SIGNAL TIMING FOR LCV TRUCKS ON A ROAD NETWORK USING REINFORCEMENT LEARNING

MOHAMMAD GHANBARI

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN CIVIL ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

August 2025

Abstract

Freight activity in urban networks is rising, and jurisdictions such as Ontario are encouraging the use of Long Combination Vehicles (LCVs) to consolidate freight loads. This thesis quantifies the delays and queueing on 16 intersections in the Region of Peel and introduces an adaptive signal-control strategy. Tested scenarios include (1) existing signal timing plans without LCVs and (2) with LCVs, (3) a single-intersection double deep q-network (DDQN) controller without LCVs and (4) with LCVs.

Introducing LCVs under existing signal timings raised network-wide delay by 14 % for all vehicles and 22 % for trucks when LCVs comprised just 1.7 % of traffic. The proposed DDQN was found to reduce average delays for all vehicles and trucks based on various conditions. Future work should extend the single intersection approach to a multi-agent framework and explore continuous-time action spaces for even finer control.

Acknowledgements

The journey toward completing this thesis has been marked by professional challenges and personal adversities. I would like to express my sincere gratitude to those who supported me throughout this journey.

First and foremost, my deepest appreciation goes to my supervisor, Dr. Kevin Gingerich, whose invaluable guidance, expertise, and unwavering support greatly contributed to the completion of this thesis. His insightful feedback and constant encouragement provided clear direction throughout my research.

I am also deeply thankful to Dr. Peter Park and Dr. Mehdi Nourinejad, whose courses enriched my academic understanding and whose continuous support and advice have always encouraged my ideas. I extend my sincere appreciation to Dr. Hina Tabassum for her critical evaluations, constructive suggestions, and expertise, which significantly enhanced the quality of my work. Additionally, my gratitude goes to Sindy Mahal for her steadfast support throughout my graduate studies.

Special thanks to the Lassonde School of Engineering for providing essential facilities, resources, and an intellectually stimulating academic environment that fostered my growth as a researcher.

I also acknowledge with appreciation the City of Mississauga and the Region of Peel for their generous provision of traffic data, which significantly contributed to achieving robust and reliable results in this project.

I express my sincere thanks to my colleagues and peers, Amir Ahmadian, Masoud Amirdadi, Mojtaba Davoodi, Mandana Hedayati, Won Mo Jeoung, Ali Khaksar, Luciano Marchesan, Tricia Martinez, Joseiry Marrero Rivas, and David Tran. Their insightful discussions and practical assistance were invaluable in navigating the various challenges encountered throughout this research.

Finally, I extend my heartfelt gratitude to my beloved family, particularly my mother, Roya, my sister, Elahe, and my brother-in-law, Masoud. Their constant emotional support, patience, and encouragement were pivotal in helping me navigate challenges. Their unconditional love and support formed the foundation of my strength and determination throughout this journey.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2D/3D	Two-Dimensional / Three-Dimensional (vehicle graphics in VISSIM)
A3C	Advantage Actor-Critic (RL algorithm)
AADT	Annual Average Daily Traffic
AASHTO	American Association of State Highway and Transportation Officials
AIMSUN	Advanced Interactive Microscopic Simulator for Urban & Non-Urban Networks
ATSC	Adaptive Traffic Signal Control
CAD	Computer-Aided Design
CKSR	Cost-Sensitive Kernel Sparse Representation (vehicle-detection algorithm)
COM	Component Object Model (VISSIM programming interface)
DDQN	Double Deep Q-Network
DLR	German Aerospace Center
DNN	Deep Neural Network
DQN	Deep Q-Network
EMF	European Modular System (for LCVs)
EMS	European Modular System (25 m vehicle configuration)
FHWA	Federal Highway Administration
FSP	Freight Signal Priority
GA	Genetic Algorithm
GDP	Gross Domestic Product
GEH	Geometric Efficiency-Hour (traffic-flow goodness-of-fit statistic)
GIS	Geographic Information System
GRG	Generalized Reduced Gradient (non-linear optimization solver in Excel)
GTA	Greater Toronto Area
GUI	Graphical User Interface
GVW	Gross Vehicle Weight
HCM	Highway Capacity Manual
HGV	Heavy Goods Vehicle
LCV	Long Combination Vehicle
LOS	Level of Service

LTR	Load Transfer Ratio
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
ML	Machine Learning
MOU	Memorandum of Understanding
MSE	Mean Squared Error (loss metric)
MTO	Ontario Ministry of Transportation
NEMA	National Electrical Manufacturers Association (ring-barrier phasing standard)
OBUs	On-Board Units (in V2X systems)
OPAC	Optimized Policies for Adaptive Control
PBS	Performance-Based Standards (for heavy vehicles)
PHF	Peak-Hour Factor
PLIDENT	Plan Identification - first generation real-time signal control (Glasgow)
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
QL	Q-Learning
RA	Rearward Amplification (LCV stability metric)
RBC	Ring-Barrier Controller (actuated signal logic)
RL	Reinforcement Learning
RSUs	Roadside Units (in V2X systems)
SARSA	State-Action-Reward-State-Action (on-policy RL algorithm)
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split Cycle Offset Optimization Technique
SGD	Stochastic Gradient Descent
SUMO	Simulation of Urban MObility (open-source micro-simulator)
SVM	Support Vector Machine
TSP	Transit Signal Priority
UCB	Upper Confidence Bound (exploration strategy)
V2X	Vehicle-to-Everything communication
VAP	Vehicle Actuated Programming (VISSIM signal module)

VISSIM	Verkehr In Städten – SIMulationsmodell (PTV microsimulation software)
VTM	Volvo Transportation Model
VPH	Vehicle Per Hour
WSDOT	Washington State Department of Transportation

Chapter 1. Introduction

1.1. Background

Truck transportation is an indispensable driver of economic growth, serving as a backbone that interconnects diverse industries and sectors. In Ontario, for instance, the trucking sector contributes 1% to the province's Gross Domestic Product (GDP). Currently, an estimated 200,000 trucks travel on Ontario’s highways, transporting goods with an estimated value of approximately $3 billion daily. It also represents the largest segment within the transportation industry, employing 36% of the workforce (Ontario Trucking Association, 2023). Ontario relies heavily on the efficiency of its trucking infrastructure to ensure the timely delivery of goods and sustain the complex web of supply chains.

One strategy for improving freight movement efficiency is the use of high-capacity vehicles to maximize load per trip. Long Combination Vehicles (LCVs) are an effective application of this approach, characterized by their extended length compared to standard trucks, which allows for transporting larger quantities of goods in a single trip. LCVs consist of a truck configuration in which more than one trailer or semi-trailer is towed in a single combination. This configuration not only enhances cost-effectiveness but also contributes to time savings, improved road safety, and a reduction in greenhouse gas emissions (Grislis, 2010; Regehr, et al., 2008).

While LCVs offer clear economic and environmental benefits, they also pose unique operational challenges. One of the primary challenges associated with LCV operations involves urban origin and destination points. The associated safety impacts and potential delays are major concerns in urban areas. Because of their extended length, navigating tight spaces can be time-consuming and pose risks to traffic safety as well as road infrastructure. Moreover, their size can block parts of the roadway, complicating lane changes for other drivers. Additionally, they can reduce visibility for surrounding vehicles, further affecting overall safety (Saha, 2021).

In urban areas, the primary bottlenecks often occur at signalized intersections, which pose challenges for all truck movements. Trucks require more time and greater distance to come to a complete stop at red lights, as well as additional time and distance to resume movement once the signal turns green. This process not only extends the total trip time for trucks but also causes delays for vehicles queued behind them. Furthermore, the frequent braking and acceleration of trucks at traffic signals lead to increased fuel consumption and a corresponding rise in air pollution (Maze, et al., 1998). LCVs experience even greater difficulties, as their larger dimensions demand more time to clear intersections and can impede other vehicles. They also face prolonged acceleration and deceleration times (Grislis, 2010).

To address the challenges posed by signalized intersections to truck movement efficiency, particularly for LCV, Freight Signal Priority (FSP) strategies have shown promising results. By optimizing signal timing in favor of freight vehicles, FSP reduces congestion, delays, and emissions, often with minimal disruption to other traffic. This approach is also relatively low-cost to implement compared to more extensive infrastructure solutions (Smith, et al., 2005).

1.2. Freight Signal Priority

Signal priority is a traffic management strategy that modifies regular signal operations to reduce stops and delays for specific vehicles (such as buses, emergency vehicles, or freight trucks) by detecting their approach and adjusting traffic light timing. Depending on the prioritized vehicle, this approach can be categorized into Transit Signal Priority (TSP), Freight Signal Priority (FSP), or Multi-modal Priority, all sharing the main objective of minimizing the number of stops and overall delays for designated vehicles. By detecting an approaching vehicle and leveraging real-time data, traffic signal controllers can be programmed to grant priority. However, it is essential to maintain a balance that preserves overall traffic flow and considers the needs of all road users.

There are three types of signal priority based on how they operate: Passive, Active, and Adaptive.

- **Passive priority** relies on historical data rather than real-time detector inputs, making it especially suitable for transit vehicles with predictable arrival times.
- **Active priority** detects vehicles in real time and adjusts signal timing accordingly, often through methods such as green extension or red truncation; this approach constitutes most signal priority deployments in the United States (Li, et al., 2011).
- **Adaptive priority** is the most recent iteration of signal priority techniques. It continuously monitors all incoming traffic and grants priority based on real-time conditions. In many cases, adaptive signal priority is implemented through Adaptive Traffic Signal Control (ATSC), which optimizes signal operations using live data (Li, et al., 2011).

Although recent research has focused on adaptive TSP, there is still a significant gap in studies examining adaptive FSP. Most FSP research so far has centered on passive and active strategies. For instance, Zhao et al. (2019) developed a traffic light control system specifically tailored to truck movements using passive and active FSP. When tested on a network of four corridors, their system achieved promising outcomes, including a 5% to 10% reduction in truck delays and fewer stops. Crucially, these improvements did not negatively impact, and in some cases even enhanced, travel times for passenger cars. However, the signalized intersections in their study operated individually, and corridor-wide coordination was not implemented.

In contrast, recent work on adaptive TSP has been especially notable. In a recent study, Othman et al. (2025) combined adaptive TSP with ATSC, extending control beyond a single intersection by introducing a novel coordination system across multiple intersections in a network. Their model uses a reinforcement learning (RL) algorithm to manage each intersection individually, aligning with emerging research that shows RL's promise for developing model-free ATSC and adaptive signal priority.

Another important issue involves differentiating the level of priority applied to various types of trucks. Iqbal et al. (2022) is among the few studies that considered multiple truck types in microsimulation by incorporating five distinct categories of freight vehicles. However, despite acknowledging these differences, they applied the same FSP approach to all trucks, leaving room for further refinement in how priority is assigned.

While adaptive signal priority has demonstrated promising results in TSP applications, to the best of our knowledge, no study has yet proposed an adaptive FSP solution. This gap is particularly significant given the fundamental differences between freight and transit vehicles, especially regarding their operational roles and frequency of travel. Adaptive TSP models typically evaluate network performance in terms of passenger-related delay, whereas freight operations are more concerned with space-related delay. Furthermore, no existing work distinguishes FSP treatments among different truck types. Taken together, these points highlight the importance of designing an adaptive FSP strategy that accommodates multiple truck types as well as general traffic. Accordingly, this study introduces a coordinated, RL adaptive traffic signal control method for an urban network, integrating FSP for various truck categories.

1.3. Research Goals and Objectives

This research aims to address gaps in the existing literature and provide novel insights into the effects of LCVs on urban networks, as well as the development of new signal timing strategies to mitigate their negative impacts. The following outlines the identified gaps and how this research seeks to address them:

1. <u>Analyze the Traffic Effects of Introducing LCVs to an Urban Network:</u>
   While previous studies have explored various impacts of introducing LCVs to urban networks, such as safety concerns (Woodrooffe, et al., 2004) and turning movements at intersections (Saha, 2021), there remains a lack of research on their broader traffic effects on the network as a whole. This research will use VISSIM microsimulation to analyze the traffic effects of LCVs introduced into a simulated urban network.
2. <u>Design RL Adaptive Signal Control Integrated with Adaptive FSP:</u>
   RL-based adaptive signal control has recently gained attention in several studies. Similarly, RL adaptive TSP has also been a focus of research. However, the application of this advanced technology to FSP remains largely unexplored. This research seeks to develop a novel signal timing strategy based on RL adaptive signal control integrated with adaptive FSP. In this approach, each intersection in the network acts as an agent capable of controlling itself.
3. <u>Assess FSP with Priorities for Individual Trucks Classes:</u>
   The literature currently lacks research on FSP strategies that account for different types of trucks. In this study, LCVs and standard trucks are treated with different levels of priority due to their distinct operational characteristics. This differentiation allows for the development of a tailored approach to freight signal priority that reflects the unique needs of various truck types.

1.4. Thesis Scope

This study aims to comprehensively explore and enhance the operational dynamics of signal priority methods within the trucking infrastructure of the Region of Peel. Serving as a strategic hub within Ontario's truck transit network, the Region of Peel boasts seven 400-series highways, Pearson international airport, and facilitates the daily transit of cargo worth approximately $1.8 billion. Projections for the year 2041 estimate that each person in the region will generate 2 to 3 truck loadings, resulting in an estimated 6 million truck origins or destinations annually in Peel (Peel Region, 2017). Notably, the busiest sector of Highway 401, between Highway 427 and Highway 410, alone witnesses the passage of up to 43,000 trucks each day (Government of Ontario, 2021). In addition, the Region of Peel consists of 26% of all LCV trips within Ontario (Saha, 2021), making this region an ideal case study for our project.

This research is structured across several distinct phases, beginning with a comprehensive literature review in Chapter 2, which synthesizes insights from prior studies on LCVs, signal priority methods, adaptive signal control systems, and RL traffic signals. Chapter 3 covers a detailed data collection process, focusing on gathering essential signal timing details and traffic metrics from the selected network intersections.

0 outlines the VISSIM simulation development in detail. It describes the calibration methods employed to ensure that the simulation accurately represents real-world scenarios. Additionally, this chapter provides background information on the reinforcement learning approach for addressing Markov process environments and introduces the Deep Q-network algorithm.

In Chapter 5, a comparison between pre- and post-calibration simulation results is presented to demonstrate the effectiveness of the calibration process. Subsequently, the baseline scenario results for the network are analyzed. A second scenario evaluates the network performance after introducing varying replacement rates of standard trucks with LCVs, highlighting differences compared to the baseline scenario.

Chapter 6 introduces a framework for RL traffic signal control incorporating FSP. As the third scenario, this RL model is implemented at a specific intersection within the network. The model is then modified to accommodate LCVs, forming the basis of the fourth and final scenario, where the intersection performance is evaluated under different LCV replacement rates. This chapter concludes with a comprehensive comparison of the outcomes across all scenarios applied at the selected intersection.

Finally, Chapter 7 provides a conclusive summary of the thesis. It begins by revisiting the problem statement, briefly reviews the data collection and methodology, and concludes with an overview of the key findings from the various analyzed scenarios.

Chapter 2. Literature Review

2.1. Long Combination Vehicles

Long Combination Vehicles are trucks that have either an extended length or both extended length and weight when compared to standard heavy truck-trailer or tractor-semitrailer combinations (Nagl, 2007). Table 1 presents a general classification of the various LCVs in operation worldwide. LCV lengths range from 25 meters up to 53.5 meters, and their gross weights span from 50 tons to an impressive 125.5 tons.

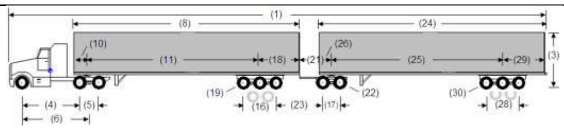
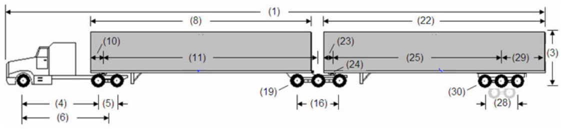
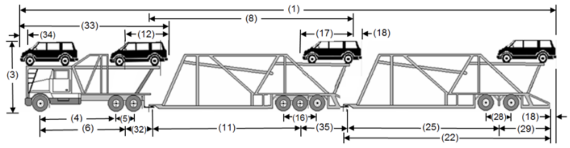
Table 1- General Classification of LCVs in Global Operations
Adapted from Grislis *(2010)*

LCV Type	Typical	Typical	Examples
Short	25 - 26	50 - 68	B-Double: Australia, North America Modular Concept Vehicle: Europe
Intermediate	26 - 30	60.5 – 85.7	Intermediate Double: USA Rocky Mountain Double: North America
Long	36 – 53.5	62 – 125.5	Road Train: Australia, Brazil Turnpike Double: North America Triple: North America

Regulatory frameworks and the precise definitions of LCVs vary by region. The European Union has policies that permit only certain short LCVs to operate (Nagl, 2007). For example, in Sweden, the European Modular System (EMS) permits LCVs up to 25.25 meters. However, Finland allows road trains that exceed this range, with permissions for lengths up to 34.50 meters (Nordic Innovation, 2020). Brazil's definition of LCVs includes tractor-trailer combinations with two or more trailers exceeding a gross vehicle weight (GVW) of 36 tons (Vieira, et al., 2013). In the United States, LCVs are categorized as truck-tractors pulling two or more trailers with a total gross weight exceeding 80,000 pounds (36,363.6 kg), and only 19 states operate the program (Perlman, et al., 2018). Meanwhile, in Canada, LCVs must not exceed the weight limits set for other articulated trucks and are defined as vehicles comprising a tractor and two or three trailers, with the total length surpassing 27.5 meters (Woodrooffe, 2010).

The LCV program in Canada began in 1969 with an initial 300 km route in Alberta. By 2016, it had extended to 17,000 km highway network spanning nine provinces, representing nearly half of Canada's total highway network (Wood, et al., 2017). The program's aim is to replace several tractor-trailers with a single LCV to reduce greenhouse gas emissions and enhance economic efficiency (Lightstone, et al., 2021). However, due to weight restrictions in Canada, LCVs primarily transport light-density goods. The Turnpike Double, Rocky Mountain Double, and Triple trailer configurations are identified as the most prevalent in Canadian operations (Ray Barton Associates Ltd., 2003). Table 2 presents the various LCV configurations permitted in Ontario along with their respective allowable carrying capacities. An A-Train Double comprises a three-axle tractor, a lead semi-trailer with tandem or tridem axles, a tandem-axle converter dolly, and a second semi-trailer with tandem or tridem axles. A B-Train Double configuration also involves a three-axle tractor and two semi-trailers with tandem or tridem axles, but it features a fifth-wheel assembly connecting the trailers instead of a dolly. Twin Stinger-Steer Auto Carriers consist of a three-axle tractor, a lead semi-trailer with triple axles (including a liftable rear-most axle), and a tandem-axle second trailer (Government of Ontario, 2023). Furthermore, the accessibility of LCVs is limited to specific highways in Ontario, with restrictions on their passage through certain roads. The primary network of highways for LCVs in the Greater Toronto and Hamilton Area is depicted in Figure 1.

Table 2- Permitted LCV Categories in Ontario
(Government of Ontario, 2023)

LCV Category	Allowable Gross Vehicle Weight	Configuration
A-Train Double	55,000 kg – 63,500 kg	
B-Train Double	63,500 kg	
Twin Stinger-Steer Auto Carrier	55,000 kg	

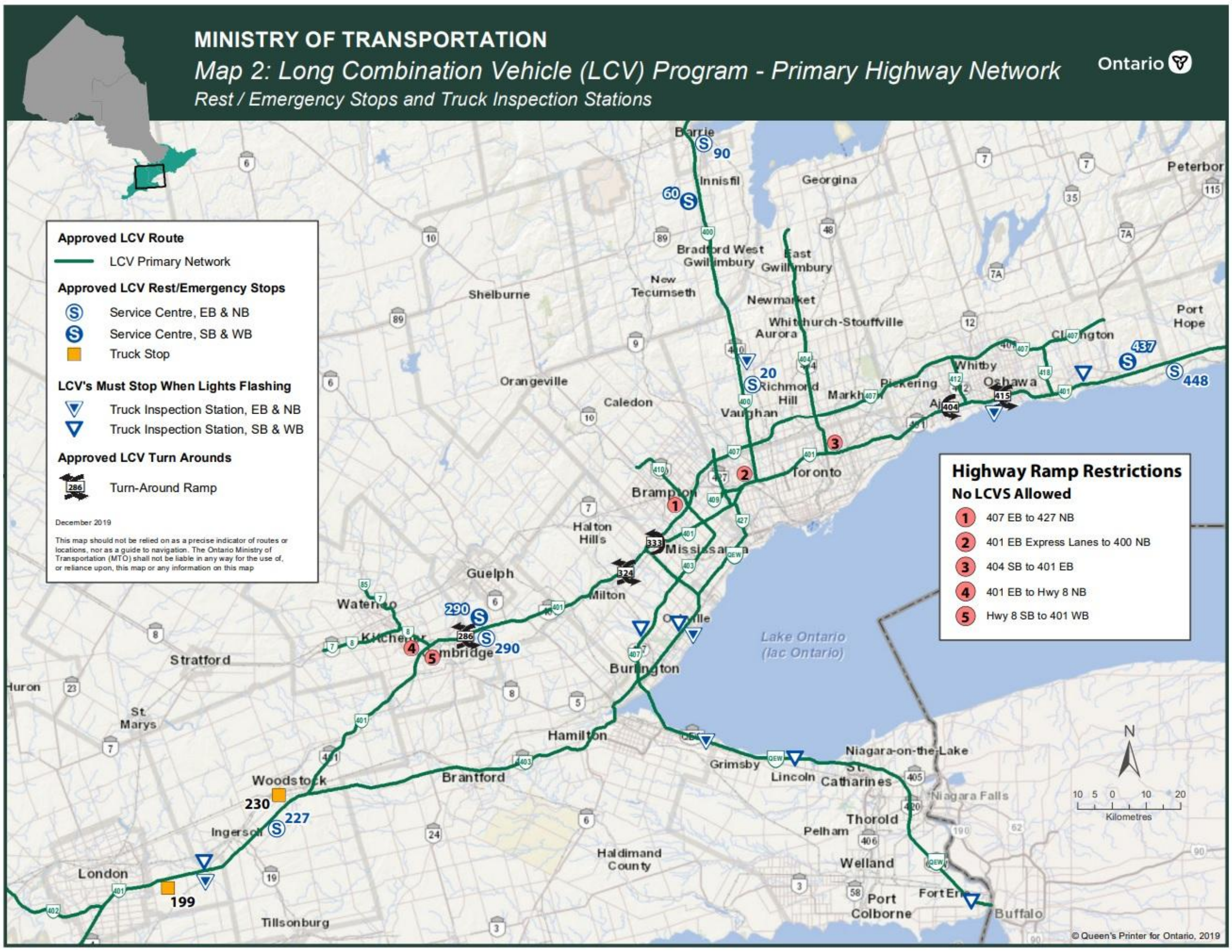


Figure 1- LCV Primary Network for The Greater Toronto and Hamilton Area
(Government of Ontario, 2023)

Long Combination Vehicles deliver remarkable benefits, including increased productivity, profitability, and improved environmental impact. They enhance transportation efficiency by reducing the number of truck movements and lowering shipping costs. For instance, firms with substantial LCV lane volumes can achieve a 16-42% reduction in total logistics costs (Middendorf, et al., 1994). In Brazil, B-Train Doubles and Road Trains enable 45% greater load capacity and 16% lower operational costs (Vieira, et al., 2013). Real-world cases like Canadian Tire's initiative, which replaced two trucks with a single LCV, demonstrate a 30% reduction in fuel usage, highlighting the tangible economic and environmental efficiencies of LCVs (Markowitz, 2022).

Despite their numerous advantages, Long Combination Vehicles pose notable challenges, particularly regarding stability and maneuverability. Kharrazi (2012) highlights the susceptibility of LCVs to rollovers and their poor lateral performance as significant safety concerns. Similarly, Wideberg, et al. (2009) observe that the increased weight and length of LCVs contribute to reduced stability. The NZ Tranport Agency (2001) emphasizes additional risks, noting that the flat sides of LCVs elevate the likelihood of side-swing incidents. Furthermore, A-train configurations are reported to be less stable compared to their B-train counterparts

Moreover, Long Combination Vehicles can negatively influence urban netwrok dynamics, with their extended length impacting traffic flow, safety, and infrastructure. Their large turning envelopes, particularly during right turns, demand adequate curb radii and lane widths to avoid conflicts with pavement edges or adjacent vehicles (Saha, 2021). In urban networks, intersections pose critical challenges for LCVs. The longer clearance times required at signalized intersections can disrupt traffic flow and elevate safety risks (Grislis, 2010). The data indicates that a majority (88%) of urban LCV collisions occur at intersections, emphasizing the need for infrastructure adjustments to accommodate these vehicles safely and efficiently (Woodrooffe, et al., 2004).

Simulation-based studies of LCVs predominantly emphasize their stability, maneuverability, and dynamic characteristics. For example, TruckSim software enables detailed simulations of LCV maneuvers and the integration of driver logic and control systems (Zhu, et al., 2022). This platform is particularly effective for evaluating performance metrics such as rearward amplification (RA), load transfer ratio (LTR), and off-tracking (Moghaddam, et al., 2022) (Sikder, 2017). Additionally, proprietary platforms like the Volvo Transportation Model (VTM) and open-source frameworks like OpenPBS are employed for specialized applications such as autonomous lane-changing and performance-based standards (PBS) evaluations (Kati, et al., 2019)

(Jacobson, et al., 2024). Despite these advancements, studies simulating the integration of LCVs with real traffic dynamics remain largely absent. In this context, a microsimulation software like VISSIM offers unique advantages for modeling LCV interactions within complex traffic networks to measure the impacts of LCVs on the network.

2.2. Traffic Signal Control

Traffic signal timing is the process of systematically scheduling traffic lights to manage vehicle flows at intersections. Effective timing aims to optimize a specific objective, such as reducing travel time delays or the number of stops, by adjusting parameters including the sequence and duration of signal phases, the cycle length, and the offset time (El-Tantawy, 2012). A phase is a specific interval during which certain movements, such as through traffic or turning vehicles, receive the right-of-way. The green time allocated to each phase determines how long these movements are served. Once the cycle completes all phases, the total duration of a single rotation is known as the cycle length, and the portion of that cycle allocated to a particular phase is called the split time, which includes both the green time and the clearance time (amber and all-red) for that phase (Federal Highway Administration , 2008).

The phase sequence determines the order and grouping of signal phases within a cycle. There are three main types of phase sequences: Fixed Phase Sequence, Selected Phase Sequence (where certain phases can be skipped based on real-time demand), and Not-Limited Phase Sequence. Figure 2 illustrates a four-legged intersection with eight phases, arranged according to the ring-and-barrier structure (Wu, et al., 2019). Under this structure, each phase belongs to a specific ring and group, and its split time is defined accordingly. Furthermore, the sum of split times for all phases in each ring must equal the cycle length.

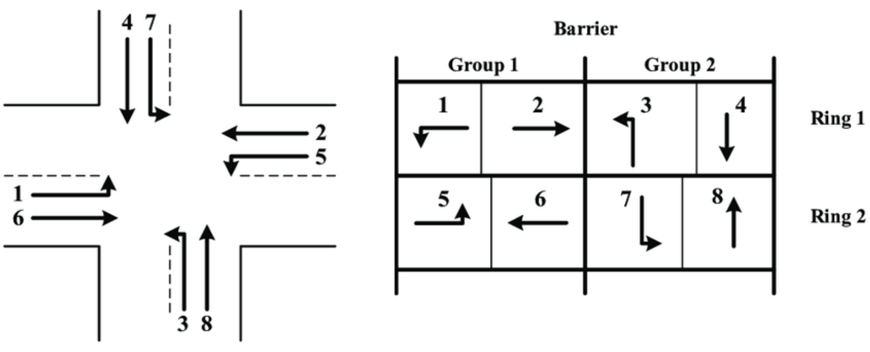


Figure 2- Sample Ring-Barrier Phase Sequence
(Wu, et al., 2019)

In optimizing signal timing parameters, the problem can be categorized into three types based on the level of coordination: isolated intersections, arterial networks, and general networks. For an isolated intersection, the objective is localized, focusing on optimizing the phase sequence and split time without considering broader system impacts. However, for arterial or general networks, the focus expands to improving system wide performance by jointly considering objectives across all intersections (Eom, et al., 2020). The concept of offset time is crucial in network level signal control and refers to the time shift (relative to a reference point) for starting a new cycle at an intersection (El-Tantawy, 2012). By coordinating offsets among adjacent signals, traffic can often pass multiple intersections without stopping, an effect known as the "green wave," which reduces both the number of stops and overall delay (Zhang, et al., 2016).

Traffic signal operations typically fall into three main categories: fixed-time control, actuated control, and adaptive control. In fixed-time systems, each signal phase is predetermined based on historical data or engineering judgment, and these timings remain unchanged regardless of current conditions. Actuated controllers introduce vehicle detection via sensors or cameras, allowing green time to be extended or phases to be skipped in response to actual traffic demand. Adaptive controllers build on this approach by continuously analyzing data from multiple detection sources and automatically adjusting signal settings to optimize traffic movement across one intersection or an entire corridor or network (Federal Highway Administration , 2008).

2.2.1. Fixed-Time Control

A fixed-time traffic control system runs according to pre-determined intervals per signal phase, having a fixed cycle independent of any variation in traffic flow. The fixed timings are normally determined by engineers via analysis of historical records, use of field observations, and conformity to established standards. The strict schedule adherence by the controller facilitates operational simplicity and maintenance, thereby providing consistent patterns that are simple to synchronize from one intersection to the next. But the inflexibility of such a system can create inefficiency in periods of fluctuating demand; low-demand routes can be green for far too long, and more-demand routes are not provided with sufficient green time. Despite such flaws, fixed-time control is usually preferred where traffic streams exhibit relative constancy and predictability, or where low-cost and simple solutions are satisfactory (El-Tantawy, 2012).

Various models have been developed for more than 60 years to optimize fixed time controllers, each with different objectives and methodologies. The Webster Method, recognized as the first signal timing approach, calculates an optimal cycle length and green time splits to minimize average vehicle delay (Webster, 1958). The Highway Capacity Manual (HCM) Method demonstrates models for delay and Level of Service (LOS), distributing green time to minimize these metrics (Transportation Research Board, 2000). Additionally, SYNCHRO and TRANSYT are software-based models that optimize fixed time signal plans for an arterial by first calculating a common cycle time and then determining the offset time at each intersection (Robertson, 1969).

2.2.2. Actuated Control

Unlike fixed time controllers, actuated controllers respond to traffic fluctuations by using sensors, such as inductive loop detectors or cameras, to detect vehicle presence. Based on these detections, the controller can skip phases without demand or extend the green interval of a phase (Eom, et al., 2020). There are two main types of actuated signals: semi actuated and fully actuated. In semi actuated systems, detectors are placed only on minor approaches (callable phases), while major approaches operate as non-callable phases. This setup is ideal for locations where a low volume minor street intersects a high-volume major street. Fully actuated systems, on the other hand, use detectors on every approach, continuously adapting cycle sequences and phase durations. This makes them particularly effective at intersections with less predictable traffic patterns across all approaches (El-Tantawy, 2012).

An actuated controller uses detector inputs to determine how long a green signal remains active for a particular movement, guided by parameters such as minimum green, maximum green, and gap time (Federal Highway Administration , 2008). The minimum green ensures that once a phase begins, it remains green long enough for queued vehicles to start moving, often accounting for driver reaction times. The gap time is the interval between consecutive vehicle detections; if it exceeds a specified threshold, the controller assumes that demand is insufficient to continue the green. The maximum green sets an upper limit on the extension of a phase, preventing a single movement from dominating the intersection. As vehicles are detected, the actual green duration can vary between these minimum and maximum limits while also adhering to any coordination constraints (see Figure 3). In the literature, numerous methods have been proposed to adjust these actuated signal control parameters, including fuzzy logic systems (Shiri, et al., 2017), the Webster method (Taale, 2002), stochastic models based on Poisson demand estimation (Zhang, et al., 2011), hybrid optimization approaches such as genetic algorithms (Kim, et al., 2003), and analytical methods (Chang, 1996).

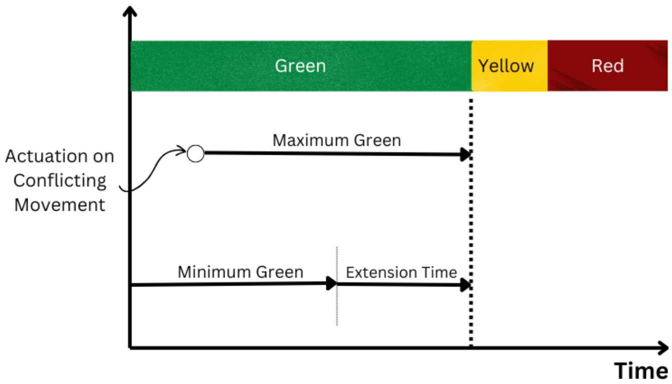


Figure 3- Actuated Controller Parameters
(Federal Highway Administration , 2008)

2.2.3. Adaptive Control

Adaptive Traffic Signal Control (ATSC) dynamically adjusts the duration of traffic lights in response to real-time traffic demand. Because it requires continuous monitoring of intersection conditions and frequent updates, implementing ATSC tends to be more expensive than fixed-time or actuated signal control. It also typically relies on advanced computational techniques to handle the complex iterative processes needed for real-time optimizations (Eom, et al., 2020). However, as traffic flow often fluctuates, ATSC has been shown to outperform other control methods in many scenarios, substantially reducing delays, congestion, and greenhouse gas emissions (Agrahari, et al., 2024).

The first concept for ATSC was proposed by Miller (1963), who introduced a real-time data-driven model for traffic signal control (Friedrich, 2002). However, this approach was not successful in field trials, and the first practical implementation was PLIDENT, introduced in Glasgow in the late 1960s (Holroyd, et al., 1971). The 2nd generation of ATSC strategies incorporated more advanced algorithms for corridor- or network-wide optimization. Two prominent examples are SCATS (Sims, et al., 1980) and SCOOT (HUNT, et al., 1981). The 2^{nd} generation models use a centralized approach, in which a central system determines signal timing parameters for multiple intersections to optimize a corridor or network. However, this approach depends heavily on reliable communication between local controllers and the central computer. In the event of communication failure, local controllers revert to independent operation, optimizing only their own intersections. Additionally, the central system itself is vulnerable as a single point of failure, and its computational burden grows significantly with the size of the network in a centralized approach (El-Tantawy, 2012).

The 3^{rd} generation models emerged as a response to the drawbacks of centralized approaches. They decompose the network-wide optimization problem into smaller, decentralized components, where each intersection operates locally but can still share critical information such as upstream vehicle arrivals (Friedrich, 2002). This structure enables each intersection to adapt its signal timing without strictly adhering to a common cycle length or offset across the corridor or network, allowing for flexible (acyclic) cycles (El-Tantawy, 2012). The 3^{rd} generation controllers follow a model-based approach, requiring an environment model that closely reflects real conditions (El-Tantawy, 2012). These methods typically incorporate detailed traffic flow equations, vehicle arrival patterns, and intersection dynamics to guide decision-making, and they rely heavily on predicting future traffic patterns (Agrahari, et al., 2024). Because the resulting models can be quite large and complex, especially when applied to corridors or networks, simplifications are often necessary (Wang, et al., 2022). Even after simplification, the optimization can still be computationally intensive, and complex algorithms are typically employed to solve these problems.

Challenges in model-based approaches led to the development of the 4^{th} generation of ATSC, often termed self-learning or model-free controllers (Wang, et al., 2022). Recent advances in reinforcement learning (RL) applications across various fields have enabled the emergence of multiple model-free ATSC methods. The general framework for RL is shown in Figure 4. In an RL approach, an agent (here, the traffic signal controller) interacts directly with the environment (the intersection), observing its current state, taking actions, and receiving rewards based on the outcomes. Over time, the agent learns to choose actions that maximize its rewards, effectively optimizing traffic signal performance without requiring an explicit traffic model.

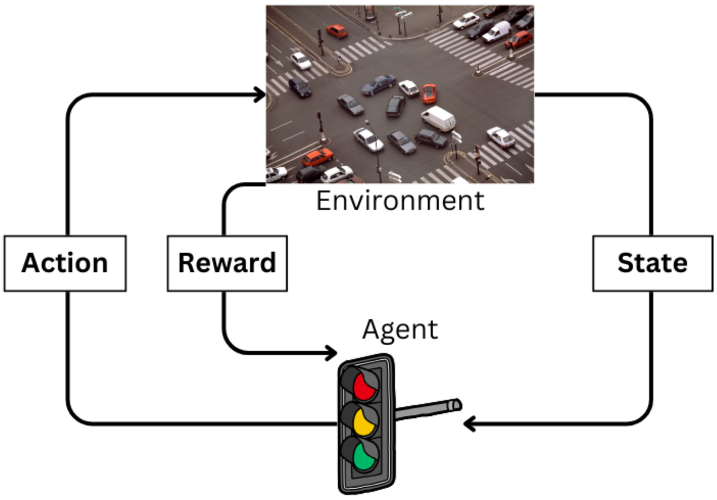


Figure 4- Reinforcement Learning Traffic Control Framework

Different RL traffic signal control methods vary in their definitions of actions, states, and rewards, resulting in distinct designs and outcomes. The definition of the state space strongly influences the learning speed and effectiveness: a more accurate depiction of actual intersection conditions fosters quicker and more robust learning (Jaakkola, et al., 1994). According to Noaeen, et al. (2022), who reviewed nearly 200 RL ATSC studies, states may include queue lengths for each approach or lane, current signal phase information, and vehicle counts or flow rates, vehicle position, and vehicle speeds. On the reward side, metrics such as total queue length, total delay, or total waiting time are commonly used improve performance.

In RL ATSC, actions typically involve defining both the phase sequence and phase duration. Two main approaches exist for setting the phase sequence: fixed and variable (Eom, et al., 2020). Under a fixed sequence, the order of phases is predetermined and cannot be changed, whereas a variable sequence allows the system to decide both the order and duration of phases at each decision point. Although variable sequencing often achieves better simulation performance (El-Tantawy, 2012), it may introduce driver confusion and raise safety concerns (Mirchandani, et al., 2001).

Additionally, Wang et al. (2022) categorize RL optimization by action frequency, distinguishing between a second-by-second and a cycle-by-cycle approach. In the second-by-second method, the controller decides each second whether to extend the current phase or switch to another, offering granular control but potentially less predictability for drivers (Medina, et al., 2012; Ma, et al., 2021).

Conversely, the cycle-by-cycle method applies actions only at the end of each cycle, defining the next cycle's sequence, split times, and cycle length (Torabi, et al., 2018; Salkham, et al., 2008). While second-by-second control can be more flexible, cycle-by-cycle adjustment is often more practical for field implementation because it presents consistent phase durations that drivers can anticipate (El-Tantawy, 2012).

To achieve corridor-wide or network-wide optimization, multi-agent reinforcement learning (MARL) can be employed, treating each intersection as an agent. Because an agent's action affects not only the state of its own intersection but also potentially impacts neighboring intersections, mutual awareness among agents becomes critical (Wei, et al., 2021). MARL approaches are typically categorized into centralized and decentralized control (El-Tantawy, 2012). In the centralized approach, a single controller receives state information from all intersections and determines a joint action for the entire network, ensuring global awareness. Although this method can be effective (Chu, et al., 2019; Prashanth, et al., 2011), it often suffers from high dimensionality, increased communication demands, and lower fault tolerance (Negenborn, et al., 2014).

The decentralized control approach divides the global optimization task into parallel subproblems, with each agent managing its own intersection. This can be implemented either with or without direct communication among agents (Wei, et al., 2021). In simpler arterial scenarios, decentralized control without communication can achieve localized mini green waves effectively (Casas, 2017; Zheng, et al., 2019). However, as scenarios grow more complex, conflicts between neighboring agents' objectives may increase. In such cases, communication among agents helps them coordinate their actions as a group (Wang, et al., 2022). Communication can take various forms, ranging from sharing state information to enhance system-wide awareness (Wang, et al., 2024), defining shared rewards to encourage cooperation (Wei, et al., 2019), or exchanging past actions (Ge, et al., 2019), or using model-based feedback (Torabi, et al., 2018).

2.3. Freight Signal Priority

Traditional signal controllers used a one-size-fits-all approach, optimizing signal timing for all vehicles without considering their individual characteristics. However, vehicles on the road differ in terms of priority, acceleration, size, weight, and their potential to contribute to air pollution. For instance, freight vehicles, due to their slower acceleration and deceleration rates, require more time and space to come to a stop at red lights and need additional time to proceed through intersections after stopping (Manta, 2019). On the other hand, transit vehicles are impacted by signalized intersections, experiencing decreases in reliability and level of service (Kaisar, et al., 2020; Hassannayebi, et al., 2024). As such, traditional signal controllers are no longer adequate, particularly in areas with significant freight traffic and high transit service frequencies. Signal priority strategies, including Transit Signal Priority (TSP) and Freight Signal Priority (FSP), are inexpensive operational improvements designed to enhance the movement of freight and transit vehicles at signalized intersections while minimizing the disruption to other road users (Smith, et al., 2005).

Freight Signal Priority (FSP) and Transit Signal Priority (TSP) are two prominent signal priority strategies with shared methodologies and implementation techniques (Murshed, et al., 2020). However, they differ significantly in practice, as highlighted by Zhao, et al. (2019). For one, freight vehicles typically exhibit varied flow patterns at intersections, especially near logistical hubs such as ports and warehouses, unlike the more uniform movement of transit vehicles. Furthermore, freight schedules are less consistent than transit timetables, leading to less predictable arrival times at signals. Lastly, freight vehicles are often assigned a lower priority compared to transit vehicles. In general, while TSP is broadly acknowledged and applied to enhance traffic network efficiency and reliability, yet research into the specific advantages of FSP alone remains limited, indicating an area for further investigation (Talukder, et al., 2022).

The strategies of signal priority can be categorized into three main types: passive priority, active priority, and real-time priority (Smith, et al., 2005). The passive signal priority operates on a continuous basis, functioning without the need for real-time traffic data. This approach is particularly effective in scenarios where traffic volumes are low, when freight and transit schedules are consistent, or where there's a substantial passage of these vehicles (Baker, et al., 2002). For instance, in London, passive priority methods outperformed active methods in terms of efficiency when bus frequency exceeded 60 buses per hour (Hounsell, et al., 1995).

Active signal priority strategies provide priority treatment after detection a specific transit or freight by altering the signal timings. It requires real-time traffic information to operate effectively, and the most implemented active strategies are the green extension and early green (Kaisar, et al., 2020). Green extension occurs when a priority vehicle is detected approaching the intersection and is anticipated to arrive just as the signal is turning red. In this case, the signal controller extends the green phase, allowing the vehicle to proceed through the intersection without stopping. The early green method is used when a vehicle approaches the intersection during the final seconds of a red signal. Here, the signal controller shortens the red phase, allowing the signal to turn green earlier and the transit or freight vehicle to continue through without stopping. Further known techniques include actuated transit or freight phases and phase insertion (Baker, et al., 2002).

Active signal priority strategies are divided into unconditional and conditional approaches, each with distinct characteristics and applications. Unconditional priority grants immediate signal adjustments to any detected transit or freight vehicle without evaluating traffic conditions or other metrics (Smith, et al., 2005). This strategy, while simple to implement, it can prove less effective in high-volume traffic, where frequent priority calls interfere with general traffic flow (Lin, et al., 2014). In contrast, conditional active priority leverages additional data, such as vehicle lateness for freight (Talukder, et al., 2022) and transit (Hu, et al., 2016), bus occupancy (Christofa, et al., 2011), or prevailing traffic conditions (Ye, et al., 2017), to selectively grant priority. This method enhances system-wide efficiency by balancing the needs of prioritized and non-prioritized traffic. In practice, major cities like Seattle, Portland, Los Angeles, and Chicago have largely adopted conditional priority methods in their active signal priority systems (Wang, et al., 2020).

Real-time signal priority can be implemented through two primary approaches: adaptive traffic signal control (ATSC) integrated with priority and standalone adaptive signal priority. ATSC with integrated priority optimizes signal timing for general traffic conditions while dynamically adjusting to provide priority upon early detection of approaching freight or transit vehicles, thereby minimizing disruptions to other traffic (Smith, et al., 2005). For instance, SCOOT-based controllers apply active measures such as green extension, early green recall, or phase skipping to minimize delays and boost passenger throughput (Dion, et al., 2005). In contrast, adaptive signal priority focuses more explicitly on balancing the trade-offs between priority vehicles and general traffic, determining when and how to grant priority based on real-time conditions (Hu, 2022). This approach can operate independently of ATSC and is compatible with other signal control methods, including actuated control systems (Li, et al., 2011).

Adaptive signal priority models use a wide range of objective functions and optimization techniques to improve traffic efficiency. Behbahani et al. (2024) and Zhou, et al. (2007) focused on multi-objective optimization to decrease delays for both buses and non-transit vehicles, utilizing genetic algorithms (e.g., parallel and nondominated sorting [NSGA]) to identify optimal timing plans along a Pareto frontier. Lee and Wang (2022) introduced a person-based adaptive TSP model that minimizes overall person delay, applying cooperative strategies to balance delays across various traffic streams through adaptive signal adjustments. In another study, Wu et al. (2023) proposed a game-theory-driven collaborative optimization method aiming to reduce both bus delays and average delays for non-transit vehicles, employing Pareto optimization to reconcile these competing goals. Although these model-based approaches have yielded promising outcomes, they also face limitations in complexity, scalability, and the need for near-perfect modeling.

Model-free adaptive signal priority overcomes many of the limitations associated with model-based approaches by optimizing in real time without relying on explicit traffic models. Reinforcement learning (RL) underpins these self-learning, model-free signal controllers, and several RL methods have been proposed for transit signal priority. Dong et al. (2024) employed a deep Q-network (DQN) approach that assigns priority by selectively switching phases or extending green time in response to queue imbalances. In another study, Hu et al. (2023) introduced a two-way priority strategy that regulates green phases to control transit headway adherence, granting early or extended greens to reduce deviations in travel schedules. Moreover, Shabestary et al. (2019) and Othman et al. (2025) emphasized person throughput by dynamically modifying signal phases based on passenger volume and sensor-derived vehicle data, granting priority to incoming transit vehicles with high ridership.

In general, signal priority systems offer significant benefits, enhancing the efficiency and appeal of transit and freight by improving schedule reliability, reducing travel times and stops, and providing a smoother ride for passengers. These systems contribute to the longevity of transit equipment and reduce road wear, leading to less pavement maintenance. In addition to increased comfort, they result in lower emissions, making transit a more competitive option compared to single-occupancy vehicles. For example, Toronto's implementation of TSP led to a reduction in tram travel time by 6-10% (Niedzielski, et al., 2024). Y. Lin (2014) notes TSP's effectiveness in diminishing bus delays, waiting times, and fleet sizes while enhancing punctuality. Zhou and Gan (2006) found that TSP is particularly beneficial under moderate traffic conditions. While applying FSP alone offers substantial savings, combining TSP and FSP yields similar advantages, supporting both transport modes effectively in high-demand scenarios (Shahadat Iqbal, 2022). Field data further supports this, indicating that the combination of TSP and FSP improved travel times for buses by 8.2% and trucks by 39.7% (Kyoungho Ahn, 2015). Therefore, TSP and FSP, when applied in conjunction, present a preferred solution to address the multifaceted demands of urban transportation.

Moreover, real-time signal priority extends beyond both active and passive strategies by incorporating dynamic traffic data and adaptive optimization, leading to more efficient and context-aware control. For instance, Zhou et al. (2024) demonstrated that their adaptive signal control model reduced person delays by 33% and queue lengths by 40% compared to the conditional active signal priority proposed by Thodi et al. (2021). Likewise, Hu et al. (2023) found that an RL adaptive priority scheme yielded better transit headways than the existing actuated controller with unconditional active transit priority in Toronto. Additionally, real-time frameworks are well-suited for multimodal environments, enabling simultaneous management of buses, freight, and pedestrian traffic based on up-to-the-moment demands and performance objectives such as person throughput and headway consistency (Othman, et al., 2025; Hu, et al., 2023).

Although numerous methods and an extensive literature base exist for TSP, FSP still requires increased attention. Research on TSP shows that real-time signal priority often outperforms active or passive models, but most FSP studies have focused on active FSP integrated with actuated or fixed-time controllers. For instance, Iqbal et al. (2022) employed conditional active priority, granting priority based on vehicle detection and demand scenarios while using fixed-time baseline timings. Similarly, Kari, et al. (2014) applied conditional active priority to a fixed-time controller, and Talukder et al. (2022) combined conditional active priority with actuated signal timings. Only a few studies have examined real-time FSP. Zhao et al. (2019) implemented a model-based ATSC featuring both conditional active and passive FSP, and Guo and Wang (2021) integrated conditional active FSP within a SCOOT-based baseline. One of the limited adaptive FSP examples is found in Das et al. (2022), who developed a model-based adaptive FSP on actuated signals. However, to the best of our knowledge, no 4th generation (model-free) ATSC integrated with FSP has been reported, and there appears to be no existing research on model-free adaptive FSP.

2.4. Microsimulation Modelling

Because RL ATSC cannot be trained directly in real traffic, a calibrated simulation that closely replicates actual conditions is essential. These simulations also serve as testing platforms before any real-world implementation. Broadly, traffic simulations are either microsimulation or macrosimulation. Microsimulation models focus on detailed vehicle interactions, such as lane changes and queuing, offering granular, real-time data that is particularly useful for RL methods (Wei, et al., 2021). Macroscopic models, on the other hand, aggregate traffic flow, capturing overall metrics like average speed and density. For RL signal control, microsimulation is generally preferred because it provides the fine-grained state information and immediate reward feedback necessary for accurate learning and decision-making.

The most widely used traffic simulator for developing RL ATSC is SUMO (Simulation of Urban Mobility) (Noaeen, et al., 2022). SUMO is an open-source, microscopic, and multi-modal simulator developed by the German Aerospace Center (DLR), capable of large-scale traffic modeling by simulating individual vehicles and their interactions within complex urban or highway networks (Krajzewicz, et al., 2002). SUMO's ability to collect detailed, real-time traffic performance metrics (e.g., travel time, queue length, vehicle throughput) makes it especially well-suited for RL signal control applications.

VISSIM is the second most commonly used microsimulation platform for developing RL ATSC (Noaeen, et al., 2022). Its features include detailed simulation of individual vehicle behaviors, precise handling of complex intersection configurations, and extensive multi-modal traffic scenarios involving pedestrians, cyclists, and public transportation. VISSIM also provides robust visualization tools, enhancing insights into traffic flow, congestion patterns, and network performance through precise metrics such as delay times, queue lengths, and stop rates. Overall, VISSIM's detailed driver behavior modeling and highly customizable interface often distinguish it for complex microsimulation tasks (Fellendorf, et al., 2010). Additionally, calibration in VISSIM is greatly simplified due to its extensive collection of realistic and granular data metrics. The simulator provides detailed information about vehicle trajectories, speeds, acceleration patterns, and queue lengths (Wisconsin Department of Transportation, 2020). This allows users to fine-tune simulation parameters to closely match observed real-world traffic conditions, facilitating reliable and validated simulation outputs necessary for effective RL model training and evaluation.

Moreover, VISSIM supports adaptive traffic signal control through several robust approaches, including the Component Object Model (COM) interface and Vehicle Actuated Programming (VAP). The COM interface allows external software such as Python scripts or C# programs to interact dynamically with simulations in real-time, enabling precise adjustments of signal timings based on evolving traffic conditions (Joelianto, et al., 2019). Additionally, VISSIM's VAP facilitates creating customized signal control logic using scripts that respond to real-time detector inputs, allowing precise adaptation of signal timings based on traffic demand (Guo, et al., 2016). Furthermore, by integrating with external RL frameworks like TensorFlow or PyTorch through the COM interface, VISSIM supports adaptive control strategies utilizing sophisticated algorithms such as Deep Q-Network (DQN) or Advantage Actor-Critic (A3C) (Rasheed, et al., 2020). These methods leverage detailed traffic state information from VISSIM, continually updating control strategies and enhancing traffic responsiveness and network performance.

The integration of these features makes VISSIM highly conducive to applying RL signal timing, offering substantial advantages in adaptive signal control studies. Its user-friendly interface, detailed calibration capabilities, and real-time adaptive control features collectively create an ideal environment for testing, evaluating, and optimizing RL algorithms, ultimately contributing to efficient and effective traffic management. For successful implementation of these processes, comprehensive data collection is critical. Essential data includes detailed traffic counts, vehicle trajectories, speed profiles, intersection geometry, traffic signal plans, and pedestrian and cyclist volumes. Accurate and granular data ensures precise model calibration, validation, and robust RL algorithm training, significantly enhancing the reliability and applicability of simulation outcomes.

Chapter 3. Study Area

3.1. Region of Peel

The Region of Peel, a regional municipality within the Greater Toronto Area (GTA), is situated to the west and northwest of Toronto. It encompasses three municipalities: Mississauga, Brampton, and Caledon. With a population exceeding 1.5 million residents, Peel is one of the most densely populated regions in Ontario (Statistics Canada, 2024). This area is a significant economic hub, hosting approximately 35,000 businesses that provide around 700,000 jobs across various sectors, including wholesale trade, transportation, retail, manufacturing, and healthcare services (Region of Peel, 2021). The goods movement industry is a significant part of the region's economy, providing 43% of jobs in Peel (Region of Peel, 2022). Peel offers a unique opportunity for the transportation of goods with major highways (Highway 401, Highway 403, Highway 410, and Highway 407), Canada's largest and busiest airport (Toronto Pearson International Airport), one of the Canada's major railroad (CN Brampton Intermodal Terminal), and the presence of large companies like Amazon and Loblaw (Region of Peel, 2012).

Trucks are one of the three primary modes of goods transportation in the Region of Peel, alongside trains and planes. In 2015, 22,500 truck trips started or ended in the Region of Peel daily, carrying goods valued at $1 billion (Region of Peel, 2015). According to the Region of Peel's Goods Movement Strategic Plan from 2012, trucking activities in Peel accounted for 25% of all truck activity in Ontario, with over 2,000 trucking companies operating in the area (Region of Peel, 2012).

The Region of Peel also supports the Long Combination Vehicles (LCVs) program by designating specific provincial highways, regional, and local roadways to accommodate their through and turning movements. LCV trips in the Region of Peel account for 26% of all LCV trips in Ontario (Saha, 2021). Figure 5 illustrates the potential and existing LCV routes in the Region of Peel as of 2019. As shown in Figure 1 and depicted in Figure 5 with red lines, most provincial highways are considered primary routes for LCVs. Additionally, Figure 5 highlights high LCV potential areas with green circles and existing LCV arterials with blue lanes. The areas marked with blue lanes and green circles align perfectly with the objectives of this study, making them ideal candidates for the case study.

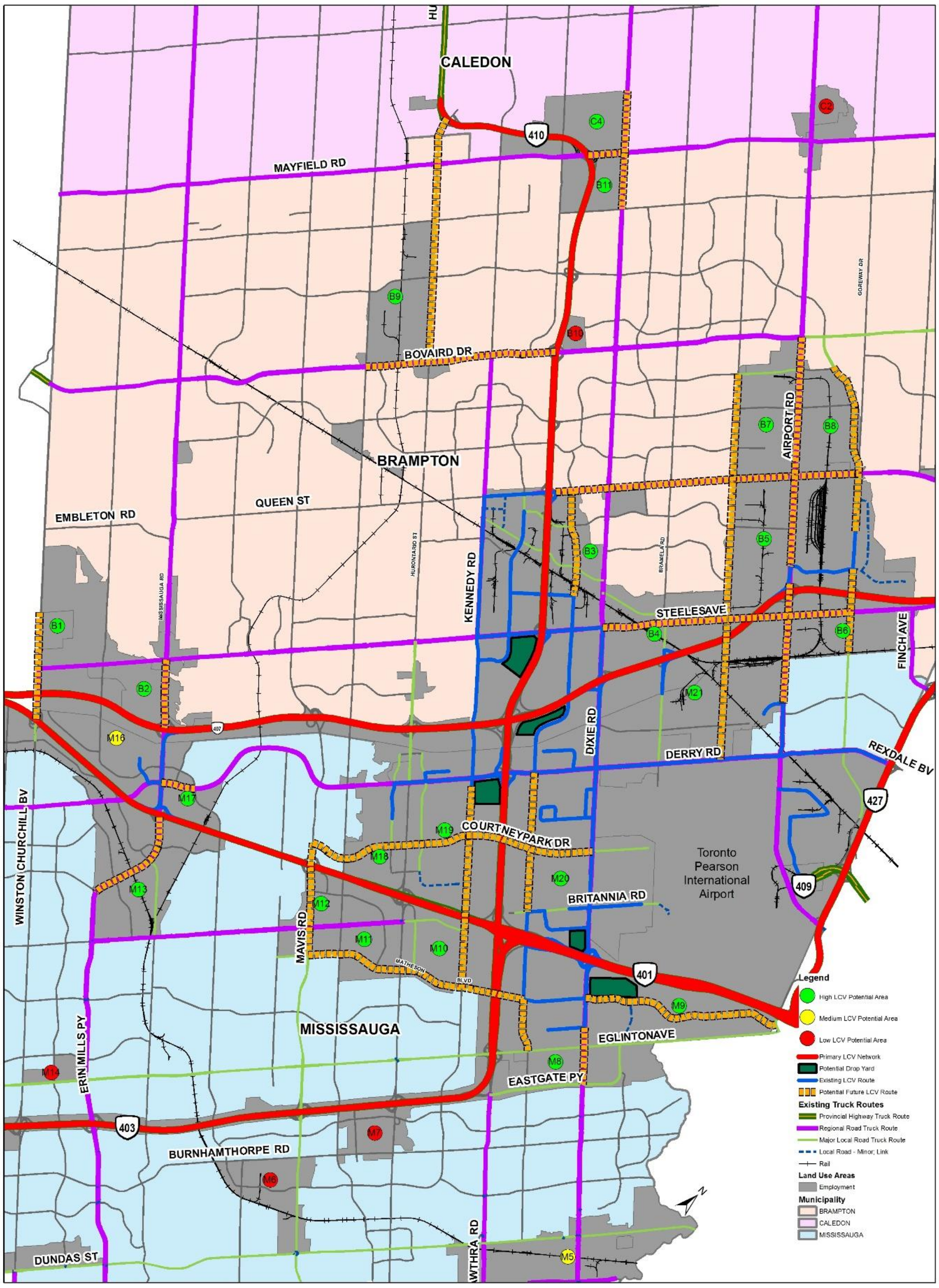
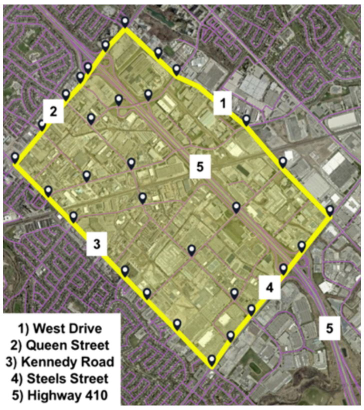


Figure 5- Existing and Potential LCV Routes
(Region of Peel, 2019)

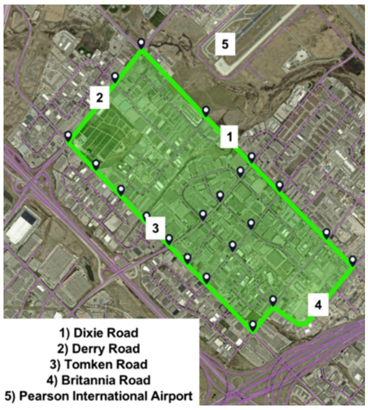
3.2. Potential Study Areas

Based on Figure 5, three potential networks within high-potential LCV areas have been selected for consideration. Figure 6 shows the location of each network in the region, as well as the location of each signalized intersection. Table 3 provides a comprehensive overview of each alternative zone, covering street boundaries, signalized intersections, truck route availability, and highway presence based on Figure 5. The count of signalized intersections both within and on the borders of each zone is particularly significant, as the primary objective of this thesis is to analyze signal priority and coordination methods within a network. Alternatives with a higher similarity between the number of border and interior intersections are more suitable as network-like alternatives.

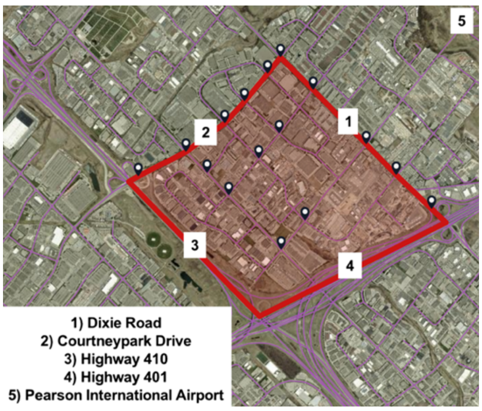
All three potential networks have significant existing and future LCV routes. However, additional factors help determine the final selection. Alternative 1 lies outside the core area bounded by Hwy. 401, Hwy. 410, Hwy. 407, and Toronto Pearson International Airport. The dense land use results in high congestion situations that are exacerbated by a large proportion of trucks. Alternatives 2 and 3 both fall within this key region. Another crucial consideration is the presence of highway interchanges, which increases the likelihood of LCV movements, since all major highways in Peel are designated LCV routes. Alternative 2 lacks any highway interchanges, making it less suitable. Additionally, using a smaller network helps minimize calibration errors and reduces computational complexities when implementing signal priority and coordination models. Consequently, Alternative 3 has been selected as the study area for the thesis.



Alternative 1



Alternative 2



Alternative 3

Figure 6- Alternative Study Area Locations

Table 3- Alternative Study Area Attributes

Alternative	Boundary	Signalized Intersections	Signalized Border Intersections	Existing LCV Routes (km)	Future LCV Routes (km)	Highway Interchanges in Zone
1	Queen St, Steels St, Kennedy Rd, West Rd	31	22	12.6	2.5	Yes; In the middle of the area
2	Dixie Rd, Britannia Rd, Tomken Rd, Derry Rd	21	16	9.6	4.4	No
3	Courtneypark Dr, Dixie Rd, Hwy 401, Hwy 410	16	10	4.5	3.2	Yes; two corners

The selected network, known as Alternative 3, is bounded by Highway 401 to the south, Dixie Road to the east, Courtneypark Drive to the north, and Highway 410 to the west. This network is strategically located near Pearson Airport and is predominantly an industrial area, encompassing numerous truck-related businesses such as transportation companies, truck repair shops, truck sales, and truck pooling facilities. Consequently, trucks constitute a significant portion of the traffic within this network. The network includes a total of 23 intersections, of which 16 are signalized and 7 are non-signalized. The intersection details are provided in Table 4, and a detailed satellite view of the study area is shown in Figure 7. The main streets and corridors in the network are as follows:

- **Dixie Road:** Dixie Road is one of the major north-south thoroughfares in the Region of Peel. Stretching approximately 38 km, it is one of the longest roads in Peel, beginning near Lake Ontario in the south and extending to the rural areas of Caledon in the north. This road intersects with major east-west arterials and highways, including Bovaird Drive, Queen Street, Steeles Avenue, Highway 401, and Highway 407.
- **Tomken Road:** Tomken Road is a north-south thoroughfare extending from Dundas Street to Highway 407 in the northern part of Mississauga. While it does not have interchanges with Highway 401 or Highway 407, it passes under or over these highways.
- **Britannia Road:** Britannia Road is divided by Highway 410 into two separate segments. The eastern segment is within the network, with one end terminating at Highway 410 and the other end within Pearson Airport. Additionally, one of the three GTA branches of the Flying J Travel Center, one of the largest trucks stop in North America, is located on this road within the network, making it a primary route for trucks.

- **Courtneypark Drive:** Courtneypark Drive is an east-west arterial road in the region that starts from Pearson Airport and connects several important roads and highways. It intersects with notable thoroughfares such as Hurontario Street, Mavis Road, and Kennedy Road, and provides access to Highway 410.
- **Meyerside Road:** Meyerside Road is an east-west local road in Mississauga, approximately 3 km in length, with most of its stretch falling within the case study area. The road is lined with industrial businesses, making it an essential corridor for trucks.
- **Highway 401:** Highway 401, one of the 400-series highways in Ontario, starts at the US-Canada border and extends to the Ontario-Quebec border. This highway is a strategic corridor that connects the economies of Ontario, Eastern Canada, and the United States. On average, 17,000 trucks commute on this highway daily, transporting goods valued at $615 million (Government of Ontario, 2021). Additionally, the segment passing through Toronto sees traffic volumes exceeding 450,000 vehicles per day, making it the busiest freeway on the continent (Hocking, 2023).
- **Highway 410:** Highway 410 is a north-south 400-series highway located entirely within the Region of Peel, starting from the north of Mississauga and ending in the north of Brampton. The highway features major interchanges with Highway 401 and Highway 407, with its northern end connecting to Highway 10 and its southern end connecting to Highway 403. The highest truck traffic on this highway is observed between the interchanges of Highway 401 and Courtneypark Drive (GTA West, 2009).

Table 4- Intersection Attributes

Intersection Number	Intersection	Signalization	Type	Major Street	Number of Thorough	Total Approach
1	Courtneypark Dr & Dixie Rd	Signalized	4 Leg	Dixie Rd	3 × 2	19
2	Courtneypark Dr & Ordan Dr	Signalized	4 Leg	Courtneypark Dr	2 × 1	8
3	Courtneypark Dr & Vipond Dr	Signalized	4 Leg	Courtneypark Dr	2 × 1	8
4	Courtneypark Dr & Shawson Dr	Signalized	4 Leg	Courtneypark Dr	2 × 1	10
5	Courtneypark Dr & Tomken Rd	Signalized	4 Leg	Tomken Rd	2 × 2	17
6	Courtneypark Dr & Hwy.410	Signalized	3 Leg	Courtneypark Dr	2	12
7	Shawson Dr & Tristar Dr	Non-Signalized	3 Leg	Shawson Dr	1	3
8	Tomken Rd & Tristar Dr	Signalized	3 Leg	Tomken Rd	2	9
9	Dixie Rd & Meyerside Dr	Signalized	4 Leg	Dixie Rd	3 × 2	16
10	Meyerside Dr & Ordan Dr	Non-Signalized	4 Leg	Meyerside Dr	1 × 1	4
11	Meyerside Dr & Vipond Dr	Signalized	4 Leg	Meyerside Dr	1 × 1	6
12	Meyerside Dr & Shawson Dr	Signalized	4 Leg	Meyerside Dr	1 × 1	8
13	Meyerside Dr & Tomken Rd	Signalized	4 Leg	Tomken Rd	2 × 1	12
14	Dixie Rd & Trinity Rd	Non-Signalized	3 Leg	Dixie Rd	3	8
15	Vipond Dr & Ordan Dr	Non-Signalized	3 Leg	Vipond Dr	1	3
16	Verbena Rd & Tomken Rd	Non-Signalized	3 Leg	Tomken Rd	2	5
17	Kestrel Rd & Verbena Rd	Non-Signalized	3 Leg	Kestrel Rd	1	3
18	Dixie Rd & Britannia Rd	Signalized	4 Leg	Dixie Rd	3 × 2	18
19	Britannia Rd & Vipond Dr	Non-Signalized	3 Leg	Britannia Rd	2	6
20	Shawson Dr & Britannia Rd	Signalized	4 Leg	Britannia Rd	2 × 1	10
21	Britannia Rd & Tomken Rd	Signalized	4 Leg	Tomken Rd	2 × 2	11
22	Dixie Rd & Shawson Dr	Signalized	4 Leg	Dixie Rd	3 × 1	15
23	Dixie Rd & Hwy. 401	Signalized	3 Leg	Dixie Rd	3	10

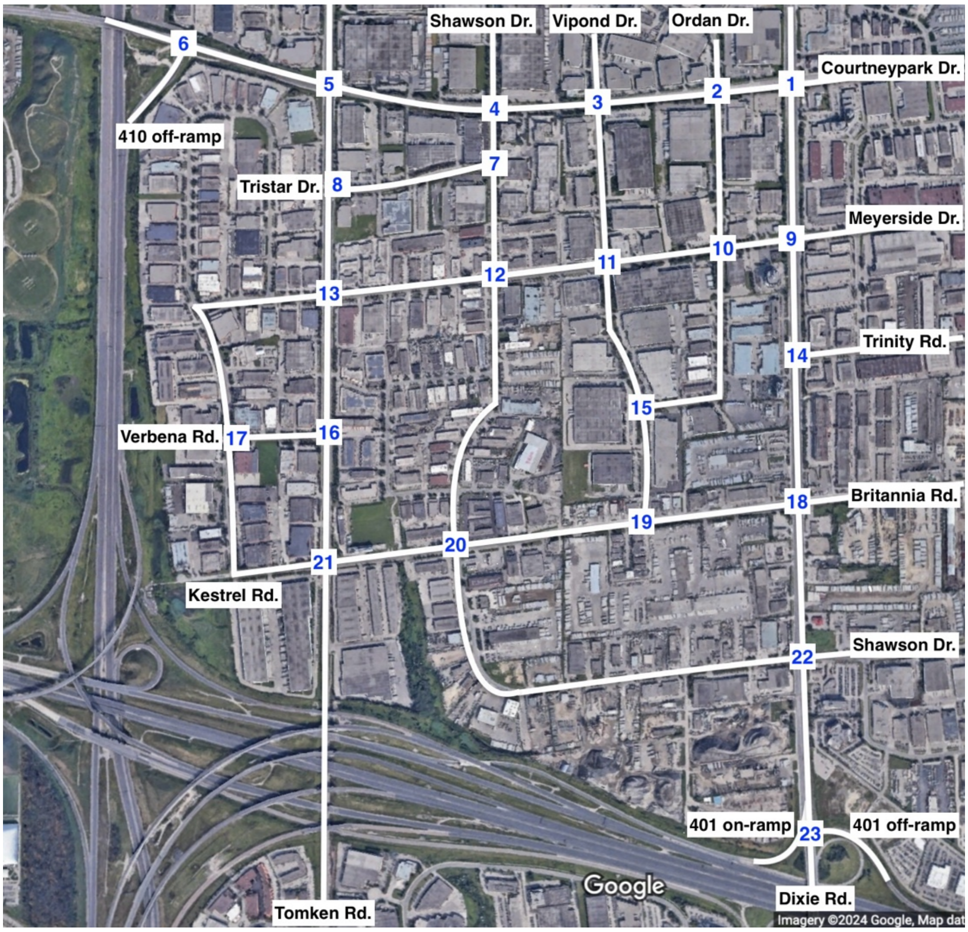


Figure 7- Satellite View of the Study Area

3.3. Traffic Volume & Signal Timing

The traffic volume of each intersection and the signal timing of signalized ones are essential for the microsimulation process of the network. These data were collected from three sources: the Region of Peel, the City of Mississauga, and a study conducted by Stantec for the City of Mississauga (Stantec, 2015). The data collection occurred in three steps:

1. Region of Peel: The first set of data was collected from the Region of Peel, which provided the signal timing and traffic volume data for all intersections along Dixie Road.
2. City of Mississauga: The second set of data was obtained from the City of Mississauga. This included the signal timing for all remaining signalized intersections within the network and the traffic volume data for most of these intersections.
3. Stantec Study: The third set of data was sourced from a study conducted by Stantec in 2015. This provided the signal timing and traffic volume data for Courtneypark Drive, filling in the remaining needed data for the microsimulation.

The detailed sources and data used in this study are presented in Table 5. The only non-signalized intersection for which data was provided is Dixie Rd. & Trinity Rd. (intersection 14); all other data pertain to signalized intersections. Although both traffic volumes and signal timing data were available from the Stantec (2015) study for all intersections along Courtneypark Drive, the data from the Region of Peel and the City of Mississauga were preferred because they are more recent. It is also noteworthy that the traffic volume data were collected on different dates, necessitating volume balancing before using them in the network.

The data from the Region of Peel has been received separately for traffic volumes and signal timing. As indicated in Table 5, the traffic volume data from Peel are the most recent compared to other sources and are considered the most reliable with minimal inconsistency (Appendix A: Volume Balancing). The traffic volume data for each intersection includes the volume of each movement of vehicles and pedestrians from each direction in 15-minute intervals. Additionally, specific data for morning, noon, and afternoon peak hours have been provided. This includes weather conditions, the percentage of heavy vehicles, light vehicles, single-unit trucks, buses, and articulated trucks for each movement, as well as the peak-hour factor (PHF) for each movement. Furthermore, a schematic representation of the peak-hour data on the intersection map is provided.

Table 5- Intersection Data Sources

Intersection Number	Intersection	Source	Traffic Volume Date	Signal Timing Date
1	Courtneypark Dr & Dixie Rd	Region of Peel	3/05/2022	3/11/2023
2	Courtneypark Dr & Ordan Dr	Stantec (Traffic Volume), Mississauga (Signal Timing)	12/11/2013	01/12/2023
3	Courtneypark Dr & Vipond Dr	Stantec (Traffic Volume), Mississauga (Signal Timing)	12/11/2013	01/12/2023
4	Courtneypark Dr & Shawson Dr	Stantec (Traffic Volume), Mississauga (Signal Timing)	12/11/2013	01/12/2023
5	Courtneypark Dr & Tomken Rd	Mississauga	13/09/2018	01/12/2023
6	Courtneypark Dr & Hwy.410	Stantec (Traffic Volume), Mississauga (Signal Timing)	12/11/2013	01/12/2023
7	Shawson Dr & Tristar Dr	N/A	N/A	N/A
8	Tomken Rd & Tristar Dr	Mississauga	06/04/2016	01/12/2023
9	Dixie Rd & Meyerside Dr	Region of Peel	3/05/2022	3/11/2023
10	Meyerside Dr & Ordan Dr	N/A	N/A	N/A
11	Meyerside Dr & Vipond Dr	Mississauga	20/04/2016	01/12/2023
12	Meyerside Dr & Shawson Dr	Mississauga	06/04/2016	01/12/2023
13	Meyerside Dr & Tomken Rd	Mississauga	29/11/2022	01/12/2023
14	Dixie Rd & Trinity Rd	Region of Peel	3/05/2022	N/A
15	Vipond Dr & Ordan Dr	N/A	N/A	N/A
16	Verbena Rd & Tomken Rd	N/A	N/A	N/A
17	Kestrel Rd & Verbena Rd	N/A	N/A	N/A
18	Dixie Rd & Britannia Rd	Region of Peel	3/05/2022	3/11/2023
19	Britannia Rd & Vipond Dr	N/A	N/A	N/A
20	Shawson Dr & Britannia Rd	Mississauga	23/03/2016	01/12/2023
21	Britannia Rd & Tomken Rd	Mississauga	29/11/2022	01/12/2023
22	Dixie Rd & Shawson Dr	Region of Peel	3/05/2022	3/11/2023
23	Dixie Rd & Hwy. 401	Region of Peel	16/06/2022	3/11/2023

The traffic signal timing parameters data provided by the Region of Peel include cycle length, offset time, and the split time of each phase for three different times: the AM peak, the PM peak, and the off-peak times. The offset time is used for the coordination process of signal controllers on major streets. Additionally, the data includes the amber time, all-red time, pedestrian minimum time, and vehicle minimum time for each phase.

The traffic volume data provided by the City of Mississauga spans a large time range, from 2016 to 2022. Similar to the data from the Region of Peel, this dataset includes the volume of each movement from each direction in 15-minute intervals. Additionally, the total number of heavy vehicles for each direction is specified for these time intervals. The daily traffic parameters for each intersection are also provided on the intersection schematics, which contain the traffic volume of each movement, the vehicle percentage for each movement, and the number of pedestrians. It is notable that pedestrian and heavy vehicle counts for each movement in each direction are not available for the time intervals; only daily data for them are provided.

The signal timing data from the City of Mississauga includes cycle time, offset time, and detailed information for each phase such as split time, pedestrian time, minimum green time, maximum green time, amber time, red clearance (all-red) time, and other parameters not utilized in this study. The data indicates that all intersections are presumed to operate with a ring-barrier controller system, as inferred from the ring assignments for each phase. Descriptions for each phase are provided to clarify which phases are enabled or need to be actuated. Vehicle presence on side streets can result in green times ranging between the minimum and maximum values, depending on demand.

The other data used in this study is from the Stantec report for Courtneypark Drive (Stantec, 2015). The traffic volume data from this report is utilized for four intersections. Although the Stantec study provided additional data, due to its outdated nature, preference was given to using more recent data from the Region of Peel and the City of Mississauga, except where no other data were available. The Stantec study offers a comprehensive dataset for each intersection, but only the peak-hour traffic volume data, including the number of cars, trucks, pedestrians, and cyclists for each movement in each direction, was used in this study. It is noteworthy that the volume data is from 2013, meaning it is over ten years old, and many intersections have since undergone changes. For example, the highway off-ramp at Courtneypark Dr E & Hwy 410 was reconfigured to merge with Courtneypark Dr from the southbound direction rather than the northbound direction during this period. These changes require a meticulous approach to volume balancing.

Based on the available data, all intersections in the network use actuated signal controllers configured in a NEMA dual-ring structure (Figure 2), except the Dixie Rd. & Hwy. 401 intersection (intersection 23), which operates on a fixed-time controller. The actuated intersections run in a semi-actuated mode, meaning the main street through phases do not require detectors, whereas left-turn phases and minor street phases do. On Dixie Road, all intersections also keep minor street through phases unactuated, while at other intersections of the network, detectors determine whether the minor street through phases are activated. Moreover, a green wave is maintained by assigning offset times to intersections along key corridors: Dixie Road, Courtneypark Drive, Tomken Road, and Meyerside Drive. These offsets differ for AM peak, PM peak, and off-peak periods to accommodate varying traffic demand patterns.

Table 6 summarizes total traffic volumes, truck volumes, and truck percentages at each intersection for both morning and afternoon peak hours. Although the data were collected from various sources and time periods and are therefore considered “unbalanced,” they still reveal key insights. Most intersections see higher traffic during the afternoon peak than in the morning. Moreover, trucks account for a sizable share of total traffic, averaging 19% in the AM peak and 15% in the PM peak. The busiest intersections by total volume are Courtneypark Dr & Tomken Rd (intersection 5) for both the AM and PM peaks, and Dixie Rd & Hwy 401 (intersection 23) for the PM peak. In terms of truck percentage, the highest values occur for the AM peak at Dixie Rd & Shawson Dr (intersection 22), Dixie Rd & Britannia Rd (intersection 18), and Shawson Dr & Britannia Rd (intersection 20).

Table 6- Unbalanced Traffic Volumes of Signalized Intersections

Intersection Number	Intersection	Total Approach Vehicles	Total Heavy Vehicles	Total Approach Vehicles	Total Heavy Vehicles
1	Courtneypark Dr & Dixie Rd	3092	676 (22%)	4079	649 (16%)
2	Courtneypark Dr & Ordan Dr	1258	202 (16%)	1456	150 (10%)
3	Courtneypark Dr & Vipond Dr	1355	175 (13%)	1526	133 (9%)
4	Courtneypark Dr & Shawson Dr	1509	186 (12%)	1727	176 (10%)
5	Courtneypark Dr & Tomken Rd	4224	577 (14%)	4770	538 (11%)
6	Courtneypark Dr & Hwy.410	2895	329 (11%)	2746	321 (12%)
8	Tomken Rd & Tristar Dr	2170	355 (16%)	2856	300 (11%)
9	Dixie Rd & Meyerside Dr	2652	569 (21%)	3395	535 (16%)
11	Meyerside Dr & Vipond Dr	685	127 (19%)	853	140 (16%)
12	Meyerside Dr & Shawson Dr	676	76 (11%)	1336	176 (13%)
13	Meyerside Dr & Tomken Rd	1896	313 (17%)	2249	296 (13%)
18	Dixie Rd & Britannia Rd	2963	763 (26%)	3646	681 (19%)
20	Shawson Dr & Britannia Rd	1120	271 (24%)	1367	243 (18%)
21	Britannia Rd & Tomken Rd	1647	274 (17%)	2271	247 (11%)
22	Dixie Rd & Shawson Dr	3366	915 (27%)	4132	763 (18%)
23	Dixie Rd & Hwy. 401	4085	790 (19%)	4251	824 (19%)

3.4. Field Data

Since the VISSIM microsimulation of the network is conducted in this research, calibration is essential to ensure the simulation closely resembles real-world conditions. Therefore, field data must be collected for both the calibration and validation processes. As the data are unbalanced and require a balancing procedure before being input into the microsimulation model, it is important to identify a suitable analytical period for the simulation and also gather field data for that specific time. Based on the available information, most intersections experience an AM peak hour from 8:00 to 9:00 a.m. and a PM peak hour from 4:00 to 5:00 p.m. However, as shown in Table *6*, the PM peak hour tends to be more congested than the AM peak hour. Given that this study focuses on a worst-case scenario, the PM peak hour is selected.

The field data collection served three primary objectives: (1) to calibrate and validate the simulation based on current network conditions; (2) to verify and refine existing signal timing data; and (3) to gather to enhance simulation accuracy. For calibration and validation purposes (Section 4.3), the maximum observed queue length for each approach at every intersection was recorded at fixed intervals during the PM peak hour (see Table 7). While the Washington State Department of Transportation (2014) “Protocol for VISSIM Simulation” suggests using the intersection’s cycle length or a default 120 seconds as the time interval to measure maximum queue length, this study employed 10-minute intervals to better capture queue fluctuations across multiple cycles and more accurately identify the peak queue length. Given the size of the network, data collection took approximately two weeks and was conducted in June 2024 under mostly sunny conditions.

In addition, signal timing parameters were documented at each intersection to ensure consistency with available records, especially where data were outdated. Information collected includes control type, phase sequence, cycle length, green times per phase, amber and all-red intervals. Further observations were made to improve simulation realism, focusing on operational characteristics such as adherence to priority rules at each approach and the turning behavior of trucks and LCVs, specifically, whether they encroach on adjacent lanes or exhibit different turning speeds.

Table 7- Maximum Queue Lengths (m), Measured at 10-Minute Intervals

Intersection	Intersection	Southbound	Northbound	Eastbound	Westbound
1	Courtneypark Dr & Dixie Rd	160	150	160	70
2	Courtneypark Dr & Ordan Dr	5	20	130	20
3	Courtneypark Dr & Vipond Dr	20	50	10	5
4	Courtneypark Dr & Shawson Dr	5	5	100	20
5	Courtneypark Dr & Tomken Rd	100	100	130	240
6	Courtneypark Dr & Hwy.410	N/A	100	110	350
8	Tomken Rd & Tristar Dr	30	40	N/A	20
9	Dixie Rd & Meyerside Dr	60	60	90	80
11	Meyerside Dr & Vipond Dr	10	10	20	30
12	Meyerside Dr & Shawson Dr	20	20	30	30
13	Meyerside Dr & Tomken Rd	190	180	50	80
18	Dixie Rd & Britannia Rd	220	150	30	70
20	Shawson Dr & Britannia Rd	40	20	50	20
21	Britannia Rd & Tomken Rd	170	160	70	250
22	Dixie Rd & Shawson Dr	400	110	20	70
23	Dixie Rd & Hwy. 401	10	60	N/A	300

Chapter 4. Methods of Analysis

4.1. Software and System

In this research, PTV VISSIM 2024 is used as the primary microsimulation tool to model the traffic network. The main objectives of the study, which are quantifying the impact of long combination vehicles (LCVs) and designing a new adaptive signal timing system, necessitate a platform with flexible vehicle definitions and online control capabilities. VISSIM meets these requirements by allowing the creation of custom vehicle types that accurately simulate LCVs and by supporting adaptive signal control through its COM (Component Object Model) interface, which enables real time manipulation of traffic signal settings. It also provides extensive calibration and validation features to ensure reliable results. The VISSIM license utilized in this project includes several add-ons such as 3D Graphics, Bing Maps, COM Interface, Emissions, Driving Simulator, Dynamic Assignments, External Driver Model, Managed Lanes, Meso Simulation, Public Transport, Street Traffic, and Test Model (Signal Control), along with signal controller features such as External, RBC Level 3, VAP, and Vissig.

VISSIM offers a user-friendly graphical user interface (GUI) for designing road network geometry and setting up simulations. While the GUI suffices for many tasks, there are scenarios, such as modifying VISSIM objects in real time during a simulation, where more flexibility is required. To address these advanced needs, VISSIM includes an additional interface based on the COM, a technology that enables interprocess communication. By leveraging the COM interface, users can automate and customize VISSIM simulations programmatically, opening the door to a wide range of integrated and dynamic simulation possibilities.

All simulations are conducted on Windows 11 (version 24H2), running on a machine with an Intel Core i7 14700F processor at 2.1 GHz and 32 GB of RAM. This hardware configuration provides the computational power required to handle multiple complex peak hour simulations and to perform real time modifications to signal control settings without encountering significant performance bottlenecks.

4.2. VISSIM Inputs

4.2.1. Network and Traffic Setup

The first step of modeling in VISSIM involves coding the streets and thoroughfares to reflect the real network. VISSIM employs a link-connector structure for this purpose. Each link represents a road segment with a constant number of lanes, constant width, and consistent lane markings, necessitating the use of multiple links to fully develop the network. However, traffic flow will not move from one link to another without a connection between them. Connectors are created to establish these connections between the links within the network. Figure *8* presents the final model of the selected network in VISSIM, where blue lanes indicate the links, and pink lanes represent the connectors.

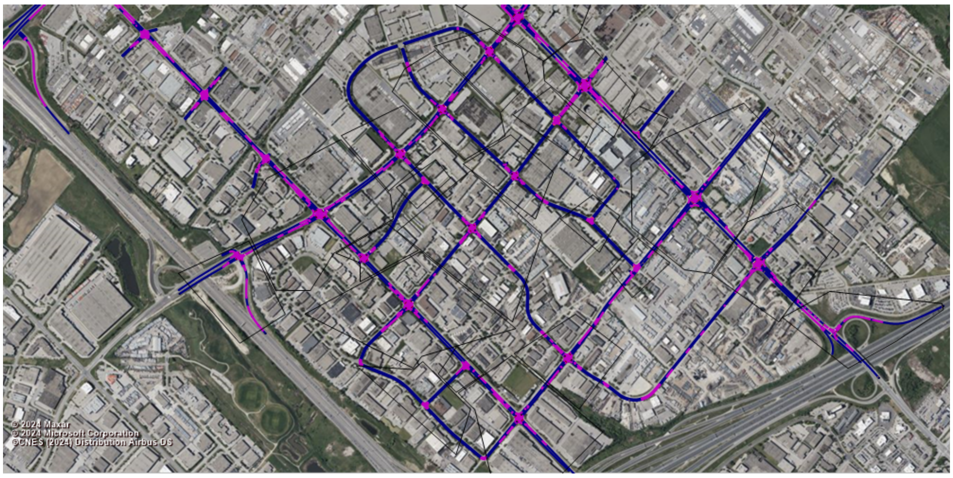


Figure 8- VISSIM Link-Connector Structure

The accuracy of network coding in VISSIM directly influences the validity of simulation outcomes. Bing Maps is utilized as the default background in the software to position links and connectors in their actual locations and widths. When lane markings are not clearly visible on the default map, Google Earth Pro is employed to measure the width and precise placement of each link. For instance, while the intersection of Dixie Rd & Britannia Rd. (intersection 18) is shown under construction in Bing Maps, detailed information about the intersection can be retrieved using Google Earth Pro (Figure *9*). Finally, the field data is used to verify the information collected from the satellite maps.

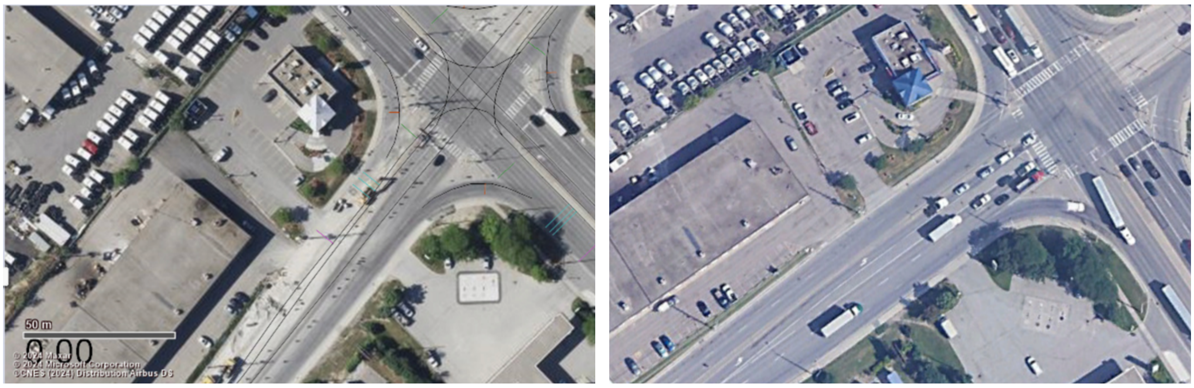


Figure 9- VISSIM Bing Map (Left Picture) vs. Google Earth Pro (Right Picture)

In the next step of coding the network structure, the desired speed for each link, priority rules, and stop signs are defined in the simulation based on data captured from Google Maps street view and field observations. A default desired speed of 60 km/h was applied to most links; exceptions reflected posted speeds observed on site: Dixie Road (70 km/h), the westbound lanes of Courtneypark Drive (70 km/h), and the Highway 410 off-ramp (50 km/h). Priority rules and stop signs are placed in their corresponding locations. Priority rules are primarily established for right turns at some of the busier intersections. Field data shows that vehicles following priority rule yield the right-of-way to the first two lanes of opposing traffic, often disregarding a potential third lane, if present.

To accurately model turning movements, reduced speed areas are defined at the sharpest points of the turns. According to guidelines from the Washington Department of Transportation (2020), these reduced speed areas are set to 3 meters in length. For right turns, the speed is set at 5 km/h for LCVs and heavy vehicles, and 15 km/h for other vehicles. For left turns, the speeds are set at 5, 15, and 25 km/h for LCVs, heavy vehicles, and other vehicles, respectively. Although the original study does not specifically address LCVs, these values are adjusted based on field observations and the capabilities of the simulation software.

The next stage of network model development involves defining the conflict areas that persist even after signal controllers are implemented. For both controlled and uncontrolled intersections, right-of-way rules are established according to Ontario’s driving regulations (Government of Ontario, 2022), ensuring that the simulation accurately reflects real-world traffic behavior.

Vehicle routing decisions are another critical component in coding the network within VISSIM. In this project, each link approaching to an intersection has a decision point, along with corresponding destination points for every possible route. Each decision–destination pair is considered a route or movement. When a decision point has multiple destinations, relative flow proportions for each route or movement must be specified. The next step involves defining vehicle compositions and vehicle inputs within the network. The vehicle compositions in this simulation include the following categories: Car, HGV (Heavy Goods Vehicle), Bus, and LCV. Unlike routing decisions, it is not feasible to define the relative flow of these vehicles at each intersection in the network. Instead, the relative flows are identified through vehicle inputs. Vehicle inputs are defined at the open-ended sides of the links, such as highway off-ramps or at the southern and northern ends of Dixie Road. Vehicle inputs are specified for each link in one-hour intervals, corresponding to the afternoon peak hour in this study. The values for flow proportions and vehicle inputs are derived from the volume balancing process.

As presented in Table *5*, the traffic volume data used in this study is obtained from three different sources at various times, increasing the likelihood of discrepancies between datasets. For example, given Figure *10*, the combined westbound left (WBL), southbound through (SBT), and eastbound right (EBR) movements at intersection 1 do not match the southbound (SB) volume at intersection 2. Volume balancing is conducted to eliminate these discrepancies between adjacent intersections while ensuring that the final adjusted traffic volumes deviate as little as possible from their original values.

The volume balancing process began by quantifying the errors in each intersection's data and removing entries that exhibited significant discrepancies with adjacent intersections. In the next step, an ideal (adjusted) table was developed, presenting detailed traffic volumes for all intersections. This adjusted table contained no inconsistencies between data points. The objective was to minimize the difference between the adjusted values and their corresponding original values. Optimization was performed using Excel's GRG Nonlinear Solver to achieve this goal. The resulting adjusted table was then directly used to define flow proportions and vehicle input volumes (Appendix A: Volume Balancing).

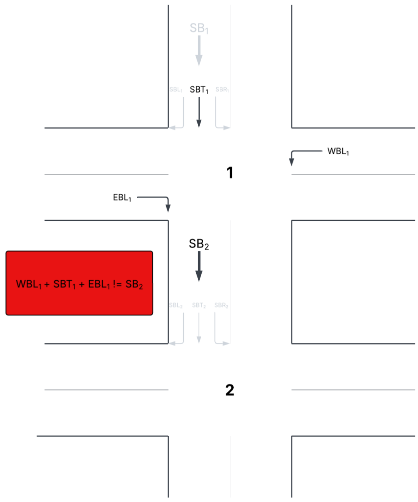


Figure 10- Example of Balancing Two Adjacent Intersections

Defining traffic signal controls is another crucial step in developing the simulation model. All intersections in the network employ actuated controllers except Dixie Rd. & Hwy. 401 (Intersection 23), which uses a fixed-time controller. VISSIM supports both fixed-time and actuated strategies, and the actuated controllers in this study follow the Ring-Barrier configuration Controller (see Figure *2*). In defining the controllers, most parameters are adopted directly from the data sources, except in a few cases where certain values (e.g., minimum green time) differ significantly from field observations. The common parameters in defining both controllers are cycle time, offset time, split times and clearance times for each phase. All intersections share a 160-second cycle time. These parameters are fixed in advance and do not change during the simulation.

Defining a fixed-time controller in VISSIM is relatively straightforward: the cycle length, offset, and split times are simply entered into the platform's built-in fixed-time controller module. In contrast, implementing the Ring-Barrier Controller (RBC) first requires specifying the phase sequence in each ring and group (Figure *2*). After choosing a network-wide cycle length, setting offsets, and assigning split times, users must also define minimum and maximum green intervals for phases that need to coordinate with other intersections. As shown in Figure *3*, actual green times for these phases vary between the predefined minimum and maximum limits, while other phases follow their assigned split times. Additionally, each phase to be activated by the RBC must have a corresponding detector, typically modeled in VISSIM as a 5-meter segment directly below the signal heads.

4.2.2. Long Combination Vehicles Integration

Modeling Long Combination Vehicles (LCVs) in VISSIM involved a multi-step process to ensure that the simulated vehicles accurately reflect the characteristics of LCVs operating in Ontario. The process began with the creation of a 2D/3D model of the LCVs within the software, followed by the definition of functions and distributions for the new LCV type. Finally, all the new parameters were assigned to a new vehicle class labeled "LCV."

As the first step, an LCV model was developed using VISSIM's built-in vehicle modeling tools. Given the limitations of the available vehicle files within VISSIM, the model was based on the closest approximation to the LCV A-train double, as defined by the Ontario Ministry of Transportation (MTO) in Table 2. The only deviation from the A-train standard is the converter dolly: VISSIM represents it as a single-axle unit, whereas the MTO specification calls for a tandem-axle dolly. The model was created by combining the "HGV - US AASHTO WB-65 Tractor," "HGV - US AASHTO WB-50 Trailer," and "HGV - US AASHTO WB-50 Trailer" files within the 2D/3D Model Segments of the software. Subsequent adjustments were made to the width and length parameters to ensure that the final vehicle aligned with MTO standards, including the dimensions of the trailer, converter, and the distances between axles in both trailers and tractors. However, the model differed from the standard in one aspect: the converter dolly in VISSIM is a single axle, whereas the MTO standard specifies a tandem axle for the A-train double. This model is depicted in Figure *11*.

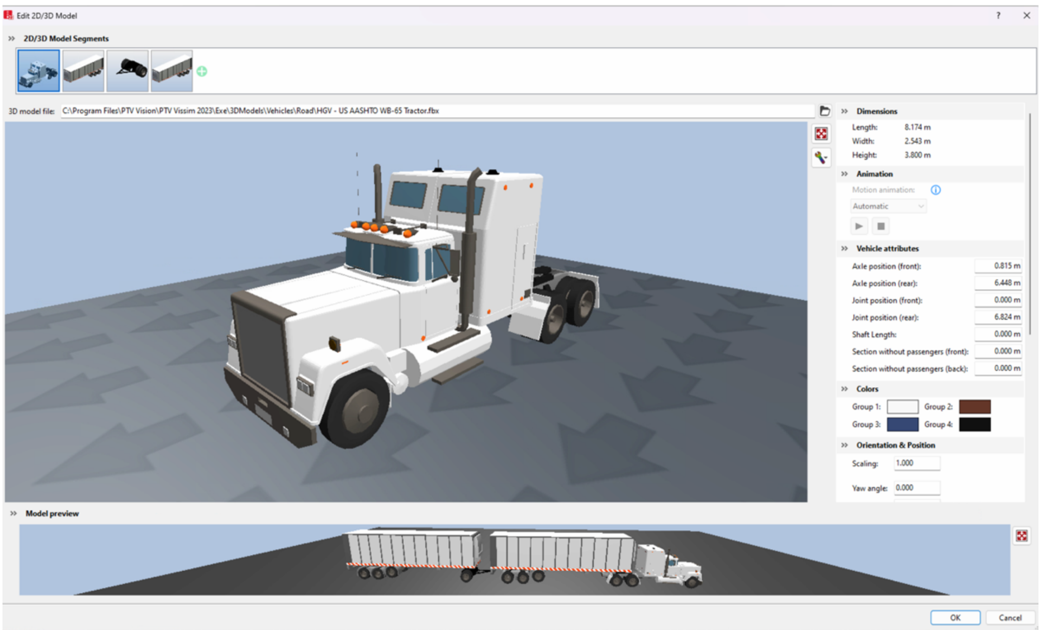


Figure 11- Long Combination Vehicle Modeled in VISSIM

Additionally, the functions and distributions for the new LCV type were defined based on a study conducted by Wisconsin (2020), which provided desired acceleration functions, power, weight, and speed distributions for various types of trucks. Due to the lack of specific function and distribution values for LCVs, the parameters defined for tractor-trailers of Wisconsin were used in this study, as they are the closest vehicle type to LCVs. Some parameters required adjustments to align with values proposed by the MTO (Government of Ontario, 2023). For instance, the power distribution needed modification since the MTO specifies a minimum engine power of 316 kW, whereas the Wisconsin study's minimum was 224 kW. To reconcile this discrepancy, all power values were scaled accordingly. Similarly, the weight distribution was adjusted to match the MTO’s maximum LCV weight of 63,500 kg. The desired acceleration functions, along with the adjusted power and weight distribution values, are presented in Appendix B: LCVs Parameters in VISSIM.

4.2.3. Assumptions

Every simulation includes a set of assumptions that differentiate it from the real-world scenario, yet these differences are structured to minimally impact the final study results. The main assumptions made in this simulation are as follows:

- No vehicles are modeled as entering or exiting the network from businesses located within the study area. The selected network comprises large companies, gas stations, and truck-related businesses, which typically experience significant vehicular traffic daily. However, due to the need for simplification and the unavailability of specific data for these businesses, their traffic has been excluded from the simulation.
- Although buses are included as one of the four vehicle types in the traffic composition, bus stops and the processes of loading and unloading passengers are not modeled in the simulation. Roads such as Tomken Road, Dixie Road, Courtneypark Drive, and Meyerside Drive do have bus stops in reality.
- The parameters predefined by VISSIM, including functions and distributions for cars, buses, and HGVs, as well as traffic car-following and lane-changing behaviors, have been left at their default settings. In reality, each network has specific conditions that would require more detailed field data analysis to appropriately modify these parameters.

- The simulation is run for the traffic volume corresponding to one and half hour of the afternoon peak hour, with results captured between 300 seconds and 5,700 seconds of the simulation runtime. In reality, some traffic volume exists in the network before the peak hour begins. To better reflect actual conditions, data capturing starts after the first 300 seconds of the simulation.

4.3. Calibration

To ensure that the simulation mimics real-world conditions as accurately as possible, this study employed a three-step calibration and validation process. First, the base model, or pre-calibration model, is validated by comparing observed queue lengths, identifying any intersections requiring further refinement. Next, simulation parameters are adjusted and traffic volumes in selected areas of the network are slightly modified. Finally, the revised model is validated by comparing queue lengths, travel times, traffic flow, truck flow, and GEH values to ensure accuracy.

The validation of queue length, traffic flow, truck flow, and GEH values in this study follows the criteria adapted from the Federal Highway Administration’s Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software (2004). Because the data available for this project was collected at the intersection level, the original guidelines were modified accordingly (see Table 8). The GEH value for an intersection i was calculated using Equation (4.1):

$$GEH_i = \sqrt{\frac{2(M_i - O_i)^2}{M_i + O_i}} \tag{4.1}$$

where M_i and O_i are the modeled and observed volume of intersection i, respectively. As summarized in Table 8, the absolute difference between the observed and modeled traffic flow at each intersection should be within 100 vehicles per hour (vph) for observed flows below 700 vph, within 15% for flows between 700 and 2,700 vph, and within 400 vph for flows above 2,700 vph. Additionally, the GEH value for individual intersection flows should be less than 5, meeting this criterion in at least 85% of intersections. For total network flow, the absolute difference between observed and modeled values should be within 5%, and the GEH value should be below 4. Finally, the absolute difference in queue lengths for each movement at each intersection should be within 20% or not exceed 12 vehicles (Dowling, et al., 2004).

Table 8- Validation Criteria
Adapted from FHWA *(2004)*

Criteria	Acceptance Target
Hourly Intersection Flows	
Flows below 700 vph	Within ±100 vph in >85% of cases
Flows between 700–2,700 vph	Within ±15% in >85% of cases
Flows above 2,700 vph	Within ±400 vph in >85% of cases
Sum of all Flows	
Total intersection flow	Within ±5%
GEH Statistics	
Intersection flows	GEH < 5 in >85% of cases
Total intersection flow	GEH < 4
Queue Length	
Queue length of movements	Within ±20% or ≤12 vehicles in

In addition to evaluating traffic flows and queues, the Federal Highway Administration (FHWA) (2004) recommends that the difference between modeled and observed travel times remain within 15% or less than one minute. For interrupted flows (e.g., signalized intersections), however, the Washington State Department of Transportation (WSDOT) (2014) provides a more precise measure. WSDOT's formula for allowable travel time variation Δ (in seconds) is given by:

$$\Delta = \frac{1}{\frac{1}{t} - \frac{528 \times S}{3600 \times L}} - t \tag{4.2}$$

where t is the real-world travel time (seconds), L is the length of the journey (feet), and S is the average free-flow speed (mph).

In the first step of the calibration process, maximum queue lengths for each movement at each intersection (as derived from the simulation) were compared against the field-observed values presented in Table 7. Simulation data were collected over a one-hour period, from 900 to 4,500 seconds of simulation time, and averaged across five runs with distinct random seed values. In contrast, field data were gathered at 10-minute intervals for each movement. Table *9* summarizes the simulation-derived maximum queue lengths along with both the percentage and absolute differences (in number of vehicles) relative to the observed values.

Table 9- Pre-Calibration Maximum Queue Lengths (m) vs. Observed

Intersection Number	Intersection	Southbound			Northbound			Eastbound			Westbound		
		Simulation	Diff (%)	Diff (Veh)	Simulation	Diff (%)	Diff (Veh)	Simulation	Diff (%)	Diff (Veh)	Simulation	Diff (%)	Diff (Veh)
1	Courtneypark Dr & Dixie Rd	143	-11	-17	172	15	22	139	-13	-21	**100**	**43**	**30**
2	Courtneypark Dr & Ordan Dr	**71**	**1320**	**66**	20	-2	0	120	-8	-10	**114**	**472**	**94**
3	Courtneypark Dr & Vipond Dr	**87**	**337**	**67**	40	-20	-10	**117**	**1067**	**107**	**118**	**2266**	**113**
4	Courtneypark Dr & Shawson Dr	**66**	**1215**	**61**	**85**	**1598**	**80**	**141**	**41**	**41**	**167**	**734**	**147**
5	Courtneypark Dr & Tomken Rd	**132**	**32**	**32**	115	15	15	121	-7	-9	289	20	49
6	Courtneypark Dr & Hwy.410		N/A		**136**	**36**	**36**	**81**	**-26**	**-29**	308	-12	-42
8	Tomken Rd & Tristar Dr	57	91	27	**63**	**58**	**23**		N/A		19	-4	-1
9	Dixie Rd & Meyerside Dr	72	20	12	**113**	**88**	**53**	**56**	**-38**	**-34**	**114**	**43**	**34**
11	Meyerside Dr & Vipond Dr	**68**	**576**	**58**	**29**	**187**	**19**	**41**	**106**	**21**	**51**	**70**	**21**
12	Meyerside Dr & Shawson Dr	**73**	**265**	**53**	**70**	**249**	**50**	20	-35	-10	**114**	**279**	**84**
13	Meyerside Dr & Tomken Rd	151	-20	-39	163	-10	-17	**37**	**-26**	**-13**	**101**	**26**	**21**
18	Dixie Rd & Britannia Rd	**154**	**-30**	**-66**	177	18	27	**46**	**54**	**16**	**97**	**39**	**27**
20	Shawson Dr & Britannia Rd	46	16	6	**58**	**188**	**38**	61	21	11	**78**	**288**	**58**
21	Britannia Rd & Tomken Rd	164	-3	-6	137	-15	-23	79	13	9	**144**	**-42**	**-106**
22	Dixie Rd & Shawson Dr	**205**	**-49**	**-195**	**167**	**52**	**57**	**91**	**354**	**71**	**270**	**285**	**200**
23	Dixie Rd & Hwy. 401	**211**	**2010**	**201**	**93**	**55**	**33**		N/A		344	15	44

The pre-calibrated data in Table *9* indicate that only 35% of the movements meet the queue length criteria specified in Table 8. This falls short of the target acceptance rate of 85%. One contributing factor is that the field data record maximum queue lengths in 10-minute intervals, whereas the simulation data were collected over a one-hour period. This discrepancy in data collection windows helps explain some of the differences, as the longer simulation interval may capture higher or more variable queue lengths. For instance, Table *9* shows that the simulated queue lengths are often higher than the observed values. However, there are also cases where the one-hour simulated queue is lower than the corresponding 10-minute observed queue, and in some instances the differences exceed 20% or 12 vehicles (e.g., southbound movements at intersections 13, 18, and 22). Consequently, the calibration process should focus on reducing these discrepancies in the highlighted intersections while also seeking to improve overall alignment with the specified acceptance criteria across the network (Appendix C: Calibration Process).

4.4. Adaptive Traffic Signal Control

4.4.1. Reinforcement Learning Background

Reinforcement Learning (RL) is a branch of machine learning that studies how an agent should take actions in an environment to maximize cumulative reward. Unlike supervised learning, which depends on static datasets, RL agents learn through dynamic interactions with their environment, receiving feedback in the form of rewards or penalties. This interactive process allows the agent to iteratively refine its decision-making strategy based on experience.

Four defining characteristics distinguish RL from other areas of artificial intelligence and machine learning: optimization, delayed consequences, exploration, and generalization. Optimization refers to the goal of finding an optimal policy, a decision-making strategy, that maximizes the agent's long-term reward. Unlike in purely reactive systems, RL must account for delayed consequences, where current actions influence future outcomes. This presents two major challenges: strategies must balance immediate and long-term effects, and it can be difficult to determine which specific action led to a particular reward or penalty when the outcome occurs much later.

Exploration is another key aspect, requiring the agent to actively investigate its environment to discover effective actions, rather than relying exclusively on prior experiences. This is essential for avoiding suboptimal behavior that may arise from incomplete knowledge. Finally, generalization ensures that the policy learned from past experiences can be applied effectively to new or previously unseen states, enabling robust performance across a wide variety of situations.

The fundamental framework of sequential decision making in RL is an interactive, closed-loop process (see Figure *4*). At each time step t, the agent selects an action a_t to execute in the environment and, in response, receives an observation o_t and a reward r_t. The goal is to choose actions that maximize the total expected future reward. Formally, the history h_t at time step t is defined as the sequence of all past actions, observations, and rewards:

$$h_t = (a_1, o_1, r_1, \dots, a_t, o_t, r_t) \qquad 4.3)$$

State refers to the information the agent uses to decide which action to take. Because the environment may contain additional data that is not visible to the agent, an agent state at time step t, denoted s_t, is often defined as a function of the entire history (Lattimore, et al., 2020):

$$s_t = f(h_t) \quad 4.4)$$

A reinforcement learning agent typically incorporates one or more of the following components to enable an intelligent decision-making process: a policy, a value function, and a model of the environment. The policy dictates the actions taken in each state, while the value function estimates the expected cumulative reward from each state (or state-action pair), guiding the agent toward decisions that yield long-term benefits. In many cases, agents may also incorporate a model that predicts state transitions and rewards; however, defining accurate transition dynamics can be challenging, particularly in domains such as traffic control, so model-free approaches are often employed.

In more detail, the policy π is a core function that guides the agent's behavior by mapping states to actions. Under a deterministic policy, we write $\pi(s) = a$, meaning that for every state s, a specific action a is chosen. Alternatively, a stochastic policy defines a probability distribution over actions, expressed as $\pi(a|s) = P[A_t = a|S_t = s]$, where the agent selects actions according to these probabilities.

The state-value function V^π measures the expected discounted sum of future rewards when following a particular policy π. Formally, for a state s:

$$V^\pi(s_t = s) = E_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots |s_t = s] \quad (4.5)$$

where $\gamma \in [0,1]$ is the discount factor that balances immediate and future rewards. If $\gamma = 0$, only the immediate reward is considered; if $\gamma = 1$, future rewards are valued as highly as immediate rewards. In addition to the state-value function, the action-value function (or Q-function) $Q^\pi\ (s, a)$ estimates the expected discounted sum of future rewards starting from state s, taking action a, and thereafter following policy π. Formally:

$$Q^{\pi}(s,a) = E_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | s_t = s, a_t = a] \quad (4.6)$$

Balancing exploration and exploitation are a central challenge in reinforcement learning. Exploitation leverages the agent's current knowledge to select actions believed to yield the highest reward, whereas exploration involves trying new or uncertain actions to discover potentially better strategies. A common approach to manage this trade-off is the epsilon-greedy strategy, where the agent chooses a random action with probability ϵ (exploration) and selects the action that maximizes the estimated value function with probability $1-\epsilon$ (exploitation). More advanced techniques, such as Upper Confidence Bound (UCB) or Thompson Sampling, adjust the balance dynamically by considering the uncertainty in action-value estimates. This careful balancing act is essential to avoid local optima and to ensure that the agent gathers sufficient information about its environment to develop a robust, high-performing policy.

4.4.2. Markov Decision Process

Real-world environments can exhibit either deterministic or stochastic behavior. In a deterministic setting, taking a particular action in a given state always leads to the same next state. However, in most real-world scenarios, such as traffic signal control, performing a particular action can result in multiple possible outcomes, making the environment stochastic. Among stochastic environments, a Markov process is one that satisfies the Markov property, meaning the probability of transitioning to the next state depends only on the current state and not on the full history of past states. Formally:

$$P[s_{t+1}|s_t] = P[s_{t+1}|h_t] \quad (4.7)$$

In principle, the Markov property can always be enforced by defining the state s_t to include all historical information h_t. However, tracking the full history is often impractical due to computational complexity. Instead, many practical applications assume that the latest observation provides sufficient information to make decisions, an assumption known as full observability. In such cases, the state can be approximated by setting:

$$s_t = o_t \quad (4.8)$$

A Markov Decision Process (MDP) extends a Markov process to include decision-making by an agent in a stochastic environment. Formally, an MDP can be defined by the 5-tuple (S, A, P, R, γ):

1. S: The set of all possible states.
2. A: The set of actions the agent can choose from.
3. $P[s_{t+1}|s_t, a_t]$: The transition probability of moving from state s_t to state s_{t+1} after taking action a_t.
4. $R(s_t, a_t)$: The reward function, which specifies the immediate reward received after transitioning from s_t via a_t.
5. $\gamma \in [0,1]$: The discount factor, balancing the importance of immediate versus future rewards.

The Markov structure allows the value function (Equation (4.5)) to be decomposed into the immediate reward plus the discounted sum of future rewards, known as the Bellman expectation equation. Given a policy π (i.e., a mapping from states to actions), the state-value function is:

$$V^\pi(s) = E_\pi[R(s_t, a_t) + \gamma V^\pi(s_{t+1})|\, s_t = s] \quad (4.9)$$

where the expectation is over both the stochastic environment transitions and the policy's action choices. Similarly, the action-value function (Equation (4.6)) would be:

$$Q^\pi(s,a) = E_\pi[R(s_t, a_t) + \gamma Q^\pi(s_{t+1}, a_{t+1})|\, s_t = s, a_t = a] \quad 4.10)$$

From Equations ((4.9) and Equation (4.10), the relationship between the state-value function and the action-value function is:

$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^\pi(s') \quad 4.11)$$

where s' is the next state, drawn according to the transition probability $P(s'|s,a)$.

The optimal value function represents the best achievable performance in the MDP. Once it is found, the MDP is considered solved. The optimal state-value and action-value functions, $V^*(s)$ and $Q^*(s,a)$, are the maximum values of $V^\pi(s)$ and $Q^\pi(s,a)$ over all possible policies:

$$V^*(s) = \max_\pi V^\pi(s); \quad Q^*(s,a) = \max_\pi Q^\pi(s,a) \tag{4.12}$$

From Equation (4.12), choosing the action that maximizes $Q^*(s,a)$ yields the optimal state-value function:

$$V^*(s) = \max_a Q^*(s,a) \qquad 4.13)$$

Combining Equations (4.11) and Equation (4.12) gives the initial Bellman optimality equation:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s\prime \in S} P(s'|s,a) V^*(s') \qquad 4.14)$$

Next, by substituting Equation (4.13) into Equation (4.14), we obtain the final form of the Bellman optimality equation:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s\prime \in S} P(s'|s,a) \max_{a\prime} Q^*(s',a') \qquad 4.15)$$

Because the Bellman optimality equation is non-linear, it generally cannot be solved in closed form. Instead, iterative methods, such as Value Iteration, Policy Iteration, Q-learning, and SARSA, are employed to approximate or converge to Q^* and V^*.

4.4.3. Deep Q-Learning Algorithm

Q-learning is a model-free, off-policy reinforcement learning algorithm designed to learn the optimal action-value function directly from interactions with the environment. It does not require a model of the environment's dynamics, making it well-suited for complex or unknown settings. By iteratively updating Q-values based on observed rewards and the estimated future rewards, as derived from the Bellman optimality equation, Q-learning gradually converges to the optimal policy.

To implement the Q-learning algorithm, a tuple (s, a, r, s') is employed, where s and a are the current state and action, r is the reward obtained from taking action a in state s, and s' is the next observed state. To evaluate and control the action-value function, Q-learning requires both the current estimate of $Q(s, a)$ and the *target* derived from the Bellman optimality equation (Equation (4.15)). Since the next state s' is observed, the Bellman equation simplifies to:

$$Q(s,a) = R(s,a) + \gamma \max_{a'} Q(s', a') \quad 4.16)$$

The iterative update rule of the Q-function is defined as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R(s,a) + \gamma \max_{a'} Q(s', a') - Q(s,a) \right] \quad 4.17)$$

where α is the learning rate that controls the size of the update step. A higher α can lead to faster learning but may cause instability, while a lower α results in more gradual, stable convergence.

As the state-action space grows larger or continuous, explicitly storing and updating all Q-values becomes impractical. Therefore, function approximation methods are used to generalize and predict Q-values across states and actions based on historical experiences. A common approach involves parameterized approximation functions, such as neural networks, decision trees, linear regressions, or nearest neighbor methods, to represent these value functions:

$$\hat{V}(s; \theta) \approx V^{\pi}(s); \quad \hat{Q}(s, a; \theta) \approx Q^{\pi}(s, a) \quad 4.18)$$

where θ denotes the parameters (e.g., weights in a neural network). This approach allows the agent to generalize from a finite set of observed transitions to unseen states and actions, making Q-learning feasible in high-dimensional or continuous domains.

In practice, differentiable function approximators, such as linear models or neural networks, are particularly popular because they can be optimized efficiently using stochastic gradient descent (SGD) or related methods. Particularly Deep Neural Networks (DNNs) provide an efficient and powerful framework for approximating these value functions. Mathematically, a DNN approximation can be expressed as a composite function of the form:

$$y = h_n\big(h_{n-1}(\dots h_1(\vec{x}) \dots)\big) \qquad 4.19)$$

where y represents the value function ($V(s)$ or $Q(s,a)$), $\vec{x}$ is the features vector, and $h_1, \dots, h_n$ are successive transformations (layers). Each transformation can be written as:

$$h_i = f_i(w_i h_{i-1} + b_i), \quad i = 1, \dots, n \qquad (4.20)$$

where w_i as the weight matrix, b_i the bias vector, and f_i a linear or nonlinear activation function (e.g., ReLU, sigmoid, tanh). Figure *12* depicts this layered structure, typically consisting of an input layer, multiple hidden layers, and an output layer. DNNs are considered universal function approximators, meaning they can approximate a wide range of functions given sufficient depth and number of nodes, often with fewer parameters than shallow architectures.

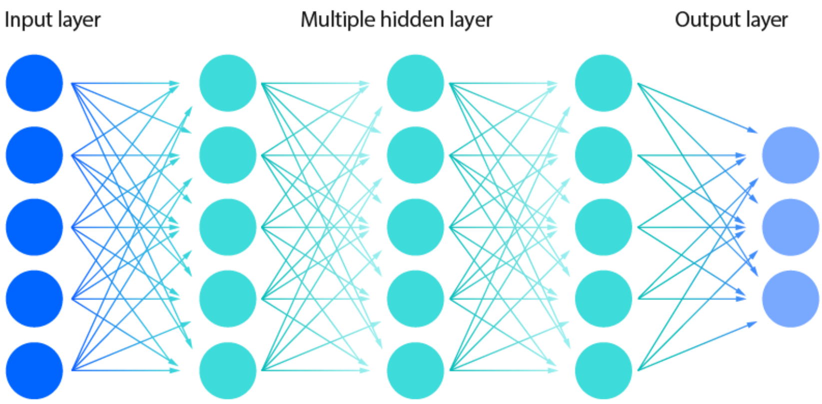


Figure 12- Deep Neural Network Configuration

Training such networks typically involves minimizing a loss function, commonly the mean squared error (MSE), which measures the discrepancy between the predicted values $\hat{Q}(s,a;\theta)$, and target values derived from experience tuples. The loss function can be formulated as:

$$J(\theta) = \left[Q(s,a) - \hat{Q}(s,a;\theta)\right]^2 \tag{4.21}$$

where $Q(s,a)$ is the target value. To minimize this loss, parameters θ are updated via Stochastic Gradient Descent (SGD). Specifically, the gradient of the loss with respect to parameters is computed, and then the parameters are adjusted in the direction of steepest descent:

$$\Delta\theta = -\frac{1}{2}\alpha.\nabla_\theta J(\theta) \qquad 4.22)$$

where α is the learning rate that controls the size of the update step. A higher α can lead to faster learning but may cause instability, while a lower α results in more gradual, stable convergence. By Substituting Equation (4.21) into Equation (4.22) gives:

$$\Delta\theta = \alpha.\left[Q(s,a) - \hat{Q}(s,a;\theta)\right].\nabla_\theta \hat{Q}(s,a;\theta) \qquad 4.23)$$

Substituting Equation (4.16) into the gradient expression, the explicit parameter update for Deep Q-Network (DQN) becomes:

$$\Delta\theta = \alpha.\left[R(s,a) + \gamma \max_{a'} \hat{Q}(s',a';\theta) - \hat{Q}(s,a;\theta)\right].\nabla_\theta \hat{Q}(s,a;\theta) \qquad 4.24)$$

Here, $\nabla_\theta \hat{Q}(s,a;\theta)$ represents the gradient of the predicted Q-value with respect to the network parameters, computed efficiently using backpropagation. This gradient-based update approach, known as Stochastic Gradient Descent (SGD), is central to modern reinforcement learning implementations involving deep function approximators.

A significant challenge in implementing Deep Q-Networks is temporal correlation between consecutive samples, causing unstable learning. A practical solution to this issue is the experience replay technique, where past experience tuples (s,a,r,s') are stored in a replay memory buffer. Instead of sequential updates, random mini batches are sampled from this buffer to update parameters via gradient descent. This procedure enhances data efficiency, breaks temporal correlations, and ensures smoother convergence.

Another issue arising from Equation (4.24) is the inherent bias caused by using the same network for action selection and action evaluation, leading to overestimation of Q-values. Double Deep Q-learning (DDQN) addresses this issue by employing two distinct Q-networks, each participating in action selection and action evaluation separately. In each iteration, both networks are used to select actions, but only one network, chosen randomly, is used to evaluate the selected action, thus reducing estimation bias. The DDQN algorithm, utilizing an epsilon-greedy exploration strategy, is detailed in Algorithm 1.

Algorithm 1: Double Deep Q-Network

1: Initialize $Q_1(s,a)$ *and* $Q_2(s,a)$*, Initialize state* $s_t = s_0$
2: ***loop***
3: *Select* a_t *using* ϵ*-greedy* $\pi(s) = \underset{a}{\operatorname{argmax}}\, Q_1(s_t,a) + Q_2(s_t,a)$
4: *Observe* (r_t, s_{t+1})
5: ***if*** *(with 0.5 probability True)* ***then***
6: $Q_1(s_t,a_t) \leftarrow Q_1(s_t,a_t) + \alpha\left[r_t + Q_1\left(s_{t+1}, \underset{a'}{\operatorname{argmax}}\, Q_2(s_{t+1},a')\right) - Q_1(s_t,a_t)\right]$
7: ***else***
8: $Q_2(s_t,a_t) \leftarrow Q_2(s_t,a_t) + \alpha\left[r_t + Q_2\left(s_{t+1}, \underset{a'}{\operatorname{argmax}}\, Q_1(s_{t+1},a')\right) - Q_2(s_t,a_t)\right]$
9: ***end if***
10: $t = t + 1$
11: ***end loop***

The following two chapters are dedicated to analyzing the results under various scenarios. Table *10* summarizes each scenario, outlining their target scope, LCV integration strategy, and corresponding signal control approach. The baseline signal timings represent the existing timing plans obtained directly from the City's data. In contrast, the Double Deep Q-Network (DDQN) controllers were developed as part of this research. The DDQN model is applied to a single intersection within the network to evaluate its localized impacts, while network-wide implementation using a multi-agent DDQN framework is identified as an area for future research. Chapter 5 presents the results for Scenarios 1 and 2, while Chapter 6 focuses on the analysis of Scenarios 3 and 4.

Table 10- Simulation Scenarios

Scenario	Target Level	LCV Integration	Signal Timing
Scenario 1	Network-wide	No	Baseline
Scenario 2	Network-wide	Yes (Various Replacement Rates)	Baseline
Scenario 3	Single Intersection	No	DDQN
Scenario 4	Single Intersection	Yes (Various Replacement Rates)	DDQN

Chapter 5. LCV Impact on the Network

5.1. Validation of the Simulation

Table *11* presents the maximum queue lengths obtained from the simulation alongside corresponding field observations. *Table 12* further analyzes these results by calculating and comparing the differences between the simulated and observed queue lengths, represented as a percentage and in terms of vehicle count. Simulation data were collected over a continuous one-hour period (from 900 to 4,500 seconds) and averaged over five simulation runs, each using different random seeds. In contrast, field data for each movement were gathered at discrete 10-minute intervals.

According to the queue length criteria in *Table 8*, the difference between simulated and observed values should remain within 20% or under 12 vehicles in at least 85% of movements. Bold cells in *Table 12* indicates instances where values exceeded these acceptance thresholds, highlighting that the criterion was met in only 65% of movements, falling short of the 85% target. One significant factor contributing to this discrepancy is the difference in measurement intervals; the simulation measures queue lengths continuously over one hour, while field observations represent shorter, 10-minute snapshots. Consequently, intersections with lower traffic volumes, which experience considerable fluctuations in queue length throughout the hour, display greater differences. For example, the observed southbound queue length at intersection 2 is 5 meters, whereas the simulated value is 38 meters (*Table 11*). If the field observations were based on a full hour rather than 10-minute increments, these values might align more closely. Supporting this, *Table 6* shows that intersection 2 is among the least busy signalized intersections in the network. In contrast, intersections experiencing higher traffic volumes are more likely to reach their maximum queue lengths consistently across shorter intervals, thus reducing discrepancies between the field and simulation measurements. Table *6* identifies intersections 1, 5, 9, 18, 22, and 23 as having volumes exceeding 3,000 vehicles per hour during the PM peak. For these intersections, the queue length criterion is successfully met in 87% of movements, surpassing the validation target of 85%.

Comparing the pre-calibration (Table *9*) and post-calibration (Table 12) results reveals significant improvements; the proportion of movements satisfying the queue length validation criterion rose from 35% to 65%. Additionally, certain discrepancies present before calibration, where the observed maximum queue length in 10 minutes exceeded the one-hour simulated value by more than 20% or 12 vehicles, no longer appear in the post-calibration results, underscoring the improvements achieved through the calibration adjustments.

Table 11- Observed and Post-Calibration Maximum Queue Lengths (m)

Intersection Number	Intersection	Southbound		Northbound		Eastbound		Westbound	
		Simulation	Observation	Simulation	Observation	Simulation	Observation	Simulation	Observation
1	Courtneypark Dr & Dixie Rd	140	160	160	150	140	160	87	70
2	Courtneypark Dr & Ordan Dr	38	5	24	20	135	130	94	20
3	Courtneypark Dr & Vipond Dr	70	20	43	50	53	10	85	5
4	Courtneypark Dr & Shawson Dr	53	5	53	5	117	100	95	20
5	Courtneypark Dr & Tomken Rd	97	100	116	100	132	130	253	240
6	Courtneypark Dr & Hwy.410	N/A		119	100	132	110	304	350
8	Tomken Rd & Tristar Dr	86	30	41	40	N/A		8	20
9	Dixie Rd & Meyerside Dr	56	60	57	60	89	90	108	80
11	Meyerside Dr & Vipond Dr	36	10	28	10	51	20	42	30
12	Meyerside Dr & Shawson Dr	37	20	53	20	25	30	61	30
13	Meyerside Dr & Tomken Rd	186	190	196	180	58	50	93	80
18	Dixie Rd & Britannia Rd	181	220	159	150	37	30	70	70
20	Shawson Dr & Britannia Rd	46	40	54	20	44	50	81	20
21	Britannia Rd & Tomken Rd	172	170	162	160	69	70	299	250
22	Dixie Rd & Shawson Dr	324	400	118	110	31	20	72	70
23	Dixie Rd & Hwy. 401	107	10	71	60	N/A		359	300

Table 12- Post-Calibration Queue Lengths vs. Observed

Intersection Number	Intersection	Southbound		Northbound		Eastbound		Westbound	
		Diff (%)	Diff (Veh)	Diff (%)	Diff (Veh)	Diff (%)	Diff (Veh)	Diff (%)	Diff (Veh)
1	Courtneypark Dr & Dixie Rd	-13	-20	7	10	-13	-20	24	17
2	Courtneypark Dr & Ordan Dr	**660**	**33**	20	4	4	5	**370**	**74**
3	Courtneypark Dr & Vipond Dr	**250**	**50**	-14	-7	**430**	**43**	**1600**	**80**
4	Courtneypark Dr & Shawson Dr	**960**	**48**	**960**	**48**	17	17	**375**	**75**
5	Courtneypark Dr & Tomken Rd	-3	-3	16	16	2	2	5	13
6	Courtneypark Dr & Hwy.410	N/A		19	19	20	22	-13	-46
8	Tomken Rd & Tristar Dr	**187**	**56**	3	1	N/A		-60	-12
9	Dixie Rd & Meyerside Dr	-7	-4	-5	-3	-1	-1	**35**	**28**
11	Meyerside Dr & Vipond Dr	**260**	**26**	**180**	**18**	**155**	**31**	40	12
12	Meyerside Dr & Shawson Dr	**85**	**17**	**165**	**33**	-17	-5	**103**	**31**
13	Meyerside Dr & Tomken Rd	-2	-4	9	16	16	8	16	13
18	Dixie Rd & Britannia Rd	-18	-39	6	9	23	7	0	0
20	Shawson Dr & Britannia Rd	15	6	**170**	**34**	-12	-6	**305**	**61**
21	Britannia Rd & Tomken Rd	1	2	1	2	-1	-1	20	49
22	Dixie Rd & Shawson Dr	-19	-76	7	8	55	11	3	2
23	Dixie Rd & Hwy. 401	**970**	**97**	18	11	N/A		20	59

In addition to examining queue lengths, comparing simulated and observed traffic flow serves as an important validation method. This comparison can be performed by directly evaluating the difference in flow volumes or by calculating the GEH statistic (see Equation (*4.1*)). Due to Table *8*, for validation purposes, the accepted thresholds require that the flow difference remains within ±100 vehicles per hour (vph) for intersections with observed flows below 700 vph, within ±15% for flows between 700 and 2,700 vph, and within ±400 vph for flows exceeding 2,700 vph.

Additionally, GEH values should remain below 5 in at least 85% of intersections. Table *13* presents an intersection-by-intersection comparison of simulated traffic flow against the post-balancing observed values. The results show that 100% of intersections meet the volume difference criteria, while 94% meet the GEH threshold, exceeding the 85% target in both cases. At the network level, the total simulated flow deviates by only 0.3% from the observed data, well within the acceptable 5% range, and the GEH statistic for the overall flow is 0.64, staying comfortably below the threshold value of 4.

Table 13- Comparison Between Observed and Simulated Traffic Flows

Intersection Number	Intersection	Total Approach Vehicles (Balanced Data)	Total Approach Vehicles (Simulation)	GEH Value	Vehicle Difference	Allowed Vehicle Difference
1	Courtneypark Dr & Dixie Rd	4465	4465	0.00	0	400
2	Courtneypark Dr & Ordan Dr	1485	1509	0.62	24	223
3	Courtneypark Dr & Vipond Dr	1722	1770	1.15	48	258
4	Courtneypark Dr & Shawson Dr	1918	1835	1.92	83	288
5	Courtneypark Dr & Tomken Rd	4019	4036	0.27	17	400
6	Courtneypark Dr & Hwy.410	3064	3240	3.13	176	400
8	Tomken Rd & Tristar Dr	1644	1819	4.21	175	247
9	Dixie Rd & Meyerside Dr	3402	3680	4.67	278	400
11	Meyerside Dr & Vipond Dr	768	826	2.05	58	115
12	Meyerside Dr & Shawson Dr	837	963	4.20	126	126
13	Meyerside Dr & Tomken Rd	1891	1759	3.09	132	284
18	Dixie Rd & Britannia Rd	3565	3685	1.99	120	400
20	Shawson Dr & Britannia Rd	1123	1145	0.65	22	168
21	Britannia Rd & Tomken Rd	2116	1924	4.27	192	317
22	Dixie Rd & Shawson Dr	3816	3625	3.13	191	400
23	Dixie Rd & Hwy. 401	3995	3676	5.15	319	400
	Sum of All Flows	**39830**	**39957**	**0.64**	**127**	**1991**

Because truck flow is a key focus of this project, the simulated truck flows were also compared with the observed data. Table *14* follows the same structure as Table *13*, except that the truck flow was not balanced separately; thus, the observed values in this table match the PM peak hour data from Table *6*. The results are again promising: 88% of intersections have truck flow differences within the permitted range, and 88% show GEH values below 5, both exceeding the target acceptance rate of 85%. At the network level, the difference between simulated and observed truck flow is 4.7%, which is under the 5% limit. Moreover, the corresponding network-wide GEH value of 3.65 is below the threshold of 4, further meeting the validation criteria. The relatively larger discrepancies in truck flows compared to overall traffic flows may be partly due to using unbalanced data for the truck flow comparison.

Table 14- Comparison Between Observed and Simulated Truck Flows

Intersection Number	Intersection	Total Approach Vehicles (Balanced Data)	Total Approach Vehicles (Simulation)	GEH Value	Vehicle Difference	Allowed Vehicle Difference
1	Courtneypark Dr & Dixie Rd	649	721	2.75	72	100
2	Courtneypark Dr & Ordan Dr	150	203	3.99	53	100
3	Courtneypark Dr & Vipond Dr	133	191	4.56	58	100
4	Courtneypark Dr & Shawson Dr	176	225	3.46	49	100
5	Courtneypark Dr & Tomken Rd	538	558	0.85	20	100
6	Courtneypark Dr & Hwy.410	321	436	5.91	115	100
8	Tomken Rd & Tristar Dr	300	264	2.14	36	100
9	Dixie Rd & Meyerside Dr	535	650	4.72	115	100
11	Meyerside Dr & Vipond Dr	140	118	1.94	22	100
12	Meyerside Dr & Shawson Dr	176	139	2.95	37	100
13	Meyerside Dr & Tomken Rd	296	258	2.28	38	100
18	Dixie Rd & Britannia Rd	681	727	1.73	46	100
20	Shawson Dr & Britannia Rd	243	169	5.16	74	100
21	Britannia Rd & Tomken Rd	247	279	1.97	32	100
22	Dixie Rd & Shawson Dr	763	694	2.56	69	114
23	Dixie Rd & Hwy. 401	824	830	0.21	6	124
	Sum of All Flows	**6172**	**6462**	**3.65**	**290**	**309**

As the final validation step, simulated travel times were compared against observed data from Google Maps. The allowable travel time variation specified by Equation (*4.2*) served as the benchmark for determining consistency between the model and real-world measurements. Table *15* presents a side-by-side comparison, with Google Maps data captured for the same analytical period, 4:00 p.m. to 5:00 p.m., averaged over five different weekdays. The simulated values, recorded from 900 to 4,500 seconds of runtime, were likewise averaged across five runs featuring distinct random seeds. The chosen routes were selected to represent a variety of paths within the network. The results show that every difference between modeled and observed travel times remains within the allowable variation range, thus satisfying the validation criteria.

Table 15- Comparison Between Observed and Simulated Travel Times

From	To	Travel Length (km)	Google Maps (Sec)	Simulation (Sec)	Absolute Difference (Sec)	Allowable Variation (Sec)
Intersection 23	Intersection 6	5.8	588	481	107	132
Intersection 22	Intersection 5	3.2	456	335	121	142
Intersection 9	Intersection 3	1.0	120	139	19	33
Intersection 13	Intersection 2	1.8	168	180	12	34
Intersection 6	Intersection 1	1.8	216	163	53	60
Intersection 3	Intersection 18	1.8	264	192	72	95
Intersection 7	Intersection 22	3.0	348	302	46	92
Intersection 4	Intersection 23	3.0	336	317	19	94
Intersection 1	Intersection 13	1.8	180	171	9	40
Intersection 2	Intersection 8	1.3	132	113	19	30

5.2. Scenario 1: Baseline Results

In the baseline scenario, the simulation outcomes reflect the results after calibration using the city's signal timings, and intersection-by-intersection findings are presented in Table *16*. All performance measures were obtained from the simulation window of 900 to 4,500 seconds and represent averages across ten runs with distinct random seeds. These results include total traffic volume (slightly different from Table *13* due to a different number of runs), truck percentage, average delay (for both all vehicles and truck-only vehicles, in seconds), average number of stops (for both all vehicles and truck-only vehicles), and Level of Service (LOS) rating for each intersection. Several notable correlations emerge from the data:

- A strong correlation (r = 0.98)[1] exists between truck delays and overall delays, and likewise between the number of stops for trucks and for all vehicles; however, certain intersections (e.g., intersection 23 with 22% trucks) show significantly higher truck delays (37% more than other vehicles).
- There is also a strong positive correlation (r = 0.89) between average delay and average stops for both trucks and all vehicles, though exceptions occur. For example, intersection 21 has a higher average delay than intersection 20, but intersection 20 records more stops. Intersection 21 experiences the highest delay among all intersections because the westbound left of this intersection encounters unusual traffic in the PM peak hour, originating from Intersection 20. The higher number of stops in Intersection 20 indicates that it is affected by its neighboring intersection (intersection 21), and most of these stops are short and are not related to its signal timing. Therefore, the number of shorter stops increases while the average delay of this intersection is less than its neighbor.
- A high correlation is observed between average delay and LOS, as low average delay (e.g., intersection 2) corresponds to LOS A, while high average delay (e.g., intersection 20) corresponds to LOS F.

[1] Pearson Correlation Coefficient:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}}$$

- Total volume and average delay show only a weak correlation (r ≈ 0.25), indicating that higher volumes alone do not guarantee higher delays, since intersection geometry, lane configuration, and signal timing also play significant roles.

Table 16- Baseline Scenario Intersection-by-Intersection Results

Intersection Number	Intersection	Volume		Avg Delay (S)		Avg Number of Stops		Level of Service	
		All	Truck Percentage	All	Trucks	All	Trucks	All	Trucks
1	Courtneypark Dr & Dixie Rd	4382	15%	41	46	0.70	0.66	LOS_D	LOS_D
2	Courtneypark Dr & Ordan Dr	1474	13%	10	12	0.26	0.27	LOS_A	LOS_B
3	Courtneypark Dr & Vipond Dr	1690	11%	11	7	0.34	0.24	LOS_B	LOS_A
4	Courtneypark Dr & Shawson Dr	1737	12%	14	15	0.40	0.41	LOS_B	LOS_B
5	Courtneypark Dr & Tomken Rd	3983	13%	40	37	0.90	0.77	LOS_D	LOS_D
6	Courtneypark Dr & Hwy.410	3125	16%	15	19	0.40	0.46	LOS_B	LOS_B
8	Tomken Rd & Tristar Dr	1731	15%	4	4	0.15	0.13	LOS_A	LOS_A
9	Dixie Rd & Meyerside Dr	3614	16%	22	20	0.47	0.39	LOS_C	LOS_B
11	Meyerside Dr & Vipond Dr	990	14%	12	11	0.45	0.34	LOS_B	LOS_B
12	Meyerside Dr & Shawson Dr	1152	15%	12	12	0.39	0.41	LOS_B	LOS_B
13	Meyerside Dr & Tomken Rd	2112	15%	39	42	1.12	1.15	LOS_D	LOS_D
18	Dixie Rd & Britannia Rd	3631	18%	34	37	0.64	0.62	LOS_C	LOS_C
20	Shawson Dr & Britannia Rd	1382	15%	55	58	2.41	2.26	LOS_F	LOS_F
21	Britannia Rd & Tomken Rd	2312	15%	81	84	2.05	2.01	LOS_F	LOS_F
22	Dixie Rd & Shawson Dr	3580	18%	29	28	0.60	0.55	LOS_C	LOS_C
23	Dixie Rd & Hwy. 401	3586	22%	30	41	0.76	1.00	LOS_C	LOS_C

The baseline simulation results for the main corridors and the overall network are summarized in Table *17* and Table *18*. Corridor-based analysis shows that Dixie Road is the busiest corridor and experiences the highest truck delays. Notably, Tomken Road has the greatest average delay for all vehicles and the highest number of stops for both passenger cars and trucks. On average, a vehicle passing through Tomken Road encounters 149 seconds of delay and over three stops, despite it being among the least congested corridors, suggesting suboptimal geometry and signal timing. Meanwhile, Table *18* reveals that trucks tend to experience more delay and lower average speeds than passenger cars, although passenger cars typically make more stops.

Table 17- Baseline Scenario Corridors Results

Corridor	Signalized Intersection	Volume		Avg Delay (S)		Avg Stops	
		All	Trucks	All	Trucks	All	Trucks
Dixie Rd.	5	18793	3353	157	172	3.17	3.22
Courtneypark Dr.	6	16391	2251	132	137	3.01	2.81
Meyerside Dr.	4	7868	1217	85	85	2.43	2.29
Tomken Rd.	4	10138	1437	164	166	4.22	4.05

Table 18- Baseline Scenario Network Results

Vehicles	Vehicles Arrived	Avg Delay (S)	Avg Stops	Avg Speed (m/s)
Truck	1673	123.8	2.87	32.7
Passenger Car	8613	116.3	3.07	35.2
All	10381	117.4	3.03	34.8

5.3. Scenario 2: Introduce LCV to the Network

To evaluate the impact of Long Combination Vehicles (LCVs) on the network, the model progressively replaces standard trucks (classified in VISSIM as Heavy Goods Vehicles, or HGVs) with LCVs. Since LCVs are roughly twice the average size of the standard HGVs, the replacement scheme involves substituting every two trucks with one LCV. To maintain consistent vehicle totals across simulations and simplify result comparisons, each replacement also adds one passenger car (so that each pair of removed trucks is replaced by one LCV plus one passenger car). For instance, if the initial composition at an entry point is 80 passenger vehicles and 20 HGVs, replacing 20% of HGVs under this scheme would result in 82 passenger cars, 16 HGVs, and 2 LCVs.

Table *19* summarizes the outcomes for varying LCV replacement rates. The 0% replacement scenario corresponds to Scenario 1 results presented in Table *18*. Under this replacement strategy, each LCV is substituted for two standard trucks and one passenger car, so as the LCV replacement rate increases, the number of trucks reaching their destinations declines, while passenger car and LCV arrivals rise (See Figure *13*). Overall, the total number of vehicles completing their trips also slightly decreases. Although each scenario introduces the same initial number of vehicles, higher network delays mean that some vehicles do not finish their journeys, creating an inverse relationship between arrivals and delay.

Moreover, Table *19* shows that average delay and stops move in parallel, while average speed trends in the opposite direction. As the LCV replacement rate increases, both delay and stops generally rise, except at 2.5% replacement, where they slightly decrease. This indicates that a small influx of LCVs (around 2.5%) does not adversely affect network performance and may even bring minor improvements. However, beyond this point, performance deteriorates with higher LCV replacement rates.

Table 19- Network Simulation Results Under Different LCV Replacement Rates

LCV Replacement	Vehicles	Vehicles Arrived	Avg Delay (Sec)	Avg Stops	Avg Speed (m/s)
0.0%	Truck	1673	123.8	2.87	32.7
	LCV	0	0.0	0.00	0.0
	Passenger	8613	116.3	3.07	35.2
	All	10381	117.4	3.03	34.8
2.5%	Truck	1563	123.9	2.83	32.7
	LCV	10	141.5	2.71	28.2
	Passenger	8717	115.2	3.01	35.4
	All	10384	116.6	2.97	34.9
5.0%	Truck	1486	130.1	3.10	31.9
	LCV	55	173.4	3.58	27.0
	Passenger	8723	119.3	3.22	34.8
	All	10360	121.0	3.20	34.3
7.5%	Truck	1410	136.8	3.34	31.2
	LCV	96	188.7	3.90	25.4
	Passenger	8745	122.1	3.30	34.4
	All	10350	124.7	3.30	33.8
10.0%	Truck	1333	147.0	3.75	30.1
	LCV	150	192.6	4.25	25.3
	Passenger	8739	127.9	3.57	33.6
	All	10319	131.2	3.59	33.0
12.5%	Truck	1248	150.9	3.84	30.0
	LCV	179	196.1	4.19	25.3
	Passenger	8770	129.0	3.65	33.5
	All	10307	134.2	3.67	32.9

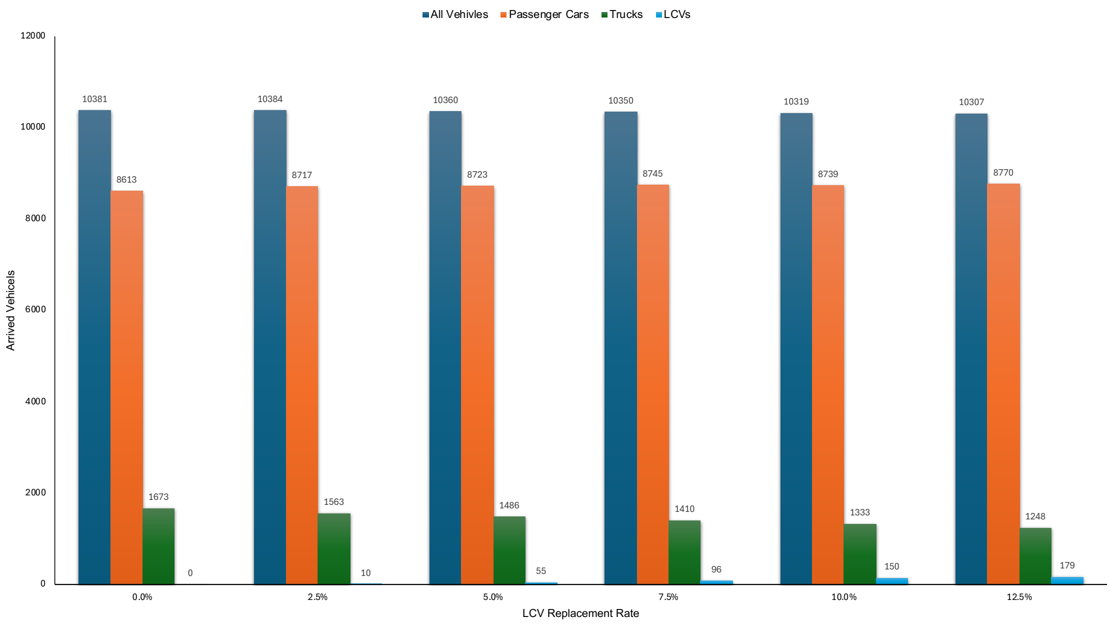


Figure 13- Number of Arrived Vehicles Under Different LCV Replacement Rates

Figure *14* offers more insight into the network's response to increasing LCV proportions by illustrating average delays for different vehicle types relative to the share of LCVs among total arrivals. Notably, even a small fraction of LCVs can significantly elevate network-wide delays. For instance, under a 12.5% replacement scenario, only 1.7% of arriving vehicles are LCVs, yet the average delay for all vehicles increases by **14.3%**, truck delays rise by **21.9%**, and passenger car delays climb by **10.9%**.

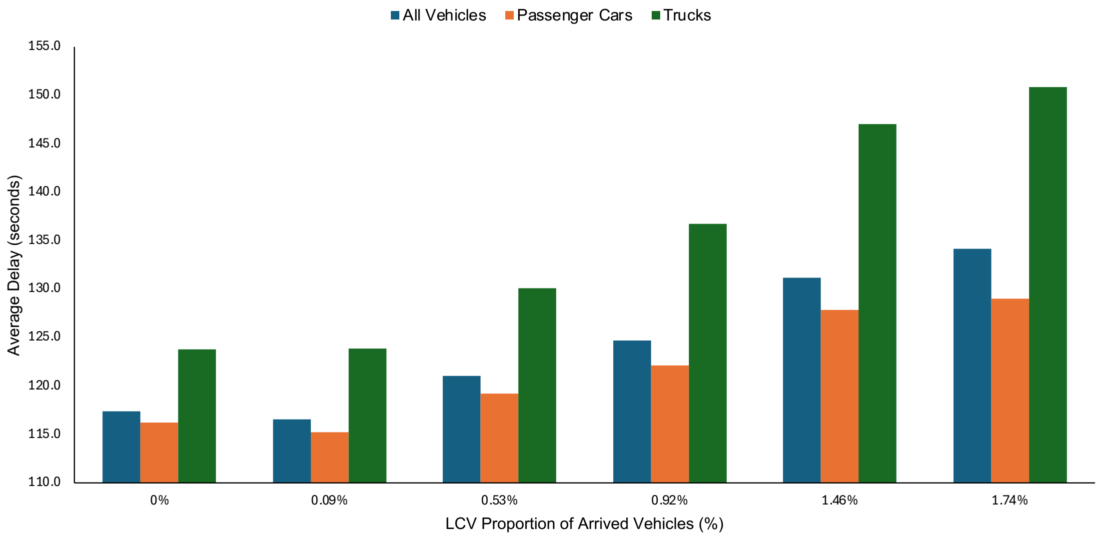


Figure 14- Average Delay Under Different LCV Arrival Ratios

To better understand the impact of Long Combination Vehicles (LCVs) on the network, intersection-level results under a 12.5% truck-to-LCV replacement scenario are presented in *Table 20*. These results are compared to the baseline conditions in Scenario 1 (Table *16*), with the comparison summarized in Table *21*. The analysis confirms that total vehicle volumes remain nearly constant, while average truck volumes decrease by approximately 25%, consistent with the replacement strategy of substituting two trucks with one LCV and one passenger car.

The introduction of LCVs negatively affects nearly all intersections, though the extent varies. For instance, intersection 23 ranked seventh in average delay under baseline conditions but rises to second following the 12.5% replacement. Here, average delay for all vehicles nearly doubles (+93%), while delay for trucks almost triples (+173%). The LOS for trucks declines sharply, dropping three full grades (e.g., from C to F).

Intersections exhibiting delay increases greater than 50% typically share geometric and operational features, namely, single-lane approaches for minor streets with short green times. In such cases, a single LCV can occupy the entire effective green interval, forcing remaining vehicles in the queue to wait for the next cycle and significantly inflating both delay and stop counts. Conversely, high-volume intersections with multi-lane approaches are less vulnerable to complete blockage but still experience modest increases in delay and stops due to longer discharge headways. At these locations, performance degradation is more proportional, with average delay increases of 5–10%, reflecting the reduced saturation flow rates associated with longer vehicle lengths.

Table 20- Intersection-by-Intersection Results Under 12.5% LCV Replacement

Intersection Number	Intersection	Volume		Avg Delay (S)		Avg Number of Stops		Level of Service	
		All	Truck Percentage	All	Trucks	All	Trucks	All	Trucks
1	Courtneypark Dr & Dixie Rd	4356	11%	45	50	0.74	0.68	LOS_D	LOS_D
2	Courtneypark Dr & Ordan Dr	1472	10%	17	23	0.48	0.54	LOS_C	LOS_B
3	Courtneypark Dr & Vipond Dr	1689	8%	12	8	0.36	0.25	LOS_B	LOS_A
4	Courtneypark Dr & Shawson Dr	1736	9%	14	15	0.39	0.37	LOS_B	LOS_B
5	Courtneypark Dr & Tomken Rd	3975	10%	42	39	0.95	0.82	LOS_D	LOS_D
6	Courtneypark Dr & Hwy.410	3118	12%	17	23	0.41	0.50	LOS_B	LOS_C
8	Tomken Rd & Tristar Dr	1727	11%	8	9	0.30	0.29	LOS_A	LOS_A
9	Dixie Rd & Meyerside Dr	3588	12%	23	21	0.49	0.41	LOS_C	LOS_B
11	Meyerside Dr & Vipond Dr	976	10%	12	11	0.42	0.33	LOS_B	LOS_A
12	Meyerside Dr & Shawson Dr	1140	11%	13	13	0.41	0.43	LOS_B	LOS_B
13	Meyerside Dr & Tomken Rd	2102	11%	49	54	1.40	1.48	LOS_D	LOS_D
18	Dixie Rd & Britannia Rd	3605	14%	34	36	0.67	0.64	LOS_D	LOS_D
20	Shawson Dr & Britannia Rd	1360	12%	55	57	2.36	2.20	LOS_F	LOS_F
21	Britannia Rd & Tomken Rd	2294	10%	82	88	2.09	2.14	LOS_F	LOS_F
22	Dixie Rd & Shawson Dr	3549	14%	31	30	0.65	0.60	LOS_C	LOS_C
23	Dixie Rd & Hwy. 401	3521	17%	58	82	2.09	2.68	LOS_E	LOS_F

Table 21- Intersection-Level Comparison Between Scenario 2 and Scenario 1

Intersection Number	Intersection	Volume Diff (%)		Avg Delay Diff (%)		Avg Stops Diff (%)		LOS Change	
		All	Truck	All	Trucks	All	Trucks	All	Trucks
1	Courtneypark Dr & Dixie Rd	-1%	-25%	10%	6%	7%	2%	0	0
2	Courtneypark Dr & Ordan Dr	0%	-23%	76%	81%	91%	102%	+2	0
3	Courtneypark Dr & Vipond Dr	0%	-22%	2%	5%	4%	4%	0	0
4	Courtneypark Dr & Shawson Dr	0%	-22%	0%	-3%	4%	-11%	0	0
5	Courtneypark Dr & Tomken Rd	0%	-24%	5%	5%	7%	7%	0	0
6	Courtneypark Dr & Hwy.410	0%	-23%	10%	4%	20%	9%	0	+1
8	Tomken Rd & Tristar Dr	0%	-26%	114%	100%	138%	126%	0	0
9	Dixie Rd & Meyerside Dr	-1%	-26%	4%	5%	4%	3%	0	0
11	Meyerside Dr & Vipond Dr	-1%	-27%	-2%	-8%	3%	-2%	0	-1
12	Meyerside Dr & Shawson Dr	-1%	-24%	7%	6%	15%	6%	0	0
13	Meyerside Dr & Tomken Rd	-1%	-28%	25%	25%	29%	29%	0	0
18	Dixie Rd & Britannia Rd	-1%	-26%	0%	5%	-2%	4%	+1	+1
20	Shawson Dr & Britannia Rd	-2%	-20%	0%	-2%	-2%	-3%	0	0
21	Britannia Rd & Tomken Rd	-1%	-28%	1%	2%	5%	6%	0	0
22	Dixie Rd & Shawson Dr	-1%	-25%	9%	9%	8%	8%	0	0
23	Dixie Rd & Hwy. 401	-2%	-25%	93%	173%	102%	169%	+2	+3

Chapter 6. Single Intersection Adaptive Traffic Signal Control

6.1. Proposed Double Deep Q-Network Framework

To integrate an RL agent into VISSIM, the first step is to define signal controllers that can modify signal timings in real time. VISSIM provides built-in signal controllers, referred to here as "Static Controllers," but these cannot be altered once the simulation begins. To create an "Adaptive Controller," Python can be linked to VISSIM via the COM interface, allowing each signal controller to be managed through Python. Although external signal controllers such as RBC or ASC/3 can also enable adaptive control in VISSIM, the Python COM interface simplifies communication with the Python-based RL agent. Further implementation details regarding the coding of these adaptive signal controllers can be found in Appendix D: Signal Controllers Coding in Python and are also available within the associated GitHub repository[1].

To develop a robust Double Deep Q-Network (DDQN) framework, several critical components must be defined clearly:

1. Action space
2. State representation
3. Reward function
4. Interactions between agents
5. Hyperparameters of the neural networks

The detailed description of each component, along with the rationale behind their selection, is provided in the following subsections.

6.1.1. Action Space Definition

The primary objective of this project is to implement adaptive signal timing that offers priority to freight vehicles. In existing active Freight Signal Priority (FSP) models, green time extensions or reductions are applied as fixed intervals after freight vehicles are detected, aiming to minimize disruptions to the traffic flow of other phases. For instance, Zhao et al. (2019) applied a fixed interval of 5 seconds, Kaisar et al. (2020) utilized intervals of either 10 or 20 seconds depending on traffic conditions, and Mahmud (2014) set this interval at 11 seconds. These modifications typically occur at the phase transition points.

[1] https://github.com/Ghanbari77/VISSIM-DDQN-TrafficControl

Considering the established approach in active FSP methods, the current project similarly adopts a fixed-interval strategy for extending or reducing green times at phase transitions. Following Zhao et al. (2019), the selected fixed interval is set at 5 seconds.

The time interval for applying actions to signal controllers is determined by the network's operational characteristics. All intersections within the studied network operate on a uniform cycle length of 160 seconds, each intersection serving two primary movements: east-west and north-south. To effectively provide priority opportunities for both main movements within each cycle, action updates are scheduled every 80 seconds, equivalent to half of the cycle duration.

Consequently, the action at each interval involves keeping, increasing, or decreasing the green phase duration by 5 seconds. Given that the green duration for certain critical phases must not fall below a predefined minimum threshold, this constraint must be verified prior to applying the actions. Mathematically, the action $a_j(t)$ for phase j at time step t is represented as follows:

$$a_j(t) = \begin{cases} 0 & g_j(t+1) = g_j(t) \\ -1 & g_j(t+1) = \max\left(g_j(t) - 5, G_{min,j}\right) \\ +1 & g_j(t+1) = g_j(t) + 5 \end{cases} \tag{6.1}$$

where $g_j(t)$ represents the green time of phase j at time step t, and $G_{min,j}$ denotes the minimum allowed green time for phase j. The action at time step t at intersection with n phases, defined as $a(t)$, is represented as the vector comprising the actions taken at each individual phase j:

$$a(t) = [a_1(t), a_2(t), \dots, a_j(t), \dots, a_n(t)] \tag{6.2}$$

Additionally, since the definition of the time step directly depends on the network's regular cycle length, significant deviations from the initial cycle length (160 seconds) are undesirable. For instance, if the cycle length becomes too short (e.g., less than 80 seconds), actions will no longer align properly with each cycle, limiting their effectiveness. Conversely, excessively increasing the cycle length results in longer split times, causing multiple actions to occur within the same phase without transitioning, which reduces their effectiveness. To address these constraints, minimum and maximum cycle lengths are defined practically for the target intersection as follows:

$$C_{min} = 120\ Seconds; \quad C_{max} = 200\ Seconds \tag{6.3}$$

6.1.2. State Representation

To effectively utilize the DDQN algorithm, the chosen state representation must satisfy the Markov property, meaning that the state should encapsulate sufficient information from the environment to accurately estimate the consequences of any potential action. Due to storage and computational constraints, only the most recent observation of the environment is considered as the state (see Equation (4.8)). Thus, it is crucial that this observation contains comprehensive and meaningful information to enable intelligent decision-making. The state representation in this study consists of three main elements:

1. Signal Timing Information:

   At each time step t, the state includes:

   - The green time for each phase $g_j(t)$, represented as a positive integer.
   - Information regarding the active phases $\delta_j(t)$, which is a Boolean indicator defined as 1 if phase j is active at time step t, and 0 otherwise.
   - The local time of the intersection $L_c(t)$, indicating how much time has elapsed within the current signal cycle at time step t.

2. Queue Length:

   Queue length is denoted as $q_j^{avg}[t-1, t]$, calculated as the average queue length for the lanes corresponding to phase j over the interval between time steps $t-1$ and t. This information provides insight into the intersection's traffic conditions during the previous time step, enabling the agent to assign more green time to phases experiencing greater congestion.

3. Traffic Flow:

   This element captures the approaching traffic volumes, categorized into:

   - The total flow of all vehicles $f_{all}(t)$,
   - The flow of trucks $f_{truck}(t)$,
   - The flow of Long Combination Vehicles $f_{LCV}(t)$, if applicable.

The traffic flows are detected by inductive loop sensors installed consistently 400 meters upstream of each intersection, based on the minimum spacing between intersections within the network. Given that the free-flow speed on the primary arterial roads is between 60 km/h and 70 km/h, the travel time from the detector to the intersection ranges from 20 to 25 seconds under free-flow conditions. Because accurate prediction of incoming traffic flow is critical for future decision-making, simply averaging traffic volumes over the entire 80-second interval from time step $t-1$ to t would not be sufficiently responsive. To enrich the temporal resolution of this data, each 80-second interval is subdivided into four shorter periods, each $\Delta T = 20 \text{ Seconds}$, approximately equivalent to the free-flow travel time. The traffic flows averaged over these shorter intervals provide more precise and actionable data for the agent's decisions.

Mathematically, if the three internal division points of the interval between $t-1$ and t are defined as ΔT_1, ΔT_2, and ΔT_3, then the traffic flows at time step t can be explicitly defined as follows:

$$\text{Traffic Flow(t)} = \begin{cases} f_{all}(t) = \{f_{all}^{avg}[t-1, \Delta T_1], f_{all}^{avg}[\Delta T_1, \Delta T_2], f_{all}^{avg}[\Delta T_2, \Delta T_3], f_{all}^{avg}[\Delta T_3, t]\} \\ f_{truck}(t) = \{f_{truck}^{avg}[t-1, \Delta T_1], f_{truck}^{avg}[\Delta T_1, \Delta T_2], f_{truck}^{avg}[\Delta T_2, \Delta T_3], f_{truck}^{avg}[\Delta T_3, t]\} \\ f_{LCV}(t) = \{f_{LCV}^{avg}[t-1, \Delta T_1], f_{LCV}^{avg}[\Delta T_1, \Delta T_2], f_{LCV}^{avg}[\Delta T_2, \Delta T_3], f_{LCV}^{avg}[\Delta T_3, t]\} \end{cases} \quad 6.4)$$

Finally, the complete state representation s_t at time step t is summarized as:

$$s_t = \begin{cases} \text{Signal Timing Information:} \quad \{g_j(t), \delta_j(t), L_c(t)\} \\ \text{Queue Length:} \quad q_j^{avg}[t-1, t] \\ \text{Traffic Flow:} \quad \{f_{all}(t), f_{truck}(t), f_{LCV}(t)\} \end{cases} \quad 6.5)$$

6.1.3. Reward Function

The reward function explicitly defines the goal that the RL agent pursues within the environment. In this project, the primary objective is to enhance the movement of freight vehicles, specifically trucks and LCVs, while simultaneously considering the overall traffic conditions. A suitable measure for assessing traffic performance at intersections is vehicle delay. The reward function, thus, is formulated based on vehicle delays as follows:

$$r = -(D_{all} + \omega(D_{truck} + D_{LCV})) \quad 6.6)$$

where D_{all} is the delay experienced by all vehicles at the intersection. D_{truck} and D_{LCV} represent delays specifically incurred by trucks and LCVs, respectively. ω is a weighting parameter controlling the priority level the agent assigns to freight vehicles. Specifically, if $\omega = 0$ no additional priority is given to trucks or LCVs beyond that given to other vehicles. Conversely, if ω is significantly large (e.g., $\omega = 10$), the agent predominantly acts to reduce delays for trucks and LCVs. This weighting parameter, referred to as the "priority value", closely aligns with the approach adopted by Zhao et al. (2019) in defining passive freight priority. Moreover, the negative sign in the equation explicitly indicates that the agent's objective is to minimize delays.

Furthermore, the exact time interval for measuring delay in Equation (6.6) must be carefully specified. Since the effectiveness of an action can only be accurately assessed over a complete signal cycle, and considering each time step in the framework corresponds to half of one cycle, the reward function is fully expressed as follows:

$$R(s_t, a_t) = r_t = -\left(D_{all}^{avg}[t, t+2] + \omega(D_{truck}^{avg}[t, t+2] + D_{LCV}^{avg}[t, t+2])\right) \quad 6.7)$$

where r_t represents the reward received when transitioning from state s_t by performing action a_t. $D_{all}^{avg}[t, t+2]$ denotes the average delay of all vehicles measured over the interval from time step t and $t+2$. Given each time step is half the cycle length, this interval equals one complete cycle duration. Hence, after the agent performs a specific action at a given step, it waits for one complete cycle to observe and obtain the corresponding reward, ensuring that the action's true effectiveness is accurately captured.

6.1.4. Interaction Between the Agents

The state, action, and reward definitions discussed so far pertain exclusively to a single intersection. However, the network includes multiple intersections, and effective interaction among the intersection-level agents can help alleviate individual deficiencies and facilitate broader network-wide coordination, such as the formation of green waves (El-Tantawy, 2012). One significant shortcoming of each isolated agent is its limited awareness of future network conditions. While current signal timing and queue lengths provide an understanding of the intersection's immediate state, the traffic flow data obtained every 20 seconds offers information only about vehicles expected to arrive within the immediate future (approximately 20 seconds ahead), given the placement of detectors roughly 20 seconds upstream. However, each agent must determine signal adjustments for the next 80-second interval, creating a gap in predictive capability.

This predictive limitation can be addressed by enabling interactions between neighboring intersection agents, specifically through sharing their local states. For example, consider an intersection k surrounded by adjacent intersections k_1, k_2, k_3, and k_4. The combined state representation for intersection k at time step t can thus be defined as:

$$S_k(t) = [s_k(t), s_{k_1}(t), s_{k_2}(t), s_{k_3}(t), s_{k_4}(t)\ ] \qquad 6.8)$$

where $S_k(t)$ is the integrated state utilized by intersection k, and $s_k(t), s_{k_1}(t)$ to $s_{k_4}(t)$ represent the local state information of intersection k and its neighboring intersections, respectively, as defined previously in Equation (6.5).

Sharing state information across intersections provides two key advantages. First, it significantly improves traffic flow prediction. By incorporating traffic flow data and queue lengths from adjacent intersections, intersection k gains enhanced insight into approaching vehicles over the full upcoming 80-second interval. Queues at neighboring intersections provide indirect yet valuable information about vehicles likely to reach intersection k in subsequent intervals, greatly enhancing the predictive accuracy of the agent's decisions. Second, this shared information supports effective signal coordination among intersections. Agents gain knowledge of their neighbors' active signal phases and local cycle timings, enabling synchronization of signal adjustments. This enhanced coordination facilitates smoother progression of vehicles through consecutive intersections, effectively establishing coordinated green waves across the network.

6.1.5. Hyperparameters

In this project, the simulation environment is implemented using VISSIM software. The model's time step is synchronized with real-time simulation, meaning each second in the RL model corresponds exactly to one simulation second in VISSIM. The RL agent applies actions at intervals of 80 seconds. However, conducting the entire training within a single, continuous simulation run is not feasible because the scenario is modeled during peak-hour conditions. Over prolonged simulations, traffic congestion rapidly accumulates, causing the environment to become excessively crowded. In such overcrowded conditions, agent actions lose their effectiveness, hindering meaningful learning. To avoid this issue, the training process is structured into separate simulation episodes. Specifically, the training dataset includes 200 episodes, each lasting 5,700 seconds (95 minutes), including an initial warm-up period of 300 seconds (5 minutes). With a decision interval of 80 seconds, the total number of training steps exceeds 14,000, ensuring sufficient experience for robust agent learning.

While the training process for this approach would require over 300 hours to develop each model under normal simulation speed, this duration can be significantly reduced by accelerating the simulation. VISSIM offers a speed-up feature that, on the current computer, reduces the duration of each episode to approximately 15–20 minutes. As a result, training each agent can be completed in roughly 48–72 hours. It is worth noting that the majority of this time is spent collecting new tuples of state, action, and reward, while the actual decision-making and action execution by the agent requires only a small fraction of the total time.

Action selection follows an epsilon-greedy exploration strategy. Initially, the exploration rate $\epsilon_0 = 1$, indicating completely random actions at the start of training. The exploration rate gradually decays toward a final value of $\epsilon_{min} = 0.01$, meaning the agent eventually selects actions randomly only 1% of the time. Given the total number of steps, the decay factor is set as $\Delta\epsilon = 0.99995$. The mathematical expression for updating the exploration rate is:

$$\epsilon_t = \epsilon_{t-1} \times \Delta\epsilon \qquad 6.9)$$

Thus, the action selected at each time step t can be formally expressed as:

$$a_t = \begin{cases} random\ action, & with\ probability\ \epsilon_t \\ arg\max_a Q_t\ (s_t, a), & with\ probability\ (1 - \epsilon_t) \end{cases} \quad 6.10)$$

Defining an optimal neural network structure is inherently challenging, as no universal guideline specifies the ideal configuration. Consequently, network architectures in this study are selected based on a similar project with analogous objectives and comparable state space dimensions (Wang, et al., 2024). Two alternative neural network configurations are considered: the first with layers containing [200, 200, 200, 100] nodes, and the second with layers containing [64, 64, 64, 32] nodes, where each number denotes the number of neurons within the respective hidden layer.

The batch size represents another crucial hyperparameter, influencing both the generalization capability of the network and computational resource requirements. Larger batch sizes typically improve generalization at the expense of increased computational load. Given computational resource limitations, the batch size is set at 64, indicating that 64 experience tuples are randomly sampled from the replay memory for each learning update. Additionally, the replay memory size is configured to store experience tuples from the last 10 episodes, ensuring adequate diversity and freshness of data during training.

Finally, the discount factor (γ) and learning rate (α) represent the last essential hyperparameters for the DDQN model. Although numerous dynamic approaches to selecting these parameters exist in the literature (Rasheed, et al., 2020), this project employs fixed values for simplicity and reliability. The discount factor is set to $\gamma = 0.99$, meaning that the reward of almost all the flowing steps in the episode are counted for a single action. Moreover, the learning rate of $\alpha = 0.01$ and $\alpha = 0.001$ both are chosen in numerous of studies (Ma, et al., 2021; Joo, et al., 2020). In this project both learning rates are applied. The summary of the hyperparameters is presented in Table *22*.

Table 22- Reinforcement Learning Hyperparameters

Category	Hyperparameter	Value
Simulation Setup	Number of Episodes	200
	Episode Duration	5700 seconds
	Warm-up Duration	300 seconds
Exploration	Initial Epsilon (ϵ_0)	1
	Minimum Epsilon (ϵ_{min})	0.01
	Epsilon Decay Rate ($\Delta\epsilon$)	0.99995
Neural Network	Hidden Layer Configuration	[200, 200, 200, 100] [64, 64, 64, 32]
	Batch Size	64
	Discount Factor (γ)	0.99
	Learning Rate (α)	0.01 0.001

6.1.6. Proposed DDQN Algorithm

The DDQN algorithm was selected for this study because the action space is small, discrete, and bounded, enabling the use of an argmax function to efficiently identify optimal actions. In contrast, if the action space were continuous, an actor–critic approach, using differentiation to optimize the action, would be more appropriate. Additionally, the traffic environment is inherently noisy, with stochastic arrivals and queue variations. In such conditions, standard DQN methods can suffer from overestimation or converge to trivial solutions. The DDQN architecture mitigates these issues, offering more stable learning and improved policy reliability in complex traffic environments.

Algorithm 2 presents the detailed steps of the proposed DDQN method. Initially, two neural networks $Q_1(s, a)$ and $Q_2(s, a)$ are initialized with random weights, and the VISSIM simulation environment is set up. The algorithm executes iteratively for a predefined number of episodes ($num\_episode$). Each episode begins with a simulation warm-up period. Afterward, at every discrete time step within the episode, the current state is first observed. Then, an action is selected using the epsilon-greedy strategy. The feasibility of the selected action is evaluated based on predefined constraints described in Equations (6.1) and (6.3). If the current time step exceeds two, the reward corresponding to the action executed two steps earlier becomes available, as defined in Equation (6.7). This reward, along with its associated state-action pair and subsequent state, is stored as a tuple (s, a, r, s') in the replay memory. These procedures continue until the episode reaches the last time step within the episode duration ($episode\_time\_step$).

For training purposes, no neural network updates occur during the first episode, which is solely dedicated to collecting initial experience data. Starting from the second episode, at each time step, a mini batch of size $batch\_size$ is sampled from the replay memory, explicitly including the two most recent experiences to maintain data freshness. The corresponding error (loss) between the predicted Q-values and target values is then computed, and neural network parameters are updated through the backpropagation technique. Lastly, the epsilon parameter, controlling the exploration-exploitation trade-off, is decreased incrementally at each training step to gradually transition from exploration to exploitation.

Algorithm 2: Proposed DDQN Framework

Initialize $Q_1(s,a)$ and $Q_2(s,a)$, VISSIM simulation environment
For $episode = 1, 2, \dots, num\_episode$ **do**
    Perform warm-up simulation
    Set $t = 0$
    For $t = 0, 1, \dots, episode\_time\_step - 1$ **do**
        Observe s_t
        Select a_t using ϵ-greedy:
            $\pi(s_t) = \underset{a}{argmax}\, Q_1(s_t, a) + Q_2(s_t, a)$ with probability $1 - \epsilon_t$, else random
        Check feasibility constraints and apply action a_t
        if $t \geq 2$ **then**
            Observe delayed reward r_{t-2}
            Store tuple $(s_{t-2}, a_{t-2}, r_{t-2}, s_{t-1})$ in replay memory
        end if
        if $episode \geq 2$ **then**
            Sample mini batch (include two most recent experiences)
            if (with 0.5 probability True) **then**
$$Q_1(s,a) \leftarrow Q_1(s,a) + \alpha\left[r + Q_1\left(s', \underset{a'}{argmax}\, Q_2(s',a')\right) - Q_1(s,a)\right]$$
            else
$$Q_2(s,a) \leftarrow Q_2(s,a) + \alpha\left[r + Q_2\left(s', \underset{a'}{argmax}\, Q_1(s',a')\right) - Q_2(s,a)\right]$$
            end if
            Update exploration rate: $\epsilon_t \leftarrow \epsilon_{t-1} \times \Delta\epsilon$
        end if
        $t \leftarrow t + 1$
    end for
end for

Finally, the complete flowchart illustrating the DDQN model processes at time step t is presented in Figure *15*. In this flowchart, it is assumed that the first episode has been completed (thus, the training process is active), and the current time step t is greater than 2, enabling storage and training procedures.

The flowchart comprises two primary loops. The first loop, representing the action selection loop, begins with the VISSIM environment providing the current state s_t to the neural networks. The neural networks then calculate the optimal policy $\pi(s_t)$ based on the state information. An action a_t^0 is subsequently selected by the Action Selector using an epsilon-greedy strategy and passed to the Signal Controller. The Signal Controller verifies the feasibility of the chosen action according to predefined constraints and outputs appropriate signal timings back into the VISSIM simulation environment.

The second loop, referred to as the training loop, starts with storing the experience tuple $(s_{t-2}, a_{t-2}, r_{t-2}, s_{t-1})$ into the replay memory. Subsequently, a mini-batch of experiences is sampled from replay memory, and the corresponding Q-values (both predicted and target) are computed by the neural networks. These computed values feed into the loss calculation step. Finally, the calculated loss is utilized to update the neural network parameters through the backpropagation method. This update step concludes the model's processes for the current time step t.

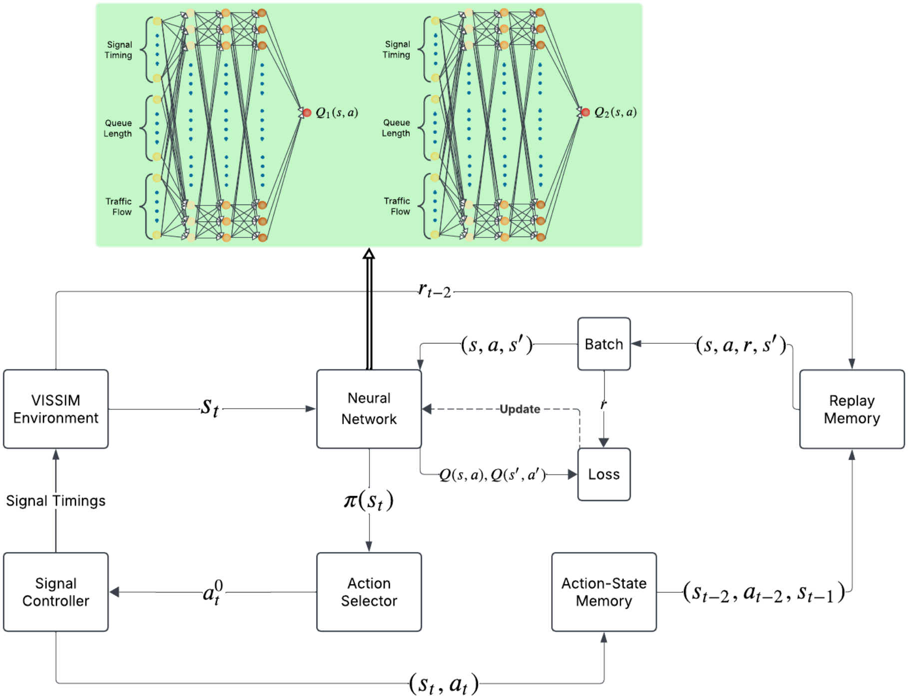


Figure 15- Proposed Double Deep Q-Network (DDQN) Flowchart

6.2. Intersection Selection and Layout

In this section, the adaptive traffic signal control methodology described in Section 6.1 is implemented at a single intersection. Intersection 22, located at Dixie Road and Shawson Drive, is selected for this implementation. As illustrated in Figure *16*, intersection 22 is closely connected to three other intersections: intersection 18 (Dixie Road and Britannia Road) to the north, intersection 23 (Highway 401 ramps and Dixie Road) to the south, and intersection 20 (Britannia Road and Shawson Drive) to the west. The distances from intersection 22 to intersections 18, 23, and 20 are approximately 443 meters, 490 meters, and 1,282 meters, respectively.

Figure 16- Location of the Target Intersection

Based on the post-calibration results of the network presented in Table *13*, intersection 22 experiences a total hourly volume of 3,580 vehicles, ranking it 5th among all intersections. Trucks constitute approximately 18% of this volume, giving it the 2nd highest truck proportion in the network. Additionally, due to the city's existing signal timing, each vehicle encounters an average delay of 29 seconds, with trucks specifically experiencing an average delay of 28 seconds.

A detailed layout of intersection 22 is provided in Figure *17*. Dixie Road, oriented north-south, serves as the major street, while Shawson Drive, oriented east-west, acts as the minor street. This intersection operates with a total of seven phases, including four through movements (northbound, southbound, eastbound, and westbound) and three left-turn movements (northbound left, westbound left, and southbound left). The northbound and southbound through movements each have three lanes; the westbound left-turn has two lanes, and all other movements are single-lane. It is important to note that the left-turn phases are actuated by vehicle detectors, and the eastbound left-turn and through movements share the same signal phase.

Figure 17- Detailed Layout and Phase Configuration of Target Intersection

6.3. Scenario 3: Single Intersection Model without LCV Integration

6.3.1. Model Training

In this scenario, the proposed model from Section 6.1 is applied to the designated intersection (Intersection 22), under the condition that no LCVs are present in the network. As a result, it is unnecessary to track LCV flows as part of the state representation (Equation (6.5)).

Table *22* summarizes two different neural network configurations tested for this scenario: a larger network with layers [200,200,200,100] and smaller network with layers [64,64,64,32]. Both configurations are drawn from the relevant literature. To identify the most suitable approach, each configuration is run for 100 training episodes. The underlying model remains almost identical to that described in Section 6.1, except that the priority parameter ω in the reward function (Equation (6.7)) is set to zero, such that the reward depends exclusively on the delay of all vehicles. Furthermore, the learning rate α is set to 0.001.

Figure *18* presents the results for both network configurations, with the y-axis indicating the negative average reward per episode. Early in training (episodes 1–30), both configurations exhibit high volatility in the reward, evidenced by pronounced spikes and dips, reflecting the exploration phase as the agent searches for effective policies under high exploration rates. As learning continues, reward fluctuations generally diminish, indicating that each network is converging toward a more stable policy.

Comparing the two configurations, the larger network [200,200,200,100] displays occasionally higher reward peaks during the initial episodes, a pattern often observed in deeper architectures, which can discover advantageous policies but may also overfit or overestimate until they have been sufficiently trained. By contrast, after approximately 30 episodes, the smaller network [64,64,64,32] exhibits more pronounced variability in its returns, likely due to a more limited capacity for approximating the Q-function. As a result, the smaller network is more sensitive to variations in traffic conditions encountered during training. Overall, the larger configuration [200,200,200,100] emerges as the preferred choice for subsequent experiments. However, it is important to provide enough training episodes to mitigate potential overfitting and ensure robust performance in the single-intersection context.

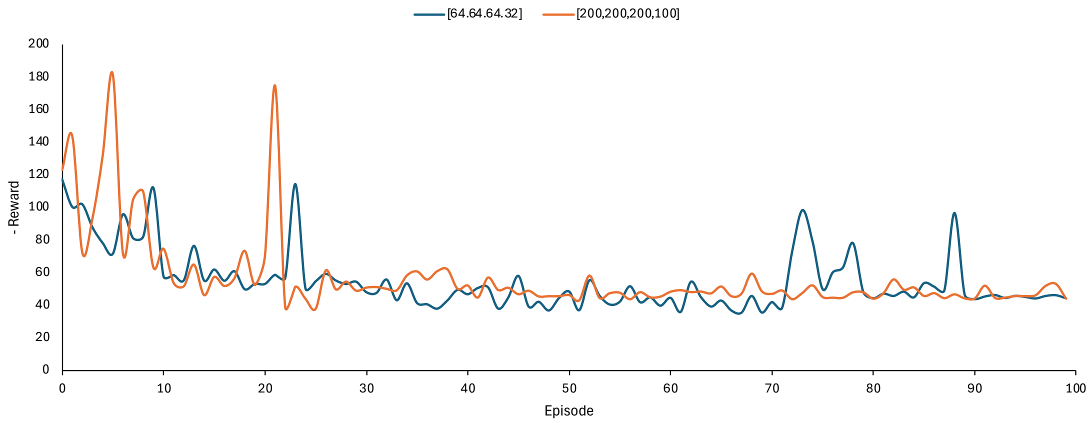


Figure 18- DDQN Learning Curves Under Different Neural Network Configuration

Table *22* introduces two learning rates, $\alpha = 0.01$ and $\alpha = 0.001$, drawn from the literature and applied here for further comparison. The model setup remains consistent with Section 6.1, except that the priority parameter ω in the reward function (Equation (6.7)) is set to zero, focusing exclusively on overall delay as the optimization objective. Figure 19 illustrates the resulting learning curves for both rates, where the y-axis indicates the average negative reward per episode.

During early training (episodes 1–30), both rates exhibit high volatility, consistent with the agent's exploration phase. However, the higher rate ($\alpha = 0.01$) yields more pronounced fluctuations, reflecting substantial parameter updates that can destabilize policy estimates. In contrast, the lower rate ($\alpha = 0.001$) converges more steadily, with reduced variance in rewards and a smoother trajectory toward improved performance.

As training progresses beyond episode 30, the $\alpha = 0.001$ model continues to display stable and consistent rewards, whereas the $\alpha = 0.01$ model shows notable oscillations, implying that large update steps may maintain instability. Despite both eventually reaching similar performance levels, the lower learning rate fosters a more reliable and gradual learning process. Overall, $\alpha = 0.001$ offers more stable and reliable convergence, making it preferable for practical implementations of the DDQN model in this context.

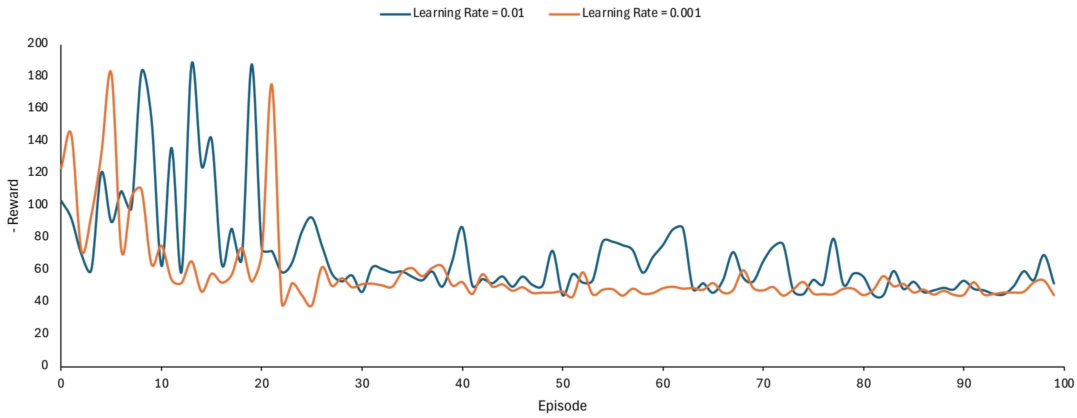


Figure 19- DDQN Learning Curves Under Different Learning Rate

Having identified the suitable neural network configuration and learning rate, the complete 200-episode training process is illustrated in Figure *20*. The figure plots the average delay at Intersection 22, where the DDQN model is applied. Notably, Intersection 22 exhibits a rapid decline in delay during the initial 30–40 episodes, suggesting that the agent quickly learns and refines an effective signal timing strategy. By around episode 50, Intersection 22's delay stabilizes at considerably lower levels, with only minor fluctuations. Between episodes 120 and 140, however, Intersection 22 experiences noticeable oscillations rather than a smooth convergence. These fluctuations may result from evolving traffic patterns (e.g., sudden spikes in arrival rates) that briefly disrupt the agent's learned policy.

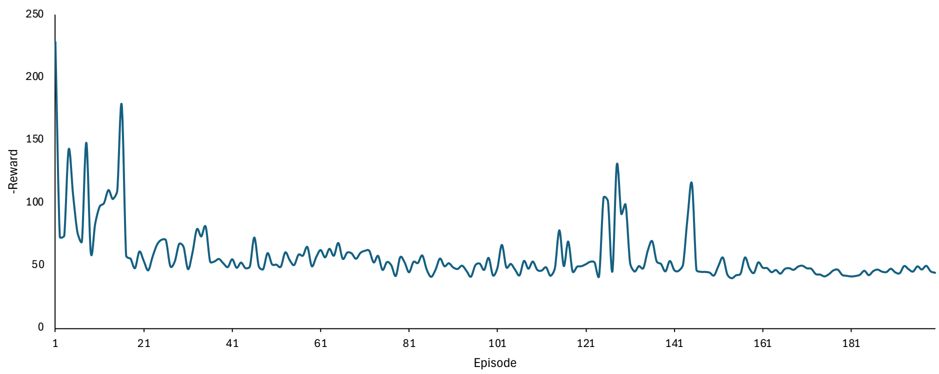


Figure 20- Target Intersection Delay During Learning Process

The next part of the results focuses on sensitivity analysis of the priority parameter ω. As described in Equation (6.7), ω determines how strongly the DDQN model prioritizes freight vehicles. A larger ω indicates a higher sensitivity to truck presence, effectively granting more priority to freight movements. In this study, five values of $\omega \in \{0, 0.5, 1, 2, 10\}$ is tested. Setting $\omega = 0$ treats trucks the same as other vehicles, the reward function depends solely on overall vehicle delay, while $\omega = 10$ places a strong emphasis on minimizing truck delays, even at the potential expense of other traffic.

Figure *21* shows four sub-figures, each comparing the baseline ($\omega = 0$) with one of the alternative ω values. Across all sub-figures, both curves exhibit large fluctuations over the first 20–30 episodes, indicating the exploration phase. Once past this initial period, each pair of curves begins to stabilize, suggesting that the agent is converging to a learned policy. However, the rate and smoothness of convergence depend on ω*.* Examining exponential moving averages of the curves reveals that $\omega = 0.5$ tracks closely with $\omega = 0$ but shows slightly improved performance and reduced volatility after episode 100. Meanwhile, $\omega = 1$ performs better than $\omega = 0$ for approximately the first 100 episodes but slips below it beyond episode 150. Of all tested values, $\omega = 2$ demonstrates the most consistent improvement over $\omega = 0$ throughout training, with relatively limited fluctuations. Finally, $\omega = 10$ ends up performing similarly to $\omega = 0$ after initial exploration. This outcome may stem from the fact that reinforcement learning algorithms, especially those using function approximation, can become unstable when reward components are excessively weighted. The agent may learn an overly specialized policy that fails to generalize well, often resulting in oscillatory behavior.

Collectively, these plots confirm the existence of a trade-off when selecting ω. Table *23* summarizes each configuration's minimum, average, and maximum rewards. Notably, $\omega = 2$ shows the best minimum, average, and maximum reward. However, these training curves alone are insufficient to identify the single best ω value for all operational conditions.

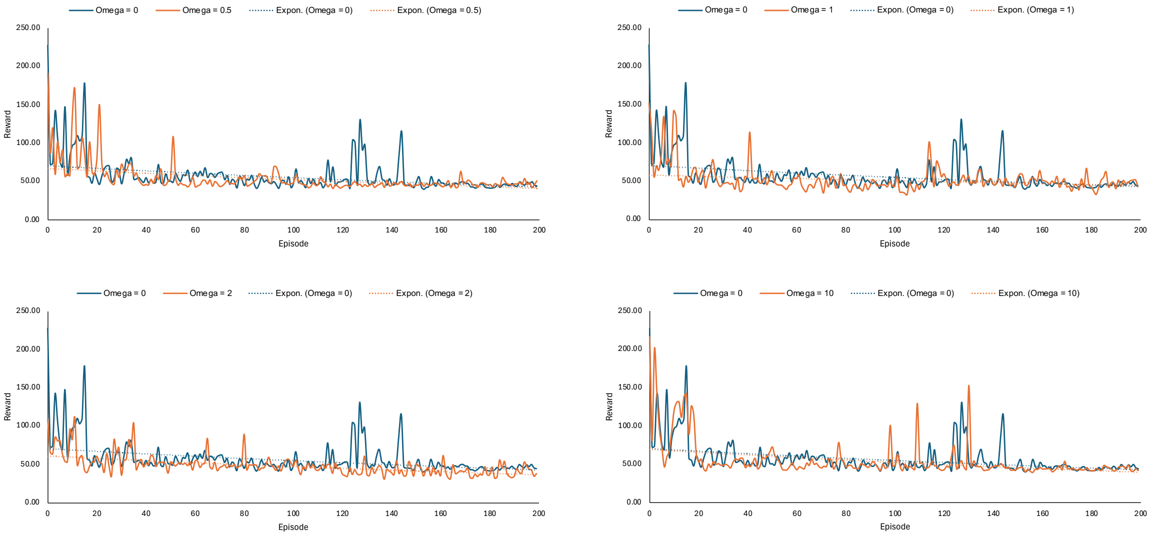


Figure 21- DDQN Learning Curves Under Different Priority Parameters

Table 23- Reward Metrics Across Different Priority Parameters

Negative Reward Metric	**Priority Parameter**				
	$\omega = 0$	$\omega = 0.5$	$\omega = 1$	$\omega = 2$	$\omega = 10$
Minimum	40.00	41.36	32.85	30.55	39.79
Average	57.96	54.74	52.72	48.86	56.02
Maximum	228.08	191.78	151.91	111.89	217.09

6.3.2. Performance Testing of the Selected DDQN Configuration

To further investigate finding the optimal freight priority parameter, we extracted each policy at the point of lowest reward in the training curve (e.g., the episode in which $\omega = 1$ yielded its minimum value) and tested under a set of varied traffic patterns. Performance metrics were collected over the simulation interval from 900 to 4,500 seconds and averaged across ten runs, each initialized with a different random seed. The selected traffic patterns replicate those used in Scenario 1, allowing for a direct comparison with earlier results. Table *24* presents the average performance across the ten patterns, offering a more comprehensive assessment of how well each ω setting generalizes beyond the training environment.

Focusing on intersection-level delay in Table *24* reveals that the baseline (Scenario 1) has an average intersection delay of 28.63 seconds for all vehicles and 27.95 seconds for trucks. In Scenario 3 with $\omega = 0$ (no special truck priority), the intersection's average delay drops to 24.49 for all vehicles and 22.82 for trucks, both notably lower than the baseline, suggesting that even a non-prioritizing RL model can outperform the City's timing plan in Scenario 1. As ω rises from 0 to 2, the intersection delay steadily decreases for both all vehicles and trucks. At $\omega = 2$, the intersection achieves its lowest average delay for both categories. This indicates that moderate priority for trucks leads to generally improved intersection performance. However, at $\omega = 10$, intersection delay rises marginally compared to $\omega = 2$. More notably, truck delay also increases despite the higher priority factor. An explanation for this paradox is that extending green phases excessively for freight vehicles can inadvertently generate longer queues for other traffic movements, potentially causing spillback and blocking maneuvers that undermine the very goal of reducing truck delay.

Overall, all ω-based scenarios (Scenario 3) outperform the baseline (Scenario 1) in both intersection and network metrics. The most substantial intersection improvements appear at $\omega = 2$, where average delays decrease by **28.8%** for all vehicles and **32.9%** for trucks. However, that does not necessarily correlate with the lowest network-wide delays. At $\omega = 2$, the intersection sees substantial gains, but the network's average delay (all vehicles) is slightly higher than that at $\omega = 1$. Specifically, $\omega = 1$ results in a 4.2% reduction in average delay for all vehicles and a 4.7% reduction for trucks across the network.

Table 24- Network and Target Intersection Delay Under Different Priority Values

Scenario		**Intersection 22**		**Network**	
		Avg Delay (s) All Vehicles	Avg Delay (s) Trucks	Avg Delay (s) All Vehicles	Avg Delay (s) Trucks
Scenario 1		28.63	27.95	117.44	123.83
Scenario 3	$\omega = 0$	24.49	22.82	115.40	122.85
	$\omega = 0.5$	22.33	21.88	114.09	122.17
	$\omega = 1$	21.64	19.69	**112.55**	**118.05**
	$\omega = 2$	**20.38**	**18.75**	113.98	121.57
	$\omega = 10$	21.94	21.11	113.53	120.86

A more detailed analysis of the best-performing adaptive freight priority models ($\omega = 1$ and $\omega = 2$) implemented at intersection 22 is presented in Table *25* and Table *26*. These tables compare the adaptive DDQN models against the baseline scenario (Scenario 1) for intersection 22 and its neighboring intersections (Intersections 18, 23, and 20), thus highlighting the local impacts of the DDQN strategies. All results represent averages over the same ten traffic patterns applied previously in Scenario 1. Specifically, Table *25* reports the total vehicles processed, average delay, average number of stops, and the intersections' overall Level of

Service for all vehicles. Table *26* provides similar metrics but focuses exclusively on truck vehicles.

The total number of vehicles passing through each intersection remains nearly constant across all signal timing scenarios, validating the statistical comparability of the performance metrics. As previously noted, at intersection 22, both average delay and average stops consistently decrease when increasing the freight priority parameter to $\omega = 2$. Specifically, when comparing $\omega = 2$ to Scenario 1 at intersection 22, the average number of stops is reduced by approximately **28.3%** for all vehicles and **29.1%** for trucks. Additionally, intersection 22's LOS improves notably from C to B under the DDQN model, indicating a significant enhancement in intersection performance for both general and freight traffic.

Performance at neighboring intersections varies with their distance from intersection 22. Intersection 18 (the northern neighbor) exhibits deteriorating conditions as the priority parameter increases, particularly for trucks. For instance, truck delay at intersection 18 increases by **8.1%** when comparing Scenario 1 to $\omega = 2$, causing a deterioration in truck LOS from C to D. Intersection 23 (the southern neighbor) experiences slight improvements in delay and average stops but a minor degradation in LOS from C to D. A potential reason for these performance declines at the northern and southern intersections is the disruption of their previous coordination with intersection 22; the city's fixed-offset coordination plan no longer aligns due to the adaptive adjustments at intersection 22.

In contrast, intersection 20 (the western neighbor), which is significantly farther away, shows substantial improvements under the DDQN implementation. Average delay at intersection 20 decreases by up to **20.3%**, and LOS notably improves from F to D. Because intersection 20 was not initially coordinated closely with intersection 22, the adaptive strategies employed at intersection 22 reduce overall congestion without negatively impacting existing coordination schemes, thus providing indirect benefits to traffic conditions further west.

Overall, these results underscore the significant role of coordination, highlighting the importance of extending the single-agent approach to multi-agent frameworks in which each intersection is individually controlled by an adaptive agent while simultaneously coordinating with neighboring intersections to maintain a synchronized green wave across the network.

Table 25- Target Intersection and Neighbors – Scenarios 1 & 3 (All Vehicles)

Intersection	Tot Vehicles			Avg Delay (s)			Avg Stops			LOS		
	Scenario 1	$\omega = 1$	$\omega = 2$	Scenario 1	$\omega = 1$	$\omega = 2$	Scenario 1	$\omega = 1$	$\omega = 2$	Scenario 1	$\omega = 1$	$\omega = 2$
Intersection 22	3580	3584	3579	28.63	21.64	20.38	0.60	0.44	0.43	C	C	B
Intersection 18 (Northern)	3631	3636	3628	34.20	34.66	35.80	0.64	0.60	0.63	C	C	C
Intersection 23 (Southern)	3586	3588	3587	30.11	27.68	28.65	0.76	0.72	0.73	C	C	D
Intersection 20 (Western)	1382	1389	1382	54.89	43.73	50.06	2.41	1.90	2.19	F	D	D

Table 26- Target Intersection and Neighbors – Scenarios 1 & 3 (Trucks)

Intersection	Tot Trucks			Avg Delay (s)			Avg Stops			LOS		
	Scenario 1	$\omega = 1$	$\omega = 2$	Scenario 1	$\omega = 1$	$\omega = 2$	Scenario 1	$\omega = 1$	$\omega = 2$	Scenario 1	$\omega = 1$	$\omega = 2$
Intersection 22	650	654	646	27.95	19.69	18.75	0.55	0.40	0.39	C	B	B
Intersection 18 (Northern)	672	682	671	36.88	36.90	39.86	0.62	0.58	0.64	C	D	D
Intersection 23 (Southern)	782	784	778	40.53	38.30	39.89	1.00	0.97	0.99	C	D	D
Intersection 20 (Western)	204	210	209	57.72	44.60	52.21	2.26	1.71	2.07	F	D	D

6.4. Scenario 4: Single Intersection Model LCV Integrated

6.4.1. Model Training

Scenario 4 is dedicated to assessing the impact of integration LCVs into traffic flow at a single intersection, using the DDQN signal timing model detailed in Section 6.1. Based on the findings from Section 6.3, the neural network employs a configuration of [200, 200, 200, 100], and the learning rate is set to $\alpha = 0.001$. Moreover, as $\omega = 2$ provided the optimal intersection-level performance in Scenario 3, it is also applied here.

Figure *22* illustrates the training performance of the DDQN models for Scenario 3 (without LCVs) and Scenario 4 (with LCVs), both utilizing a freight priority parameter of $\omega = 2$ are presented in Figure *22*. In Scenario 4, 12.5% of trucks are replaced with LCVs, a practical and representative ratio designed to introduce sufficient complexity for the model to differentiate between LCVs and standard trucks. The reward depicted in the figure is the episodic average delay of all vehicles at the intersection, calculated over each 80-second time step.

Initially, both scenarios exhibit significant volatility, reflecting extensive exploration. However, Scenario 4 consistently demonstrates higher average delays, particularly in the first 40 episodes, indicating that the model integrating LCVs learns more slowly and is less effective in reducing intersection delays compared to Scenario 3 (without LCVs). This outcome likely arises from the increased complexity and larger state space introduced by LCVs. As training progresses, delays in both scenarios stabilize after episode 50, yet Scenario 4 continues to exhibit slightly higher delays throughout subsequent episodes. This persistent difference may result from the inherently longer delays associated with accommodating LCVs compared to standard trucks.

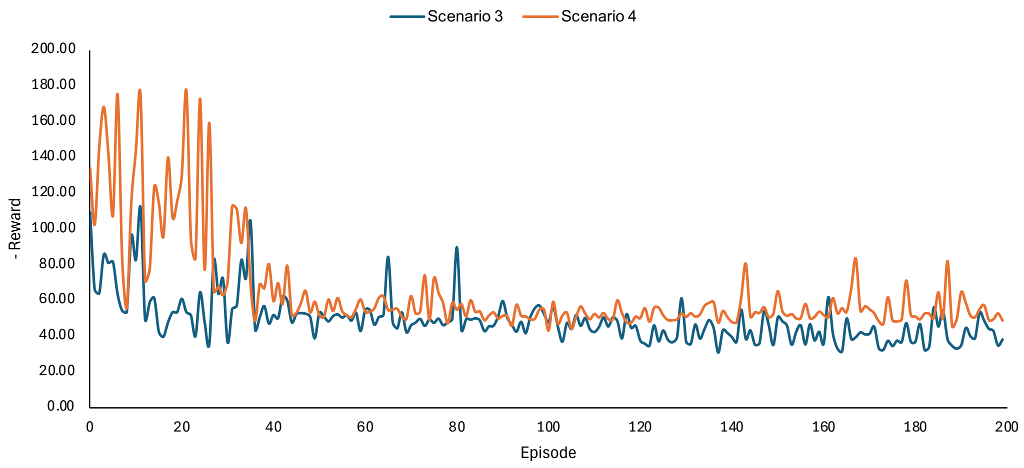


Figure 22- Target Intersection Delay During Learning With/Without LCV

6.4.2. Performance Testing of the Selected DDQN Configuration

Table *27* compares the performance of the DDQN model (with a 12.5% LCV presence) against a baseline signal timing across varying LCV replacement rates, focusing on average delays for all vehicles, trucks, and LCVs. Total numbers of vehicles, trucks, and LCVs remain similar between baseline and DDQN scenarios, ensuring that the comparison of delays and improvement metrics is valid and unaffected by substantial fluctuations in demand or throughput. However, the proportion of trucks decreases and LCVs increases as the replacement rate rises.

According to Table *27*, the DDQN model consistently outperforms the baseline across all tested LCV replacement rates. The resulting delay improvements range from **16% to 19%** for all vehicles, and **14% to 20%** for trucks, underscoring the effectiveness of adaptive signal control when integrating freight priority. As observed in Scenario 2, higher LCV percentages (0% to 12.5%) generally lead to increased delays under both baseline and DDQN due to the added complexity of longer, slower vehicles. Nonetheless, the DDQN approach still achieves substantial reductions in delays relative to the baseline.

Additionally, while LCV-specific delays improve by 21% at a 2.5% replacement rate, these gains diminish slightly (to 9–11%) as the replacement rate rises to 7.5–12.5%, reflecting the greater challenge posed by higher proportions of LCVs in the traffic mix. Even so, the DDQN method continues to yield significant LCV delay reductions compared to the baseline, confirming its capability to handle increasingly mixed fleets effectively.

Table 27- Target Intersection: DDQN vs. Baseline at Varying LCV Rates

LCV Replacement	Signal Timing	Total Vehicle	Avg Delay (s) All Vehicles	Improvement All Vehicles	Total Trucks	Avg Delay (s) Trucks	Improvement Trucks	Total LCVs	Avg Delay (s) LCVs	Improvement LCVs
0.0%	Baseline	3585	28.65	19%	645	27.99	19%	N/A	N/A	N/A
	DDQN	3602	23.24		651	22.66		N/A	N/A	
2.5%	Baseline	3589	28.44	19%	611	27.65	20%	2	37.34	21%
	DDQN	3599	23.13		612	21.99		1	29.68	
5.0%	Baseline	3582	28.83	19%	576	28.06	20%	21	32.46	19%
	DDQN	3597	23.25		583	22.31		23	26.27	
7.5%	Baseline	3575	28.42	17%	549	27.48	18%	35	32.92	11%
	DDQN	3587	23.47		552	22.44		35	29.23	
10.0%	Baseline	3558	29.51	18%	516	28.53	14%	59	37.42	9%
	DDQN	3572	24.12		518	24.43		61	33.91	
12.5%	Baseline	3564	29.32	16%	486	29.06	17%	68	38.72	11%
	DDQN	3576	24.71		490	24.15		72	34.64	

Finally, Table *28* presents a comprehensive comparison of four scenarios at Intersection 22:

1. **Scenario 1:** Baseline signal timing without LCVs,
2. **Scenario 2:** Baseline signal timing with LCV integration (12.5% replacement rate),
3. **Scenario 3:** DDQN adaptive signal timing without LCVs ($\omega = 2$),
4. **Scenario 4:** DDQN adaptive signal timing with LCV integration (12.5% replacement rate, $\omega = 2$).

Across all four scenarios, the total number of vehicles is nearly constant, ensuring a fair comparison of average delay and stops. However, because trucks are replaced by LCVs in Scenarios 2 and 4, the number of trucks in Scenario 1 matches Scenario 3, and Scenario 2 matches Scenario 4.

From Table *28*, the lowest average delays and stops, both for all vehicles and trucks, occur in Scenario 3, where adaptive control is applied without LCVs. Notably, only Scenario 3 achieves an improved LOS rating (from C to B) for both truck and general traffic flows. Nevertheless, Scenario 4 still reduces average delay by **13.7%** compared to Scenario 1, demonstrating that the DDQN approach enhances intersection performance even when LCVs are added.

Additionally, comparing Scenario 1 and Scenario 3 reveals 28.8% and 32.9% improvements in delay for all vehicles and trucks, respectively. By contrast, between Scenarios 2 and 4 (both with LCV integration), the improvements are 21% (all vehicles) and 20.2% (trucks). The smaller gains in Scenario 4 highlight the added complexity of incorporating LCVs into the state space, which makes the optimization process more challenging. However, the adaptive model still yields a considerable advantage over baseline timing when LCVs are present.

Table 28- Target Intersection Performance Under Different Scenarios

Scenario	Total Vehicles	Avg Delay (s) All Vehicles	Avg Stops All Vehicles	LOS All Vehicles	Total Trucks	Avg Delay (s) Trucks	Avg Stops Trucks	LOS Trucks
Scenario 1	3580	28.63	0.60	LOS_C	650	27.95	0.55	LOS_C
Scenario 2	3549	31.28	0.65	LOS_C	485	30.27	0.60	LOS_C
Scenario 3	3579	20.38	0.43	LOS_B	646	18.75	0.39	LOS_B
Scenario 4	3576	24.71	0.61	LOS_C	490	24.15	0.55	LOS_C

Chapter 7. Conclusion

7.1. Thesis Summary

This research evaluated the impacts of integrating Long Combination Vehicles (LCVs) within an urban transportation network and proposed an innovative adaptive traffic signal control strategy based on a Double Deep Q-Network (DDQN) reinforcement learning framework to mitigate associated negative effects. Using PTV VISSIM as the primary microsimulation tool, the study accurately modeled traffic behaviors, rigorously calibrated the simulation with observed data, and validated the results to ensure reliability and robustness in subsequent analyses.

The study area was strategically selected within the Region of Peel, known for substantial freight activities and potential future LCV routes, and encompassed 16 signalized intersections. Comprehensive data collection involved extensive field observations and gathering of traffic volume and signal timing data from authoritative sources, including the Region of Peel and the City of Mississauga. These efforts laid a solid foundation for accurately representing existing traffic conditions and calibrating the microsimulation model. Post-calibration validation significantly improved the model's accuracy, confirming its effectiveness in replicating real-world traffic dynamics.

Analysis of the baseline scenario (Scenario 1) revealed essential insights regarding intersection-specific and network-wide performance. Notably, intersections displayed strong correlations between delays experienced by trucks and general vehicles, highlighting certain intersections' heightened sensitivity to truck volumes. Additionally, the findings indicated that while increased traffic volumes generally contributed to greater delays, intersection geometry, lane configurations, and signal timing strategies also substantially influenced performance.

Introducing LCVs into the traffic mix (Scenario 2) introduced added complexities owing to their larger size and slower maneuverability. Even modest proportions of LCVs significantly increased average vehicle delays and stops. For instance, with LCVs constituting just 1.7% of the total vehicles, average delays rose by 14.3% for all vehicles, 21.9% for trucks, and 10.9% for passenger cars. These findings underscored the necessity for sophisticated signal control methods capable of effectively accommodating larger freight vehicles without substantially disrupting general traffic flow.

To address these challenges, this study proposed an adaptive traffic signal control strategy utilizing a DDQN-based reinforcement learning model. This adaptive framework dynamically adjusted signal timings based on real-time traffic states, incorporating comprehensive state representations that included signal timing information, queue lengths, and approaching traffic volumes. The model also provided explicit prioritization for freight vehicles via a flexible weighting parameter within its reward function, effectively balancing overall traffic efficiency and targeted freight prioritization.

The key distinction between this model and adaptive, model-free TSP methods in the literature lies in the reward definition. TSP approaches often achieve their best results when the reward is based on passenger delay (Othman, et al., 2025). However, for freight vehicles, particularly LCVs, this metric is less appropriate. LCVs occupy substantial road space at intersections yet carry relatively few passengers. Therefore, defining the reward solely in terms of passenger delay is not meaningful in this context. Instead, the reward in this study is defined as the total delay of all vehicles, with differentiated priority weights assigned to trucks and LCVs. Additionally, the action design in this methodology is informed by an existing freight signal priority approach proposed by Zhao et al. (2019).

The implementation of the DDQN model at Intersection 22, characterized by high freight vehicle volumes, demonstrated significant performance improvements compared to traditional fixed-time signal control. Scenario 3, which excluded LCVs, illustrated substantial reductions in average delays and number of stops for both general traffic and trucks when utilizing adaptive signal control. Specifically, compared to Scenario 1 (baseline), intersection delays improved by 28.8% for all vehicles and 32.9% for trucks. Sensitivity analysis further identified moderate freight prioritization as optimal, significantly enhancing intersection-level performance without negatively impacting broader network conditions.

Scenario 4 assessed the DDQN adaptive control strategy with integrated LCV traffic. Compared to Scenario 2 (baseline with LCVs), results showed improvements ranging from 16% to 19% in average delays for all vehicles and 14% to 20% for trucks, depending on the proportion of LCVs. Furthermore, even with the added complexity of LCVs, Scenario 4 maintained a 13.7% reduction in intersection delays relative to Scenario 1, emphasizing the adaptive model’s robustness.

In conclusion, this study effectively demonstrates the practicality and efficacy of a DDQN-based adaptive traffic signal control system for managing intersections with significant freight vehicle presence. This approach not only improves traffic efficiency but also robustly accommodates evolving freight transportation trends, such as increased utilization of LCVs, providing a promising strategy for future traffic management applications.

7.2. Limits and Recommendations

While this study offers valuable insights into the integration of LCVs within urban traffic systems and demonstrates the effectiveness of a DDQN adaptive traffic signal control framework, several limitations were encountered in the data collection process, simulation modeling, reinforcement learning implementation, and the generalization.

- <u>Data Collection Limitations:</u> The traffic volume and signal timing data used in this study were sourced from different periods ranging from 2013 to 2023. Although volume balancing procedures were applied to reconcile inconsistencies, using fully up-to-date and temporally consistent data would likely improve simulation reliability. Furthermore, while field observations provided the most recent insights into network performance, the calibration process was constrained by the format of the observed data. Specifically, queue lengths were recorded in 10-minute intervals, whereas simulation outputs reflected continuous one-hour periods. A one-hour observational dataset would have yielded more precise calibration results and better alignment between simulated and observed performance.
- <u>Simulation Model Limitations:</u> The VISSIM simulation did not incorporate vehicle entries or exits from local businesses within the study area, potentially underrepresenting localized congestion and micro-level interactions. Additionally, while buses were included in the vehicle composition, their operational dynamics, such as stopping, boarding, and alighting, were not modeled, which could affect the accuracy of performance measures. Regarding LCV modeling, although the simulation utilized composite data from various sources to approximate acceleration, deceleration, power, and weight distributions, a lack of standardized and localized parameters for LCVs limits the precision of vehicle behavior in the model. Future research would benefit from targeted empirical studies to derive accurate LCV dynamics under urban driving conditions.

- <u>DDQN Model Limitations:</u> The adaptive DDQN model used fixed 5-second green time adjustments, a common practice in freight signal priority studies. However, such discrete intervals may not yield optimal results under rapidly changing traffic conditions. Additionally, identifying the batch size, memory size, and the state representation was influenced by hardware and software constraints; traffic sensor data were captured every 20 seconds, whereas finer-grained updates (e.g., every 1 second) could provide richer, more responsive input for the learning model.
- <u>Scope and Generalizability:</u> This research focused exclusively on the PM peak hour, chosen to reflect the worst-case traffic conditions. While the results were promising, evaluating the model across other time periods, including AM peak and off-peak hours, would enhance the generalizability of the findings. Moreover, the DDQN adaptive control was implemented at a single intersection. Since traffic performance is often influenced by upstream and downstream intersection interactions, a broader multi-agent approach would better reflect the complexities of real-world signal coordination.

7.3. Future Scope

Building on the insights and outcomes of this study, several opportunities exist for extending and enhancing the research to better support the integration of LCVs and adaptive traffic signal control in urban transportation systems.

A natural extension of this research involves scaling the DDQN model beyond a single intersection to a corridor- or network-level implementation. While the single-agent framework demonstrated promising results at one location, many traffic inefficiencies emerge from the interactions between closely spaced intersections. Future work could explore multi-agent reinforcement learning (MARL) strategies in which agents representing different intersections communicate or coordinate actions. Potential coordination structures include centralized, decentralized, and hierarchical control approaches. Such strategies could facilitate the formation of green waves along freight corridors and enhance overall network stability, particularly in scenarios with high truck volumes and interdependent freight routes.

In addition to broader spatial implementation, the real-world deployment of the proposed DDQN framework remains an important research direction. Although the model was trained and tested in a simulated environment using VISSIM, field application introduces new challenges such as real-time data acquisition, computational constraints, and hardware-software integration. Future efforts should focus on embedding trained DDQN models within traffic signal controllers or dedicated edge computing units, with access to real-time detector data. Such implementations would require designing robust fallback mechanisms and ensuring consistent performance under a variety of traffic and environmental conditions.

The action space in this study is structured around a fixed-interval adjustment scheme, in which green times are modified by ±5 seconds during each half-cycle interval. While this approach is commonly adopted in the freight signal priority literature, it may lack the flexibility needed to fully address the complexity of highly dynamic traffic environments. Future research could investigate the use of continuous or variable-length signal adjustments, as well as reducing the action interval from half-cycle time to as low as one second. These enhancements could enable more responsive and adaptive control strategies, potentially improving both delay reduction and fairness across different vehicle classes.

With respect to LCV simulation, while the LCV model provided a reasonable estimate for simulation purposes, future research should incorporate empirical data on LCV acceleration, weight distribution, and turning behavior in urban environments. Dedicated studies collecting trajectory and operational data for LCVs could greatly improve the fidelity of simulation inputs and enhance the accuracy of impact assessments.

Temporal generalization is another critical dimension. This study focused on the PM peak hour, which typically represents one of the most congested periods in urban networks. However, to assess the broader applicability and robustness of the adaptive model, future work should include additional peak periods (e.g., morning, midday) and off-peak conditions. Further, the model's response to atypical scenarios, such as incidents, temporary signal malfunctions, or weather-induced capacity changes, should be examined to evaluate the resilience of the learned policies.

Together, these future research directions can lead to the development of more scalable, data-driven, and context-aware adaptive traffic signal control systems, better aligning with the evolving demands of urban mobility and freight transportation.

Chapter 8. References

[1] **Agrahari Anurag [et al.]** Artificial Intelligence-Based Adaptive Traffic Signal Control System: A Comprehensive Review [Journal]. - [s.l.] : Electronics, 2024. - 19 : Vol. 13.

[2] **Baker Ronald J. [et al.]** An Overview of Transit Signal Priority [Report]. - [s.l.] : Advanced Traffic Management Systems Committee and Advanced Public Transportation Systems Committee of the Intelligent Transportation Society of America (ITS America), 2002.

[3] **Behbahani Hamid and Poorjafari Mohammad** Passenger-based adaptive transit signal priority for BRT systemswith multiple loading areas [Journal]. - Tehran, Iran : IET Intelligent Transport Systems, 2024. - Vol. 18. - pp. 1089-1108.

[4] **Casas Noe** Deep Deterministic Policy Gradient for Urban Traffic Light Control [Journal] // Neural and Evolutionary Computing. - [s.l.] : Computer Science, 2017.

[5] **Chang Edmond Chin-Ping** Guidelines for Actuated Controllers in Coordinated Systems [Journal]. - [s.l.] : Transportation Research Record, 1996. - 1 : Vol. 1554. - pp. 61-73.

[6] **Christofa Eleni and Skabardonis Alexander** Traffic Signal Optimization with Application of Transit Signal Priority to an Isolated Intersection [Journal]. - [s.l.] : Transportation Research Record, 2011. - 1 : Vol. 2259. - pp. 192-201.

[7] **Chu Tianshu, Wang Jie and Lara Codecà Zhaojian Li** Multi-Agent Deep Reinforcement Learning for Large-scale Traffic Signal Control [Journal] // Machine Learning. - [s.l.] : Computer Science, 2019. - pp. 1-10.

[8] **Das Debashis [et al.]** Traffic Signal Priority Control Strategy for Connected Emergency Vehicles with Dilemma Zone Protection for Freight Vehicles [Journal]. - [s.l.] : Transportation Research Record, 2022. - 1 : Vol. 2676. - pp. 499-517.

[9] **Dion François and Rakha Hesham** Integration of transit signal priority within adaptive traffic signal control systems [Conference] // 84th Annual Meeting of the Transportation Research Board. - Washington, DC : [s.n.], 2005.

[10] **Dong Yuxuan [et al.]** Adaptive Transit Signal Priority Control for Traffic Safety and Efficiency Optimization: A Multi-Objective Deep Reinforcement Learning Framework [Journal]. - [s.l.] : Mathematics, 2024. - 3994 : Vol. 12.

[11] **Dowling Richard, Skabardonis Alexander and Alexiadis Vassili** Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Software [Report]. - Washington, D.C. : Federal Highway Administration, 2004.

[12] **El-Tantawy Samah** Multi-Agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC) // Doctor of Philosophy Thesis. - [s.l.] : University of Toronto, 2012.

[13] **Eom Myungeun and Kim Byung-In** The traffic signal control problem for intersections: a review [Journal]. - [s.l.] : European Transport Research Review, 2020. - 50 : Vol. 12.

[14] **Federal Highway Administration** Traffic Signal Timing Manual [Report]. - Washington, DC : U.S. Department of Transportation, 2008.

[15] **Fellendorf Martin and Vortisch Peter** Microscopic Traffic Flow Simulator VISSIM [Book Section] // Fundamentals of Traffic Simulation / book auth. Barceló Jaume. - [s.l.] : International Series in Operations Research & Management Science, 2010. - Vol. 145.

[16] **Friedrich Bernhard** Adaptive signal control: an overview [Conference] // 13th Mini Euro Conference–Handling Uncertainty in the Analysis of Traffic and Transportation systems. - 2002.

[17] **Ge Hongwei [et al.]** Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control [Journal]. - [s.l.] : IEEE Access, 2019. - Vol. 7. - pp. 40797-40809.

[18] **Government of Ontario** Driving through intersections [Online] // Ontario.ca. - August 2022. - https://www.ontario.ca/document/official-mto-drivers-handbook/driving-through-intersections#section-1.

[19] **Government of Ontario** Long Combination Vehicle Program Conditions [Online] // Ontario. - 2023. - https://www.ontario.ca/page/long-combination-vehicle-program-conditions.

[20] **Government of Ontario** Traffic Volume Data 1988-2019 [Report]. - [s.l.] : Ministry of Transportation, 2021.

[21] **Grislis Aivis** Longer combination vehicles and road safety [Journal]. - [s.l.] : Transport, 2010. - 3 : Vol. 25. - pp. 336–343.

[22] **GTA West** GTA West Corridor Planning & Environmental Assessment Study (Draft Overview of Forecasting Travel Demand Analysis) [Report]. - [s.l.] : MRC; URS; AECOM; Ecoplans, 2009.

[23] **Guo Ge and Wang Yunpeng** Eco-Driving of Freight Vehicles With Signal Priority on Congested Arterial Roads [Journal]. - [s.l.] : IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, 2021. - 5 : Vol. 70. - pp. 4225-4237.

[24] **Guo Yaru and Ma Jihui** An Improved Actuated Signal Control of Intersection Based on VISVAP [Conference] // International Conference on Sensor Network and Computer Engineering. - [s.l.] : Atlantis Press, 2016. - pp. 2352-5401.

[25] **Hajbabaie Ali and Benekohal Rahim F.** A Program for Simultaneous Network Signal Timing Optimization and Traffic Assignment [Journal]. - [s.l.] : IEEE Transactions on Intelligent Transportation Systems, 2015. - 5 : Vol. 16. - pp. 2573-2586.

[26] **Hassannayebi Erfan [et al.]** Data-Driven Multi-Criteria Assessment Framework for Analyzing the Reliability of Bus Services [Journal]. - [s.l.] : Transportation Research Record, 2024. - 4 : Vol. 2678. - pp. 415-439.

[27] **Hocking Kyle** Is Highway 401 really North America’s busiest highway? [Online] // Toronto CityNews. - March 2023. - https://toronto.citynews.ca/2023/03/28/401-north-america-busiest-highway/.

[28] **Holroyd John Baker and Hillier John A.** The Glasgow experiment: Plident and after [Report]. - 1971.

[29] **Hounsell N.B. and Wu J.** Public transport priority in real time traffic control systems [Conference] // Applications of Advanced Technologies in Transportation Engineering / ed. Stephanedes Yorgos J. and Francesco Filippi. - Capri, Italy : [s.n.], 1995. - pp. 71-75.

[30] **Hu Jia, Park Byungkyu Brian and Lee Young-Jae** Transit signal priority accommodating conflicting requests under Connected Vehicles technology [Journal]. - [s.l.] : Transportation Research Part C: Emerging Technologies, 2016. - Vol. 69. - pp. 173-192.

[31] **Hu Wen Xun [et al.]** Deep Reinforcement Learning Two-Way Transit Signal Priority Algorithm for Optimizing Headway Adherence and Speed [Journal]. - [s.l.] : IEEE Transactions on Intelligent Transportation Systems, 2023. - 8 : Vol. 24. - pp. 7920-7931.

[32] **Hu Wenxun** Adaptive Transit Signal Priority Algorithms for Optimizing Bus Reliability and Travel Time Using Deep Reinforcement Learning // A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy. - [s.l.] : Department of Civil and Mineral Engineering University of Toronto, 2022.

[33] **HUNT P B [et al.]** SCOOT-a traffic responsive method of coordinating signals [Journal]. - [s.l.] : Transport and Road Research Laboratory, 1981.

[34] **Iqbal Shahadat [et al.]** Developing Guidelines for Implementing Transit Signal Priority and Freight Signal Priority Using Simulation Modeling and a Decision Tree Algorithm [Journal]. - [s.l.] : Transportation Research Record, 2022. - 4 : Vol. 2676. - pp. 133-144.

[35] **Jaakkola Tommi, Singh Satinder P. and Jordan Michael I.** Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems [Report] / Department of Brain and Cognitive Sciences ; Massachusetts Institute of Technology. - 1994.

[36] **Joelianto Endra and Sutarto Herman** Simulation of Traffic Control Using VissimCOM Interface [Journal]. - [s.l.] : Internetworking Indonesia Journal, 2019. - 1 : Vol. 11. - pp. 55-61.

[37] **Joo Hyunjin, Ahmed Syed Hassan and Lim Yujin** Traffic signal control for smart cities using reinforcement learning [Journal]. - [s.l.] : Computer Communications, 2020. - Vol. 154. - pp. 324-330.

[38] **Kaisar Evangelos I. [et al.]** Evaluation of Freight and Transit Signal Priority Strategies in Multi-Modal Corridor for Improving Transit Service Reliability and Efficiency [Report]. - Florida Atlantic University : Freight Mobility Research Institute, 2020.

[39] **Kari David, Wu Guoyuan and Barth Matthew J.** Eco-Friendly Freight Signal Priority Using Connected Vehicle Technology: A Multi-Agent Systems Approach [Conference] // IEEE Intelligent Vehicles Symposium (IV). - Michigan, USA : [s.n.], 2014.

[40] **Kharrazi Sogol** Steering Based Lateral Performance Control of Long Heavy Vehicle Combinations // Thesis for the Degree of Doctor of Philosophy in Machine and Vehicle Systems. - Gothenburg, Sweden : Chalmers University of Technology, 2012.

[41] **Kim Jin-Tae and Courage Kenneth G.** Evaluation and Design of Maximum GreenTime Settings for Traffic-Actuated Control [Journal]. - [s.l.] : Transportation Research Record, 2003. - Vol. 1852. - pp. 246-255.

[42] **Krajzewicz Daniel [et al.]** SUMO (Simulation of Urban MObility) - An open-source traffic simulation [Conference] // Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002). - [s.l.] : 4th Middle East Symposium on Simulation and Modelling, 2002. - pp. 183-187.

[43] **Lee Wei-Hsun and Wang Hsuan-Chih** A Person-Based Adaptive Traffic Signal Control Method with Cooperative Transit Signal Priority [Journal]. - [s.l.] : Journal of Advanced Transportation, 2022. - Vol. 2022. - pp. 1-17.

[44] **Li Meng [et al.]** Modeling and Implementation of Adaptive Transit Signal Priority on Actuated Control Systems [Journal]. - [s.l.] : Computer-Aided Civil and Infrastructure Engineering, 2011. - 4 : Vol. 26. - pp. 270-284.

[45] **Lightstone Adrian, Belony Tony and Cappuccilli Jean-François** Understanding Goods Movement in Canada: Trends and Best Practices [Report]. - Ottawa : Transportation Association of Canada, 2021.

[46] **Lin Y. [et al.]** Transit signal priority control at signalized intersections: a comprehensive review [Journal]. - [s.l.] : Transportation Letters, 2014. - 3 : Vol. 7. - pp. 168-180.

[47] **Ma Zibo [et al.]** Adaptive Optimization of Traffic Signal Timing via Deep Reinforcement Learning [Journal]. - [s.l.] : Journal of Advanced Transportation, 2021. - Vol. 2021. - pp. 1-14.

[48] **Mahmud Maisha** Evaluation of Truck Signal Priority at N Columbia Evaluation of Truck Signal Priority at N Columbia Blvd and Martin Luther King Jr. Blvd Intersection: Blvd and Martin Luther King Jr. Blvd Intersection: VISSIM Micro Simulation Analysis of Truck Signal VISS [Report]. - [s.l.] : Portland State University Portland State University , 2014.

[49] **Manta Stavroula** Evaluation of Freight and Transit Signal Priority Strategies for Improving Transportation Operations in Urban Corridors // A Thesis Submitted to the Faculty of College of Engineering and Computer Science In Partial Fulfillment of the Requirements for the Degree of Master of Science on Civil Engineering . - [s.l.] : Florida Atlantic University , 2019.

[50] **Markowitz Thomas H.** Reducing Greenhouse Gas Emissions From Ontario Diesel Highway Trucks [Report]. - Toronto : Ontario Society of Professional Engineers, 2022.

[51] **Maze T. H. [et al.]** Electronic Screening in the I-35 Corridor [Journal] // Transportation Quarterly. - 1998. - 1 : Vol. 52. - pp. 45-55.

[52] **Medina Juan C. and Benekohal Rahim F.** Agent-based traffic management and reinforcement learning in congested intersection network. [Report] / University of Illinois at Urbana-Champaign. - [s.l.] : USDOT Region V Regional University Transportation Center, 2012.

[53] **Middendorf David P. and Bronzini Michael S.** The productivity effects of truck size and weight policies [Report] / Center for Transportation Analysis, Energy Division, Oak Ridge National Laboratory. - [s.l.] : Prepared for the Federal Highway Administration, U.S. Department of Transportation, 1994.

[54] **Miller Alan J** A Computer control system for traffic networks / Alan J. Miller [Book]. - Birmingham, England : University of Birmingham, Graduate School in Highway & Traffic Engineering, 1963.

[55] **Mirchandani Pitu and Head Larry** A real-time traffic signal control system: architecture, algorithms, and analysis [Journal]. - [s.l.] : Transportation Research Part C: Emerging Technologies, 2001. - 6 : Vol. 9. - pp. 415-432.

[56] **Murshed Md Tausif [et al.]** Freight Signal Priority (FSP): Assessment of Environmental and Mobility Impact and Feasibility for Implementation [Conference] // TRB Annual Meeting. - Washington : [s.n.], 2020.

[57] **Nagl Philipp** Longer Combination Vehicles (LCV) for Asia and the Pacific Region: Some Economic Implications [Report] : UNESCAP Working Paper. - [s.l.] : Economic and Social Commission for Asia and the Pacific (UNESCAP), 2007.

[58] **Negenborn R.R. and Maestre J.M.** Distributed Model Predictive Control: An Overview and Roadmap of Future Research Opportunities [Journal] // IEEE Control Systems Magazine. - 2014. - 4 : Vol. 34. - pp. 87-97.

[59] **Niedzielski Michał A., Goliszek Sławomir and Górka Anna** Signals, tracks, and trams: public transport signal priority impact on job accessibility over time [Journal]. - [s.l.] : Scientific Reports, 2024. - 23459 : Vol. 14.

[60] **Noaeen Mohammad [et al.]** Reinforcement learning in urban network traffic signal control: A systematic literature review [Journal]. - [s.l.] : Expert Systems With Applications, 2022. - Vol. 199.

[61] **Nordic Innovation** Next Nordic Green Transport Wave - Large Vehicles [Report]. - [s.l.] : Nordic transport regulations for large-scale hydrogen transport, 2020.

[62] **Noyce David A. and John W. Shaw PE** Automated Optimal Balancing of Traffic Volume Data for Large Access- Controlled Highway Networks and Freeway-to-Freeway Interchanges [Conference] // TRB Annual Meeting. - [s.l.] : University of Wisconsin, 2014. - Vols. 14-3565.

[63] **NZ Tranport Agency** Heavy combination vehicle stability and dynamics [Report]. - New Zealand : [s.n.], 2001.

[64] **Oketch Timothy and Carrick Marzenna** Calibration and Validation of a Micro-Simulation Model in Network Analysis [Conference] // TRB Annual Meeting. - Washington DC : [s.n.], 2005. - Vols. 05-1938.

[65] **Ontario Trucking Association** Ontario Budget Notes Importance of Trucking; Invests in Infrastructure and Training [Online]. - March 2023. - https://ontruck.org/ontario-budget-notes-importance-of-trucking-invests-in-infrastructure-and-training/.

[66] **Othman Kareem [et al.]** Multimodal adaptive traffic signal control: A decentralized multiagent reinforcement learning approach [Journal]. - [s.l.] : Multimodal Transportation, 2025. - 1 : Vol. 4.

[67] **Peel Region** GOODS MOVEMENT STRATEGIC PLAN 2017-2021 [Report]. - 2017.

[68] **Perlman David [et al.]** Review of the Federal Motor Carrier Safety Regulations for Automated Commercial Vehicles [Report]. - Washington, D.C. : Intelligent Transportation Systems Joint Program Office and Federal Motor Carrier Safety Administration, 2018.

[69] **Prashanth L A and Bhatnagar Shalabh** Reinforcement learning with average cost for adaptive control of traffic lights at intersections [Conference] // 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). - Washington, DC, USA : [s.n.], 2011. - pp. 1640-1645.

[70] **Rasheed Faizan [et al.]** Deep Reinforcement Learning for Traffic Signal Control: A Review [Journal]. - [s.l.] : IEEE , 2020. - Vol. 8. - pp. 208016-208044.

[71] **Ray Barton Associates Ltd.** Literature Review of the Safety of Long Combination Vehicles and their Operation in Canada. [Report]. - Ottawa : Canada Safety Council, 2003.

[72] **Regehr Jonathan D., Montufar Jeannette and Rempel Garreth** Safety performance of longer combination vehicles relative to other articulated trucks [Journal]. - [s.l.] : Canadian Journal of Civil Engineering, 2008. - Vol. 36. - pp. 40-49.

[73] **Region of Peel** 2015: a Look at Peel's Transportation Division [Report]. - 2015.

[74] **Region of Peel** Long Combination Vehicle (LCV) Usage Study [Report]. - 2019.

[75] **Region of Peel** Peel employment survey [Online] // Peel Region. - 2021. - https://data.peelregion.ca/pages/peel-employment-survey.

[76] **Region of Peel** Peel Region Goods Movement Strategic Plan 2012 to 2016 [Report]. - 2012.

[77] **Region of Peel** Sustainable and integrated transportation [Online] // Peel Region. - 2022. - https://www.peelregion.ca/advocacy/sustainable-transportation.asp.

[78] **Robertson D. I.** TRANSYT: A Traffic Network Study Tool [Report]. - [s.l.] : Road Research Laboratory, 1969.

[79] **Saha Ucchas** A Swept Path Analysis of Intersection Designs for Long Combination Vehicles. - [s.l.] : York University, 2021.

[80] **Salkham As'ad [et al.]** A Collaborative Reinforcement Learning Approach to Urban Traffic Control Optimization [Conference] // IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. - Sydney, NSW, Australia : [s.n.], 2008. - pp. 560-566.

[81] **Shabestray Soheil Mohamad Alizadeh and Abdulhai Baher** Multimodal iNtelligent Deep (MiND) Traffic Signal Controller [Conference]. - Auckland, New Zealand : IEEE Intelligent Transportation Systems Conference (ITSC), 2019. - pp. 4532-4539.

[82] **Shiri M. J. Shirvani and Maleki H. R.** Maximum Green Time Settings for Traffic-Actuated Signal Control at Isolated Intersections Using Fuzzy Logic [Journal]. - [s.l.] : International Journal of Fuzzy Systems, 2017. - 1 : Vol. 19. - pp. 247-256.

[83] **Sims A. G. and Dobinson K. W.** The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits [Journal]. - [s.l.] : IEEE Transactions on Vehicular Technology, 1980. - 2 : Vol. 29. - pp. 130-137.

[84] **Smith Harriet R. [et al.]** Transit Signal Priority (TSP): A Planning and Implementation Handbook [Book]. - [s.l.] : United States Department of Transportation, 2005.

[85] **Stantec** Courtneypark Drive East Class EA Transportation and Traffic Analysis Report [Report]. - [s.l.] : City of Mississauga, 2015.

[86] **Statistics Canada** Peel, RM [Census division], Ontario and Peel, RM [Census division], Ontario [Online] // Statistics Canada, 2016 Census of Population. - 2024. - https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/prof/index.cfm?Lang=E.

[87] **Taale H.** Comparing methods to optimise vehicle actuated signal control [Conference] // Eleventh International Conference on Road Transport Information and Control. - London, UK : [s.n.], 2002. - Vol. 486. - pp. 114-119.

[88] **Talukder Md Abu Sufian [et al.]** Analytical and Empirical Evaluation of Freight Priority System in Connected Vehicle Environment [Journal]. - [s.l.] : Journal of Transportation Engineering, Part A: Systems, 2022. - 6 : Vol. 148.

[89] **Thodi Bilal Thonnam, Chilukuri Bhargava Rama and Vanajakshi Lelitha** An analytical approach to real-time bus signal priority system for isolated intersections [Journal]. - [s.l.] : Journal of Intelligent Transportation Systems, 2021. - 2 : Vol. 26. - pp. 145-167.

[90] **Torabi Behnam, Wenkstern Rym Z. and Saylor Robert** A Self-Adaptive Collaborative Multi-Agent based Traffic Signal Timing System [Conference] // IEEE International Smart Cities Conference (ISC2). - Kansas City, MO, USA : [s.n.], 2018. - pp. 1-8.

[91] **Transportation Research Board** Highway Capacity Manual [Book]. - Washington, DC : Transportation Research Board, 2000.

[92] **Vieira Rodrigo De Souza, Moreno Gonzalo Guillermo and Martins Daniel** Stability and Road Safety of Long Combination Vehicles (LCV): Issues and Models [Conference] // 22nd International Congress of Mechanical Engineering. - Ribeirão Preto, SP, Brazil : [s.n.], 2013.

[93] **Wang Qinzheng [et al.]** Field Evaluation of Connected Vehicle-Based Transit Signal Priority Control under Two Different Signal Plans [Journal]. - [s.l.] : Transportation Research Record, 2020. - 7 : Vol. 2674. - pp. 172-180.

[94] **Wang Xiaoyu [et al.]** eMARLIN: Distributed Coordinated Adaptive Traffic Signal Control with Topology-Embedding Propagation [Journal]. - [s.l.] : Transportation Research Record, 2024. - 4 : Vol. 2678. - pp. 189-202.

[95] **Wang Xiaoyu, Abdulhai Baher and Sanner Scott** A Critical Review of Traffic Signal Control and A Novel Unified View of Reinforcement Learning and Model Predictive Control Approaches for Adaptive Traffic Signal Control [Journal] // Systems and Control. - [s.l.] : Electrical Engineering and Systems Science, 2022.

[96] **Washington State Department of Transportation** PROTOCOL FOR VISSIM SIMULATION [Report]. - [s.l.] : Washington State Department of Transportation; CH2MHill; HDR; DKS, 2014.

[97] **Washington State Department of Transportation** PROTOCOL FOR VISSIM SIMULATION [Report]. - 2014.

[98] **Webster F V** Traffic Signal Settings [Report]. - London, England : Road Research Technical Paper No 39, 1958.

[99] **Wei Hua [et al.]** PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network [Conference] // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. - New York, NY, USA : Association for Computing Machinery, 2019. - pp. 1290–1298.

[100] **Wei Hua [et al.]** Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation [Journal] // SIGKDD Explor. Newsl.. - New York, NY, USA : Association for Computing Machinery, 2021. - 2 : Vol. 22. - pp. 12–18.

[101] **Wideberg J. and Dahlberg E.** A Comparative Study of Legislation and Stability Measures of Heavy Articulated Vehicles in Different Regions [Journal]. - [s.l.] : International Journal of Heavy Vehicle Systems, 2009. - Vol. 16. - pp. 354-361.

[102] **Wisconsin Department of Transportation** WisDOT Vissim Vehicle Fleet Study [Report]. - [s.l.] : AECOM, 2020.

[103] **Wood Steven and Regehr Jonathan D.** Regulations governing the operation of longer combination vehicles in Canada [Journal] // Canadian Journal of Civil Engineering. - 2017. - 10 : Vol. 44. - pp. 838-849.

[104] **Woodrooffe John H. F.** Review of Canadian Experience with the Regulation of Large Commercial Motor Vehicles [Book]. - [s.l.] : Transportation Research Board, 2010.

[105] **Woodrooffe John, Anderson Darryl and Ash Lloyd** THE INFLUENCE OF POLICY ON CRASH RATE OF LONG COMBINATION VEHICLES [Conference] // Proceedings 8th International Symposium on Heavy Vehicle Weights and Dimensions . - Johannesburg, South Africa : [s.n.], 2004.

[106] **Wu Bing [et al.]** A Collaborative Transit Signal Priority Method Considering Gaming among Traffic Participants [Journal]. - [s.l.] : Journal of Advanced Transportation, 2023. - Vol. 2023.

[107] **Wu Wei, Huang Ling and Du Ronghua** Simultaneous Optimization of Vehicle Arrival Time and Signal Timings within a Connected Vehicle Environment [Journal] // Sensors. - 2019. - 191 : Vol. 20. - pp. 1-17.

[108] **Ye Zhirui and Xu Mingtao** Decision Model for Resolving Conflicting Transit Signal Priority Requests [Journal]. - [s.l.] : EEE Transactions on Intelligent Transportation Systems, 2017. - 1 : Vol. 18. - pp. 59-68.

[109] **Zhang Guohui and Wang Yinhai** Optimizing Minimum and Maximum Green Time Settings for Traffic Actuated Control at Isolated Intersections [Journal] // Transactions on Intelligent Transportation Systems. - [s.l.] : IEEE, 2011. - 1 : Vol. 12. - pp. 164-173.

[110] **Zhang Lihui [et al.]** Signal coordination models for long arterials and grid networks [Journal]. - [s.l.] : Transportation Research Part C: Emerging Technologies, 2016. - Vol. 71. - pp. 215-230.

[111] **Zhao Yanbo and Ioannou Petros** A co-simulation, optimization, control approach for traffic light control with truck priority [Journal]. - [s.l.] : Annual Reviews in Control, 2019. - Vol. 48. - pp. 283-291.

[112] **Zheng Guanjie [et al.]** Diagnosing Reinforcement Learning for Traffic Signal Control [Journal] // Machine Learning. - [s.l.] : Computer Science, 2019.

[113] **Zhou Bin [et al.]** Transit Signal Priority Strategy With Heterogeneous Graph-Based Deep Reinforcement Learning for Autonomous Public Transit Vehicles [Journal]. - [s.l.] : IEEE Transactions on Intelligent Vehicles, 2024.

[114] **Zhou Guangwei, Gan Albert and Shen L. David** Optimization of Adaptive Transit Signal Priority Using Parallel Genetic Algorithm [Journal]. - [s.l.] : Tsinghua Science & Technology, 2007. - 2 : Vol. 12. - pp. 131-140.

[115] **Zhou Guangwei, Gan Albert and Zhu Xiaoxia** Determination of Optimal DetectorLocation for Transit Signal Priority with Queue Jumper Lanes [Journal]. - Washington : Transportation Research Record, 2006. - pp. 123-129.

Appendices

Appendix A: Volume Balancing

To quantify the discrepancies of data between the adjacent intersections, the GEH formula, as proposed by Noyce, et al. (2014) is employed, expressed as:

$$GEH = \sqrt{\frac{2(v - \hat{v})^2}{v + \hat{v}}}$$

where in this project, v is the predicted hourly volume, and $\hat{v}$ is the observed hourly volume. For instance, in Figure *10*, the combined volumes of EBR, SBT, and WBL are summed to obtain v, whereas the SB volume at the downstream intersection is $\hat{v}$. The interpretation of GEH values, based on the study by Oketch, et al. (2005), is as follows: GEH between 0 and 5 indicates a good fit, between 5 and 10 warrants further investigation, and values over 10 signify a substantial error.

Table *29* using the GEH formula for the traffic volume of the analytical period (afternoon peak hour), compares the data of each intersection with its adjacent ones and outlines the errors associated with each data source and movement. In this table, N/A cells denote movements or intersections for which data is unavailable or non-existent. The last column shows the accuracy of each data source, calculated as the average of the relevant GEH values involving that source. For example, the average error for the Dixie Rd. & Shawson Dr. (intersection 22) intersection data from the Region of Peel is 3, obtained by averaging the GEH values for the southbound, northbound, and eastbound movements at that intersection (excluding the westbound data, which is non-existent), the southbound movement at Dixie Rd & Hwy. 401 (the southern intersection), and the northbound movement at Dixie Rd & Britannia Rd. (the northern intersection).

Table 29- Quantified Data Source Errors

Intersection Number	Source	Intersections	Southbound	Westbound	Northbound	Eastbound	Average Error of Each Source
1	Region of Peel	Courtneypark Dr & Dixie Rd	N/A	N/A	1	0	2
2	Stantec	Courtneypark Dr & Ordan Dr	N/A	5	N/A	2	2
3	Stantec	Courtneypark Dr & Vipond Dr	N/A	3	3	1	2
4	Stantec	Courtneypark Dr & Shawson Dr	N/A	0	N/A	1	3
5	Mississauga	Courtneypark Dr & Tomken Rd	N/A	8	21	0	7
6	Stantec	Courtneypark Dr & Hwy.410	N/A	7	N/A	N/A	3
8	Mississauga	Tomken Rd & Tristar Dr	5	N/A	16	N/A	11
9	Region of Peel	Dixie Rd & Meyerside Dr	0	N/A	2	N/A	2
11	Mississauga	Meyerside Dr & Vipond Dr	2	N/A	N/A	2	2
12	Mississauga	Meyerside Dr & Shawson Dr	N/A	2	13	6	5
13	Mississauga	Meyerside Dr & Tomken Rd	1	6	N/A	N/A	7
14	Region of Peel	Dixie Rd & Trinity Rd	4	N/A	2	N/A	3
18	Region of Peel	Dixie Rd & Britannia Rd	4	N/A	1	N/A	2
20	Mississauga	Shawson Dr & Britannia Rd	4	N/A	5	1	4
21	Mississauga	Britannia Rd & Tomken Rd	N/A	1	N/A	N/A	1
22	Region of Peel	Dixie Rd & Shawson Dr	1	N/A	3	5	3
23	Region of Peel	Dixie Rd & Hwy. 401	1	N/A	N/A	N/A	2

Based on the last column in Table *29*, most of the data appear reliable except for the Mississauga data at Tomken Rd. & Tristar Dr. (intersection 8), which showed an average GEH value over 10. Therefore, it is advisable to disregard this data and treat the intersection as if no data is available. The data for Courtneypark Dr. & Tomken Rd. (intersection 5), Meyerside Dr. & Shawson Dr. (intersection 12), and Meyerside Dr. & Tomken Rd. (intersection 13), all from Mississauga, also warrant further investigation. Comparing Table *5*'s collection dates with Table *29*'s results confirms that data sources with collection dates closer to those at adjacent intersections generally yield lower errors. In this regard, the Dixie Road corridor data from the Region of Peel demonstrated notably low error rates.

The next step involves determining the adjusted traffic volumes for each movement at every intersection in a way that eliminates discrepancies between intersection data, while minimizing the difference between the adjusted and original data. Two tables are created: the first contains the original traffic volume data, or unbalanced data, for each intersection during the afternoon peak hour, directly sourced from available data. Since this data is gathered from multiple sources, the mentioned discrepancy in Table *29* still exists between the values for adjacent intersections. The second table represents the adjusted or balanced data, where discrepancies between intersection values are resolved, but the values differ from the original data.

To generate the adjusted table, decision and dependent variables are defined. The adjusted table represents the final balanced data, as shown in Table *30*. In this table, green cells represent the decision variables, while the other cells are dependent variables. Red cells indicate intersections and movements where data is unavailable. The objective is to adjust the decision variables in such a way that the total differences between the corresponding values of the original and adjusted tables are minimized. To quantify the differences, a new GEH model is applied. The mathematical expression for this objective, based on the study by Noyce, et al. (2014), is as follows:

$$Min \sum \sqrt{\frac{2(A_{ij} - \hat{A}_{ij})^2}{(A_{ij} + \hat{A}_{ij})}}$$

where:

A_{ij} = Unadjusted volume approaching from direction i to the intersection j

$\hat{A}_{ij}$ = Adjusted volume approaching from direction i to the intersection j

Table 30- Example of Adjusted Traffic Volumes

	Southbound				Westbound				Northbound				Eastbound			
Intersections	**L**	**T**	**R**	**Sum**	**L**	**T**	**R**	**Sum**	**L**	**T**	**R**	**Sum**	**L**	**T**	**R**	**Sum**
Courtneypark Dr E & Dixie Rd	265	820	226	1314	37	70	138	245	137	713	60	914	161	325	215	701
Courtneypark Dr E & Ordan Dr	23	5	6	34	26	338	68	433	9	5	13	27	11	665	91	767
Courtneypark Dr E & Vipond Dr	4	2	13	19	45	275	33	353	34	25	73	133	86	690	67	843
Courtneypark Dr E & Shawson Dr	4	2	36	42	50	266	6	322	69	48	111	228	108	727	71	906
Courtneypark Dr E & Tomken Rd	555	355	87	997	115	173	83	371	243	500	117	860	55	233	112	400
Courtneypark Dr E & Hwy.410	0	0	0	0	0	502	0	502	38	0	51	89	0	349	0	349
Shawson Dr & Tristar Dr	0	103	3	106	0	0	0	0	1	265	0	266	6	0	15	21
Tomken Rd & Tristar Dr	8	550	24	582	3	0	1	4	36	837	6	878	22	0	21	43
Dixie Rd & Meyerside Dr	173	833	68	1075	64	27	56	147	130	802	159	1135	55	67	105	227
Meyerside Dr & Ordan Dr	62	27	33	122	75	130	19	225	219	5	27	251	3	138	108	248
Meyerside Dr & Vipond Dr	44	40	29	114	16	311	55	382	10	57	8	75	21	196	16	232
Meyerside Dr & Shawson Dr	31	55	32	118	15	277	58	350	16	122	21	159	48	180	17	245
Meyerside Dr & Tomken Rd	28	456	90	574	155	68	102	325	222	741	190	1103	35	27	84	149
Dixie Rd & Trinity Rd	0	1046	0	1046	0	0	52	52	0	1083	102	1186	0	0	0	0
Vipond Dr & Ordan Dr	51	21	0	72	190	0	20	210	0	55	200	255	0	0	0	0
Verbena Rd & Tomken Rd	0	530	165	695	0	0	0	0	119	539	0	658	564	0	91	656
Kestrel Rd & Verbena Rd	342	38	0	381	200	0	85	284	0	64	314	378	0	0	0	0
Dixie Rd & Britannia Rd E	143	814	88	1046	117	53	76	246	166	1060	234	1461	47	67	138	252
Britannia Rd E & Vipond Dr	211	0	0	211	0	307	0	307	0	0	0	0	255	40	0	295
Shawson Dr & Britannia Rd E	61	8	18	87	276	0	31	307	17	100	171	288	28	63	1	92
Britannia Rd E & Tomken Rd	33	431	157	621	2	13	20	35	208	520	25	753	118	34	86	238
Dixie Rd & Shawson Dr	108	895	65	1068	194	20	76	290	203	1327	299	1829	58	42	185	285
Dixie Rd N & Hwy. 401	0	1098	177	1275	837	0	994	1831	0	835	0	835	0	0	0	0

The dependent variables in the adjusted table are derived from the decision variables. First, the decision variables and the rationale for their selection should be explained, followed by the process for determining the dependent variables based on the decision variables. Three types of decision variables are defined: input volumes entering the network, decision variables representing intersections with unavailable data (indicated by red cells), and decision variables introduced to overcome circular dependencies among variables. The approach aims to define the minimum number of decision variables necessary for the process.

In calculating the dependent variables, two key assumptions are considered: 1) The sum of traffic volumes approaching an intersection from one direction equals the corresponding movement at the adjacent upstream intersection. For example, in Figure *10*, the southbound (SB) volume at the downstream intersection should equal the combined westbound left (WBL), southbound through (SBT), and eastbound right (EBR) movements at the upstream intersection. 2) The ratio of vehicles for each movement from a given approach is assumed to be equal with the ratio observed in the original data. For instance, if 168 out of 1,459 vehicles approaching Dixie & Britannia from the north turned left in the original data, then the adjusted northbound left-turn volume is set as $\frac{168}{1459}$ of the total northbound approach volume.

Excel's GRG Nonlinear optimization solver is used to solve the optimization problem. The constraint of the problem is that all variables, both decision and dependent, must remain positive. The optimized table obtained from this process is then directly applied to define the vehicle inputs and relative flows in the decision routing within the software.

Appendix B: LCVs Parameters in VISSIM

In modeling LCVs in VISSIM, most parameters mirror those of standard heavy goods vehicles (HGVs), but additional modifications are incorporated to capture the unique characteristics of LCVs. These include a customized desired acceleration function, power distribution, and weight distribution. Table *31*, Table *32*, and Table *33* detail the parameter settings, ensuring a more accurate representation of LCV behavior within the simulation.

Table 31- Desired Acceleration Function for LCV

Speed (km/h)	Acceleration (m/s^2)	Min Acceleration (m/s^2)	Max Acceleration (m/s^2)
0	0.762	0.305	2.500
10.0	0.762	0.305	2.152
20.0	0.762	0.305	1.856
30.0	0.762	0.305	1.597
40.0	0.738	0.305	1.378
50.0	0.576	0.305	1.186
60.0	0.448	0.250	1.021
70.0	0.351	0.190	0.881
80.0	0.271	0.150	0.759
90.0	0.213	0.110	0.652
100.0	0.165	0.080	0.564
110.0	0.128	0.000	0.485
120.0	0	0	0.1
130.0	0	0	0
140.0	0	0	0

Table 32- Power Distribution for LCV

Power (kW)	Cumulative Distribution
316	0
366	0.05
380	0.1
396	0.2
408	0.3
418	0.4
428	0.5
437	0.6
447	0.7
459	0.8
475	0.9
489	0.95
539	1

Table 33- Weight Distribution for LCV

Weight (kg)	Cumulative Distribution
25500	0
27500	0.002
29500	0.008
31500	0.022
33500	0.06
35500	0.136
37500	0.232
39500	0.317
41500	0.394
43500	0.469
45500	0.547
47500	0.632
49500	0.717
51500	0.812
53600	0.92
55500	0.967
57500	0.988
59500	0.995
61500	0.999
63500	1

Appendix C: Calibration Process

The initial calibration step involved modifying certain traffic volumes and adding new entrance or exit points in the network. Specifically, traffic volumes were adjusted as follows: the southbound approach at Intersection 5 was decreased by 10%, the eastbound approach at Intersection 6 was increased by 50%, the northbound approach at Intersection 21 was increased by 20%, and the westbound approaches at Intersections 22, 23, and 9 were reduced by 25% and 10%, respectively. Additionally, an entrance point was added between Intersections 19 and 20 with a flow of 300 vehicles per hour, and another source was introduced between Intersections 3 and 11 with a flow of 100 vehicles per hour.

The subsequent calibration phase involved refining key parameters to better reflect observed driver behavior and intersection operations. First, the emergency stop positions before each connector were modified in accordance with the WSDOT study (2014), so that the last possible point at which a vehicle can change direction is set to the turn bay length minus 10 meters. In addition, rather than coding stop signs at non-signalized intersections (which can lead to overly conservative driver behavior and thus inflated queues), those intersections were modeled with reduced speed areas, as recommended by WSDOT (Washington State Department of Transportation, 2014).

To more accurately depict real-world safety conditions, additional parameter adjustments were also introduced. After incorporating LCVs into the network, observations revealed that LCVs traveling in the outer lane of a double left-turn movement frequently encroached on the adjacent lane. To mitigate this issue, the outer lane of all double left-turn movements was restricted for LCVs. Moreover, certain right-turn movements, such as the westbound right turn at Intersection 23, required more explicit control mechanisms to prevent potential collisions between vehicles turning right from a minor street and those traveling straight through on a main street. In these cases, priority rules were defined in conjunction with conflict areas to ensure safe and efficient traffic operations.

Appendix D: Signal Controllers Coding in Python

```
Import os
import win32com.client as com
from scipy.optimize import linprog

# Create a COM object to interface with the Vissim software
Vissim = com.Dispatch("Vissim.Vissim")

# Specify the paths for the Vissim network file (.inpx) and layout file (.layx)
filename = "filepath.inpx"
layoutfile = "filepath.layx"

# Load the Vissim network
Filename = os.path.join(os.path.abspath(os.getcwd()), filename)
flag_read_additionally = False
Vissim.LoadNet(Filename, flag_read_additionally)
Vissim.LoadLayout(layoutfile)


def Ring_Barrier(splits):
    """
    Uses linear programming to adjust ring-barrier signal timings based on given minimum
    split times (green intervals). The objective is to minimize a weighted sum of green times
    under two equality constraints that ensure certain movements have balanced durations.

    Parameters
    ----------
    splits: list of int or float
    A list of 8 values representing minimum split times for the following movements:
    [0] Min_Split_NBL
    [1] Min_Split_SB
    [2] Min_Split_SBL
    [3] Min_Split_NB
    [4] Min_Split_EBL
    [5] Min_Split_WB
    [6] Min_Split_WBL
    [7] Min_Split_EB
     If any entry is 0, that movement is forced to remain at 0 seconds.
     Otherwise, the movement is bounded below by the provided value with no upper bound.

     Returns
```

```
    result: OptimizeResult
     The result object returned by scipy.optimize.linprog, which contains information such as
    the optimized split times, solver status, etc.
   """
   # Objective function coefficients: these weights correspond to each movement in 'splits'.
  # The goal is to minimize the sum c[0]*x0 + c[1]*x1 + ... + c[7]*x7.
   c = [2, 1, 2, 1, 2, 1, 2, 1]

   # Build the bounds for each movement:
   bounds = []
   for split in splits:
     if split == 0:
       bounds.append((0, 0))  # Fix this movement's green time to 0
     else:
       bounds.append((split, None))  # Lower bound = given split, upper bound = None (no
limit)

  # Equality constraints:
  #   1) NBL + SB == SBL + NB
  #   2) EBL + WB == WBL + EB
  A_eq = [
    [1, 1, -1, -1, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 1, -1, -1]
  ]
  b_eq = [0, 0]  # Each equality constraint must sum to zero when rearranged

  # Solve the linear programming problem using the 'highs' solver
  result = linprog(c, A_eq=A_eq, b_eq=b_eq, bounds=bounds, method='highs')
  return result


def Dixie_401_Signal(Sim_time, offset, Min_green_SB_NB, Min_green_WB,
End_of_Last_Cycle_Time):
    """

    Controls the traffic signal states for the Dixie_401 intersection using Vissim
SignalControllers.

    Parameters
    ----------
    Sim_time: int or float
      The current simulation time in seconds.
    offset: int or float
      Offset time to synchronize this signal with other intersections.
    Min_green_SB_NB : int or float
```

```
        The minimum green duration (in seconds) for the Southbound/Northbound (SB_NB)
    approach.
    Min_green_WB: int or float
        The minimum green duration (in seconds) for the Westbound (WB) approach.
    End_of_Last_Cycle_Time: int or float
        The simulation time at which the previous cycle ended. If 0, it indicates the first cycle.

    Returns
    -------
    End_of_Last_Cycle_Time : int or float
        Updated simulation time at which the current cycle ends.
    Local_time : int or float
        Time within the current cycle (i.e., how many seconds have elapsed since the last
    cycle ended).

    Notes
    -----
    - SigState values correspond to:
       1 -> Red,
       3 -> Green,
       4 -> Yellow.
    - If End_of_Last_Cycle_Time is zero, we apply an offset correction at the beginning of the
    simulation.
    """
    # Retrieve the specific intersection (with Key=1) from the global SignalControllers (scs).
    Dixie_401 = scs.ItemByKey(1)

    # Access individual signal groups for southbound/northbound (Key=1) and westbound
    (Key=2).
    SB_NB = Dixie_401.SGs.ItemByKey(1)
    WB = Dixie_401.SGs.ItemByKey(2)

    # Initialize both approaches to Red.
    SB_NB.SetAttValue("SigState", 1)
    WB.SetAttValue("SigState", 1)
    # --- Handle first cycle offset ---
    # If this is the very first cycle (End_of_Last_Cycle_Time == 0), adjust the current
simulation time to incorporate the offset.
    if End_of_Last_Cycle_Time == 0:
    Sim_time = Sim_time + offset - 1
```

```python
    # --- Compute Local_time (time since last cycle started) ---
    # If we are exactly at the end of the last cycle, reset local time to zero.
    # Otherwise, local time is how many seconds have passed since the last cycle ended.
    if Sim_time == End_of_Last_Cycle_Time and Sim_time != 0:
        Local_time = 0
    else:
        Local_time = Sim_time - End_of_Last_Cycle_Time

    # --- Set Signal States by Time Intervals ---
    # Below conditions define when SB_NB or WB signals turn green or yellow based on the local time and minimum green durations.

    # Red clearance time of WB = 3 seconds
    if 3 <= Local_time < Min_green_SB_NB + 3:
        SB_NB.SetAttValue("SigState", 3)  # Green
    # Yellow time of SB_NB = 4 seconds
    elif Min_green_SB_NB + 3 <= Local_time < Min_green_SB_NB + 7:
        SB_NB.SetAttValue("SigState", 4)  # Yellow
    # Red clearance time of SB_NB = 4 seconds
    elif Min_green_SB_NB + 11 <= Local_time < (Min_green_SB_NB + Min_green_WB + 11):
        WB.SetAttValue("SigState", 3)  # Green
    # Yellow time of WB = 4 seconds
    elif (Min_green_SB_NB + Min_green_WB + 11) <= Local_time < (Min_green_SB_NB + Min_green_WB + 15):
        WB.SetAttValue("SigState", 4)  # Yellow

    # --- End of Cycle Update ---
    # If Local_time matches the end of both SB_NB and WB green+yellow intervals, the cycle ends.
    # Then, update End_of_Last_Cycle_Time to mark the new end of cycle.
    if Local_time == (Min_green_SB_NB + Min_green_WB + 14):
        # For the first cycle, reverse the initial offset adjustment once it ends.
        if End_of_Last_Cycle_Time == 0:
            Sim_time = Sim_time - offset + 1
        End_of_Last_Cycle_Time = Sim_time + 1

    return End_of_Last_Cycle_Time, Local_time
```

```python
def Dixie_Shawson_Signal( Sim_time, Last_Barrier_Time, Ring_1, RBC_Function_Result,
offset, Initial_Splits, End_of_Last_Cycle_Time, change):
  """
  Controls signal phases at the Dixie-Shawson intersection based on a ring-barrier
approach.
  Updates signal group (SG) states by considering detector presence, timing offsets,
  ring-phase transitions, and optimized split results from the Ring Barrier function.

  Parameters
  ----------
  Sim_time : float
    Current simulation time (in seconds).
  Last_Barrier_Time : float
    The simulation time when the previous ring barrier finished.
  Ring_1 : bool
    A flag indicating which ring is currently active: True if Ring 1, False if Ring 2.
  RBC_Function_Result : OptimizeResult
    The result of the ring-barrier calculation (from Ring_Barrier), containing
    optimized split times in RBC_Function_Result.x.
  offset : float
    An offset for synchronizing the start time of this signal (applied if first cycle).
  Initial_Splits : list of float
    List of initial green split times for up to 8 signal phases or movements.
  End_of_Last_Cycle_Time : float
    The recorded simulation time when the last ring cycle began (used for logging or
analysis).
  change : bool
    Indicates whether splits have been updated and a new ring-barrier calculation
    needs to be performed mid-cycle.

  Returns
  -------
  Last_Barrier_Time : float
    Updated simulation time when the last ring barrier finishes.
  Ring_1 : bool
    Flipped if the cycle transitions from Ring 1 to Ring 2 or vice versa.
  RBC_Function_Result : OptimizeResult
    Updated ring-barrier result.
  End_of_Last_Cycle_Time : float
    Updated cycle start time.
  current_phase : list of int
    A list of length 8 indicating which phases are currently green (1) or off (0).
  Local_time : float
```

```
Time within the current ring (i.e., how many seconds have elapsed since the last ring
ended).
    """

    # Make a local copy of Initial_Splits so we don't mutate the original list.
    splits = Initial_Splits.copy()

    # Retrieve the signal controller for the Dixie-Shawson intersection (Controller Key=2).
    Dixie_Shawson = scs.ItemByKey(2)

    # Access the individual signal groups (SGs) controlling each movement (SG keys 1–8).
    NBL = Dixie_Shawson.SGs.ItemByKey(1)
    SB  = Dixie_Shawson.SGs.ItemByKey(2)
    WB  = Dixie_Shawson.SGs.ItemByKey(4)
    SBL = Dixie_Shawson.SGs.ItemByKey(5)
    NB  = Dixie_Shawson.SGs.ItemByKey(6)
    WBL = Dixie_Shawson.SGs.ItemByKey(7)
    EB  = Dixie_Shawson.SGs.ItemByKey(8)

    # Initialize all signals to Red (SigState=1).
    NBL.SetAttValue("SigState", 1)
    SB.SetAttValue("SigState", 1)
    WB.SetAttValue("SigState", 1)
    SBL.SetAttValue("SigState", 1)
    NB.SetAttValue("SigState", 1)
    WBL.SetAttValue("SigState", 1)
    EB.SetAttValue("SigState", 1)

    # current_phase[i] = 1 if movement i is currently green; otherwise 0.
    current_phase = [0] * 8

    # --- Adjust for first cycle offset ---
    # If no barrier cycle has occurred yet (End_of_Last_Barrier_Time == 0), apply the offset.
    if Last_Barrier_Time == 0:
        End_of_Last_Cycle_Time = End_of_Last_Cycle_Time + offset - 1

    # --- Cycle boundary checks ---
    # If the simulation time matches the end of the last barrier, check detectors to see if any
approach has zero vehicles. If so, set its split to 0, then recalculate RBC_Function_Result.
    if Sim_time == Last_Barrier_Time and Sim_time != 0:
        if net.Detectors.ItemByKey(50).AttValue("Presence") == 0:
            splits[0] = 0   # NBL movement
        if net.Detectors.ItemByKey(51).AttValue("Presence") == 0:
            splits[6] = 0   # WBL movement
```

```
    if net.Detectors.ItemByKey(53).AttValue("Presence") == 0:
        splits[2] = 0  # SBL movement

    # Recompute optimized splits based on any zeros now inserted.
    RBC_Function_Result = Ring_Barrier(splits)

    # Since we are at the start of a new ring, local time resets to 0.
    Local_time = 0

    # If we are entering Ring 1, change the start time of the new cycle.
    if Ring_1:
        End_of_Last_Cycle_Time = Sim_time


else:
    # If a new ring hasn't been started, find how long we've been in the current ring.
    Local_time = Sim_time - Last_Barrier_Time

# --- Mid-cycle split changes ---
# If 'change' is True, the RBC_Function_Result might need recalculation again here.
if change:
    for i in range(8):
        if RBC_Function_Result.x[i] == 0:
            splits[i] = 0
    RBC_Function_Result = Ring_Barrier(splits)

# --- Ring 1 Logic ---
# Movements: NBL, SB, SBL, NB (indices in RBC_Function_Result.x are 0, 1, 2, 3).
if Ring_1:
    # Safety/truncation: Ensure after changes in the mid-cycle the Local_time doesn't
exceed the current ring time.
    if Local_time > RBC_Function_Result.x[0] + RBC_Function_Result.x[1] - 1:
        Local_time = RBC_Function_Result.x[0] + RBC_Function_Result.x[1] - 1

    # NBL: Green then Yellow
    # Yellow time of NBL = 3
    # Red Clearance time of NBL = 0
    if Local_time < RBC_Function_Result.x[0] - 3:
        NBL.SetAttValue("SigState", 3)  # Green
        current_phase[0] = 1
    elif RBC_Function_Result.x[0] - 3 <= Local_time < RBC_Function_Result.x[0]:
        NBL.SetAttValue("SigState", 4)  # Yellow

    # SB: Green then Yellow
```

```
    # Yellow time of SB = 4
    # Red Clearance time of SB = 3
    if RBC_Function_Result.x[0] <= Local_time < (RBC_Function_Result.x[0] +
RBC_Function_Result.x[1] - 7):
        SB.SetAttValue("SigState", 3)  # Green
        current_phase[1] = 1
    elif (RBC_Function_Result.x[0] + RBC_Function_Result.x[1] - 7) <= Local_time <
(RBC_Function_Result.x[0] + RBC_Function_Result.x[1] - 3):
        SB.SetAttValue("SigState", 4)  # Yellow

    # SBL: Green then Yellow
    # Yellow time of SBL = 3
    # Red Clearance time of SBL = 0
    if Local_time < RBC_Function_Result.x[2] - 3:
        SBL.SetAttValue("SigState", 3)  # Green
        current_phase[2] = 1
    elif RBC_Function_Result.x[2] - 3 <= Local_time < RBC_Function_Result.x[2]:
        SBL.SetAttValue("SigState", 4)  # Yellow

    # NB: Green then Yellow
    # Yellow time of NB = 4
    # Red Clearance time of NB = 3
    if RBC_Function_Result.x[2] <= Local_time < (RBC_Function_Result.x[2] +
RBC_Function_Result.x[3] - 7):
        NB.SetAttValue("SigState", 3)  # Green
        current_phase[3] = 1
    elif (RBC_Function_Result.x[2] + RBC_Function_Result.x[3] - 7) <= Local_time <
(RBC_Function_Result.x[2] + RBC_Function_Result.x[3] - 3):
        NB.SetAttValue("SigState", 4)  # Yellow

    # If we hit the end of Ring 1's cycle, set a new barrier time and switch to Ring 2.
    if Local_time == (RBC_Function_Result.x[0] + RBC_Function_Result.x[1] - 1):
        Last_Barrier_Time = Sim_time + 1
        Ring_1 = False

  # --- Ring 2 Logic ---
  # Movements: (WB, WBL, EB). Indices in RBC_Function_Result.x are 5, 6, 7.
  else:
    # Safety/truncation: Ensure after changes in the mid-cycle the Local_time doesn't
exceed the current ring time.
    if Local_time > RBC_Function_Result.x[5] - 1:
        Local_time = RBC_Function_Result.x[5] - 1

    # WB: Green then Yellow
```

```
    # Yeow time of WB = 4
    # Red Clearance time of WB = 4
    if Local_time < (RBC_Function_Result.x[5] - 8):
      WB.SetAttValue("SigState", 3)
      current_phase[5] = 1
    elif (RBC_Function_Result.x[5] - 8) <= Local_time < (RBC_Function_Result.x[5] - 4):
      WB.SetAttValue("SigState", 4)

    # WBL: Green then Yellow
    # Yellow time of WBL = 3
    # Red Clearance time of WBL = 2
    if Local_time < (RBC_Function_Result.x[6] - 5):
      WBL.SetAttValue("SigState", 3)
      current_phase[6] = 1
    elif (RBC_Function_Result.x[6] - 5) <= Local_time < (RBC_Function_Result.x[6] - 2):
      WBL.SetAttValue("SigState", 4)

    # EB: Green then Yellow
    # Yellow time of EB = 4
    # Red Clearance time of EB = 4
    if RBC_Function_Result.x[6] <= Local_time < (RBC_Function_Result.x[6] +
RBC_Function_Result.x[7] - 8):
      EB.SetAttValue("SigState", 3)
      current_phase[7] = 1
    elif (RBC_Function_Result.x[6] + RBC_Function_Result.x[7] - 8) <= Local_time <
(RBC_Function_Result.x[6] + RBC_Function_Result.x[7] - 4):
      EB.SetAttValue("SigState", 4)

    # If we hit the end of Ring 2's cycle, update barrier time and return to Ring 1.
    if Local_time == RBC_Function_Result.x[5] - 1:
      Last_Barrier_Time = Sim_time + 1
      Ring_1 = True

  return (
    Last_Barrier_Time,
    Ring_1,
    RBC_Function_Result,
    End_of_Last_Cycle_Time,
    current_phase,
    Local_time
  )
```