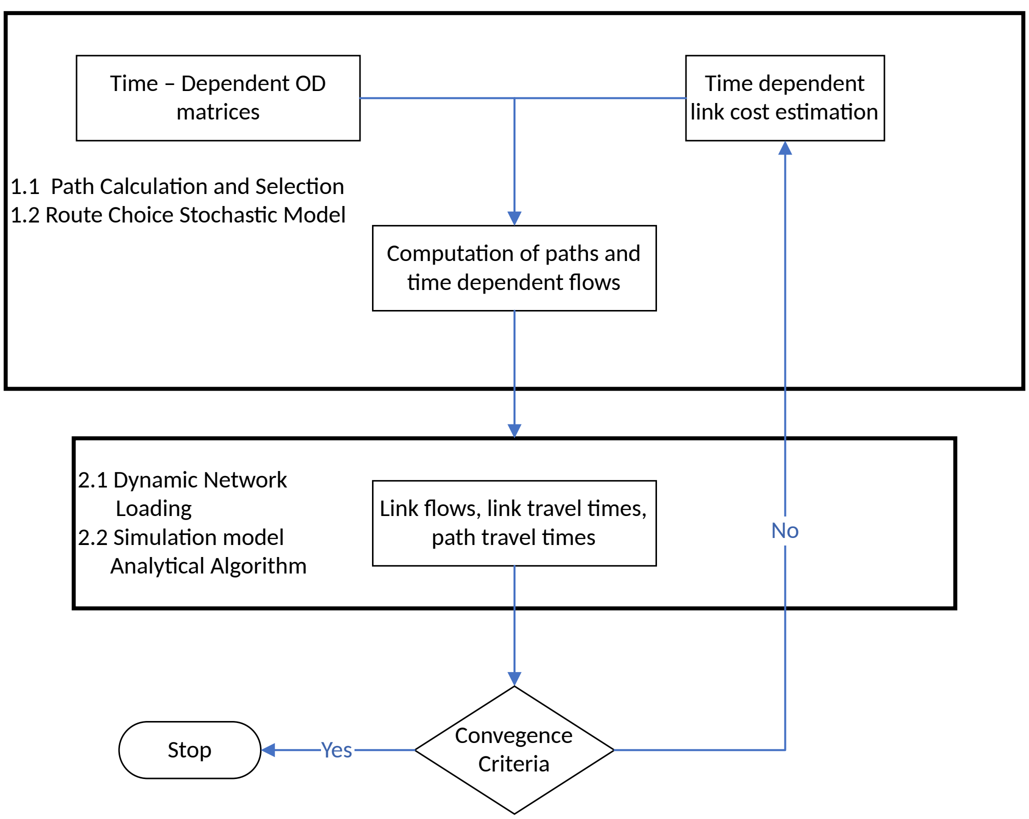


Figure 3.1: SUMO Traffic Assignment Algorithm

3.2.1 Model Building in SUMO

As SUMO is open-source software, the network and the vehicle demand are not predefined in the software package. SUMO has different tools to import road networks and generate traffic. In this thesis, the network map is derived using Open Street Maps(OSM). The network map derived from OSM does not have multiple lanes. The Java editor for OSM(JOSM) is used to match the satellite imaging(Bing Maps), and then the network is finally edited in NETEDIT to correct the missing lanes. The fig 3.2 depicts the file structure and data requirements for a SUMO simulation setup. The subsequent sections will explain each of the data requirements in detail.

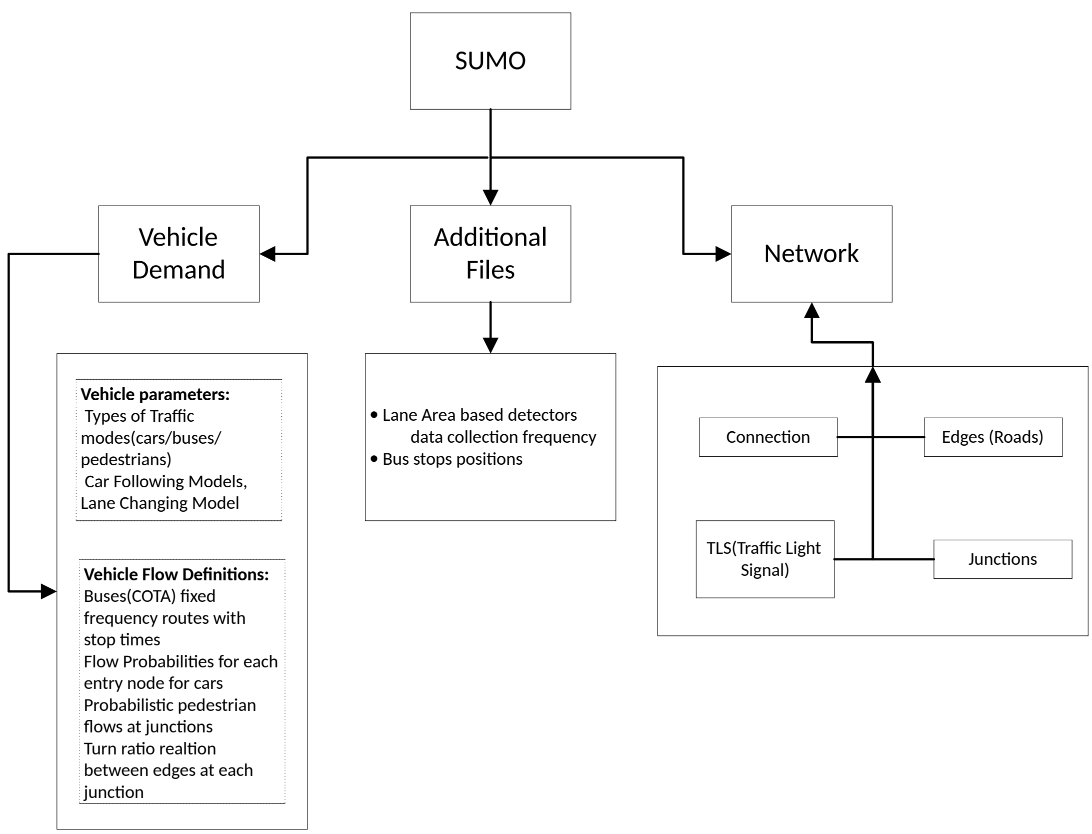


Figure 3.2: SUMO File Structure Visualization and Data Requirement

3.2.2 Network

Network in sumo is a directed graph consisting of roads(defined as edges in sumo), junctions, connections between edges and junctions, and traffic light definition. The network represents the infrastructure on which the different traffic modes interact.SUMO supports multiple types of network files import like open drive, OSM and XML. OSM is used in this thesis because of its better compatibility with the SUMO NETGENERATE(network import tool that converts osm to XML format). The network is the most critical component of the simulation.

Traffic Lights : Traffic light actuation is an important component to capture the traffic movement in simulation accurately. In SUMO, the traffic lights are defined in the network file using NETEDIT.

3.2.3 Additional Files(detectors)

As defined in fig 3.2 the additional files contain two components. The detector information and the bus stop positions in the network. There are multiple types of detectors in the SUMO that can be used to determine the traffic state in the simulation. Most of these detectors give output in an XML file structure, which can extract the information for traffic parameters like, total wait time, average travel time, average stops, etc. The various types of detectors in SUMO are defined below:

- **Induction Loop Detector(e1)** : An inductor loop detector detects the presence of a vehicle. This type of detector is usually used to generate aggregate counts/flows of vehicles on a particular lane. Please refer to fig 3.4a with yellow marked e1 detectors on multiple lanes.

- **Lane Area Detectors(e2)**: This type of detector is used to determine the traffic on an area along a lane or lanes. There are two types of e2 detectors. One is a single lane, and another one with multiple lane coverage. In a real environment, e2 detectors are similar to camera sensors on intersections. Lane detectors have a specific length determined by start and end positions. These detectors are usually used to detect jams and queue lengths near traffic signalized control intersections. Please refer to fig 3.4b in which the e2 detector is marked with blue color

- **Multi-Entry-Exit Detectors(e3)**: The descriptions of these types of detectors include an exit and entry point on the network. There is no restriction on the number of entry and exit points. The e3 detectors in a real environment can be assumed to be roadside units(RSU) that need additional sensors to work and collect data Please refer to fig 3.4c with entry-point marked. The yellow line connects the entry to the exit point for the e3 detector.

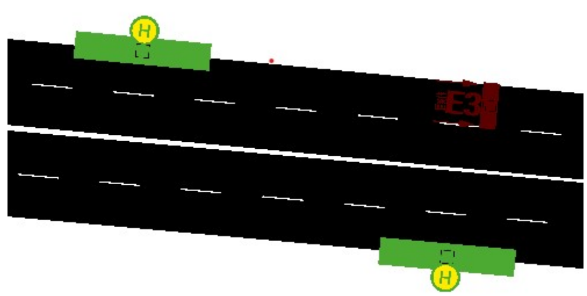
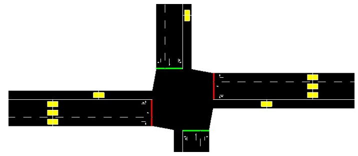
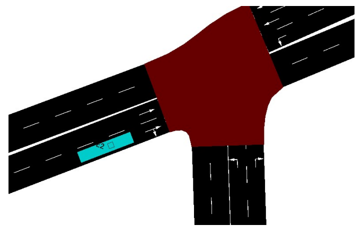


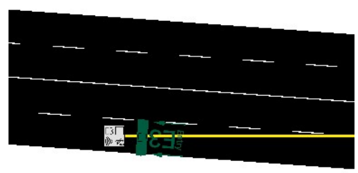
Figure 3.3: SUMO Network Bus Stop



(a) Induction Loop Detector:e1



(b) Lane Area Detector:e2



(c) Multi Entry-Exit Detectors:e3

Figure 3.4: Various detectors in SUMO

3.2.4 Vehicle Demand

Once the network setup is complete in NETEDIT, the user can define the traffic/vehicle demand. There are many ways of generating traffic demand in SUMO, as explained below:

- **Trip Definitions**: In SUMO, a trip definition consists of at least starting edge, ending edge, and the departure time. For converting trips to routes, a SUMO program called Dynamic user assignment(Duarouter) is used. Duarouter can access many types of routing algorithms to calculate the optimal path for the vehicle based on the edge's priority. Routing can be done using two different modes. The first mode is called routing by travel time and edge priority. The second mode is routing by effort. In SUMO, the objective of the routing algorithm is to minimize the travel time between origin and destination. The travel time can be calculated either from the speed limit and vehicle maximum speed or computed during the run-time of the simulation. The run-time option allows a vehicle to know beforehand if it will face a jam on some edges. This is similar to the human behavior model. Due to tools like Google Maps, humans on the road can dynamically detect the traffic and adjust their routes.

- **Flow Definitions**: This approach is very similar to the trip definition approach, but the benefit of using this approach is many vehicles having the exact origin and destination can be grouped in one flow definition. Flow definition needs starting and ending edge, the flow starts and end time, and also flow parameters like vehicle number per hour/probability of vehicle generation per second.

- **Random Demand**: This one uses a custom inbuilt script in SUMO called "randomTrips.py". This approach is an easy way to generate random traffic in the simulation network, but the results are usually highly unrealistic.

- **Origin-Destination (O-D) matrices**: This is the most used approach by traffic engineers to generate traffic in simulations. O-D matrices are usually available from traffic authorities. O-D matrices contain information for each vehicle's origin and destination. O-D matrices are used in SUMO in conjecture with "od2trips.py," which converts O-D definitions to trips. Due to the nonavailability of O-D matrices on the OSU campus author chooses another approach in this thesis.

- **Flow Definitions combined with turning ratios**: This approach modifies the flow definition approach. If destination edges are skipped and instead turn ratios at each junction are defined, this approach can be used at multi-junction types of network.

- **Detector data(observation points)**: In the USA, most of the city's transportation departments use inductor loops to generate large data sets of traffic counts on various roads.These inductor loop counts can be used by SUMO "dfrouter" to generate traffic demand.

- **Custom traffic demand by hand**:If the user wants, they can define trips using XML file format. This approach is not suitable for generating traffic demand for more extensive networks.

- **Population statistics**: To define traffic, users can define inhabitants, households, car rate, incoming traffic, outgoing traffic, etc. in a statistics file which is then used by a program in SUMO called "activitygen" to generate routes

3.2.5 Car Following Model

As explained in section 3.1 SUMO is a microscopic traffic simulator. To model individual vehicle driving physics, various types of car-following models are implemented in SUMO which is explained in the following subsections:

Krauss Model

The car-following model's basics are that velocity v of a vehicle i is affected by its leader velocity $i+1$ and the position gap between the two vehicles. There are other parameters like driver reaction time or sensitivity τ. (Krauss 1998). The default car following model in SUMO is the Krauss model. A vehicle can have two types of longitudinal motion in a traffic simulation: free-motion, i.e. no leader, and interacting motion. In free motion, as no leader is bounding the speed of the vehicle is determined by the speed limit of the road given by the equation3.4.

$$\frac{\partial v_i(t)}{\partial t} = f(v_{i+1}(t), x_{i+1}(t - x_i(t), \tau...)) \tag{3.3}$$

$$v \leq v_{max} \tag{3.4}$$

If the motion type is interacting, then there would be a leader in front of the vehicle. The car-following model must be collision-free thus, the equation determines the velocity of the vehicle. 3.5 and equation 3.6

$$v \leq v_{safe} \tag{3.5}$$

$$v_{safe}(t) = v_l(t) + \frac{g(t) - v_l(t)\tau}{\frac{v}{b(v)} + \tau} \tag{3.6}$$

$$t: \text{ time step}$$

$$v_l(t): \text{ velocity of leading vehicle at time } t$$

$$g(t): \text{ gap between vehicle and leading vehicle } i \text{ at time } t$$

$$\tau: \text{ reaction time of the driver}$$

$$b(v): \text{ deceleration function}$$

In the real world , vehicle dynamics(acceleration/deceleration) are affected by many road load factors like rolling friction, aerodynamic drag, and inertia forces. A desired v_{des} is calculated based on the powertrain limits. The desired speed of each vehicle is given by equation3.7. Human drivers have an imperfection in real life, and to capture this effect, a random error is subtracted from the desired speed v_{des} given by equation 3.8

$$v_{des}(t) = min[v_{safe}(t), v(t) + a, v_{max}] \tag{3.7}$$

$$v_(t) = max[0, rand[v_{des}(t) - a\epsilon, v_{des}(t)]] \tag{3.8}$$

Intelligent Driver Model(IDM)

The IDM car following model is derived from Optimal Velocity Model (OVM) (Treiber and Kesting 2010):

$$\dot{v} = \frac{v_{opt}(s) - v}{\tau} \tag{3.9}$$

$$s: \text{ current gap to leading vehicle}$$

$$v_{opt}(s): \text{ optimal velocity depends on gap } s$$

$$\tau: \text{ Time to adapt to new speed}$$

OVM model could not react to speed to the leading vehicle. The parameters it reacts are only relative distance. As OVM does not account for leading vehicle velocity, it is susceptible to accidents. IDM was modeled to correct this error. IDM is modeled on acceleration equation 3.10

$$\dot{v} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s*(v,\Delta v)}{s}\right)^{2}\right] \tag{3.10}$$

$$v: \text{ current velocity}$$

$$v_0: \text{ desired velocity}$$

$$s^*: \text{ desired gap}$$

The desired gap s^* is calculated by below equation:

$$s^*(v,\Delta v) = s_0 + max\left[0, \left(vT + \frac{v\Delta v}{2\sqrt{ab}}\right)\right] \tag{3.11}$$

Each vehicle type can have different values for the parameter of the model.

$$T: \text{ the time headway (between 0.8-2 seconds)}$$

$$s_0: \text{ the minimum gap}$$

$$a: \text{ acceleration (between 1-2 } m/s^2\text{)}$$

$$b: \text{ deceleration (between 1-2 } m/s^2\text{)}$$

3.3 Traffic Simulation Setup for OSU Campus

3.3.1 Network

The “Woody Hayes Drive" section of the campus is selected as the road network. A selection of the Woody Hayes Drive is selected for this study because:

1. This road section is within most Campus Area Bus Service (CABS).

2. This road connects student and staff parking spaces at Buckeye lot and Carmack on the west campus to the north east campus.

3. Woody Hayes provides a connection to US315 with north-campus surface parking, and parking garages. US315 is the major highway that connects suburban areas with OSU campus and Columbus downtown.

Figure 3.6 exhibits the OSM import for Woody Hayes Drive, and the complete network is divided into sections. Figure 3.5explains the steps taken to improve the default network imported from OSM. A JAVA-based tool called JOSM is used to superimpose bing maps for the network. JOSM helped to accurately map the lanes, which are very critical for traffic movement across junctions. The speed limit defined in the OSM maps is not updated. To correct the speed limit NETEDIT has been used to define the road limits for each edge of the network. The connections define the turn-only lanes in the network; for connections reference check fig 3.9. There are five controlled traffic light junctions on the network. The network is divided into six sections from west to east direction. The objective for diving the network into sections is to capture the traffic movement across controlled traffic lights accurately.

Figure 3.5: Network Import Process

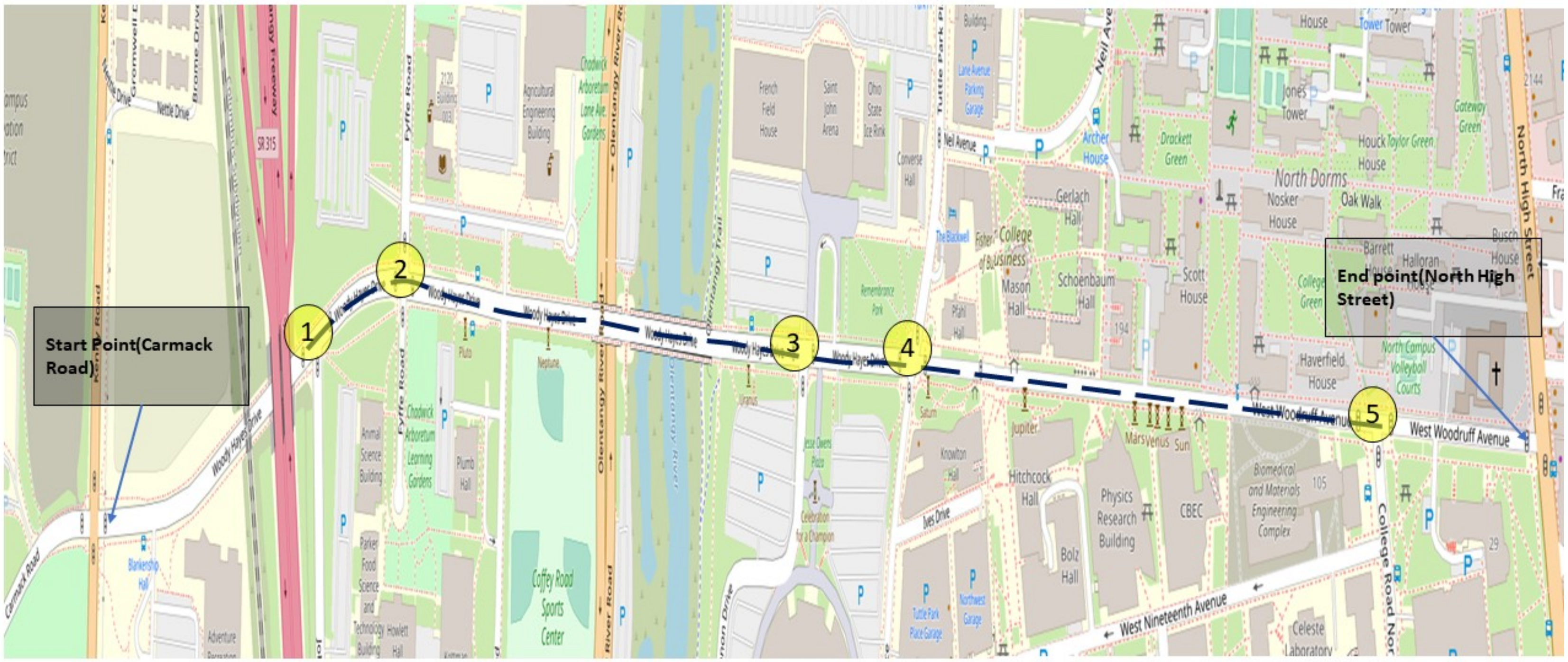


Figure 3.6: Woody Hayes Drive Network Junctions

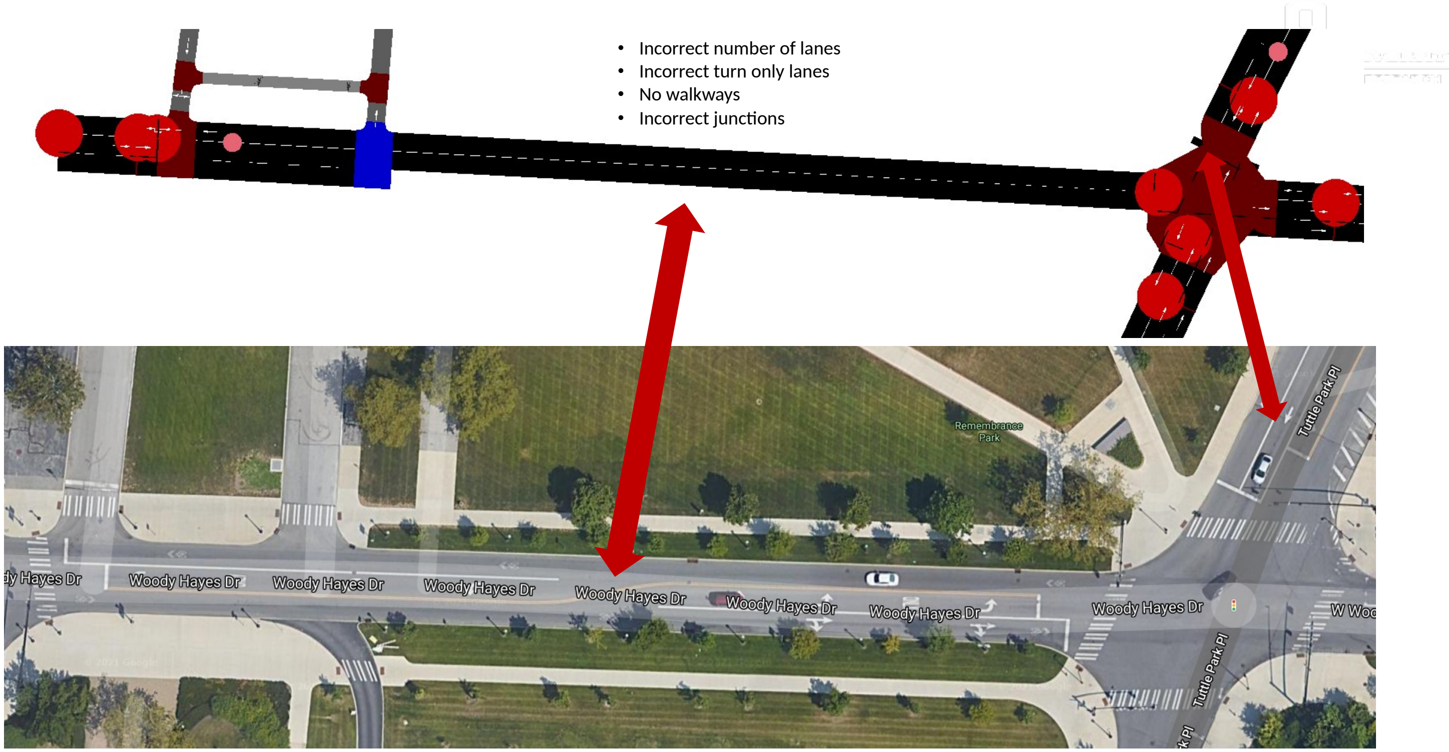
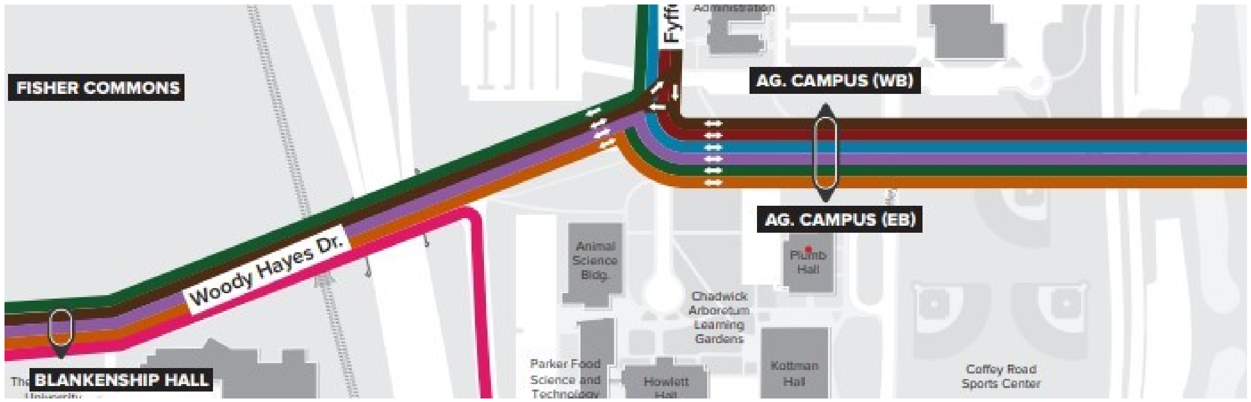
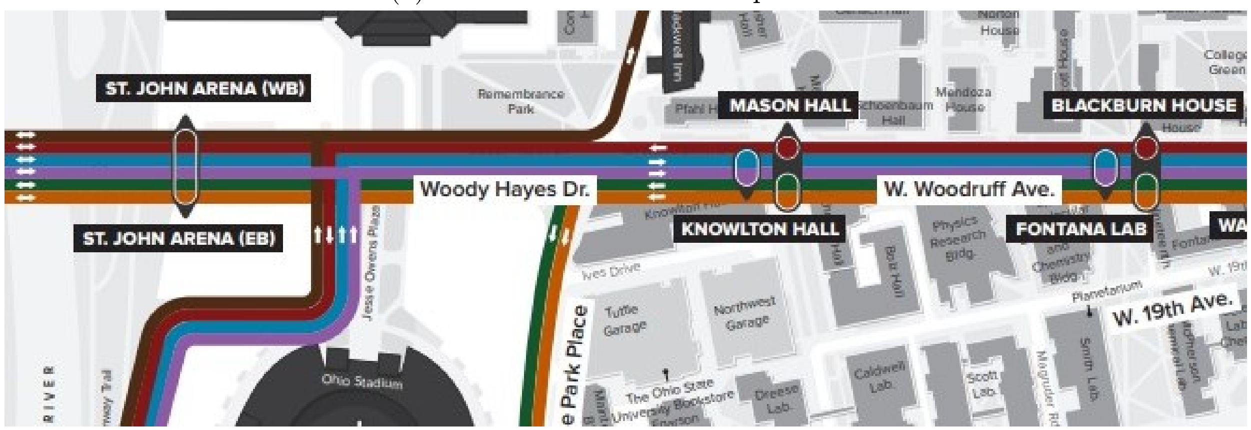


Figure 3.7: Missing Features in OSM Network



(a) CABS Routes & Bus Stops:Part1



(b) CABS Routes & Bus Stops:Part2

Figure 3.8: CABS Routes & Bus Stops on Woody Hayes Drive(Network Image in Two Parts)

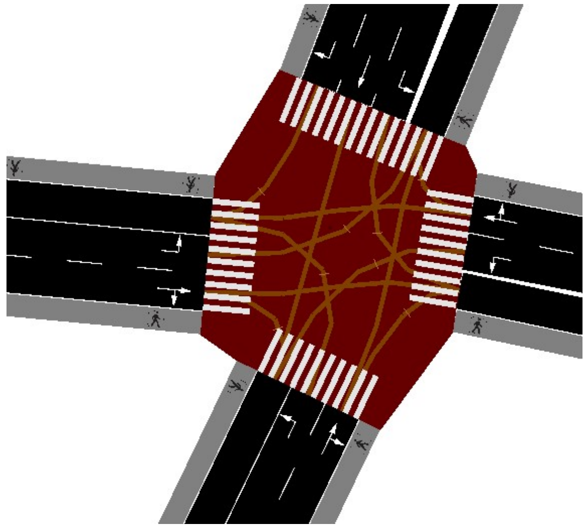


Figure 3.9: Connections

3.3.2 Traffic Lights

The author focuses on both kinds of traffic lights. The first four intersections mentioned in fig 3.6 work in a coordinated manner and the fifth junction has vehicle actuated traffic lights.

The traffic light controller timings for the five intersections are provided by Transportation and Traffic Management (TTM) OSU. The traffic lights on the Woody Hayes section can be divided into two categories:

- **Coordinated Traffic Lights**: Phase actuation is not dynamic. The green phases for each direction are already fixed in the controller. The coordination between different junctions is achieved by synchronizing the green phases for the west to the east direction in the network. There is a fixed offset time for the first phase of each traffic light predetermined in the controller to implement green waving in the network.

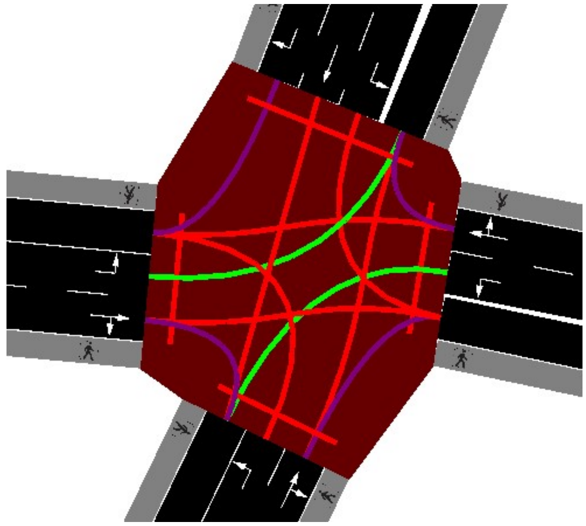


Figure 3.10: Traffic Light Phases:SUMO

- **Vehicle actuated Traffic lights**: Actua3ted traffic lights are based on using different detectors to verify the presence of a vehicle in the serviced lane. In OSU campus, cameras are used to detect the presence of vehicle fig 3.11. Actuated traffic lights have a minimum green phase timing, and the phase extends if the camera detects a vehicle on the lane. For each vehicle detected, the green phase extends. There is a maximum green time for each lane.

3.3.3 Virtual Detectors

In this thesis, e3 detectors are used to determine the average travel time for 12 segments in the network. There are 12 detectors placed in the simulation, giving the output of average travel time in E-W and W-E directions. The entry and exit points are selected so that each segment includes a traffic light junction, but no bus stops are included in the segment. An e3 detector calculates the travel time for a vehicle detected at both entry and exit points.

Figure 3.11: Camera on Traffic Light

There are nine bus stops for CABS routes on Woody Hayes Drive depicted in fig 3.8. The GPS location of the bus stops is mapped in the SUMO network. The SUMO network BUS stops are depicted in fig 3.3 highlighted with green color.

3.3.4 Vehicle Parameters & Car Following Model

The objective of the project is to generate a multi-modal traffic simulation scenario. There are three different modes of mobility interacting in the simulation. In the simulation setup, passenger cars, Buses and pedestrians have been used as mobility agents. There can be other modes as well, but these are the most prominent ones. As one can observe from fig 3.12 cars and buses are more than 90% of the total traffic on Woody Hayes drive.In sumo, various vehicle types are defined as vehicle classes.

The classes used in this simulation setup are explained below with their respective parameters are explained in the Table: 3.2

Table 3.2: Traffic Modes Parameters

vClass	**length width height** (m)	**minGap** (m)	**Maximum Acceleration** (m/s^2)	**Maximum Deceleration** (m/s^2)	**Emergency Braking Deceleration** (m/s^2)	**Maximum Speed** (km/h)	**Speed Deviation**
pedestrian	0.215 0.478 1.719	0.25	1.5	2	5	5.4	0.1
passenger cars	4.3 1.8 1.5	2.5	2.9	7.5	9	180	0.1
bus	12 2.5 3.4	2.5	1.2	4	7	85	0.1

This thesis uses the IDM model because it is more accurate in depicting actual-world driving. The IDM model selection is based on a literature survey. In the article, [39] authors have done a comprehensive analysis and compared the performance of different car following models to real-world driving data. As shown in fig 3.13 the IDM model average speed and average passing time is nearest to the real data captured by L.Bieker et. al. in their work.

3.3.5 Vehicle Flow

For CABS routes flow definitions have been defined with bus stop timings and frequency-based in data received from TTM for various CABS routes on Woody Hayes Drive. The flow definition aspect of simulation will be explained in detail in Chapter 4.

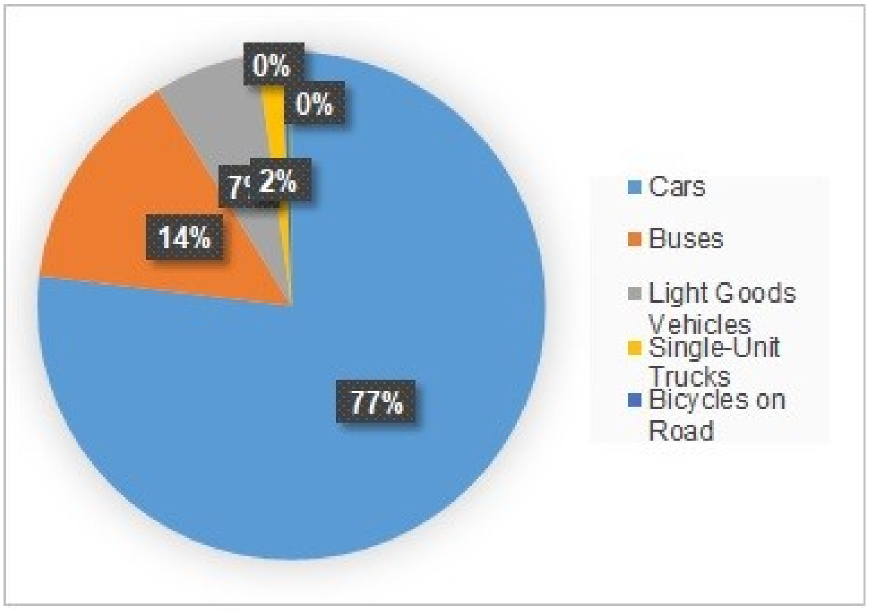
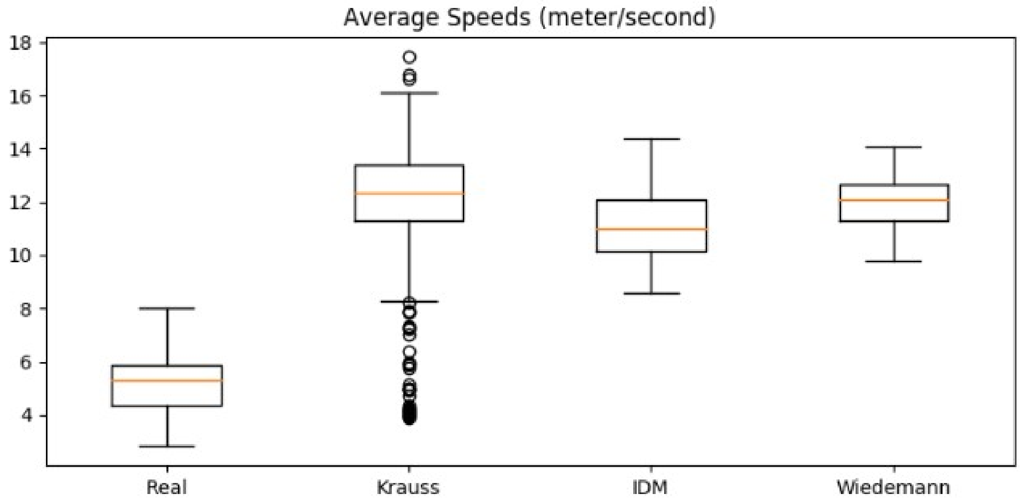


Figure 3.12: Traffic Modes Distribution

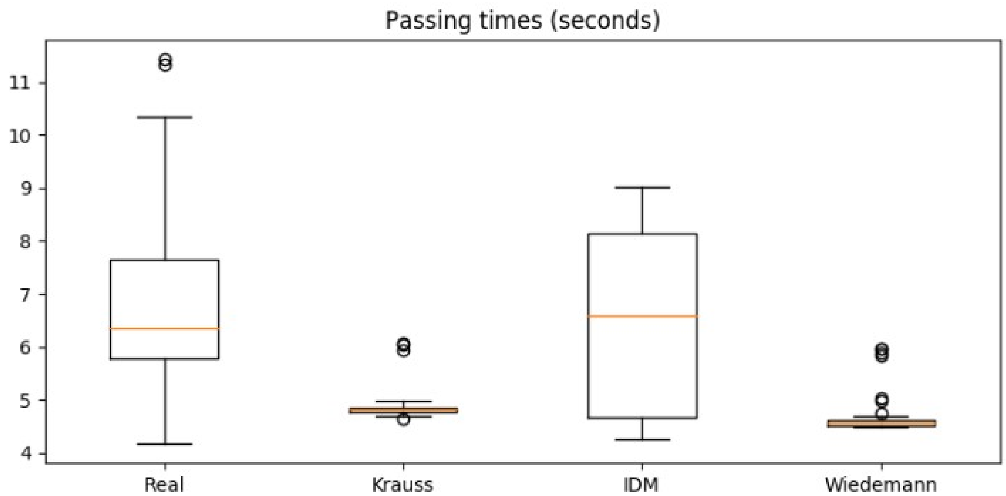
This thesis uses this approach as the turn ratios are fixed for each edge, and the optimization algorithm varies the starting flow probability. For passenger cars flow probability with turn ratio definition has been used. For CABS, this approach is not required as they have a fixed route defined. In SUMO, a program called "jtrrouter" is used to generate traffic from flow and turn ratio files. This approach will be explained in detail in chapter 4.jtrrouter send trip definitions to duarouter which uses shortest path algorithm called "Dijkstra."

3.4 Chapter Summary

Traffic simulation setup in SUMO was introduced. This chapter explained the critical components of a microscopic traffic simulation. Different traffic modes were explained. The reasoning behind the selection of Woody hayes Drive for the network was presented. Traffic lights and their types were introduced. The working of different



(a) Speed Comparison



(b) Travel Time Comparison

Figure 3.13: Comparison of car following model with real driving data at intersection [39]

detectors in SUMO and how they were linked to traffic measurement were explained in this chapter. The vehicle demand subsection explains various traffic methods, and the selection criteria for the turn-based algorithm were presented.In last section car following models **Krauss** and **IDM** were explained. IDM was selected for the car-following model in this simulation based on the literature review.

Chapter 4: Traffic Simulation Calibration

A microscopic traffic simulation model needs a variety of information as inputs . The model's fidelity is linked to the input data . The model accuracy increases as sample size for data collection from real environment is increased . It isn't easy to collect sufficient data sets in many scenarios that accurately depict the traffic state.In case of limited data, the traffic simulation engineer has to set specific targets for the simulation to match the input calibration data set. There are challenges in data acquisition for traffic state estimation because of cost & infrastructure restrictions..

The target for the calibration process for traffic models is to sufficiently match the simulation results with calibration data sets, representing the traffic state. A calibration process includes the following stages:

- Gathering data for the traffic simulation calibration which represent the traffic state in real environment.
- Definition of criteria to evaluate the performance of simulation in terms of the objective function.
- Selection of parameters for the calibration.
- Design of an optimization algorithm to minimize the objective function.

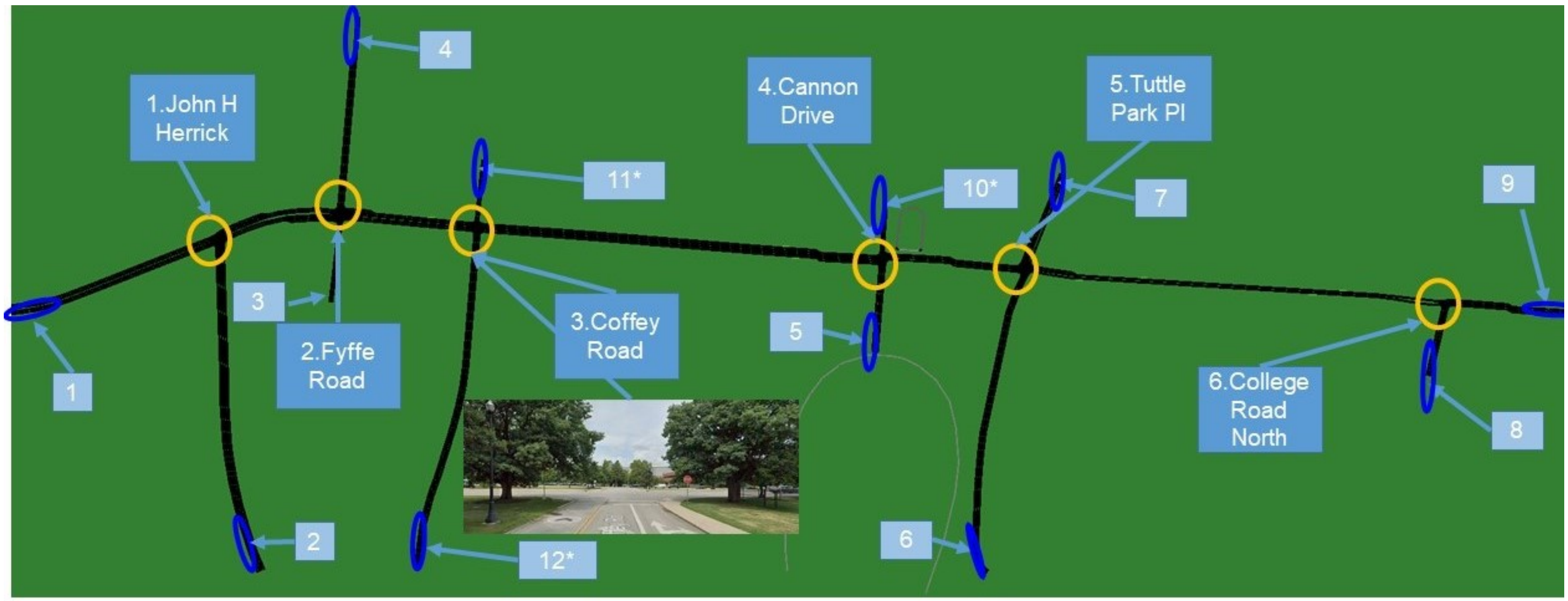
This chapter will introduce all the above stages for calibration.

4.1 Input Data

4.1.1 CABS GPS Data

OSU TTM provides the primary data source used in this thesis. The data used is GPS logged points from the CABS busses. There are various CABS routes on the Woody Hayes Drive network, but the challenge with this data is that the frequency for data capture is not consistent. The average interval for data capture from CABS GPS is 30 seconds. Refer fig 4.1 depicting the position and speed for a CAB route. The sparse arrows marked in the figure show the less frequent GPS logging for a route . The parameter used for determining the traffic state is the travel time between sections of the entire network . The network as a whole is divided into 12 sections 4.2and 12 entry points. Out of 12 entry points, nine are used to estimate the traffic flow. For the other three marked with stars (refer fig 4.2) in the network, the flow probability is constant(0.001) for the simulation .

Figure 4.1: CABS Route GPS marked



- Total six intersections are included in the network.
- Total 12 segments are used for error calculation. 6 in West to East direction and 6 in the opposite way.
- The Coffey Road Intersection in not signal regulated there is only a stop sign for both NB and SB direction.
- Other 5 intersections are signalized and except college road north all other works in coordination.
- Sumo needs 12 directional flows but entry 10* is from parking lot C(very few vehicles coming outside).

Figure 4.2: SUMO Network With Sections and Entry Points

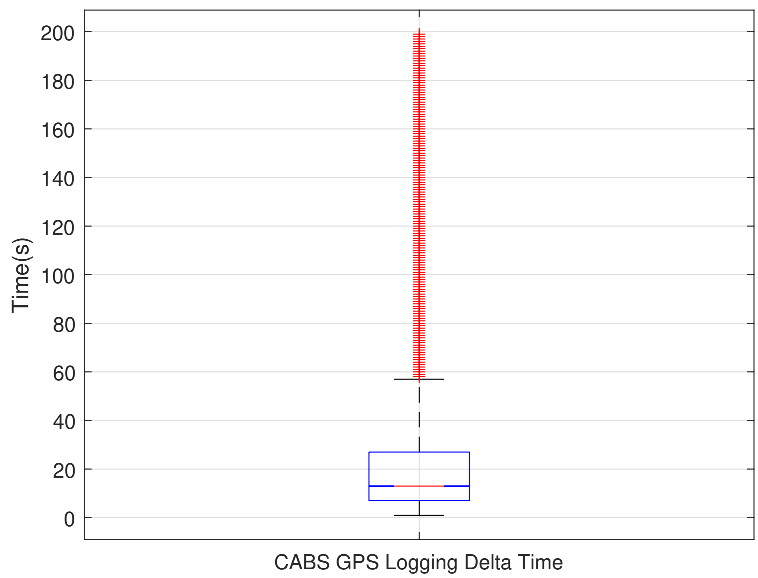


Figure 4.3: CABS GPS Logging

The fig.4.3 shows the distribution of delta time(time gap between consecutive GPS timestamps for a CABS route).It can be observed from the figure that the logging frequency is very low and inconsistent which increases uncertainty in determining the exact position and speed for CABS.In the fig 4.4,each box bar represents the distribution in the travel time for one hour from CABS GPS data.

As explained in the previous section due to inconsistency in the frequency of logging GPS locations, an algorithm is developed to calculate the average travel time for the specified section length. The algorithm defines two reference points in the CABS route. The algorithm process the GPS data file and search for the nearest GPS point logged during the trip. The distance and timestamps between the two GPS points are extracted to calculate the average speed. This average speed is then used to compute the average travel time between the specified section length. The

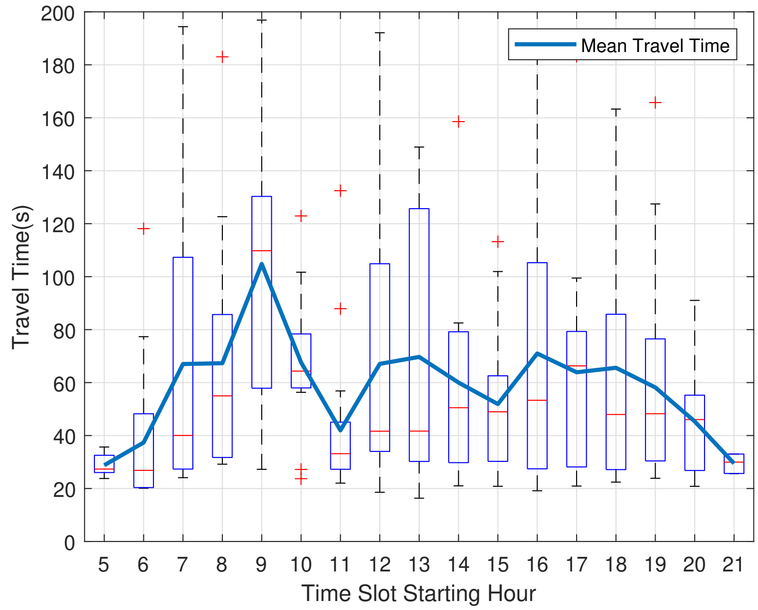


Figure 4.4: Travel Time for Tuttle Park Intersection

fundamental for the travel time algorithm is depicted in fig 4.5. The **R** used in the algorithm is 30 meters. The sections in the network are selected so that each section has one traffic light, and no section included the bus stop as stop time at the bus stop can add noise in the travel time calculation.

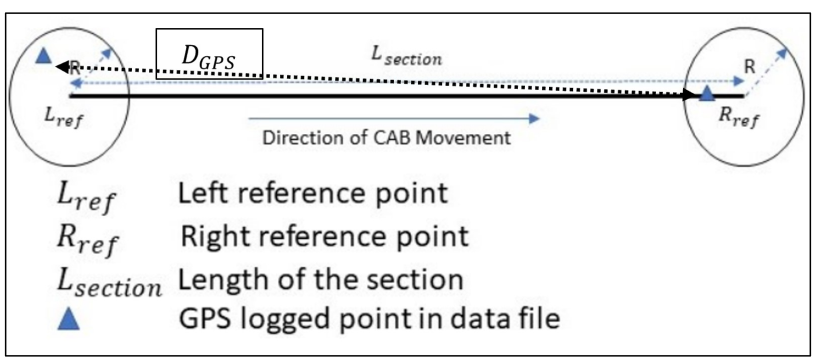


Figure 4.5: Travel Time Section for CABS

$$\overline{V} = \frac{D_{GPS}}{\Delta T_{timestamp}} \tag{4.1}$$

$$\overline{T} = \frac{L_{section}}{\overline{V}} \tag{4.2}$$

$\overline{V}$: Average Speed

$\Delta T_{timestamp}$: Difference between detected GPS logged timestamps

$\overline{T}$: Average Travel Time from CABS GPS for a section j

$L_{section}$: Section length in m

4.1.2 Count Data at Intersection

The travel time data is not sufficient enough to determine the state of traffic because CABS are less than 20% in the network. thus, traffic count data provided by *Carpenter Marty (CM) Transportation Inc.* is used to calculate the turn ratios at each intersection The fig 4.6 shows an example of how the data that is gathered around an intersection. This count data is used to generate the probability ratio, the turn at each intersection (example figure, for turn ratio of one lane 4.7).In the fig 4.8 the turn ratio comparison for different times of the day can be observed.

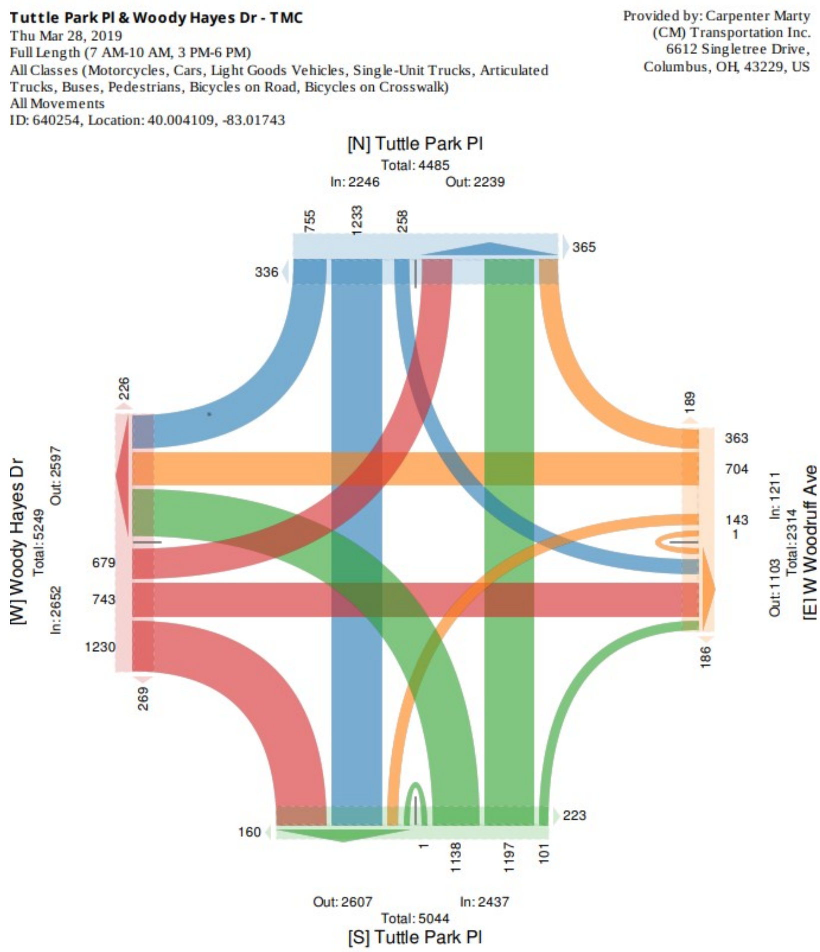


Figure 4.6: Tuttle Park Intersection Traffic Counts

4.1.3 Traffic Light Controller Data

The travel time for a vehicle is dependent on the traffic light timing. If a vehicle arrives at green phase during crossing the intersection its travel time would be less. To calibrate the model ,actual traffic phase timings used in Woody Hayes Drive traffic controllers is required to be emulated in the simulation. TTM also provided traffic light controller data for this work. The controller data is also verified in the field to check the accuracy of phase timings. The traffic controller in the OSU campus

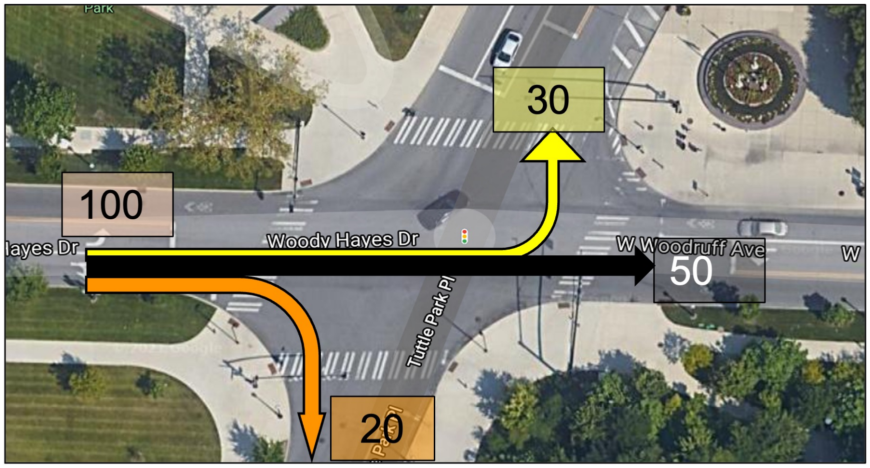


Figure 4.7: Turn ratio Example for a lane

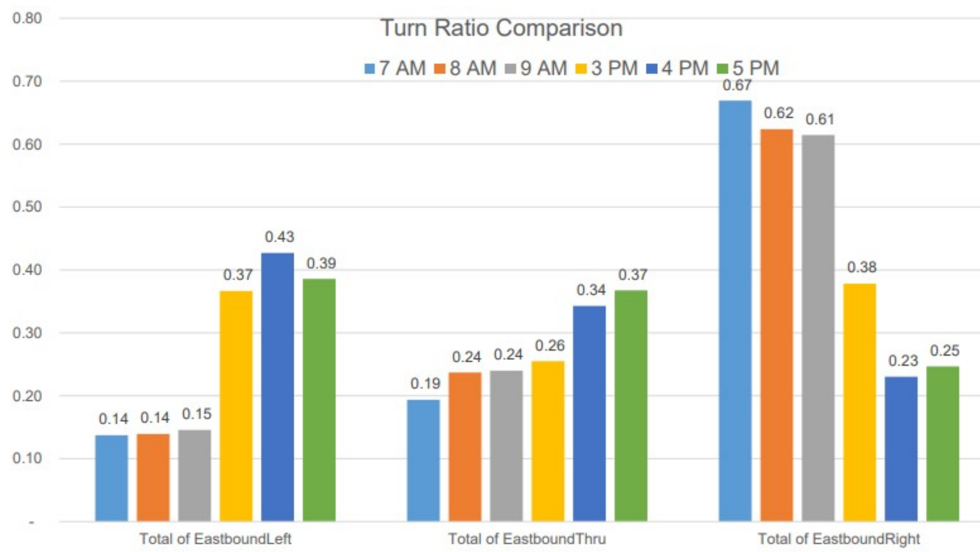


Figure 4.8: Turn Ratio for Tuttle Park East-Bound Traffic

works in two different modes during a weekday. From 8:00 a.m to 9:00 a.m first five intersections work in a coordinated manner(fixed phase timings), and the sixth intersection has vehicle actuated traffic lights 4.2. The sample data sheet used to extract controller phase timings is shown in fig. 4.9 The table 4.1 shows the extracted controller green phase timings for turn directions at Tuttle Park intersection. This green phase timings is imported in the simulation. The phases which are in same column occur concurrently in the traffic controller. The right turn moving traffic has to always yield to traffic moving from other direction.

```
OSU NORTH (WOODYHAYES@TUTTLE PARK PL) (9601)  FIBER OPTIC WOODY HAYES @ TUTTL

Coordination Patterns
-----------------------------------------------------------------
Pattern 1
Cycle Length . .  100   COS . . . . .  111
Offset . . . . .    0
Vehicle Permissive . . [1]    0   [2]    0
Vehicle Perm 2 Displacement   0   Phase Reservice. .  NO
Splits:     Phase 1-  15  2-  35  3-  15  4-  35
            Phase 5-  15  6-  35  7-  15  8-  35
            Phase 9-   0 10-   0 11-   0 12-   0   Split Sum: 0
Split Extension/Ring   [1]    0   [2]    0
Split Demand Pattern   [1]    0   [2]    0
XRT Pattern. . . 0
   Phase Number:  1   2   3   4   5   6   7   8   9  10  11  12
Coord Phases . . .    X   .   .   .   X   .   .   .   .   .   .
Veh Recall . . . .    X   .   .   .   X   .   .   .   .   .   .
Veh Max Recall . .    .   .   .   .   .   .   .   .   .   .   .
Ped Recall . . . .    X   .   .   .   X   .   .   .   .   .   .
Veh Omit . . . . .    .   .   .   .   .   .   .   .   .   .   .
Alt Sequence . .  A:  .   B:  .   C:  .   D:  .   E:  .   F:  .
                    ----------------------------------------
Pattern 2
Cycle Length . .  100   COS . . . . .  211
Offset . . . . .    0
Vehicle Permissive . . [1]    0   [2]    0
Vehicle Perm 2 Displacement   0   Phase Reservice. .  NO
Splits:     Phase 1-  15  2-  29  3-  15  4-  41
            Phase 5-  15  6-  29  7-  22  8-  34
            Phase 9-   0 10-   0 11-   0 12-   0   Split Sum: 0
Split Extension/Ring   [1]    0   [2]    0
Split Demand Pattern   [1]    0   [2]    0
XRT Pattern. . . 0
   Phase Number:  1   2   3   4   5   6   7   8   9  10  11  12
Coord Phases . . .    X   .   .   .   X   .   .   .   .   .   .
Veh Recall . . . .    X   .   .   .   X   .   .   .   .   .   .
Veh Max Recall . .    .   .   .   .   .   .   .   .   .   .   .
Ped Recall . . . .    X   .   .   .   X   .   .   .   .   .   .
Veh Omit . . . . .    .   .   .   .   .   .   .   .   .   .   .
Alt Sequence . .  A:  .   B:  .   C:  .   D:  .   E:  .   F:  .
```

Figure 4.9: Traffic Signal Controller Phase Time Data

4.2 Objective Function

In the section 2.2.2 different objective functions used in literature for traffic simulation calibration are introduced. In this thesis, the following objective function

Table 4.1: Green Phase Timings for Tuttle Park Intersection

Direction (Phase Number)	**Green Phase Time(s)**	**Direction (Phase Number)**	**Green Phase Time(s)**	**Direction (Phase Number)**	**Green Phase Time(s)**	**Direction (Phase Number)**	**Green Phase Time(s)**
East Bound Left Turn (1)	15	West Bound Thru (2)	29	South Bound Left Turn (3)	15	North Bound Thru (4)	41
West Bound Left Turn (5)	15	East Bound Thru (6)	29	North Bound Left Turn (7)	22	South Bound Thru(8)	34

is used to calibrate the traffic simulation. Point Mean Relative Error (PMRE). The selection of PMRE is based on the criteria that only travel time could be extracted. The absolute difference between the average travel time of simulation and calibration data is divided by calibration data travel time to properly scale for each lane. This scaling gives relative error for each intersection and ensures that the objective function is not biased for any section.

$$PMRE(T) = \sqrt{\sum_{j=1}^{j=N} \left(\frac{T_{obs,j} - T_{sim,j}}{T_{obs,j}} \right)^2} \tag{4.3}$$

Each vehicle type can have different values for the parameter of the model.

j : section number

N : Total Number of Sections(12)

$T_{obs,j}$: Average Travel Time from CABS GPS for section j

$T_{sim,j}$: Average Travel Time from SUMO Simulation for section j

The SUMO traffic simulation time is two hours. The $T_{sim,j}$ is calculated for the middle one hour in the sumo simulation.The first half-hour and last half-hour are not used to calculate the travel time because vehicles start entering and exiting during these periods. Thus the traffic has not reached an equilibrium state during the first and last half hours.The middle one hour is used when the traffic flow in the network is in equilibrium. For a particular SUMO simulation run, the average travel time across a section of the network can be defined by the below function:

$$T_{sim,j} = f(V_{count}, P_{counts}, SpaT, T_{stop}, Car_{following}) \tag{4.4}$$

V_{count} : Number of Vehicles in lane

P_{counts} : Number of Pedestrian Crossing an Intersection

$SpaT$: Signal Phase and Timing

T_{stop} : Stop Time for Cabs at Bus Stops

$C_{following}$: IDM Car Following Model Parameters

$T_{sim,j}$: Average Travel Time from SUMO Simulation for section j

Assumptions

Due to the shortage of traffic data, there are some assumptions in the traffic simulation. The assumptions are listed below:

- Average CABS bus stop time is 20 seconds for each bus stop (CABS have to stop at every bus stop irrespective of the riders).

- Pedestrian is generated with a consistent flow of one pedestrian every 60 seconds at every lane of the intersection crossing. Pedestrian interaction is focused at the Tuttle Park intersection. Tuttle park is selected for generating pedestrian because this intersection has the highest travel time and the right turning vehicles have to yield for any pedestrian. To capture effect of pedestrian on right turning vehicles this intersection is selected.

- Section lengths are small enough so that the average travel time for CABS and passenger cars should be comparable, i.e., due to high traffic density, passenger cars will not reach the acceleration limits in the network.

- IDM is used as the car-following model(refer section 3.3.4).

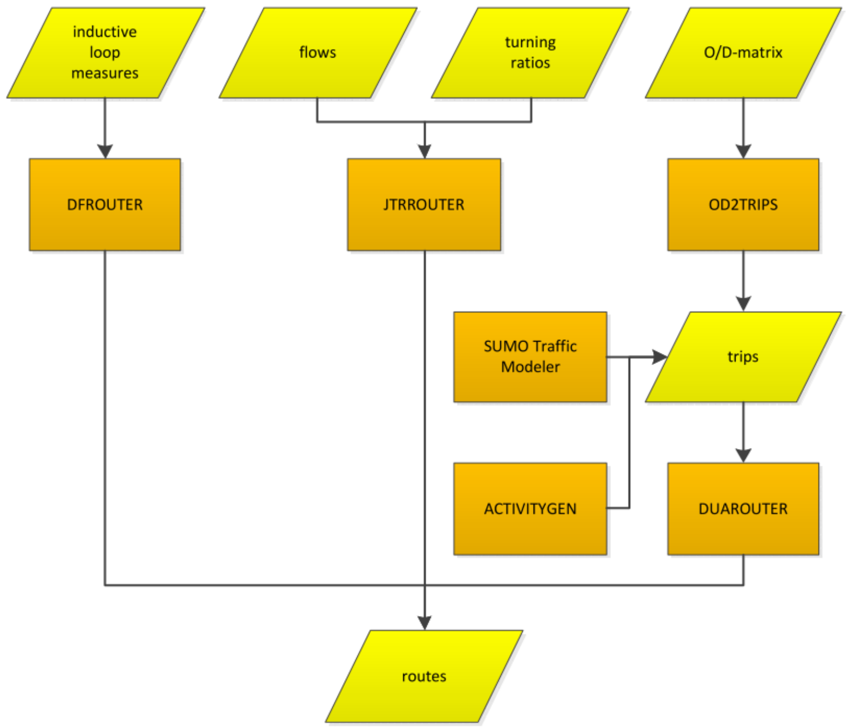


Figure 4.10: SUMO Routes Generation

4.3 Optimization Parameters

The objective of the calibration is to minimize the difference between CABS GPS data travel time and simulation travel time. The optimization algorithm minimizes the objective function 4.3. In the equation 4.4, only vehicle count is the variable that the optimization algorithm can change; other parameters are inputs from data or assumed to be constant. In the SUMO simulator, there are various ways to generate vehicles fig 4.10. In this thesis, the author has used “JTRROUTER" for routes generation. Turning ratios are generated from the count data, and the algorithm optimizes the “flow" to minimize the objective function. In section,3.3.5 this approach is explained. The below equation describes the optimization flow:

The variable P_k decides the vehicle flow per hour. For example, if P_k is 0.10, then 180 vehicles will be generated in one hour from that entry point.

Optimisation problem :

$$Minimize\, PMRE(T_{sim}) = \sqrt{\sum_{j=1}^{j=N} \left(\frac{T_{obs,j} - T_{sim,j}}{T_{obs,j}} \right)^2} \tag{4.5}$$

$$T_{sim,j} \propto \frac{1}{P_k}$$

where:

$$P_k \in [0.05, 0.20] \tag{4.6}$$

P_k : Probability of Vehicle Generation from k^{th} entry in network

k : k is the entry number in network from 1 to 9

4.4 Optimization Algorithm

The optimization algorithm used to minimize the objective function 4.3 is a heuristic Genetic Algorithm(GA). The objective function 4.3 is nonlinear and discontinuous; thus, a gradient-based optimization algorithm cannot minimize the PMRE. Matlab's GA implementation is used to solve the problem. The objective function, data extraction from SUMO detectors, is also programmed in MATLAB. The parameters used for GA are described in the table 4.2. At every generation, 150 candidate solutions are generated as a population. The SUMO model is run through MATLAB for every candidate in the population. Once the SUMO execution is complete, e3 detectors in the SUMO network generate XML files.The e3 detector data collection frequency is 1Hz. The data from XML for each vehicle travel time across the section is parsed using MATLAB script. The parsed travel time data is used to calculate the average travel time for one hour. The best candidate solution is then used to generate population candidates for the next generation; this process is repeated till the target for ten generations is reached. The flow chart 4.11 explains the calibration process with the GA as an optimization algorithm.

Table 4.2: GA parameters

Parameter	**Value**
Population per Generation	150
Maximum Generation	10

4.5 Calibration Results

The GA optimization algorithm provides a sub-optimal solution for the objective function minimization. The fig 4.12 depicts the evolution for the best candidate solution among the population. At the end of 10^{th} generation, the best fitness(minimum objective function value) candidate(vehicle flow probability) is stored as the best solution4.3.

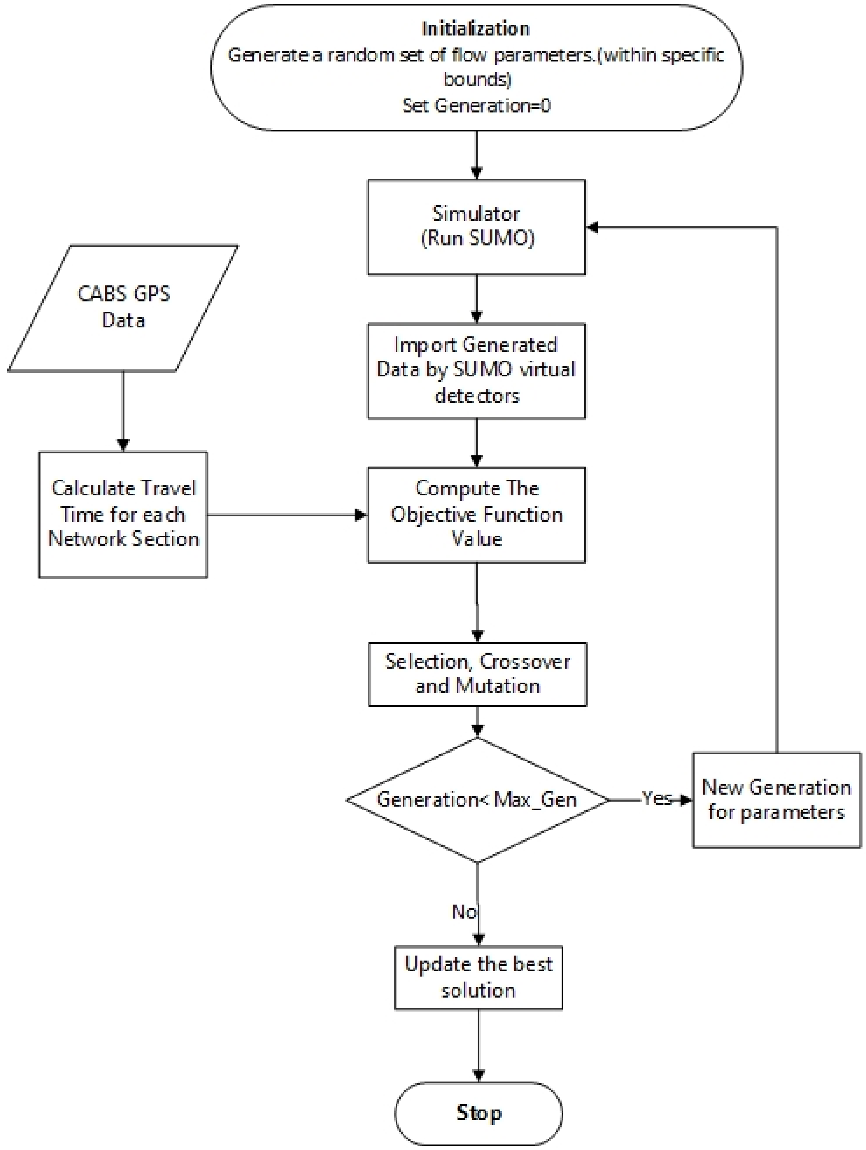


Figure 4.11: Calibration Process Flow Chart

From the fig 4.13the difference between the simulation and the actual(data extracted from CABS GPS) can be observed. For sections number 3,6,7,8, and 11, the error is above 30%. The absolute error is calculated using the below equation:

$$Error(e) = \frac{|T_{obs,j} - T_{sim,j}|}{T_{obs,j}} \tag{4.7}$$

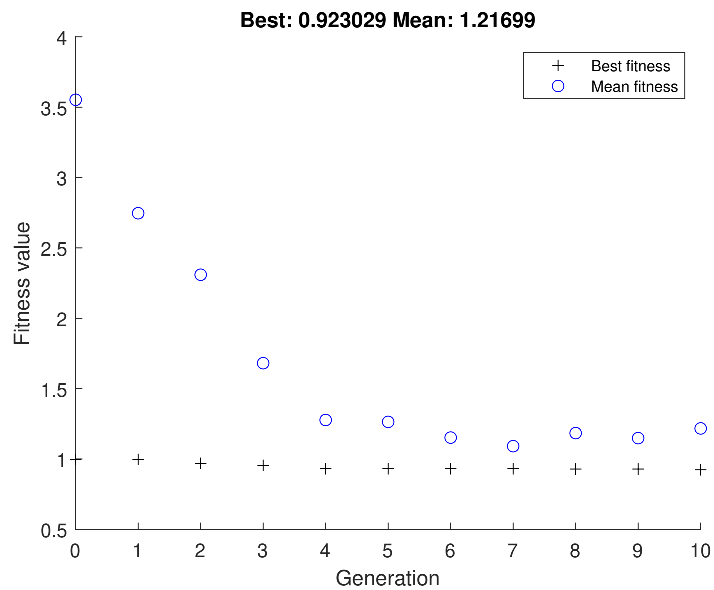


Figure 4.12: Genetic Algorithm Best Solution Evolution

Table 4.3: Flow Values from Network Entry After Calibration

Optimization Parameter Flow Probability (Entry Number)	**Value**	**Vehicle Flow per hour(vph)**
P(1)	0.05	180
P(2)	0.011	40
P(3)	0.09	324
P(4)	0.13	468
P(5)	0.07	252
P(6)	0.025	90
P(7)	0.016	58
P(8)	0.013	47
P(9)	0.082	295

In the fig 4.14 ,the frequency of travel time for simulation and calibration data is compared for section number 5 at tuttle park.It can be observed that the peak of the curve fit to both simulation and actual travel time is offset by 20 seconds. The reason for this offset is due to the assumption in the section 4.2. More data is required to

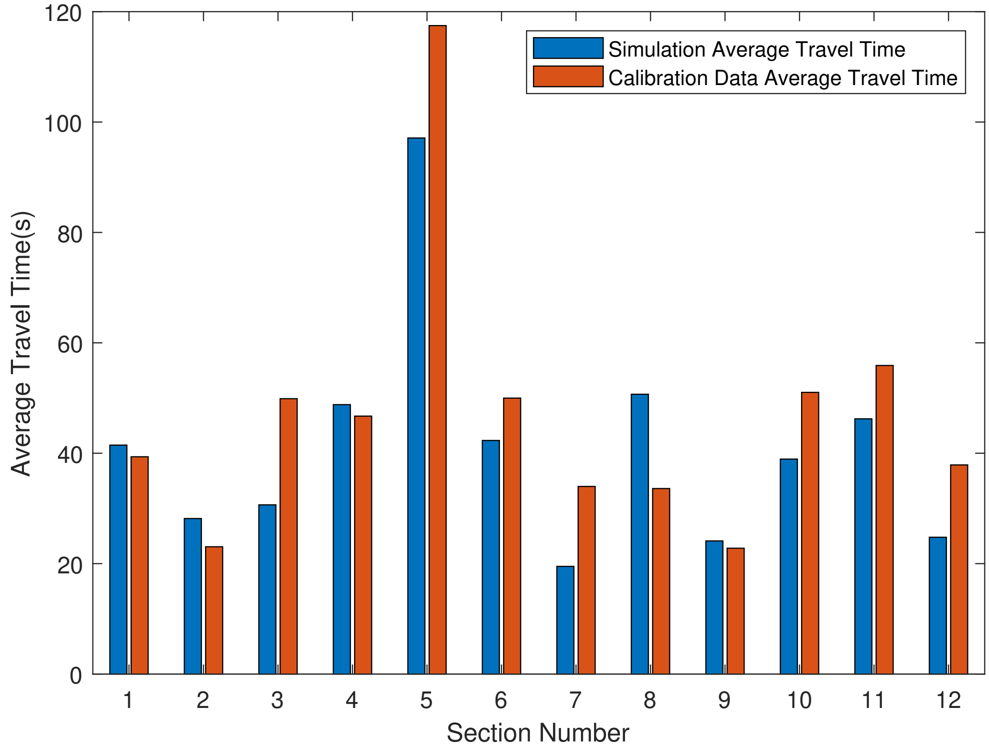


Figure 4.13: Average Travel Time: Simulation vs Data

determine the traffic state. The data required will improve the calibration by removing the constant stop time and constant vehicle flow.If vehicle count data can be gathered from the camera sensors located on each traffic light, 3.11 then the objective function can be improved by adding vehicle count and travel time.

4.6 Chapter Summary

This chapter summarized the work on calibration algorithm. It can be observed from the results that travel time can be used as a state to determine traffic in a network. Due to inconsistency in GPS data-set the calibration is not very accurate but if GPS logging frequency can be increased the travel time algorithm will work more

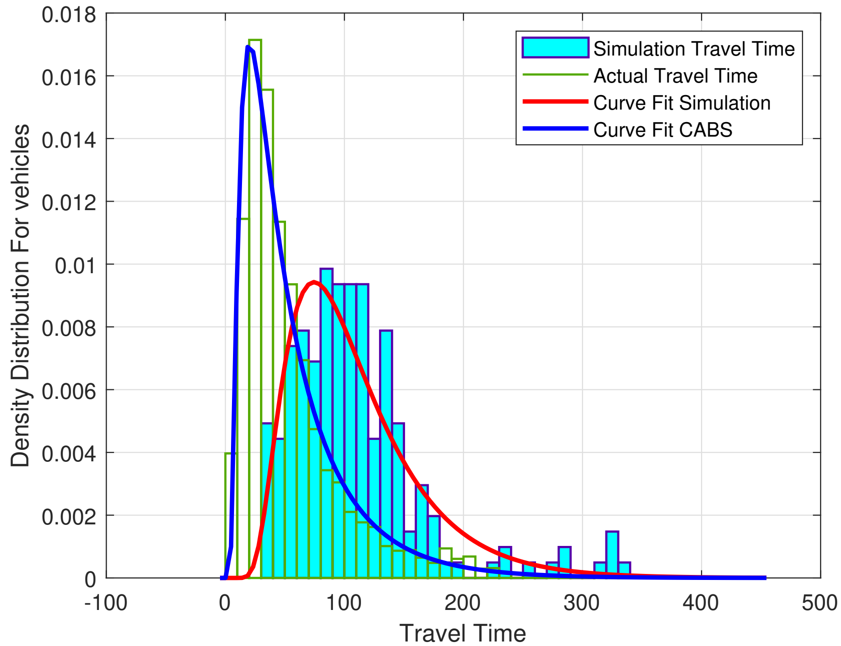


Figure 4.14: Simulation vs Actual Travel Time(Section 5)

efficiently to remove uncertainty in calculation of CABS position. Also pedestrian count data needs to be captured on OSU campus intersections so that the constant flow rate assumptions at crossings (for pedestrian counts) can be removed.

Chapter 5: Multi-Agent Reinforcement Learning for Traffic Signal Control

5.1 Traffic Signal Control Systems

The function of a traffic signal control system is to safely and efficiently manage traffic flows at intersections via alternating the green, yellow, and red lights. The traffic signal control problem is usually solved on two different levels: local and arterial.

Local traffic signal control only considers the local traffic conditions at an isolated intersection. In arterial control, several traffic light controllers coordinate to manage the traffic flow. In arterial control, the common goal is to generate the "green wave" scenario. A series of traffic lights are synchronized to allow a platoon of vehicles to move through several intersections. The fig 5.1 shows the impact if coordination is not done properly. The vehicles arrive at the green phase for Intersection A, but they face red light for consecutive B & C. In a perfect scenario, all the vehicles should face green light at three (A, B,&C) intersections. In Woody Hayes Drive, traffic light control works in an arterial control manner to provide green waving for traffic movement from East to West and West to East. The control logic for each traffic light is predetermined(fixed phase timings) and doesn't change with the dynamic traffic conditions. As in arterial control, the signal phasing are predetermined in this chapter.

An intelligent agent-based control strategy is introduced, which can adapt the signal timings by learning the traffic conditions in its vicinity. The fig 5.2 shows the various control strategy for traffic signal systems. In the OSU campus, the phasing approach is “Group-Based," and the signal timings are either “Fixed" or “Vehicle Actuated." The focus of this thesis is to adopt a signal timing strategy that is adaptive to traffic response.

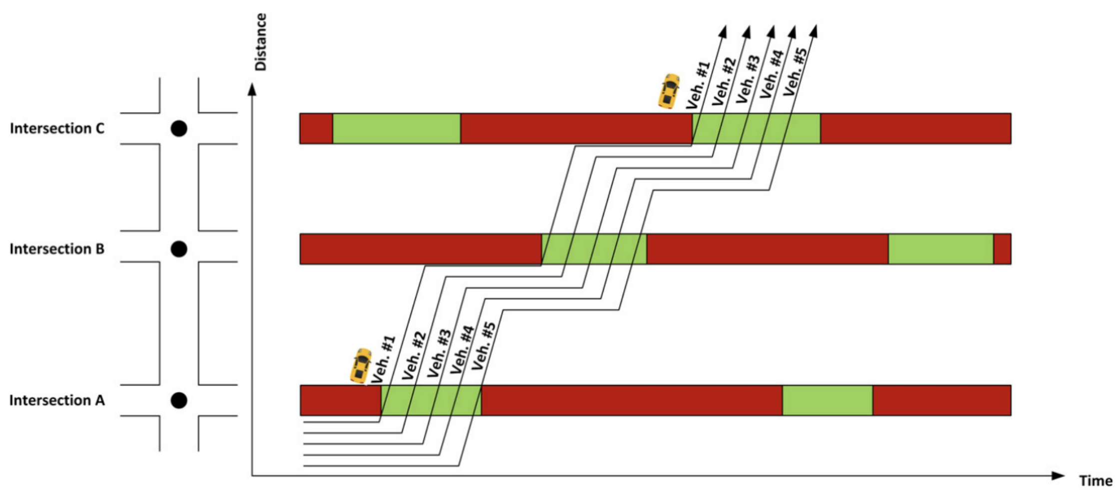


Figure 5.1: Ineffective Static Coordinated Green Phases for Green Wave [40]

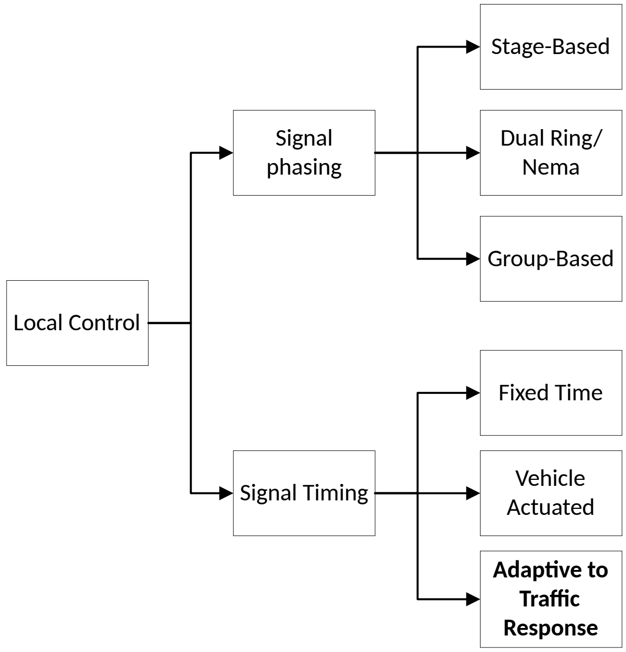


Figure 5.2: A Summary of Signal Control Structure [40]

5.2 Problem Formulation

Traffic signal optimization is a stochastic and continuous nonlinear constrained problem. Simulation-based optimization is the best candidate for solving this problem since in the real world, the changes in the traffic controller policy may cause safety concerns.

Simulation based optimisation problem can be formulated as:

$$\min_{x} f(x, v) = \mathbb{E}[F(x, v)]$$

$$s.t.\ c_1(x, v, u) = 0$$

$$c_2(x, v, u) \leq 0$$

The expected value of a predefined network performance measure F is the objective function f. F depends on the deterministic decision variables x (traffic controller phase timings) and exogenous parameters v. The feasible space is defined by c_1 and c_2 , two sets of deterministic and analytical constraints.

For example, a cumulative stop time of vehicles in traffic can be an objective function F for a signal optimization problem. The decision variables are the green duration for the signal phases. Network topology and the traffic demand are captured in v. The traffic demand is based on probability of flow from each entry to network which affect the traffic counts during traffic simulation run. The flow probability is the parameter which creates variability in the traffic pattern. For microscopic traffic simulation , u defines the car-following model and lane change model parameters.

Calibrated microscopic traffic simulation can act as a Software in Loop(SIL) for the traffic signal controller strategy development. The fig 5.3 shows the general framework of **SIL** for traffic control problems.

5.3 Intelligent local traffic signal control

A traffic signal controller can be designed as an agent-based framework. In an agent-based framework, each signal controller can be modeled as an intelligent agent which learns from its environment and outcomes of its past actions.

Figure 5.5 shows the process of decision-making by the intelligent agent. The agent can receive the state from the external traffic environment at every decision step by two methods. The first method involves direct measurement based on sensors deployed in a traffic surveillance system. The second method is to use the state estimation approaches to determine traffic states from the observations. The first

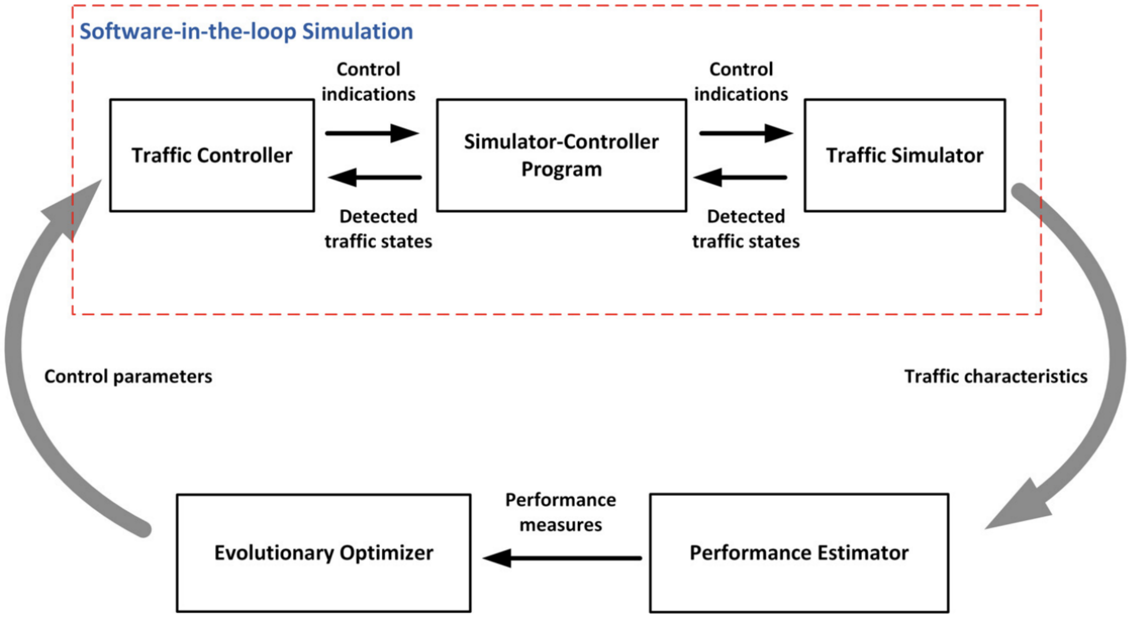


Figure 5.3: Optimization Framework for Traffic Signal Control [40]

method is dependent on a lot of data collection(e.g., traffic counts), while the second method can be performed with fewer data(e.g., lane queue in a specific area). After the agent has received the traffic state, it acts and receives a reward as feedback from the environment. The agent is assumed to follow the Markov decision process(MDP). The MDP assumption is that the future state depends only on the current state and control instead of a sequence of events that are precursors to the state-control pair.

Markov Decision Process

The traffic light agent process is MDP. The components of MDP are summarized in fig 5.4. MDP is defined by a state-set $\mathcal{X}$, action set $\mathcal{U}$, one step dynamics, and transition from state action (x, u) to the next state x', and reward (Puterman, 2014). The transition dynamics $T(x'|x, u)$ and reward $r(x, u)$ are given by

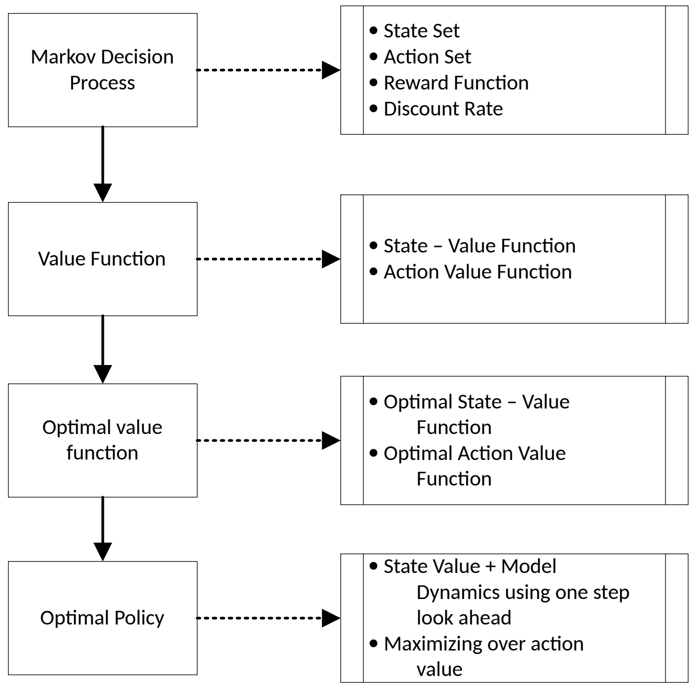


Figure 5.4: A flowchart for the Markov Decision Process

$$T(x'|x,u) = \mathbb{P}[X_{t+1} = x'|X_t = x, U_t = u] \tag{5.1}$$

$$r(x,u) = \mathbb{E}[R_{t+1}|X_t = x, U_t = u] \tag{5.2}$$

The agent's knowledge is represented by either state-based $V(x)$ or action-based Q-function ($Q(x,u)$). These terminology is derived from Bellman equations. Bellman equations for state-based and action-based value functions are:

$$V_\pi(x) = \mathbb{E}_\pi[R_{t+1} + \gamma V_\pi(X_{t+1})|X_t = x] \tag{5.3}$$

$$Q_\pi(x,u) = \mathbb{E}_\pi[R_{t+1} + \gamma Q_\pi(X_{t+1}, U_{t+1})|X_t = x, U_t = u] \tag{5.4}$$

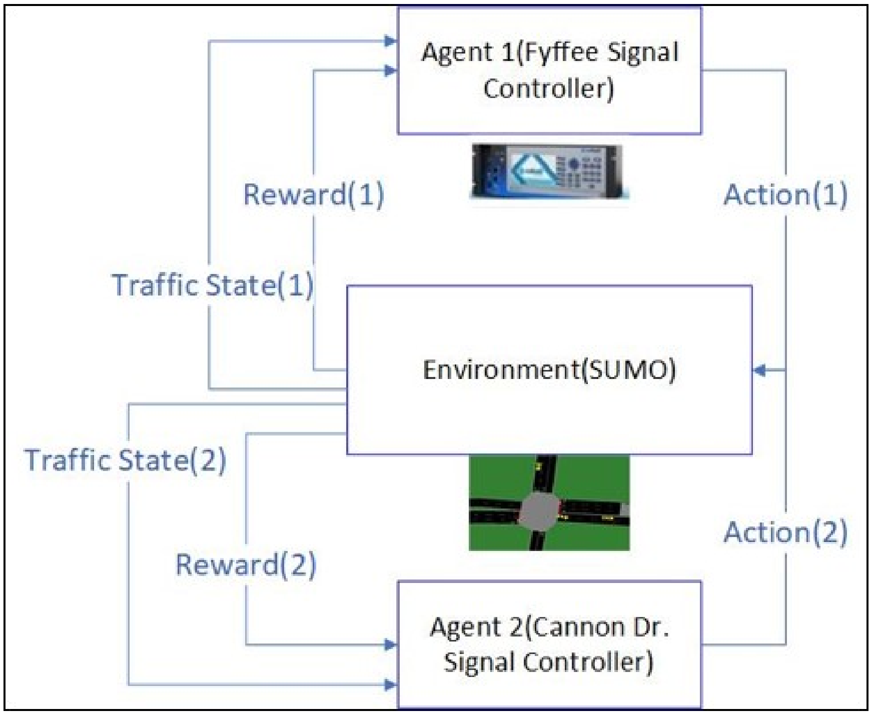


Figure 5.5: Description of agent-environment interaction

The optimum policy is decided by the value functions. In this thesis action-based Q-function is used as the value function for finding the optimum policy. A policy π_1 is better than π_2 if $Q_{\pi_2}(x,u) \leq Q_{\pi_1}(x,u)$ for the state-action pair (x,u). For MDP there always exist an optimal policy denoted by π_* with the corresponding value function as $Q_*(x,u)$. The optimal policy for action-based value function is found by below equation:

$$\pi_*(\mathbf{u} \mid \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{u} = \arg\max_{\mathbf{u} \in U} Q_*(\mathbf{x}, \mathbf{u}) \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

5.3.1 Traffic Light Agent Framework

The fig 5.7 shows the intelligent controller implementation. The agent updates its action and receives a reward every 100 seconds from the environment(SUMO traffic simulation). The MDP parameter for the agents are defined below:

- **States**: The states for the traffic agent are the green phase timing, individual lane density, and individual queue length (vehicle count in the lane). We can take the example of junction 1 (Fyffe) with three grouped green phases and serve nine lanes in all directions combined (fig 5.6). The states are denoted by $X \in (PhaseTimings, Q_{lanes}(k), Q_{density}(k))$, where k is the lane.

- **Control Action**: The action for the controller is the next cycle green phase duration for the grouped phases.The action taken by the agent is every 100 seconds. The controller doesn't interrupt the ongoing traffic phase cycle. Once the last cycle is complete new actions are applied. The action space is denoted by:

  $$\mathcal{U} = (G_t : where\, G_t = 15 + 5m \, and \, m \in (1, 10) \; s.t. \; m \in \mathbf{N})$$

  Thus the bounds for actions are (15,75) seconds.

Two junctions are selected for this study on intelligent traffic light control (5.8). The subsequent sections will introduce the MDP solution for the RL problem.

Learning Algorithm

Temporal difference (TD) is used to solve the MDP problem for traffic light control. TD implements bootstrapping to make updates to Q-function, and its updates are:

$$Q\left(\mathbf{x}_t, \mathbf{u}_t\right) \leftarrow Q_t\left(\mathbf{x}_t, \mathbf{u}_t\right) + \alpha \delta_t\left(\mathbf{x}_t, \mathbf{u}_t\right) \tag{5.6}$$

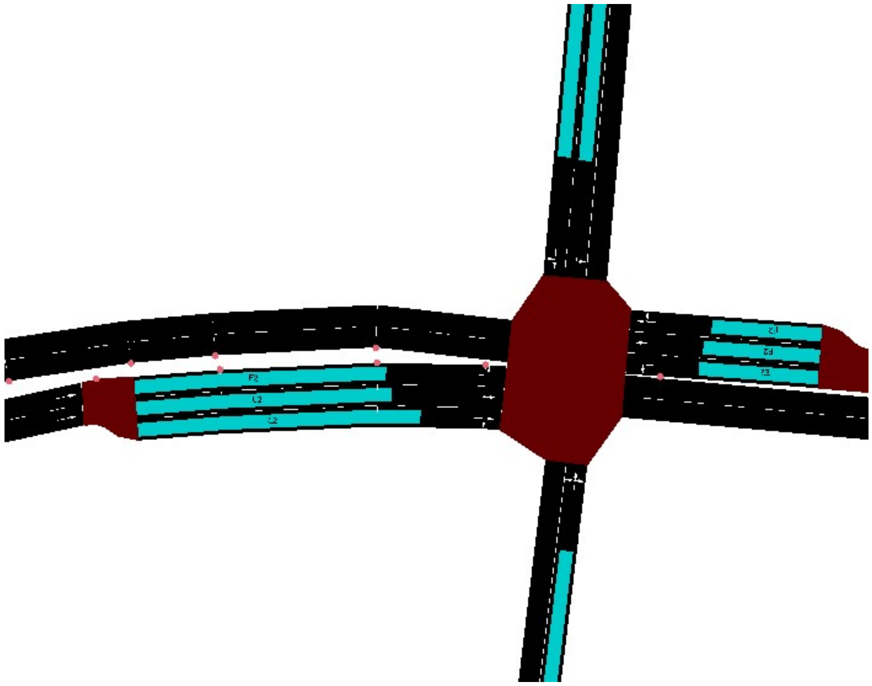


Figure 5.6: Fyffe Junction in SUMO Network(lanes highlighted)

where $\alpha \in [0,1]$ is the learning rate and δ_t is TD error.

TD error is the difference between TD estimation and the corresponding action-value. The calculation method for TD used in this work is on-policy **State Action Reward Next State Next Action**(SARSA). For SARSA the TD error is defined by below update:

$$\delta\left(\mathbf{x}_t, \mathbf{u}_t\right) = r_{t+1} + \gamma Q_t\left(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}\right) - Q_t\left(\mathbf{x}_t, \mathbf{u}_t\right) \tag{5.7}$$

TD algorithms use tabular format for storing the Q-function value $Q(x, u)$. This represents scalability issues for large environments with multiple traffic lights. The number of state-action pairs increases in the order of magnitudes requiring a lot of memory to store the Q-function.Due to memory constraints, only two traffic lights are considered in this work.

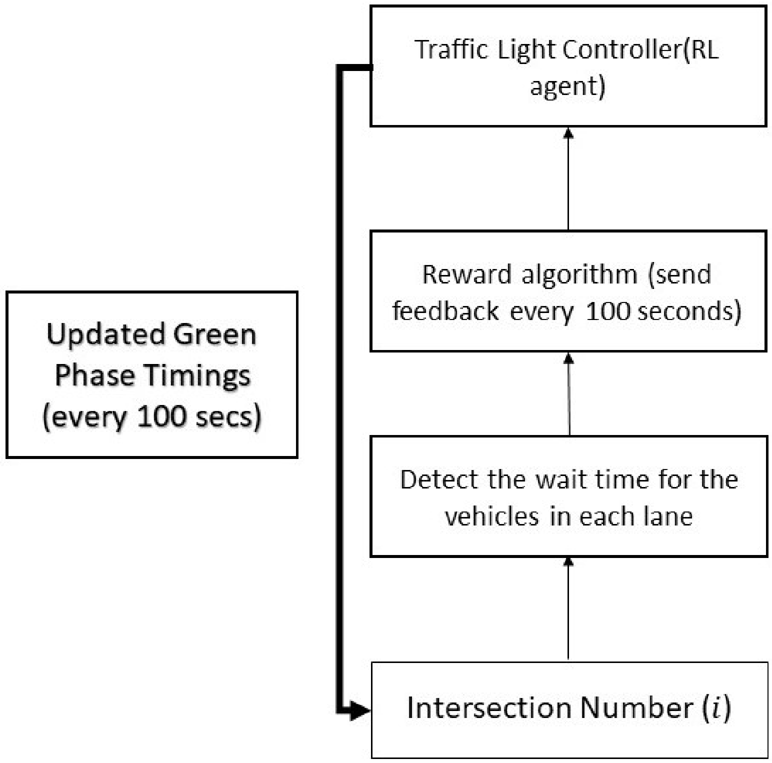


Figure 5.7: Agent Framework For Traffic Light

Agent Policy

The action in the $\mathbf{u}_{i,t+1}$ is determined by the decaying epsilon greedy policy. In this policy a random action is selected with $\epsilon_{i,t}$(exploration parameter) probability while a greedy action is selected with $1 - \epsilon_{i,t}$ probability.

$$\mathbf{u}_{i,t+1} = \begin{cases} \text{random } \left(\mathbf{u}_i\right), & \text{if } \xi < \epsilon_{i,t}; \\ \arg\max_{\mathbf{u}_i} Q_{i,t}\left(\mathbf{x}_{i,t+1}, \mathbf{u}_i\right), & \text{otherwise} \end{cases} \quad \mathbf{u}_i \in \mathcal{U}_i \tag{5.8}$$

The ϵ used in this work is decaying with each step. The following equation shows the decays nature of exploration parameter:

$$\epsilon_{i,t} = \max[\lambda^{\text{step}}\epsilon_{\text{o}}, \epsilon_f] \tag{5.9}$$

Figure 5.8: Two Intersections for Intelligent Agent

$\epsilon_{i,t}$: Exploration parameter for i^{th} junction at time t

λ : 0.999, decaying parameter

$step$: step number for actions by controller; $t/100$

ϵ_o : 0.05, initial epsilon for the exploration

ϵ_f : 0.005 minimum epsilon value for exploration

Reward Function

The reward function is the feedback from the traffic environment after the traffic light agent takes an action. For traffic, the light agent receives a reward from the environment every 100s. The below equation denotes the reward function.

$$r_{i,t} = \frac{T_{\text{wait,step-1}} - T_{\text{wait,step}}}{100} \tag{5.10}$$

where :

$$T_{wait,step-1} : \text{ Sum of wait time in all lanes for } (step-1) \text{ 100 seconds}$$

$$T_{wait,step} : \text{ Sum of wait time in all lanes for previous 100 seconds}$$

This reward function gives the agent a positive reward if the action was taken by it(green phase timings) reduces the total wait time(wait time is determined by vehicles in a standstill on lanes served by traffic signal).By implementing this reward function, the agent must take optimum actions to reduce the wait time for vehicles.

5.4 Results

The fig 5.9 and 5.10 shows the results of training the two RL agents in the SUMO traffic simulation environment. The x-axis shows the simulation time in hours, and the y-axis is the sum of vehicle stop time for every 100s in minutes. It can be observed that till the **Equill. Line** the agent is exploring different actions, but after this line, the actions are optimized to reduce the vehicle stop time.The two agents are independent and are not sharing traffic state information to each other. This type of RL is called independent agent based framework.The fig 5.11 is the distribution of sum of wait time in all lanes for two junctions. The mean for RL agent is 56 seconds and for static control is 172 seconds. The RL agent traffic control is much better in reducing the traffic congestion.

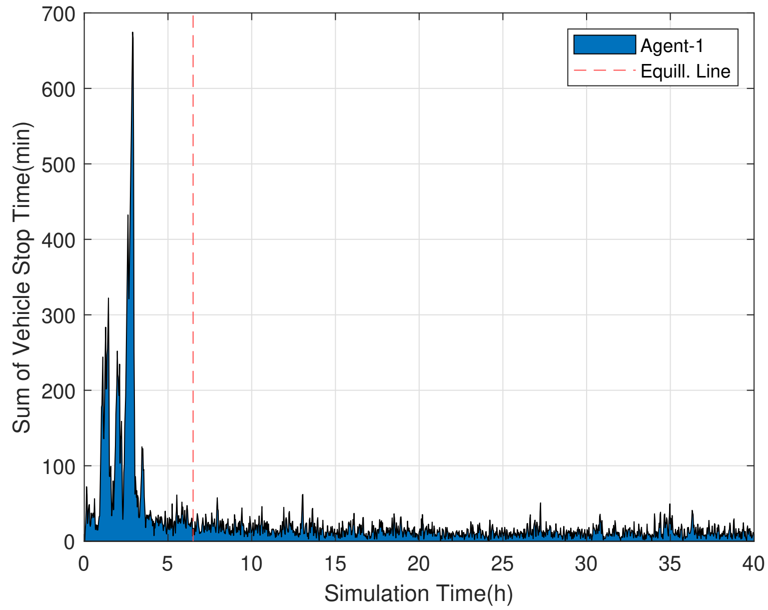


Figure 5.9: Sum of wait time for vehicles Agent 1

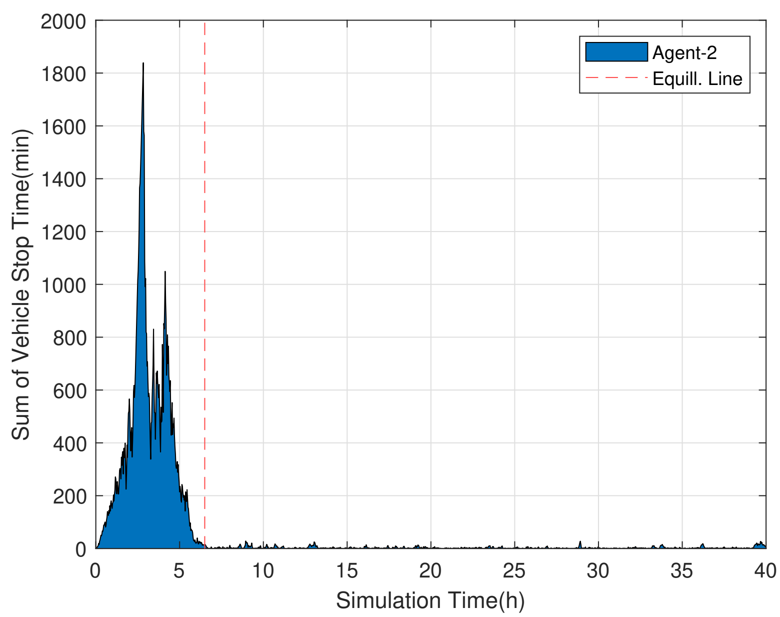


Figure 5.10: Sum of wait time for vehicles Agent 2

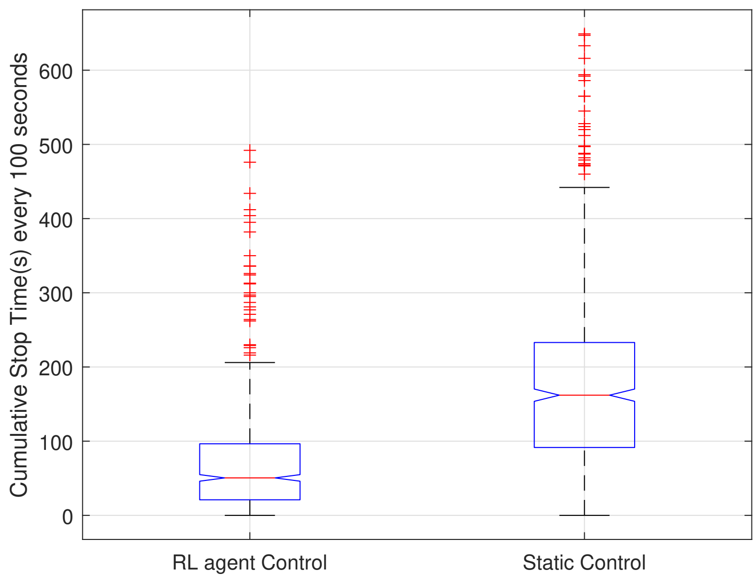


Figure 5.11: Comparison of sum of wait time for two controller policy

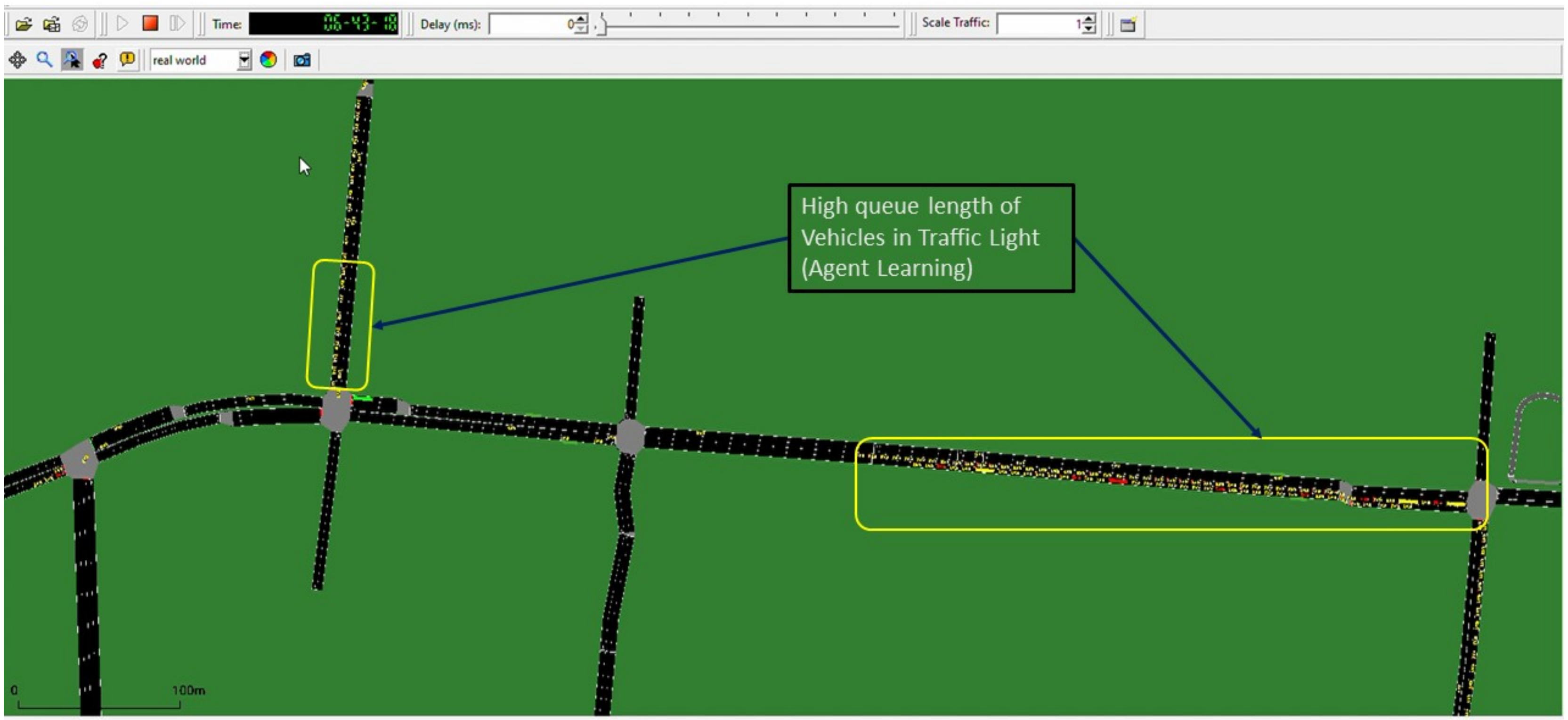


Figure 5.12: Initial learning phase for RL agent

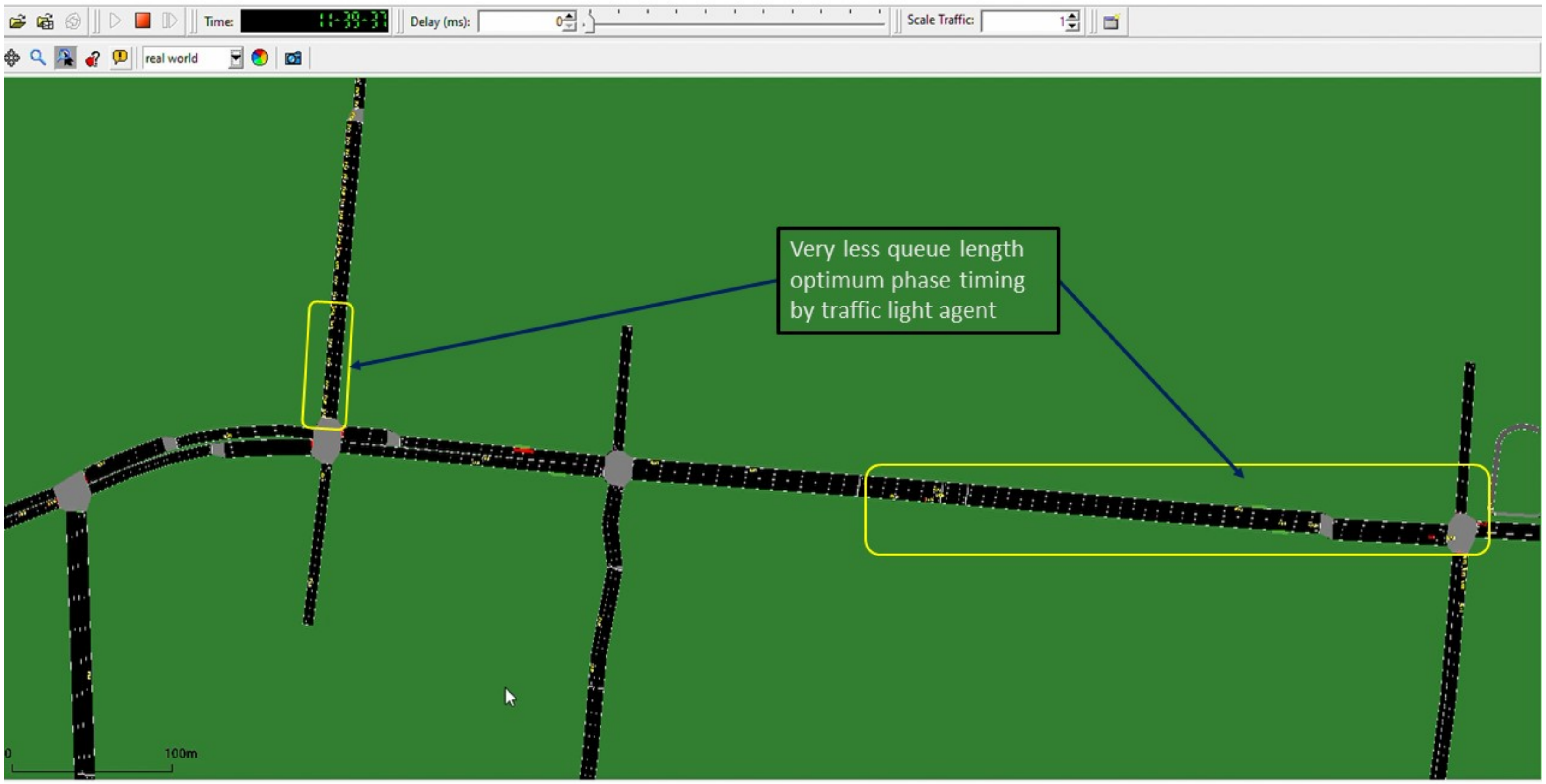


Figure 5.13: The optimized policy used by RL agent

5.4.1 Discussion on RL Controller

In this study the updated green timings for a traffic signals is updated every 100 seconds. The update time for the controller is varied from 5 seconds to 1000 seconds. The aim of higher update time is to achieve multiple traffic signal cycles in between consecutive control updates. During this multiple traffic signal cycles the signal timings are not changed. The benefit of higher update time is multiple cycles can help in achieving in isolation of control design decision for a particular choice of green phase timings from the last controller decision. The challenges with higher update time is agent needs much more time to learn in the environment and the second challenge is the high traffic flow gets stuck in tuttle park intersection(section 5) if the agent takes an action which is not optimum during the learning phase. In further extension of this work the traffic flow jam can problem can be alleviated by using smaller time step for SUMO solver(e.g 0.1s)

5.5 Chapter Summary

This chapter introduced the concept of MDP for traffic light controller. An intelligent Reinforcement agent based controller strategy was adopted to improve the traffic flow across two intersections. The comparison between RL agent and the static controller depicted a significant improvement in reducing the traffic congestion. The simulation approach can be extended to the more intersection and can also be implemented in real time online learning agents.

Chapter 6: Traffic Co-Simulation with Real-Time Hardware in Loop Setup

This chapter introduces an integration framework for the dSpace Automotive Simulation Model (ASM) with the traffic simulator (SUMO). This framework aims to link high-fidelity vehicle dynamics, powertrain, and autonomous vehicle systems models with low-fidelity traffic models. This framework works in Real-Time using the Hardware in Loop(HIL) setup (SCALEXIO) provided by dSpace.

Engineers and researchers use real-time simulation to develop, test, and validate many areas of vehicle testing, for example, vehicle dynamics, chassis control systems, powertrain control strategies. The recent research in the area of implementing Artificial Intelligence (AI) based methods for connected and autonomous vehicles(CAVs) controls also needs a lot of testing. Testing CAVs can be performed either in the real world with all the necessary hardware or parts of the environments could be simulated using virtual representation. The simulation-based approach is a much faster, safer and efficient alternative compared to actual road testing. A virtual simulation environment can also be used for human drivers training using the driver in the loop with the simulation environment. The simulation for emulating the real environment involves a critical component of replicating the real traffic conditions in terms of network and nearby vehicles to ego vehicles (the vehicle on which different algorithms are being tested).

A framework that integrates real-time HIL and a large scale traffic simulator need the following three components:

- High-Fidelity Simulation for ego vehicle dynamics compared to traffic simulators.
- A communication interface between the above two components exchanges the information of ego vehicle and fellow vehicles between two simulators in real-time.

In this chapter, all the above-mentioned components will be discussed. The primary focus of this thesis is to develop a real-time communication interface between two simulators.

6.1 Ego Model Simulation Environment

The ego for ADAS and powertrain is developed in Simulink. The fig 6.1 shows the basic blocks in the Simulink model. The dSPACE IO block communicates with the HIL bench (SCALEXIO).

6.2 Ego Vehicle Interface(EVI)

The real-time execution of the simulation also adds a layer of complexity to the simulation as the SCALEXIO hardware has enough computation power to compute the high fidelity model for ego vehicle in real-time but extracting information from SUMO for traffic is not optimized by the TRACI API. The EVI block decides the region of interest(ROI) in the vicinity of the ego vehicle, and then the nearest ten vehicles from traffic are chosen as fellow vehicles. Fellow vehicle speed, position, and

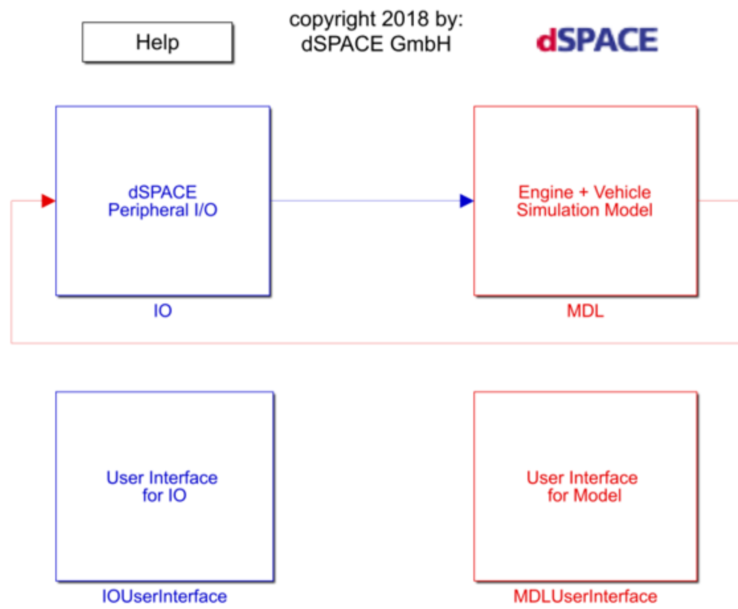


Figure 6.1: Automotive Simulation Models
Vehicle Dynamics with Traffic

pose(yaw angle) are transferred through EVI to the HIL bench in real-time. fig 6.4 shows steps 1-5 for order and information flow between the three components.

The integrator reset(6.2) updates every 200 ms and exchanges information about EGO and traffic between SUMO and ASM.As the update rate of ASM is 1ms which is very fast compared to SUMO 200 ms. There is an extrapolation algorithm in the EVI which predicts the future position of fellow traffic vehicles till the next SUMO step is available for updated fellow positions. Thus there is an “extrapolation algorithm" in the “SUMO position block," which finds the (x,y) coordinates for every 1ms step. This extrapolation algorithm performance is discussed in the results section.

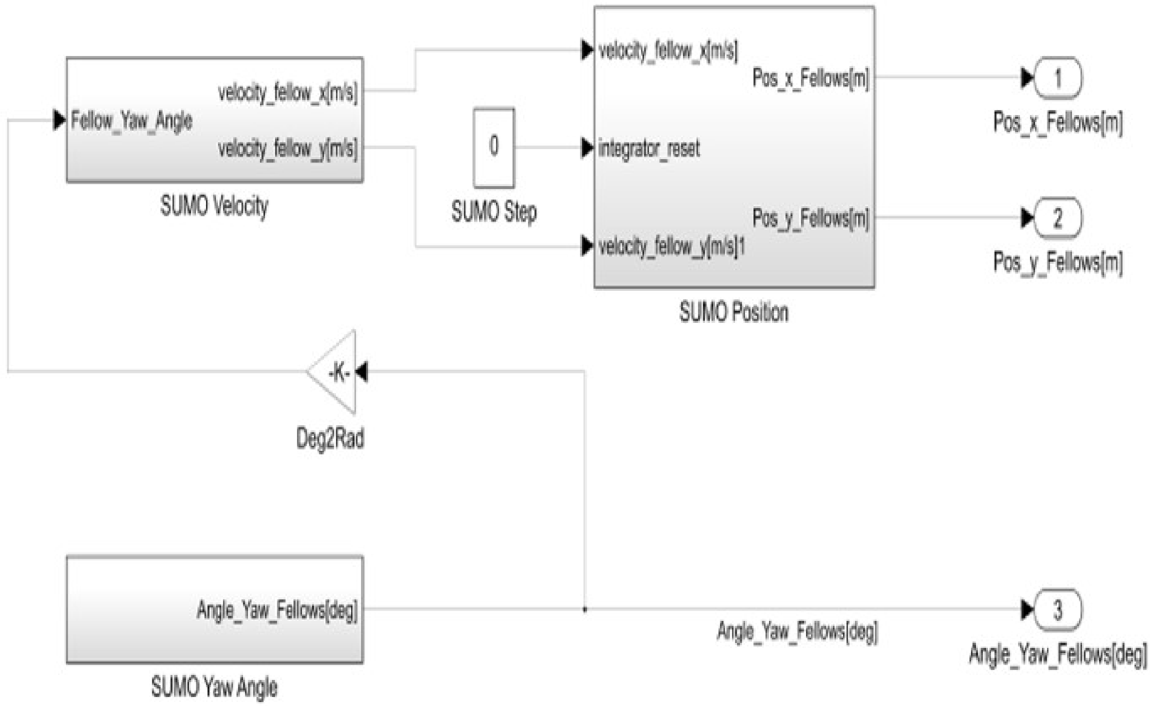


Figure 6.2: Simulink Blocks for Information Exchange

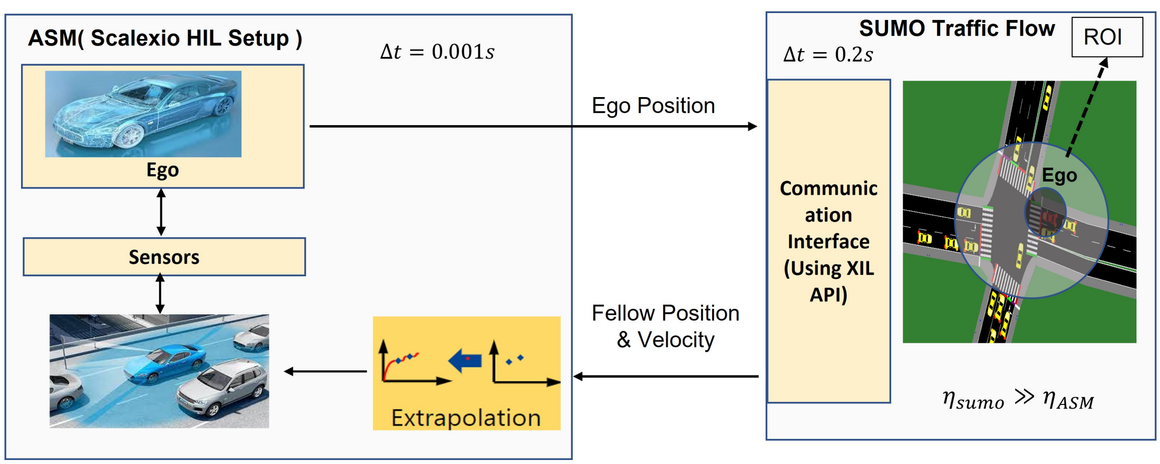


Figure 6.3: Co-Simulation Framework

6.2.1 XIL API

XIL [41] is an API standard for communication between test automation and test benches. The standard is developed by "Association for Standardization of Automation

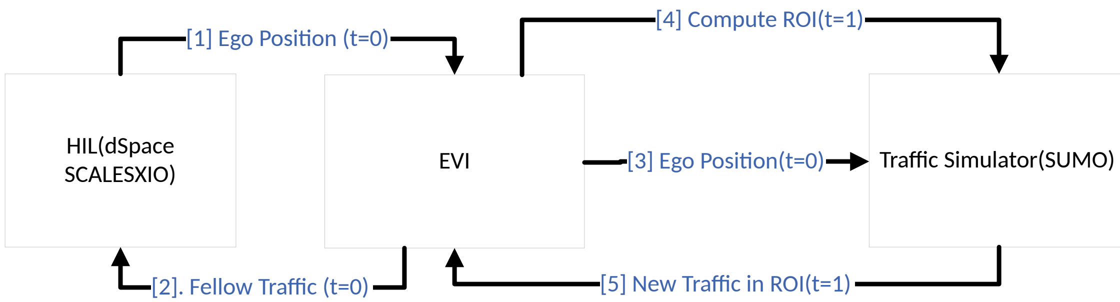


Figure 6.4: Order for information flow between three components: Steps Repeat every 200ms

and Measuring Systems" (ASAM). It was launched in June 2009 . The standard supports all stages of testing and development like model-in-the-loop (MIL), software-in-loop (SIL), and hardware-in-the-loop (HIL). This standard can be used for all “in-the-loop" systems and is thus called “XIL." The benefit of this standard is it can be used with any HIL bench irrespective of supplier. This benefits in migrating the hardware to different HIL bench.

The XIL API supports the following features:

- Access to the metadata of complex variables.
- Capturing support of complex data types (vector and matrix).
- Signal Generation support for complex data types.
- Stimulate variables with raw or physical values.
- Pausing the simulation.
- Read and write data from a testbench variable as raw or physical value.
- Discard fetched data.

- Specify the XIL version of the Testbench / Framework implementation to be instantiated.

XIL API has multiple ports, as shown in fig 6.5. In this thesis, the “MAPort" has been used with Traci in a python script for the EVI interface.

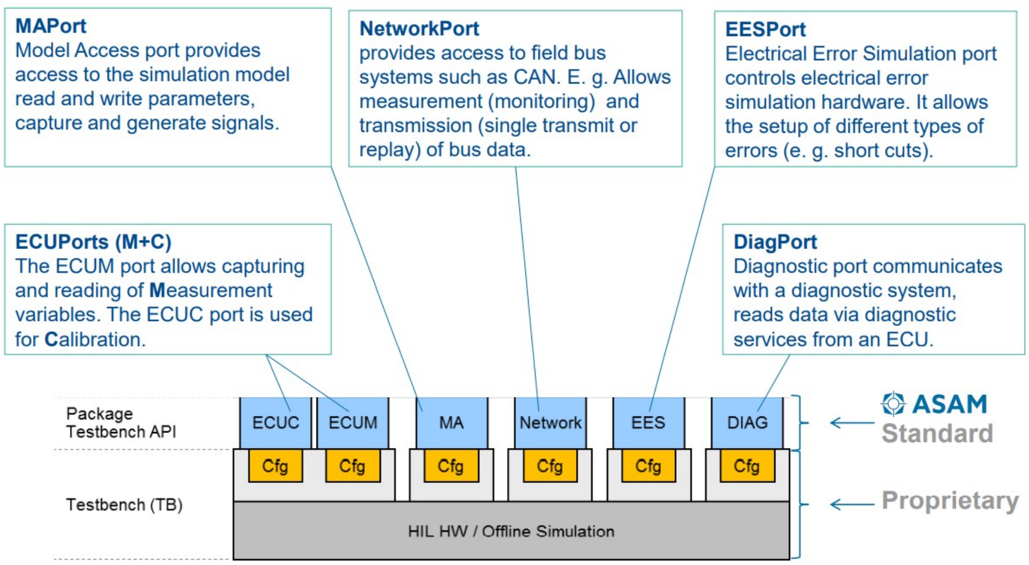


Figure 6.5: Concept of Ports in XIL[42]

6.2.2 TraCI

TraCI is a TCP-based client/server to provide access to SUMO. TraCI is called the “Traffic Control Interface". The benefit of this interface is it can communicate with SUMO in online mode and start/stop simulation or extract/write information to the SUMO model. The API commands used in this work are listed below:

Table 6.1: TraCI commands for EVI

TraCI Command	Information
traci.start()	Start the SUMO simulation
traci.simulationStep()	Progress simulation by one time step
traci.simulation.getTime()	Get current simulation clock time from SUMO
traci.vehicle.getIDList()	IDs for all vehicle in SUMO scenario
traci.vehicle.getNextTLS('ego')	Get State Information for next traffic light in front of Ego
traci.vehicle.moveToXY()	Move Ego Vehicle to position specified by ASM
traci.vehicle.getPosition()	Get XY position for the fellow traffic in SUMO network
traci.vehicle.getSpeed()	Get longitudinal speed in m/s
traci.vehicle.getLateralSpeed()	Get lateral speed in m/s
traci.vehicle.getAngle()	Get Yaw angle in degrees for fellow vehicles
traci.close()	Close the SUMO simulation

Scenario Demo

The demo scenario used for co-simulation purpose is the calibrated sumo network (Woody Hayes Drive) from 4. The Ego vehicle is inserted 600 seconds after the SUMO simulation has started. During first 600 seconds SUMO is not in synchronisation and computes the traffic in network till 600 seconds in approximately 10 seconds. Once the insertion time(600s) has reached the ASM starts and then the EVI controls the SUMO steps. After insertion time Ego vehicle starts its route from West to East in Woody Hayes Drive. The route is predetermined in the SUMO and ASM simulators. The ego vehicle dynamics are controlled by ASM and the traffic vehicles are controlled by SUMO. The task for EVI is to exchange information every 200ms for the fellow vehicles and ego vehicle in ASM and sumo respectively. The benefit of insertion time is that user can define any insertion time and the network will get pre-populated with traffic before the start of ego vehicle route. This can help in generating different traffic conditions for the same route.

Results

This subsection shows the results of synchronization between SUMO and ASM in real-time execution. At every step start, the clock information from ASM is extracted, which is then used to proceed with the SUMO simulation step. SUMO calculates the position for fellow vehicles and caches the result to EVI within 150ms. The information for fellow traffic vehicles in ROI is then passed to ASM at 180ms. This step is repeated till the end of the scenario for the simulation. (refer fig 6.6). The fig 6.8 shows the distribution of response time in milliseconds for the EVI and the SUMO simulation step. This plot shows that the meantime is less than 100ms, but as the outlier is around 150ms, the simulation δT for SUMO is 200ms. If the outliers can be removed, the δT for SUMO can be reduced to 100ms. The fig 6.8 shows the distribution of time to write EVI data to the HIL bench in ASM. The mean time is around 2ms, but outliers are in the range of 20ms. This is the reason caching is started at 180 ms(20ms before the sumo step time). This can create a small offset error and can be improved if outliers are removed.

The fig 6.7 is the plot for (x,y) coordinates of a fellow vehicle in ROI. The extrapolation algorithm and SUMO output at every synchronisation step(in this plot sumo is running at $\delta T = 100ms$) is exchanging information.The error in the position predicted by extrapolation and SUMO is maximum 0.3 metres at urban speeds

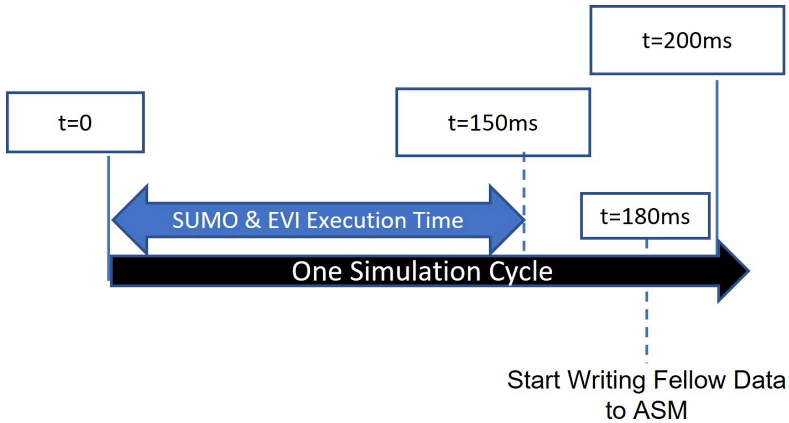


Figure 6.6: One Cycle of Simulation Synchronisation: Repeats every 200ms

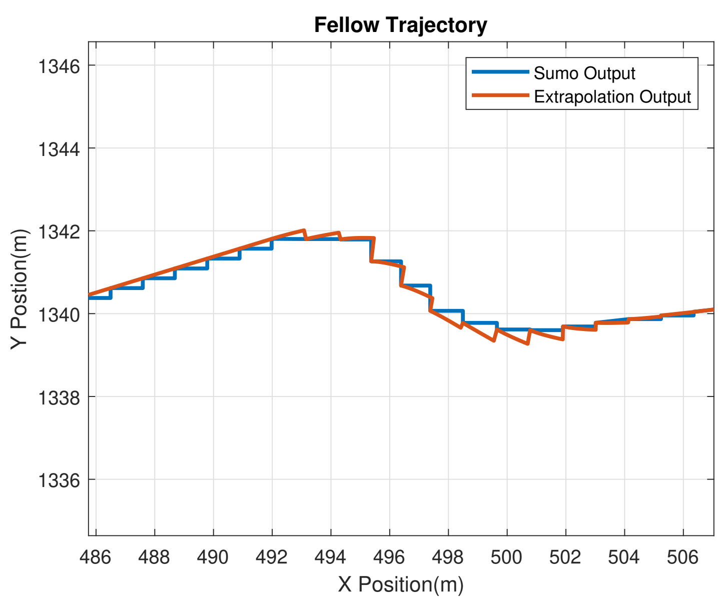


Figure 6.7: Sumo & Extrapolation Algorithm Fellow Trajectory

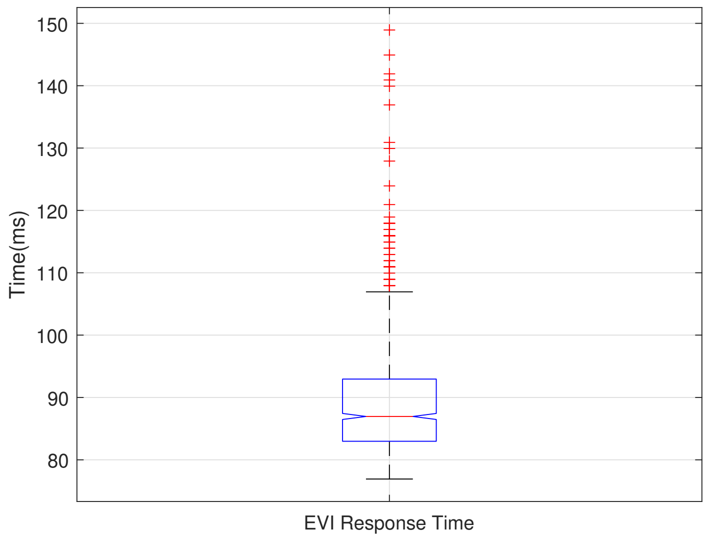


Figure 6.8: EVI execution response time

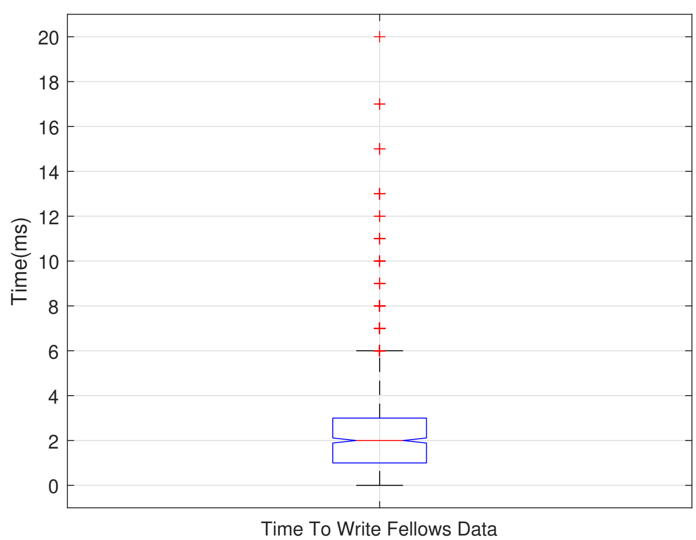


Figure 6.9: Caching Data to HIL Execution Time

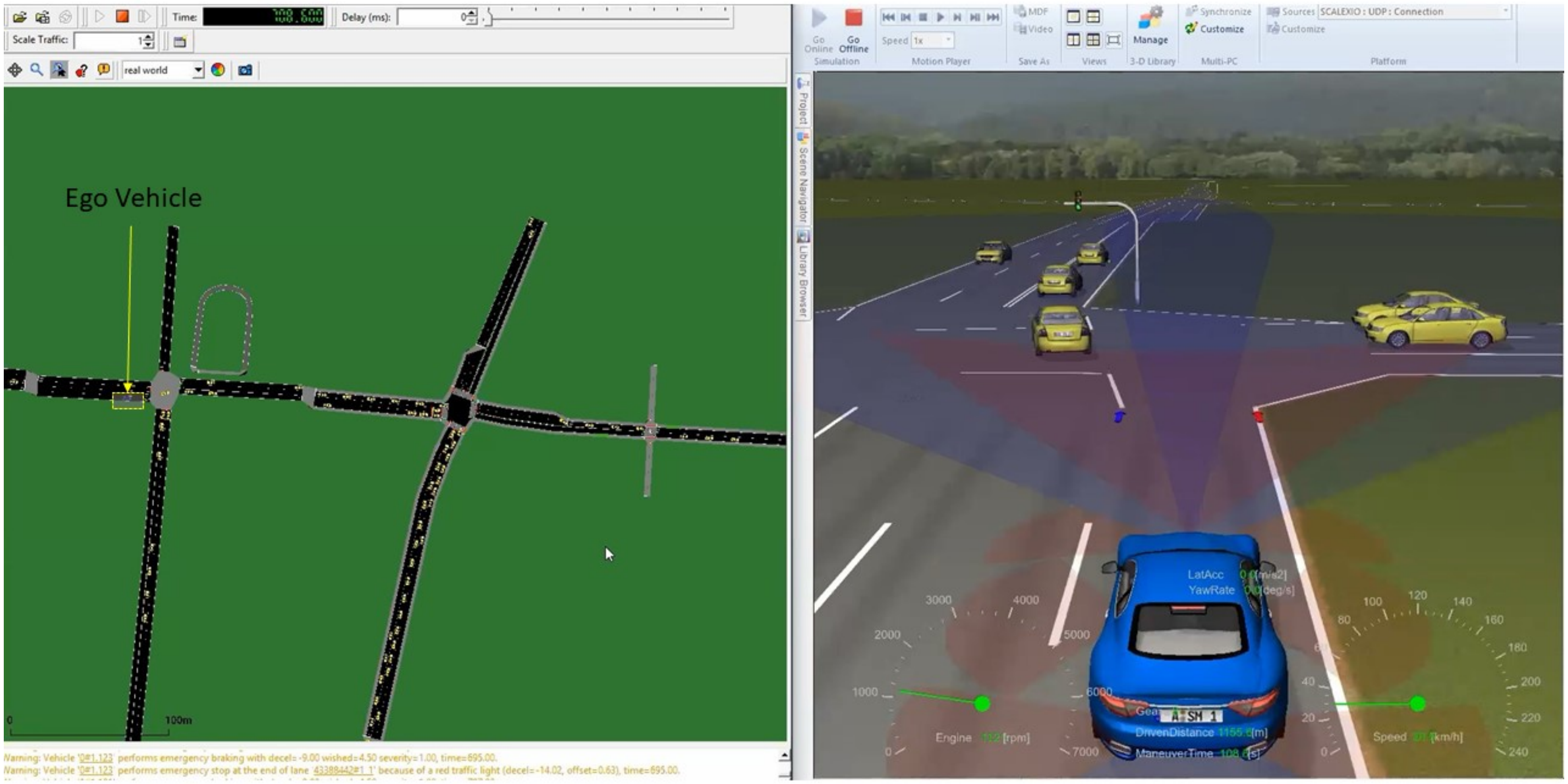


Figure 6.10: Co-Simulation Snapshot: ASM & SUMO

6.2.3 Chapter Summary

In this chapter, a framework was established using the dSpace ASM and SUMO traffic simulator. The results of the extrapolation and synchronization algorithm(EVI) show that this framework can be used to test powertrain and ADAS functionality in an urban environment with high traffic conditions. The extrapolation algorithm in EVI handles the data exchanging between two simulators efficiently with minimum error in the future state prediction for traffic vehicles.

Chapter 7: Conclusion & Future Research

7.1 Conclusions

Microscopic traffic simulations and their calibrations present a promising means to help the development of intelligent mobility systems. Using existing data sources to calibrate the traffic simulations can help understand the challenges in solving this complex problem of nonlinear traffic dynamics. This thesis examined the use of existing data sources in the OSU campus to build a framework for microscopic traffic simulation in one section(Woody Hayes Drive). The final chapter concludes this thesis and summarizes some possible future research directions.

In chapter 4, the optimization framework was introduced, which can be used to calibrate microscopic traffic simulation using GPS data from the vehicles. From GPS data, traffic state variables make travel time extracted and used as an objective criterion for simulation calibration. The CABS count was less than 20% in the "Woody Hayes network" thus, the approach was to use multiple days data-set to capture many travel routes in the network. The challenges in terms of GPS logging inconsistency introduced errors in the calibration process which can be improved by having better quality GPS logging. The objective function can be enhanced if the count data for vehicles are available as it removes the uncertainty for vehicle flow in the network.

This can be done with the current infrastructure on the campus using a video camera mounted on traffic signals.

In chapter 5, a smart RL agent-based traffic light optimization framework was explored. The framework is governed by an evolutionary optimizer (RL) and SUMO traffic simulator. An arterial traffic light control with two junctions was considered. Comparing the RL agent traffic controller with the static controller showed scope for improvement in traffic congestion.

Chapter 6 extended the idea of linking traffic simulator with hardware in loop simulator DSpace ASM. An ego vehicle can be modeled with high-fidelity vehicle and powertrain models compared to the SUMO vehicle dynamics model. The communication platform can be used to test the various algorithm in the field of energy management and ADAS for vehicle testing.

7.2 Future Work

There can be some extensions to work presented in this thesis. The future work is divided into specific problems tackled in this thesis.

Traffic Calibration

The work in the field of traffic calibration presented in this thesis was offline calibration. This calibration can be extended for online cases in which the calibration is adapting to real-time traffic information. For real-time traffic information, a framework can be developed which seeds real-time data to traffic simulation. The benefit of this framework would be to predict the traffic for the nearby horizon. Prediction of traffic can help in mitigating traffic congestion or using to optimize the energy management strategy for connected vehicles.

Intelligent Signal Control

The Multi-agent RL traffic signal control introduced in this thesis was independent control. The two traffic agents were not sharing knowledge about traffic state in their vicinity to each other. A study can be performed in which some traffic states can be shared between the agent. For traffic control, dynamic lane assignment or dynamic speed limit study can be performed. The fig 7.1 shows this implemented in an intersection.

Figure 7.1: Dynamic message signs indicate the lane use at the intersection in Utrecht, The Netherlands(Courtesy of Anema(2015))

Traffic Co-simulation

The brief work on co-simulation has proved that real work traffic scenarios can be generated using SUMO and dSpace ASM. The integration and the extrapolation algorithm are very accurate for urban traffic speeds to predict the position of traffic vehicles. The following improvements can be implemented:

- Additional modes of mobility from SUMO can be implemented in ASM; pedestrian and various vehicle types (buses, vans, etc.) improve the traffic scenario in ASM.

- Adding the infrastructure elements like buildings, pedestrian footpaths represent the terrain from the real environment more accurately. Some work has already started to work in this direction(refer to fig 7.2).

Figure 7.2: 3D render of buildings in OSU campus(extracted from Google Earth)

Appendix A: EVI Communication

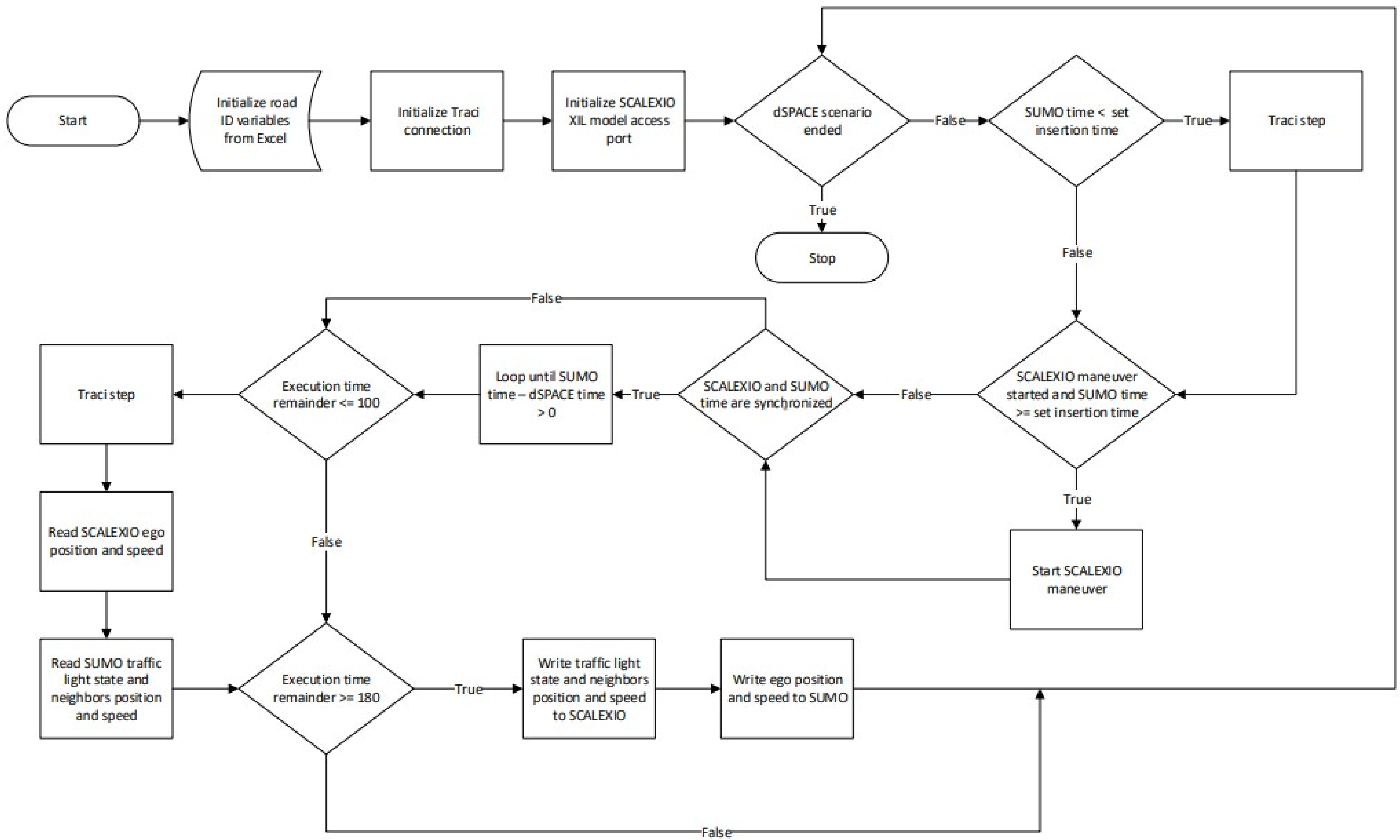


Figure A.1: EVI communication block flow chart

Bibliography

[1] Maria Coelho, Mark A Austin, and Mark Blackburn. "Semantic Behavior Modeling and Event-Driven Reasoning for Urban System of Systems Semantic Behavior Modeling and Event-Driven Reasoning for Urban System of Systems". In: December (2017).

[2] *Travel Trends - January 2020 - Policy | Federal Highway Administration*. URL: https://www.fhwa.dot.gov/policyinformation/travel_monitoring/20jantvt/page2.cfm.

[3] Von Der Fakultät V Verkehrs- Maschinensysteme and Yu Chen. "Adding a comprehensive Calibration Methodology to an Agent-Based Transportation Simulation". In: (2013).

[4] Robert E. Shannon. *Systems Simulation - The Art and Science*. Prentice Hall, 1975. ISBN: 0138818398.

[5] Martin Treiber and Arne Kesting. *Traffic flow dynamics: Data, models and simulation*. 2013, pp. 1–503. ISBN: 9783642324604. DOI: 10.1007/978-3-642-32460-4.

[6] Fajar Yoseph Chandra et al. "Improving the accuracy of real-time traffic data gathered by the Floating Car Data method". In: *2014 2nd International Conference on Information and Communication Technology, ICoICT 2014* November 2016 (2014), pp. 391–396. DOI: 10.1109/ICoICT.2014.6914094.

[7] *MS2 Portal*. URL: https://odot.ms2soft.com/TDMS.UI_Core/portal.

[8] Richard Dowling et al. "Guidelines for Calibration of Microsimulation Models: Framework and Applications:" in: *https://doi.org/10.3141/1876-01* 1876 (Jan. 2004), pp. 1–9. DOI: 10.3141/1876-01. URL: https://journals.sagepub.com/doi/10.3141/1876-01.

[9] Biagio Ciuffo, Vincenzo Punzo, and Vincenzo Torrieri. "Comparison of Simulation-Based and Model-Based Calibrations of Traffic-Flow Microsimulation Models:" in: *https://doi.org/10.3141/2088-05* 2088 (Jan. 2008), pp. 36–44. DOI: 10.3141/2088-05. URL: https://journals.sagepub.com/doi/10.3141/2088-05.

[10] *Transportation research. Part F : an international journal. (eJournal / eMagazine, 1998) [WorldCat.org]*. URL: https://www.worldcat.org/title/transportation-research-part-f-an-international-journal/oclc/936515472.

[11] Alexander Paz et al. "Calibration of traffic flow models using a memetic algorithm". In: *Transportation Research Part C: Emerging Technologies* 55.4 (2015), pp. 432–443. ISSN: 0968090X. DOI: 10.1016/j.trc.2015.03.001. URL: http://dx.doi.org/10.1016/j.trc.2015.03.001.

[12] J.M. de Jong. "Delft Center for Systems and Control". PhD thesis.

[13] Jingtao Ma, Hu Dong, and H. Michael Zhang. "Calibration of microsimulation with heuristic optimization methods". In: *Transportation Research Record* 1999 (Jan. 2007), pp. 208–217. ISSN: 03611981. DOI: 10.3141/1999-22. URL: https://journals.sagepub.com/doi/10.3141/1999-22.

[14] Seung-Jun Kim, Wonho Kim, and L. R. Rilett. "Calibration of Microsimulation Models Using Nonparametric Statistical Techniques". In: *Transportation Research Record: Journal of the Transportation Research Board* 1935.1 (Jan. 2005), pp. 111–119. ISSN: 0361-1981. DOI: 10.1177/0361198105193500113. URL: https://journals.sagepub.com/doi/10.1177/0361198105193500113.

[15] Byungkyu Park and Hongtu Qi. "Development and evaluation of a procedure for the calibration of simulation models". In: *Transportation Research Record* 1934 (Jan. 2005), pp. 208–217. ISSN: 03611981. DOI: 10.1177/0361198105193400122. URL: https://journals.sagepub.com/doi/10.1177/0361198105193400122.

[16] Sandro Chiappone et al. "Traffic simulation models calibration using speed–density relationship: An automated procedure based on genetic algorithm". In: *Expert Systems with Applications* 44 (Feb. 2016), pp. 147–155. ISSN: 0957-4174. DOI: 10.1016/J.ESWA.2015.09.024.

[17] Sandeep Menneni, Carlos Sun, and Peter Vortisch. "Microsimulation calibration using speed-flow relationships". In: *Transportation Research Record* 2088 (Jan. 2008), pp. 1–9. ISSN: 03611981. DOI: 10.3141/2088-01. URL: https://journals.sagepub.com/doi/10.3141/2088-01.

[18] Kh Abdalhaq and Maher I Abu Baker. "Using Meta Heuristic Algorithms to Improve Traffic Simulation". In: *Journal of Algorithms* 2.4 (2014), pp. 110–128.

[19] Wei Fan and Zegeye Gurmu. "Combined decision making of congestion pricing and capacity expansion: Genetic algorithm approach". In: *Journal of Transportation Engineering* 140.8 (2014). DOI: 10.1061/(ASCE)TE.1943-5436.0000695.

[20] Wei Fan and Randy B. Machemehl. "Optimal Transit Route Network Design Problem with Variable Transit Demand: Genetic Algorithm Approach". In: *Journal of Transportation Engineering* 132.1 (Jan. 2006), pp. 40–51. DOI: 10.1061/(ASCE)0733-947X(2006)132:1(40). URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-947X%282006%29132%3A1%2840%29.

[21] Ruey-Long Cheu et al. "Calibration of FRESIM for Singapore Expressway Using Genetic Algorithm". In: *Journal of Transportation Engineering* 124.6 (Nov. 1998), pp. 526–535. DOI: 10.1061/(ASCE)0733-947X(1998)124:6(526). URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-947X%281998%29124%3A6%28526%29%20https://ascelibrary.org/doi/abs/10.1061/%28asce%290733-947x%281998%29124%3A6%28526%29.

[22] Tao Ma and Baher Abdulhai. "Genetic Algorithm-Based Optimization Approach and Generic Tool for Calibrating Traffic Microscopic Simulation Parameters:" in: *https://doi.org/10.3141/1800-02* 1800 (Jan. 2002), pp. 6–15. DOI: 10.3141/1800-02. URL: https://journals.sagepub.com/doi/10.3141/1800-02.

[23] Jing Zhao et al. "Optimal operation of displaced left-turn intersections: A lane-based approach". In: *Transportation Research Part C: Emerging Technologies* 61 (Dec. 2015), pp. 29–48. ISSN: 0968090X. DOI: 10.1016/j.trc.2015.10.012. URL: https://arizona.pure.elsevier.com/en/publications/optimal-operation-of-displaced-left-turn-intersections-a-lane-bas.

[24] Byungkyu Park, Carroll J. Messer, and II Thomas Urbanik. "Traffic Signal Optimization Program for Oversaturated Conditions: Genetic Algorithm Approach:" in: *https://doi.org/10.3141/1683-17* 1683 (Jan. 1999), pp. 133–142. DOI: 10.3141/1683-17. URL: https://journals.sagepub.com/doi/10.3141/1683-17.

[25] B. Park, Carroll J. Messer, and T. Urbanik. "Enhanced genetic algorithm for signal-timing optimization of oversaturated intersections". In: *Transportation Research Record* 1727 (Jan. 2000), pp. 32–41. ISSN: 03611981. DOI: 10.3141/1727-05. URL: https://journals.sagepub.com/doi/10.3141/1727-05.

[26] Byungkyu Park and Joyoung Lee. "Optimization of coordinated-actuated traffic signal system: Stochastic optimization method based on shuffled frog-leaping algorithm". In: *Transportation Research Record* 2128 (Jan. 2009), pp. 76–85. ISSN: 03611981. DOI: 10.3141/2128-08. URL: https://journals.sagepub.com/doi/abs/10.3141/2128-08.

[27] Aleksandar Stevanovic and Peter T. Martin. "VisSim-Based Genetic Algorithm Optimization of Signal Timings:" in: *https://doi.org/10.3141/2035-07* 2035 (Jan. 2007), pp. 59–68. DOI: 10.3141/2035-07. URL: https://journals.sagepub.com/doi/10.3141/2035-07.

[28] Aleksandar Stevanovic et al. "Optimizing Traffic Control to Reduce Fuel Consumption and Vehicular Emissions: Integrated Approach with VISSIM, CMEM, and VISGAOST". In: *https://doi.org/10.3141/2128-11* 2128 (Jan. 2009), pp. 105–113. DOI: 10.3141/2128-11. URL: https://journals.sagepub.com/doi/10.3141/2128-11.

[29] Aleksandar Stevanovic, Jelka Stevanovic, and Cameron Kergaye. "Optimization of traffic signal timings based on surrogate measures of safety". In: *Transportation Research Part C: Emerging Technologies* 32 (July 2013), pp. 159–178. ISSN: 0968-090X. DOI: 10.1016/J.TRC.2013.02.009.

[30] Lihui Zhang, Yafeng Yin, and Shigang Chen. "Robust signal timing optimization with environmental concerns". In: (2013). DOI: 10.1016/j.trc.2013.01.003. URL: http://dx.doi.org/10.1016/j.trc.2013.01.003.

[31] Pengfei Li et al. "Simulation-Based Optimization of Maximum Green Setting under Retrospective Approximation Framework:" in: *https://doi.org/10.3141/2192-01* 2192 (Jan. 2010), pp. 1–10. DOI: 10.3141/2192-01. URL: https://journals.sagepub.com/doi/10.3141/2192-01.

[32] Thomas L Thorpe and Charles W Anderson. *Traffic Light Control Using SARSA with Three State Representations*. Tech. rep. 1996.

[33] Kok Lim Alvin Yau et al. *A survey on Reinforcement learning models and algorithms for traffic signal control*. June 2017. DOI: 10.1145/3068287. URL: https://dl.acm.org/doi/abs/10.1145/3068287.

[34] Mohamed A. Khamis and Walid Gomaa. "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework". In: *Engineering Applications of Artificial Intelligence* 29 (2014), pp. 134–151. ISSN: 09521976. DOI: 10.1016/j.engappai.2014.01.007. URL: http://dx.doi.org/10.1016/j.engappai.2014.01.007.

[35] Agon Memedi and Falko Dressler. "Vehicular visible light communications: A survey". In: *IEEE Communications Surveys and Tutorials* 23.1 (2021), pp. 161–181. ISSN: 1553877X. DOI: 10.1109/COMST.2020.3034224. URL: http://purelifi.com/case-studies/.

[36] Pablo Piñol Peral P. Pablo Garrido Abenza Manuel P. Malumbres. *Towards Simulation for Autonomous Mobility*. 2017, pp. 41–50.

[37] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. Vol. 2018-Novem. Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 2575–2582. ISBN: 9781728103235. DOI: 10.1109/ITSC.2018.8569938.

[38] Daniel L Gerlough and Matthew J Huber. *Traffic Flow Theory A Monograph*. Tech. rep.

[39] Laura Bieker-Walz et al. "Evaluation of car-following-models at controlled intersections". In: *31st Annual European Simulation and Modelling Conference 2017, ESM 2017* Krauss (2017), pp. 238–243.

[40] Junchen Jin. "Advance Traffic Signal Control Systems with Emerging Technologies". PhD thesis. KTH,Sweden, 2018. ISBN: 9789177297581. URL: https://www.diva-portal.org/smash/get/diva2:1205236/FULLTEXT01.pdf.

[41] *XIL API - dSPACE*. URL: https://www.dspace.com/en/inc/home/applicationfields/our_solutions_for/xil_api.cfm.

[42] Christian Sczyrba-Neumann. "ASAM XIL 2.2 Agenda Introduction Motivation for New Release". In: (2020). URL: https://www.asam.net/index.php?eID=dumpFile&t=f&f=3487&token=5a1adf3b6ee0c2b01462d3311b5fddc5e425ca2e.