REAL-TIME CONTROL FOR INTERSECTION TRAFFIC SIGNALS

A Dissertation

by

XIAOWEI CAO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Xiubin B.Wang
Committee Members,	Yunlong Zhang
	Alireza Talebpour
	Swaroop Darbha
Head of Department,	Robin Autenrieth

December 2019

Major Subject: Civil Engineering

ABSTRACT

This research focused on proposing a real-time control for intersection traffic signals at different roadway levels, including isolated intersections, corridors, and grid networks. First, the real-time control algorithm (DORAS) for isolated intersections was derived, and was tested on a real-life intersection with actuated control and OPAC. The results showed that DORAS was able to significantly reduce average vehicle delay and stops. Secondly, an advanced control algorithm (MADM) for corridors and grid networks was introduced, which was developed based on DORAS and Max Pressure (MP) method by Varaiya. The algorithm was tested in simulation for corridor and network case with other methods. The results showed that MADM outperformed the chosen real-time control methods, as well as the classic bandwidth maximization method in terms of average vehicle delay and stops. Finally, a relatively independent and isolated chapter, a mathematical algorithm that considers probabilistic vehicle presence was developed to treat the left-turn spillback at intersections given the locations of connected vehicles (CVs). Two consecutive intersections in College Station, TX were tested through simulation, in which the downstream intersection was subject to left-turn spillback. The results showed that the algorithm was able to significantly reduce the average vehicle delay at downstream intersection without impact the performance of upstream intersection. Future work includes testing the proposed real-time control algorithm on real corridors and networks, improving the computational efficiency, and comparing with other real-time control methods.

ACKNOWLEDGMENTS

I would like to thank my dissertation advisor, Dr. Bruce Wang, and the committee members, Drs. Yunlong Zhang, Alireza Talebpour, and Swaroop Darbha, for their patience and support throughout the course of this research.

I am indebted to the graduated members of Dr. Wang's group, including Kai Yin, Lisa Zhong, and Wen Wang, who gave me much needed advice to help me through difficulties. Our group member Chaolun Ma also collaborated on the numerical test about network control, which is incorporated into this dissertation. Thanks also to the transportation program faculty, Drs. Mark Burris, Gene Hawkins, Luca Quadrifoglio, and Dominique Lord for teaching me and making my time at Texas A&M University a rewarding experience. I'd extend my gratitude to the Zachry Department of Civil & Environmental Engineering for financially supporting me through the teaching assistantship during my Ph.D. study.

Last, but not the least, is my deep gratitude to my parents for their patience and love. They are my source of strength.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a dissertation committee consisting of Professor Bruce Wang (chair), Yunlong Zhang and Alireza Talebpour of the Zachry Department of Civil & Environmental Engineering and Professor Swaroop Darbha of the J. Mike Walker ’66 Department of Mechanical Engineering.

The algorithm derivation in Chapter 3 were published in 2017 co-authored with Professor Bruce Wang.

The simulation part in Chapter 4 was conducted in part by Chaolun Ma of the Zachry Department of Civil & Environmental Engineering.

The left-turn spillback estimation in Chapter 5 was published in 2019, and was partly conducted by Jian Jiao of the Zachry Department of Civil & Environmental Engineering.

All the rest work for the dissertation was completed independently by the student.

Graduate study was supported by a teaching assistant fellowship from the Zachry Department of Civil & Environmental Engineering at Texas A&M University.

NOMENCLATURE

DORAS	Dynamic, Optimal, Real-time Algorithm for Signals
ROW	Right of Way
OPAC	Optimization Policies for Adaptive Control
RHODES	Real-time Hierarchical Optimized Distributed Effective System
ACTS	Adaptive Traffic Control Systems
HCM	Highway Capacity Manual
TTI	Texas A&M Transportation Institute
TxDOT	Texas Department of Transportation
FHWA	Federal Highway Administration
TRANSYT-7F	Traffic Network Study Tool Version 7F
PASSER	Progression Analysis and Signal System Evaluation Routine
MP	Max Pressure
MADM	Mixed Algorithm of DORAS and MP
GPS	Global Positioning System
HCS	Highway Capacity Software
NEMA	National Electrical Manufacturers Association
CV	Connected Vehicle
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
SCOOT	Split Cycle Offset Optimization Technique
SCATS	Sydney Coordinated Adaptive Traffic System

CACC	Cooperative Adaptive Cruise Control
CVIC	Cooperative Vehicle Intersection Control
CBD	Central Business District
COM	Component Object Model
vph	Vehicle Per Hour

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

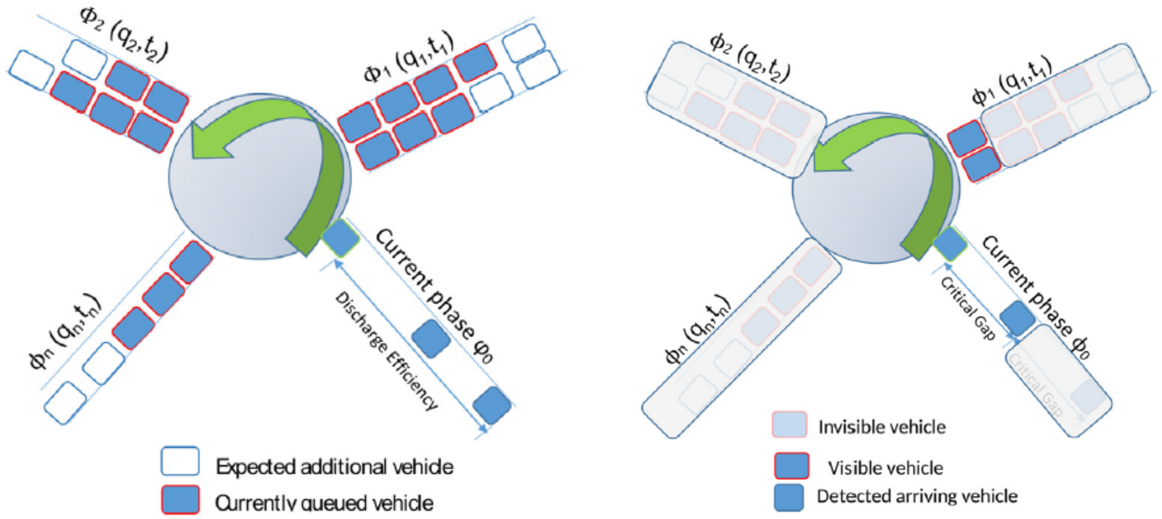
LIST OF TABLES

1. INTRODUCTION

Roadway intersection traffic signal control plays an important role in urban life. Millions of urban travelers experience delay at signalized intersections daily. In 2013, drivers in United States wasted 6.8 billion hours and 3.1 billion gallons of fuel due to traffic congestion [1]. Congestion continues to worsen, with a 400% increase in the total estimated cost of congestion since 1982, and a 28% increase since 2000 [1]. The Federal Highway Administration (FHWA) estimates that approximately half of all congestion is due to special events, poor signal timing, work zones, or traffic incidents [2], suggesting that signal control is a major contributor to urban congestion.

Means of achieving optimal signal control, a fundamental challenge in traffic operations, has been studied for decades. The impact from an improved traffic control is almost immediately observable and tangible. The widespread usage of GPS, cellular phones, sensors, vehicle to vehicle communication, and vehicle infrastructure integration combine to yield a data-rich transportation environment, providing great opportunities for real-time, information-driven intersection traffic control, which were not possible before the digital age. Researchers are able to access real-time traffic arrival or queuing information much more conveniently than ever before.

This research attempts to answer the question: given information about intersection queuing and traffic arrivals, what is the optimal timing policy for signal control at an isolated intersection as well for coordinated arterials? This question has been intermittently approached in a long history of literature and much remains open to further exploration. This dissertation, in general, attempts to propose a real-time, globally adaptable policy for signal control optimization. In contrast, the traditional controls are usually parametric, requiring custom parameters restricted to specific control schemes such as setting up the critical gap for the actuated control. Figure 1.1b illustrates ignorance of most approach traffic in an actuated control compared with a full stream data environment of Figure 1.1a as in an optimum based study.



(a) Stream Data Control

(b) Partial Data Control

Figure 1.1: Full Stream Data Environment versus. Partial Data for Actuated Control.

1.1 Brief History of Signal Control

Signal control has a long history that evolves with traffic and transportation. The first traffic signal was installed at the intersection of Great George Street and Bridge Street in London, England, in 1868 by the railway engineer J. P. Knight [3]. Based on railway controls, the first signals were mechanical semaphores during the day and colored gas-powered lamps at night. The earliest signals were developed and managed by local police, and required an officer to operate the signal and enforce compliance.

The rise of traffic signals is tied to the rapid rise of automotive traffic. In 1914, Cleveland's police department installed a red and green traffic control light at the corner of the 105th street and Euclid Avenue, which is the first permanent installation in the world. In 1917, Detroit police officer made a major improvement in traffic lights, adding a yellow caution light to guide pedestrians and allow traffic to clear the intersection between changes. Later, the first automatic timers was applied to traffic lights in Houston in 1922, and coordinated signals followed shortly after on 16th Street in Washington DC's in 1926 [4]. Electromechanical signals remained the prevalent form of control until the introduction of computer-controlled signals in the late 1960s.

Traffic signal standardization began in 1920 when the American Association of State Highway Officials asked the National Bureau of Standards to create a uniform design code for traffic rules and signs [4]. Standardization of hardware continued with the National Electrical Manufacturers Association (NEMA) standardization of controller hardware in 1976 [5]. The NEMA standard also codified a common phase numbering system, with phases 2 and 6 designated as mainline through movements.

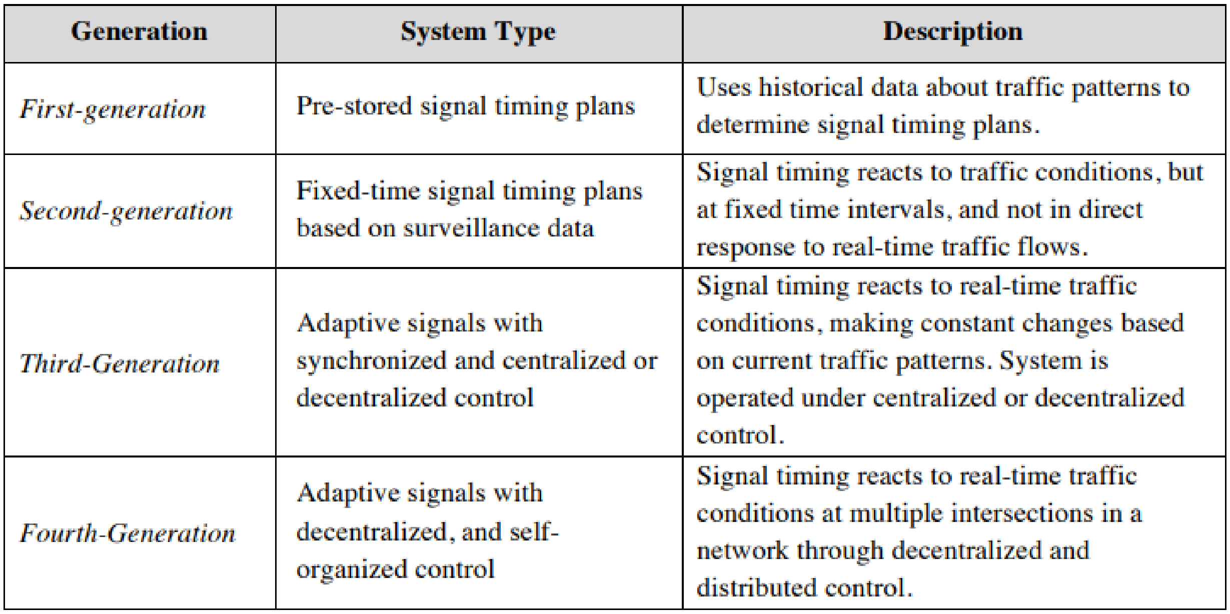
For decades, a tremendous amount of research has been devoted to signal control, a summary of which is seen in Gartner [6].

There are three basic forms of traffic signal control: pre-timed, semi-actuated, and fully-actuated. The latter is further subdivided in to fully actuated with or without volume density control. According to Orcutt [7], pre-timed control is used mainly in the central business district (CBD), especially where a network of signals must coordinated. The pre-timed control functions with pre-determined signal cycle length and split, and thus does not respond to real-time traffic, although literature in this area, such as Webster's method [8], also have had significant impact on the setup of maximum green time of each phase for the actuated control.

Actuated control, based on Orcutt's definition, can respond to actual traffic demand of one or more movements registered by detectors. Using infrastructure-based sensors, such as loop and video detectors, infrared detectors, or radar, activated controllers collect real-time traffic data. If the actuator has the ability to detect all movements at an intersection, the control is said to be fully actuated. Orcutt also states that fully actuated control should normally be used at isolated intersections. The actuated signal control is a typical example of parametric control. Parametric control is operated through improved setup for parameters, often restricted to specific schemes disregarding the global optimally of the employed scheme.

Vehicle-actuated control is either semi-actuated or fully-actuated. The former uses loop detectors implemented on the minor approach; signal switching from the major approach is not actuated unless vehicle arrivals are detected on the minor approach. The latter implements loop detectors on all approaches, an improvement over the former. In the latter, each phase controls the green duration within a range of minimum and maximum limits. Within this range, the signal switch is determined by vehicle arrivals at the advance loop detector. If a vehicle is detected arriving, the green interval is extended by a minimal amount, called the critical gap, just enough to allow the detected vehicle to pass through the intersection. The green indication is terminated either when no vehicle is detected to be arriving within the time of gap or when green time maxes out.

Modern traffic signals leverage sensors to adapt to prevailing conditions at the intersections. Such systems that response to current traffic conditions are known as adaptive traffic control systems (ATCS). ATCS make signal timing decision at short intervals based on real-time traffic conditions. These systems haven been called third-generation systems [9]. ATCS contrast with first-generation systems, which use historical data about traffic patterns to determine signal timing plans. ACTS are also an improvement over the second-generation systems, which make use of signal timing plans based on surveillance data, but are restricted as to how frequently the response is generated [10]. Goel et al. [11] introduced the four signal control generations in details, which are summarized in Figure 1.2.



Generation	System Type	Description
First-generation	Pre-stored signal timing plans	Uses historical data about traffic patterns to determine signal timing plans.
Second-generation	Fixed-time signal timing plans based on surveillance data	Signal timing reacts to traffic conditions, but at fixed time intervals, and not in direct response to real-time traffic flows.
Third-Generation	Adaptive signals with synchronized and centralized or decentralized control	Signal timing reacts to real-time traffic conditions, making constant changes based on current traffic patterns. System is operated under centralized or decentralized control.
Fourth-Generation	Adaptive signals with decentralized, and self-organized control	Signal timing reacts to real-time traffic conditions at multiple intersections in a network through decentralized and distributed control.

Figure 1.2: Signal System Types.

1.2 Research Objectives

This dissertation investigates possible adaptive signal control strategies, as well as methods for coordinated arterial signal control.

The specific objectives of this research are as follows:

1. To develop a real-time dynamic intersection control algorithm that can minimize the overall intersection delay.

2. To apply the proposed algorithm to isolated signalized intersection case and compare with some of the most popular adaptive signal control methods.

3. To develop a real-time control method for coordinated arterial network to minimize delay/stops of a whole system.

4. To test the proposed algorithm for coordinated arterial network and compare with other coordinated signal control methods.

5. To propose a control strategy to solve the left-turn spillback problem for approaches with high left-turn volume.

6. To prove the accuracy of the proposed left-turn spillback algorithm with varying traffic and geometry conditions, and to test it in real case.

1.3 Research Contributions

It is intended that this research will produce several major contributions to the state-of-art knowledge of traffic control. These contributions include:

1. Development of a dynamic, optimal and real-time traffic signal control algorithm for isolated intersections. A continuous model is developed to characterize the queuing process as a function of control policy and vehicle arrival processes. The continuous model applied to the general intersection appears to be the first in literature and allows to examine the problem in a way that previous literature does not allow, such as optimal condition.

2. Development of algorithms for coordinated arterial network signal control. The proposed algorithm for isolated intersections is combined with another popular method to generate a mixed method. The mixed method is able to control the arterial network more efficiently than several existing methods in isolation in terms of average vehicle delay.

3. Development of a left-turn spillback control algorithm. The algorithm will be based on estimating the accurate left-turn spillback probability, which can be achieved by the proposed equation. The algorithm is designed to apply to real cases(downstream/upstream intersection) to effectively reduced the extra delay caused by left-turn spillback.

1.4 Report Organization

The remainder of the dissertation is organized as follows.

- Chapter 2 summarizes the review of literature, Some of the investigated research topics include actuated control, adaptive control, and coordinated signal control.

- Chapter 3 is about isolated intersection control. The basic idea is to first model intersection vehicle delay by assuming continuous vehicle arrival and departure, and presenting the optimal condition for green signal switch. Two proposed numerical algorithms, optimum-based (DORAS) and queue-based (DORAS-Q), are expected to generate optimal signal switching policy. These two proposed algorithms are compared with OPAC and actuated control under different traffic situations in simulation.

- Chapter 4 extends the results and insights from Chapter 2 by developing coordinated signal control algorithm, the purpose of which is similar to that of isolated intersections, which is to minimize overall/network average vehicle delay or stops. Four algorithms are compared in a 3x3 arterial network grid under different traffic volume cases, including a proposed mixed algorithm between DORAS-Q and MP.

- Chapter 5 investigates the common left-turn spillback problem. This chapter diverges from earlier chapters as it deals with improvements to existing practices based on new technology.

This chapter first introduces calculation of left-turn spillback probability. Then it proposes a conceptual control algorithm focusing on solving the left-turn spillback problem, which could reduce the average vehicle delay for certain intersections. Different simulation cases are implemented to test the proposed algorithm.

- Chapter 6 summarizes the results and presents conclusions, then briefly discusses the limitations of the study, and presents formulation and directions for future work.

2. LITERATURE REVIEW

Traffic signals are likely the most intensively studied and argued-over topic in the history of traffic engineering. A vast literature exists on this research area and a very large number of engineers and scientists have made great endeavor to study and apply it, some even spending their entire academic careers on this problem.

In this chapter, we first review the literature to show the status of research trends in order to position the dissertation study, leaving relevant literature regrading the specific study topics to their respective chapters later on, such as optimal control on isolated signal and coordinated or network signal control.

2.1 Actuated Signal Control

The research area of actuated vehicle control is literature-rich largely because for decades this method has been considered among the most adaptive and most practically feasible.

Newell [12] investigates the vehicle-actuated signal at the intersection of two two-way streets (four-way intersection) at which there is no turning traffic. The main conclusion is that the high efficiency of a vehicle-actuated signal as compared with a fixed-cycle signal for one-way streets does not necessarily extend to the case of two-way streets. In particular, if the flows in opposite direction on the two-way street are nearly equal and the intersection is nearly saturated, then it is very inefficient (even worse than a fixed-cycle) for a vehicle-actuated signal to follow a policy of holding a signal green until the last of the two discharging queues has vanished. (It is even worse to switch when only one queue has vanished.) In all cases, the competition between incompatible desirable policies for the two opposing traffic streams invites compromise strategies that give considerably higher delays on all approaches than occur on one-way streets.

Starting in the 1960s and persisting until now, many analytically rigorous studies have been conducted on the subject of traffic signal control. Cowan [13] proposes an improved model for signalized intersection with vehicle-actuated control by assuming more general and realistic traffic arrival processes. The results show that his model is able to satisfy both the random "bunch-gap" structure of traffic and the fact that vehicles have non-zero length. Dunne [14], on the other hand, applied a binomial process to generated vehicle arrivals to further improve the delay formulas for an isolated intersection. The results indicate that the binomial arrival pattern is a desirable one, particularly in view of the relatively simple analytic signal control strategies.

Later, Akcelik [15] introduced an analytical method for estimating average green times and cycle time at vehicle actuated signals. He discusses the arrival headway distributions as well, since the estimation of arrival headways is fundamental to the modeling of actuated signal timing. This method provides essential information for predicting the performance characteristics of intersections controlled by actuated signals and for investigating the optimization of actuated controller settings.

Lin [16] also developed a model for estimating the average green durations that result from full-actuated signal control. The model is developed primarily on the basis of probabilistic interactions between traffic flows and the control. The method has proved to be readily and manually used. It can also be implemented in the form of a simple computer program that requires limited computing facilities.

A few studies in particular examine the optimal size of the critical gap in the actuated signal control (Newell [17] and Wang et al. [18]). Critical gap is the control parameter determining closeness of two consecutive arrivals, disregarding traffic situations at other intersections. In practice, the critical gap is typically set up between 2 and 3 seconds; however, no matter how "optimal" the critical gap is, it is a parameter restricted to this particular control scheme, rather than being a candidate for global optimality of the intersection performance.

2.2 Adaptive Signal Control

Adaptive signal control is about adapting signal by at least partially considering the near-future data to incrementally optimize the local signal timing, mainly about cycle length and splits, not necessary for global optimum.

Miller's work [19] represents one of the early adaptive signal control methods. Later, Ross et al. [20] builds upon Miller's foundation by proposing a computer-control scheme that concentrates upon providing real-time, traffic-responsive correction for a critical intersection. The critical intersection represents a point in the network at which primary driver routes or those where congestion has arisen for other reasons. The sub-optimal scheme of this method attempts to minimize the total, measured in vehicle-seconds. Within upper and lower limits on the cycle length, the control method is able to provide variable split and cycle length for traffic signal in response to traffic needs.

A summary of adaptive signal control is respectively seen in an NCHRP Synthesis Report by Stevanovic [21] as well as in Zhao et al. [9], in which the few famous adaptive systems are summarized, including OPAC developed by Gartner [6, 22] and RHODES developed by Head et al. [23, 24].

Notably, Dunne and Potts [25, 26] propose a control method based on queue lengths of approaches. The algorithm is represented by computed control functions depending on the number of vehicles queued and demanding service at the intersection. It shows that the algorithm gives dynamic rather than static control in that traffic demand drives the traffic signal phase by phase. A queue clearance policy is analyzed as well in which signal switches when the queue is cleared for the current phase and when the signal has not reached its green maximum.

Zijverden and Kwakernaak [27] propose a rolling horizon short-term optimization method, which is perfected along that line of OPAC series.

Worthy of special mention, there is an branch of literature investing optimal signal control that deals with tapping the maximum potential for handling traffic at an intersection. Optimal control here is not about optimal fixed signal timing, or optimal actuated scheme, but optimal over all possible schemes given practical constraints such as min/max green times. These studies hold the greatest promise to improve the adaptive signal control to its maximum potential. Specially, Gartner [6, 28] proposes a dynamic programming model, later termed OPAC I in [29], that defines a control policy to be a series of signal switch points. Optimization of these switch points is equivalent to optimizing intersection control. This definition of policy is also adopted by Sen and

Head [30]. They propose a general purpose algorithm for real-time traffic control to optimize various performance indices such as delay, strops and queue lengths. Furthermore, optimal phase sequencing is a direct by-product of their method. The results show consistent reduction in delay by adopting the proposed algorithm compared with fully and semi-actuated control.

The control progresses by sequentially determining signal switch points according to traffic situation, which is consistent to the proposed methodology in this thesis. Gartner's [6] model is intended for implementation in a real time, dynamic traffic situation. The optimal policy defined in OPAC I presents a critical important benchmark to other approximation algorithms. However, the optimal decision at any moment depends on future optimal decisions, which is hard to implement when the optimization horizon is long and the state space as partially represented by the queue lengths at the intersection is large. Therefore, OPAC III and OPAC IV are proposed [6, 28, 29] as simplified optimization. OPAC II sets a stage of fixed length for optimization and moves the control forward stage by stage, where each stage roughly has the length of a signal cycle. OPAC IV, however, moves the control forward on a rolling time horizon using the virtual-fixed-cycle concept. In a alternative way, some earlier literature tries to examine the structure of the optimal policy with restrictive assumptions intersection geometry.

In particular, Grafton and Newell [31] examines the optimal policy by assuming deterministic, uniform, and continuous traffic arrivals and departures at an intersection between two one-way streets, which reveals that the queue clearance policy proposed by Dunne and Potts [26] may not be optimal, especially for a large initial queue situation. The authors find that the queue clearance policy is sub-optimal when the saturation flow rates between approaches have a significant difference. Grafton and Newell's findings may be easily explained in this thesis in the following chapters. In addition, Sen and Head [30] apply a discrete dynamic programming (DP) model to the intersection control, and only exhaustive search is implied to find the optimal solution.

As early as 2004, Huang and Miller [32] proposed the concept of a smart intersection making use of wireless communication. A simple and reliable protocol for electronic traffic signaling systems was the basis for construction of a sample application: a red-light alert system. Although the system was not tested at the field intersection, this work provides the motivation to explore the area of wireless technologies for adaptive traffic control systems.

An emerging technology, connected vehicles (CVs), can communicate with each other (V2V) and with the infrastructure (V2I) through dedicated short-range communications (DSRC). Connected vehicles combines several emerging technological advances, such as advanced wireless communications, on-board computer processing, advanced vehicle sensors, GPS navigation, and smart infrastructure to provide a networked environment. Compared to the traditional detectors, CV technology can provide real-time information (such as, position, speed, acceleration, and other traffic data) necessary for evaluating traffic conditions on a road network. Connected vehicle technology has the potential to reduce travel time by 37%, reduce emissions by 30% and improve safety indicators by 45% [33].

Collecting connected vehicle data is significantly less expensive than installing and maintaining a suite of detectors (e.g., loop, radar or video). If one or more connected vehicles cannot communicate with the infrastructure due to one communication failure or the other, it will decrease the market penetration rate only on a road network and will not have a large impact to the total signal control system performance. If the infrastructure is out of order by chance, the intersection control strategy can restore to the traditional actuated or fixed time signal control quickly [34].

Several studies have been implemented on the applications of CV technology in adaptive traffic signal control. Some papers [34, 35] concentrated on phase optimization-based methods to optimize the signal control and other [36–39] employed queue-based methods to model and achieve the signal control system optimization

Development a new real-time, dynamic signal control algorithm involves translation of delay minimization directly into intersection efficiency maximization in order to determine optimal switch points. These derived optimal conditions for signal switch suggest a new direction for control algorithms.

2.3 Coordinated Signal Control

Coordinated signal control is used to synchronize multiple intersections to enhance the operation of one or more directional movements in a roadway system. Coordinated signal control is typically applied on corridors or arterials with closely spaced intersections, and where there is evidence of desire for traffic platoons. Although there are numerous objectives for the coordination of traffic signals, a common goal is given briefly in the National Report Card [40]: The intent of coordinating traffic signals is to provide smooth flow of traffic along streets and highways in order to reduce travel times, stops and delays.

Other advantages of signal coordination include [41]:

1. A higher level of traffic service in terms of higher overall speed and less stops.
2. More uniform vehicle speeds.
3. Fewer accidents because platoons arrive in green time, therefore reducing the chance of collisions.
4. Greater obedience to the signal commands from both motorists and pedestrians.
5. Through traffic tends to stay on the arterial streets instead of diverting onto parallel minor streets in search of alternative routes.

Coordinated signal control has three fundamental components: a cycle length, the splits at each intersection, and a set of offsets that controls the starting times of movements relative to other signals in the system.

Traditional coordination basically adds a layer of signal controller logic to the basic actuated logic. The method requires a consistent cycle length for the corridor, as well as splits and offsets for each timing plan.

Due to the development of adaptive signal control, however, some advanced methods have been emerged, which are able to maintain coordination without having to explicitly define cycle length, splits or offsets. They are capable of adjusting timing plans based on measured traffic conditions.

The detailed discussion of these two categories of coordinated signal control methods are left to their respective chapters and sections.

2.4 Queue Length Estimation and Spillback

The main focus of this section is improved left-turn queue estimation using the connected vehicle technology with an application to signal timing. A number of studies have been conducted on queue length estimation with the connected vehicle (or probe vehicle) technology.

Prior to the advent of connected vehicles, researchers had studied queue estimation using traditional technologies, such as loop detectors [42, 43], which are mainly about cross-section data. Today's new technologies, such as connected vehicles, can generate trajectory data [44], which brings new opportunities and presents new challenges to modeling. Comert and Cetin's [45, 46] proposed analytic formulas are able to estimate expected queue length and its variance based on two variables, the location of the last connected vehicle and the market penetration ratio, along with the probability distribution of the arrival process. Tiaprasert et al. [38] proposed another mathematical model for queue length estimation that does not require queue characteristics to be known and applied a discrete wavelet transformation to enhance the accuracy and consistency of the estimation under differently equipped vehicle penetration ratios. An event-based model by Li et al. [47] was developed together with a data fusion method that combined both trajectory data and loop detector data. Queue length can also be estimated by shockwave methods with trajectory data, as proposed by Cheng et al. [48]. Badillo et al. [49] presented an algorithm that combined loop detector data with real-time vehicle probe data from connected vehicles for the queue length estimation.

A few studies have been made on left-turn capacity, spillback and the resulting through-traffic blockage (also referred to as left-turn blockage). One of the early works was performed by Messer and Fambro [50]. The relationship between left-turn capacity and the geometric layout was studied and guidelines to avoid spillbacks were developed. The problems with short left-turn bays were further investigated by Yin et al. [51] and Zhang and Tong [52]. To model left-turn blockage, a cell transmission model was developed in Wang et al. [53].

For more information about vehicle connectivity and potential challenges, readers are referred to Lu et al. [54], who gave a review of the state-of-the-art literature. In a more general signal control area, new signal control strategies have been developed to take information from connected vehicles into account, such as the study conducted by Guler et al. [39]. Goodall et al. [5] developed a predictive microscopic simulation algorithm with a rolling-horizon strategy and used real data to optimize an objective function over a short period of time. Feng et al. [55] proposed a framework that combined adaptive signal control with the dilemma zone protection, multi-modal signal priority and coordination. They also used detector data to improve the framework's performance under low market-penetration rates. Zohdy et al. [56] developed a signal control algorithm based on the latest cooperative adaptive cruise control (CACC) system to reduce intersection delay and fuel consumption. Lee et al. [57] further presented a cooperative vehicle intersection control (CVIC) algorithm that does not even require a traffic signal. In comparison, our paper is focused on left-turn traffic spillback estimation in a new data environment of connected vehicles.

3. MODELING ISOLATED INTERSECTION FOR OPTIMAL SIGNAL CONTROL*

What is the relationship between isolated and networked signal control? The literature has not clarified this relationship. Neither do the control mechanisms indicate any connection between the two. We believe the two shall and must have an inherent connection that ultimate solution of one implies solution to the other.

This chapter tries to develop a real-time dynamic signal control algorithm for an isolated intersection to minimize the overall vehicle waiting time and stops by selecting the optimal phase switching points, given all the necessary constraints that are shown below.

Notations

$(\mathbf{n}, t)$ intersection state variable, where $\mathbf{n} = n_i$ with n_i representing queued vehicles for approach t. t represents the time before the end of the control horizon.

θ control policy that determines signal switch from one approach to another during the entire control horizon. With slight notional abuse, $\theta(t)$ is used for the specific signal control at time t.

Φ set of phases. N is the total number of phase in Φ. The phases are indexed from 0 to $N-1$ sequentially according to the order in a signal cycle with 0 being the current one with green signal. The index is therefore on a rotational basis.

ϕ_k set of approaches on phase $k \in \Phi$. $k = 0$ means current approach with ROW.

*Reprinted with permission from “Dynamic optimal real-time algorithm for signals (DORAS): Case of isolated roadway intersections”by X. B. Wang, X. Cao, and C. Wang, 2017. *Transportation Research Part B: Methodological*, vol. 106, pp. 433-446, Copyright 2017 by Elsevier.

$D_i^\theta(t_1, t_2)$ total arrivals of vehicles from approach i during time interval (t_1, t_2) under policy θ. $D_i(\Delta t)$ means total arrivals in the past period Δt, which slight notional abuse.

Ω set of constraints for signal timing, which includes minimum/maximum green times, sequence of phases, etc.

$\hat{q}_i$ currently observed queue length for approach i at the decision point.

Δq_i expected additional vehicles to be discharged beyond $\hat{q}_i$ for approach i during a normally run phasing in the current cycle forward.

q_i the total number of vehicles discharged for approach i during the current signal cycle that starts from the current phase, $q_i = \hat{q}_i + \Delta q_i$.

s_i saturation flow rate for approach i.

L green time loss due to signal switch, referred to as all-red interval in this paper.

ε_0 current intersection efficiency under the current green phase.

ε_1 next (or, switch-to) phase intersection efficiency equivalent.

$\lambda_i^\theta(t)$ continuous vehicle arrival rate from approach i at time t under policy θ.

$d_i^\theta(t)$ vehicle discharge rate approach i at time t. $d_i(t) = \lambda_i(t)$ when approach i has the ROW and when no queue exist for approach i. $d_i(t) = 0$ when approach i does not have the ROW. $d_i(t) = s_i(t)$ when approach i has the ROW and when a queue exists being cleared. In summary, the controller assigns situational value to d_i.

$t_\dagger$ critical point of time at which the signal switches from one phase to the next.

$w^{\theta}(\mathbf{n}, t)$ total intersection vehicle waiting time from state $(\mathbf{n}, t)$ till the end of the control under policy θ.

$t+$ limit to time $t = \lim_{\Delta \to +0} t + \Delta$.

$t-$ limit to time $t = \lim_{\Delta \to +0} t - \Delta$.

$\pi_i(t_k, t_{k+1})$ green duration from time t_k to t_{k+1} for phase i. In phase i, the pre-determined set of approaches get the right of way.

α_j/β_j minimum/maximum green time limit for phase j.

3.1 Theoretical Framework of Intersection Control

The optimal policy is clarified first. The control policy consists of a series of signal phases that last till the end of the control horizon, which sequentially takes ROW (except for the all-red interval π_R). Policy θ is defined as a set $\{\pi_1(t_1, t_2), \pi_R(t_2, t_2 + L), \pi_2(t_2 + L, t_3), \pi_R(t_3, t_3 + L), \pi_3(t_3 + L, t_4) \ldots\}$, each $\pi_i(t_k, t_{k+1})$ donates a green indication for phase $i \in \Phi$ for the time interval (t_k, t_{k+1}) except for the all-red π_R. Beware that all phases in Φ are mutually exclusive, meaning conflict of ROW between any two of them. Under a policy θ, $\pi_i(t_k, t_{k'})$ and $\pi_i(t_{k'}, t_{k+1})$ are not considered two phases but one $\pi_i(t_k, t_{k+1})$, where $t_{k'} \in (t_k, t_{k+1})$. Note that phase π_R is specially for the all-red interval, an interval that follows almost every signal switch from other phases. Or equivalently, a control policy may be represented by a series of time points for signal switches between phase, e.g. $\{t_1, t_2, \ldots, t_n\}$. The key is to find the signal switch points in order to determine the control policy. In the modeling that follows, it is assumed that the time clock goes from time t backward to time zero, where time zero represents the end of the control horizon. This also means that in a time interval (t_k, t_{k+1}), we have $t_k > t_{k+1}$. In Figure 3.1, a policy θ is illustrated, in which $t^{\dagger}$ is a switch point for phase π_k to switch to phase π_{k+1}. Note that the developed policy is general, which allows, but does not require, an all-red interval for a phase change. Additionally, $s(\mathbf{n}, t)$ is used for an intersection state to start with, in which $\mathbf{n}$ represents queues for all approaches at time t while t is the time prior to the end of the control horizon. Theoretically, a control policy θ is the function of current state $s(\mathbf{n}, t)$, and should be written as $\theta(n, t)$. To simplify the presentation, it is simply denoted as θ.

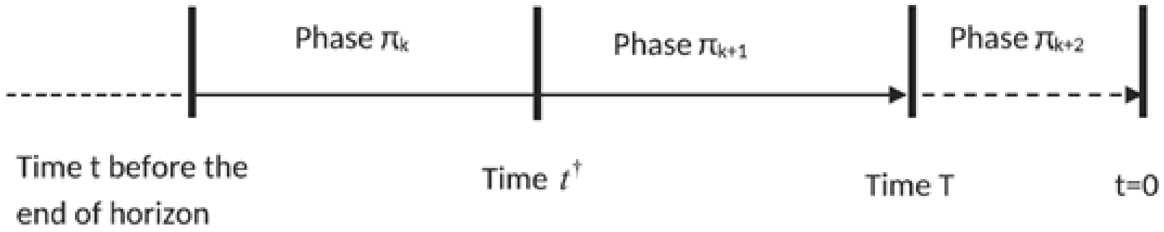


Figure 3.1: A Policy θ Over a Control Horizon.

The intersection signal control problem may be described in Equation 3.1 below, which assumes policy θ as illustrated in Figure 3.1.

$$\begin{aligned}
w^\theta(\mathbf{n}, t) &= \sum_{\forall i} \int_T^{t^\dagger} \Big(n_i + \int_{t^\dagger}^{t} \lambda_i^\theta(\tau_1) d\tau_1 - \int_{t^\dagger}^{t} d_i^\theta(\tau_1) d\tau_1 + \int_{\tau}^{t^\dagger} \lambda_i^\theta(\tau_1) d\tau_1 - \int_{\tau}^{t^\dagger} d_i^\theta(\tau_1) d\tau_1 \Big) d\tau \\
&+ \sum_{i} \int_{t^\dagger}^{t} \Big(n_i + \int_{\tau}^{t} \lambda_i^\theta(\tau_1) d\tau_1 - \int_{\tau}^{t} d_i^\theta(\tau_1) d\tau_1 \Big) d\tau \\
&+ w^\theta \Big(\Big(\mathbf{n} + \int_T^t \lambda_i^\theta(\tau_1) d\tau_1 - \int_T^t d_i^\theta(\tau_1) d\tau_1 \Big), T \Big). \qquad (3.1)
\end{aligned}$$

where $w^\theta(\mathbf{n}, t)$ is the total intersection vehicle waiting time from time t to the end of control time horizon, given a green phase π_k at time t, which will change to phase π_{k+1} at time point $t^\dagger$, where $t^\dagger$ is within $[t, T]$. Here $t^\dagger$ is called a pivotal point of signal. Additionally green phase π_{k+1} changes to π_{k+2} at time T. $w^\theta(\mathbf{n_0}, T)$ is the salvage waiting time at time T with a resultant state $(\mathbf{n_0}, T)$. For modeling, t and T are both assumed to be given so that we mat study the condition for $t^\dagger$. $\lambda_i^\theta(t)$ is the arrival rate from approach i under policy θ, usually equal to the actual arrival rate $\lambda_i(t)$, unless the intersection queuing capacity restricts arriving vehicles from entering the queue so that those vehicles are lost to elsewhere. $d_i^\theta(t)$ is the discharge rate under policy θ for approach i. d_i^θ takes the following situational values:

- Case I: $d_i^\theta(t) = s_i$, the saturation flow rate when the signal is green AND when a queue exists for approach i.
- Case II: $d_i^\theta(t) = 0$ when the signal is red for approach i.
- Case III: $d_i^\theta(t) = \lambda_i^\theta(t)$, the arrival rate when the signal is green and when no vehicle queue is present at the intersection for approach i.

Because this methodology is developed for a full information environment, the departure rates are assumed to be continuously monitored and be know through sensors. If technologies are able to monitor the departure situation to tell if the departure line is blocked by downstream vehicles, more situational values may be assigned to d_i^θ, which is beyond the normal discussion of this dissertation. Another notion, under any control policy θ the departure rate d_i^θ is set such that the cumulative departure is no more than the cumulative arrivals at any given time, that is, $n_i +$ $+ \int_x^t \lambda_i^\theta(x)dx - \int_x^t d_i^\theta(x)dx \geq 0$, $\forall \tau < t$, $\forall i$, starting from an initial state $(\mathbf{n}, t)$. Readers shall according understand the valuation of $d_i^\theta(t)$. For simplicity, no notation for how to get values for $d_i^\theta(t)$ in Equation 3.1.

In Equation3.1, the first term is for waiting time after the pivotal point $T^\dagger$ till T when green phase switches from π_k towards π_{k+1}. The big parenthesis with integral is for the total waiting time vehicles a time $\tau \in (t^\dagger, T)$. The second integral is for waiting time before the pivotal point $t^\dagger$ till t. The third term is a salvage value term, which also has to do with the choice of pivotal point $t^\dagger$ because $t^\dagger$ results in the queue $\mathbf{n_0}$ at time T.

It appears that a solution approach is to find the optimal pivotal point $t^\dagger$ backward from very end of the solution horizon, assuming a total cost of queues at the end of the optimization is zero according to the traditional dynamic programming technique, which mimics that in [28] and [30]. First, the signal switch point is examined. Optimal pivotal point $t^\dagger$ is a time point such that shifting it backward or forward would both increase the total waiting time, which leads to the optimal condition below.

Taking first order derivative of Equation 3.1 over $t^\dagger$ gives the following results,

$$\frac{\partial w^\theta(\mathbf{n},t)}{\partial t^\dagger} = \sum_{\forall i} \int_T^{t^\dagger} [-(\lambda_i^\theta(t^\dagger+) - d_i^\theta(t^\dagger+)) + \lambda_i^\theta(t^\dagger-) - d_i^\theta(t^\dagger-)]d\tau + \frac{\partial w^\theta(\mathbf{n_0},T)}{\partial t^\dagger}. \quad (3.2)$$

In the above, $t+$ and $t-$ are used as defined earlier. Because both λ and d are assumed continuous, $w^\theta(\mathbf{n},t)$ is differentiable at $t^\dagger$. Naturally, the arrival/departure rates at both sides of time $t^\dagger$ under policy θ by $\lambda^\theta(t^\dagger+)$ vs. ${}^\theta(t^\dagger-)$ and $d^\theta(t^\dagger+)$ vs. $d^\theta(t^\dagger-)$, respectively. Equation 3.2 becomes Equation 3.3, which essentially establishes the equivalence between maximizing intersection (net) throughput and minimizing intersection vehicle delay.

$$\begin{aligned} \frac{\partial w^\theta(\mathbf{n},t)}{\partial t^\dagger} &= \sum_{\forall i} \Big[-\underbrace{(\lambda_i^\theta(t^\dagger+) - d_i^\theta(t^\dagger+))} + \underbrace{\lambda_i^\theta(t^\dagger-) - d_i^\theta(t^\dagger-)} \Big](t^\dagger - T) + \frac{\partial w^\theta(\mathbf{n_0},T)}{\partial t^\dagger} \\ &= 0. \end{aligned} \quad (3.3)$$

Clearly, Equation 3.3 is identical to Equation 3.4.

$$\sum_{\forall i}[d_i^\theta(t^\dagger+) - \lambda_i^\theta(t^\dagger+)] = \sum_{\forall i}[d_i^\theta(t^\dagger-) - \lambda_i^\theta(t^\dagger-)] - \frac{w^{'\theta}(n_0,T)}{t^\dagger - T}. \quad (3.4)$$

Equation 3.4 represents a necessary condition for the general case of optimal signal switch. The left side of Equation 3.4 is the net discharge rate under the current phase while the right side is the equivalent net discharge rate of the switch-to phase. By the term equivalent, it means to include the converted rate from salvage function $w^{'\theta}(n_0,T)$, the last term in Equation 3.4. The pivotal point $t^\dagger$ shall be chosen such that the net gain of the total vehicle discharge rate from the before to after switch shall sum up to off set the resulting marginal change in the subsequent intersection waiting time. To better understand it, two cases are shown below.

3.1.1 A Special Case

Just for obtaining some insights, assume that the last term in Equation 3.3, the salvage value $\frac{\partial w^\theta(\mathbf{n_0},T)}{\partial t^\dagger} = 0$, which gives rise to the following equation.

$$\frac{\partial w^\theta(\mathbf{n},t)}{\partial t^\dagger} = \sum_{\forall i} \Big[-\underbrace{(\lambda_i^\theta(t^\dagger+) - d_i^\theta(t^\dagger+))} + \underbrace{\lambda_i^\theta(t^\dagger-) - d_i^\theta(t^\dagger-)} \Big](t^\dagger - T) \tag{3.5}$$

In this special case, when the first order derivative is zero, the second order derivative is also zero, which means indefinite regarding whether $t^\dagger$ is a minimum point. Therefore, we have to further examine the first order derivative. Assume the arrivals $\lambda_i^\theta(t^\dagger+) = \lambda_i^\theta(t^\dagger-)$ for all i without losing much generality. Whenever we have $\sum_{\forall i} \lambda_i^\theta(t^\dagger + \delta) < \sum_{\forall i} \lambda_i^\theta(t^\dagger - \delta)$, where $\delta \to 0$ and $\delta > 0$, signal switch at time $t^\dagger$ will lead to less total waiting time than otherwise. Therefore, $\sum_{\forall i} \lambda_i^\theta(t^\dagger + \delta) < \sum_{\forall i} \lambda_i^\theta(t^\dagger - \delta)$ for very small $\delta > 0$, is a necessary and sufficient condition for signal switch at time $t^\dagger$. This concludes that s switch shall result in a higher intersection discharge rate.

3.1.2 The General Case

In the general case, the following is a sufficient and necessary condition in light of the insights from the special case above:

$$\sum_{\forall i}[d_i^\theta(t^\dagger + \delta) - \lambda_i^\theta(t^\dagger + \delta)] - \sum_{\forall i}[d_i^\theta(t^\dagger - \delta) - \lambda_i^\theta(t^\dagger - \delta)] + \frac{w'^\theta(n_0, T)}{t^\dagger - T} < 0. \tag{3.6}$$

Translate this observation into the case in which an effective time loss exists in the form of an equivalent all-red period due to signal switch. In this case, the switch-to phase i s ALL-RED, implying immediate switch-to efficiency is zero, *i.e.* $\sum_{\forall i} d_i^\theta(t^\dagger-) = 0$. Within a certain time period to come, the average discharge rate equivalent shall be larger than that prior to signal switch when the objective function is to minimize the total vehicle delay, which is expressed in Equation

3.7 below ad further simplified equation from Equation 3.6.

$$\sum_{\forall i} d_i^\theta(t^\dagger +) < -\frac{w'^\theta(\mathbf{n_0}, T)}{t^\dagger - T}. \tag{3.7}$$

We call $-\frac{w'^\theta(\mathbf{n_0}, T)}{t^\dagger - T}$ as *switch-to efficiency equivalent*. Note that $+\frac{\partial w^\theta(\mathbf{n_0}, t)}{\partial t^\dagger}\Big|_{t^\dagger - T}$ is usually negative (considering the time is counted backward), which means earlier switch allows earlier release of the waiting vehicles at subsequent phases. $-\frac{\partial w^\theta(\mathbf{n_0}, t)}{\partial t^\dagger}\Big|_{t^\dagger - T}$ is therefore the (marginal) net gain from signal switch. Equivalently, if the current phase maintains a higher rate, the green signal shall not switch to the next phase. Therefore, Proposition 1 below is proposed to summarize the findings so far.

Proposition 1. *At roadway intersection signal control, where all vehicles have equal weight and where the objective is to minimize the total vehicle waiting time at the intersection, the green indication switches if and only if the current (e.g. ongoing) service rate is less than the switch-to efficiency equivalent.*

Proposition 1 illustrates the optimal control structure. In other words, green signal switch shall lead to a higher intersection service rate *equivalent*.

In next section, discussion of the calculation of the current rate and switch-to rate equivalent will be given in details. The derivations have used continuous arrivals and departures because the author believes they can approximate the actual discrete processes well enough. In other words, any discrete process may be approximated by a continuous one of vehicles without causing nontrivial differences. In the subsequent algorithms proposed, this study will have to resort to approximations using discrete measures for practical implementations.

3.2 Efficiency Equivalent Marginal Effect

This section discusses the intersection efficiency equivalent, $\frac{\partial w^\theta(\mathbf{n_0}, t)}{\partial t^\dagger}\Big|_{t^\dagger - T}$ in Equation 3.3, which represents the marginal effect of the pivotal point $t^\dagger$ on subsequent phases' waiting time. Here the entire remaining section of the full signal cycle starting from, but not including, the current phase in green is treated as a big virtual phase considering a fixed sequence of phases. The marginal effect of $t^\dagger$ on vehicle waiting time as calculated below is over this period also with the last all-red at the end of the cycle being excluded. The marginal effect beyond the current cycle is assumed to be unclear and is expected zero. However, one can extend in a similar fashion this period of consideration till a certain phase beyond the current cycle as long as the marginal effect on the phases considered is non-trivial and is quantifiable, which is beyond the scope of this dissertation.

Simply but, the marginal effect of $t^\dagger$, which can be certain at $t^\dagger$, is mainly due to the (earlier release of) queued vehicles. If a signal switch point is made earlier by a small amount of Δ, the total reduced vehicle delay at the subsequent phase in the case of a fixed sequence of phases may be approximated by $\sum_{i\neq 0} q_i^\theta(t_i) * \Delta$, where q_i^θ represents the total (potentially) *queued* vehicles to be cleared for phase i during a normal run cycle under policy θ. $\sum_{i\neq 0} q_i^\theta(t_i) * \Delta$ means that all the observed and forecast queues for approaches in the current cycle forward that are cleared during the phases may be cleared earlier by Δ if the signal switches earlier by this amount Δ and if each subsequent phase runs as it should be running. Not that policy θ does not necessary guarantee clearance of queued vehicles for all approaches. On the other hand, θ may dictate clearance of more vehicles than those queued in order for a higher intersection efficiency. However, q_i^θ *normally* only represents a subset of queued vehicles that are cleared during phasing under policy θ.

The marginal effect of Δ is approximated in the following way. w' in the right hand of Equation 3.2 may be re-written as follows.

$$\begin{aligned} w' &= w^\theta\Big(\Big(\mathbf{n} + \int_T^t \lambda_i^\theta(\tau_1)d\tau_1 - \int_T^t d_i^\theta(\tau_1)d\tau_1\Big), T\Big) \\ &= w^\theta\Big(\Big(n_i + \int_T^{t^\dagger} \lambda_i^\theta(\tau_1)d\tau_1 - \int_T^{t^\dagger} d_i^\theta(\tau_1)d\tau_1 + \int_{t^\dagger}^t \lambda_i^\theta(\tau_1)d\tau_1 - \int_{t^\dagger}^t d_i^\theta(\tau_1)d\tau_1\Big), T\Big) \quad (3.8) \end{aligned}$$

$$\begin{aligned}\frac{\partial w^{\theta}(\mathbf{n_0},t)}{\partial t^{\dagger}} &= \sum_{\forall i:i\in\phi_k,\phi_k\in\Phi}\frac{\partial w^{\theta}(\mathbf{n_0},t)}{\partial n_{0i}}\times\frac{\partial n_{0i}}{\partial t^{\dagger}}+\sum_{\forall i}d_i^{\theta}(t^{\dagger})*C(t^{\dagger})\\ &= \frac{\partial w^{\theta}(\mathbf{n_0},t)}{\partial n_0}\times\frac{\partial n_0}{\partial t^{\dagger}}+\sum_{\forall i}d_i^{\theta}(t^{\dagger})*C(t^{\dagger})\\ &= -\sum_{\forall i:i\notin\phi_0}q_i^{\theta}+\sum_{\forall i}d_i^{\theta}(t^{\dagger})*C(t^{\dagger}) \qquad (3.9)\end{aligned}$$

In Equation 3.9, n_{0i} represents the queue for approach i at time T. Specially, n_0 is the simplification for n_{00}, meaning for the current approach at time T. Regarding the first term on the right hand side of the above derivation, note that $\frac{\partial n_{0i}}{\partial t^{\dagger}}=0$ for all $i\notin\phi_0$ because queues are building in all other queues than the current one regardless of the current signal switch. Here $C(t^{\dagger})$ represents a full signal cycle under the optimal policy starting from time $t^{\dagger}$. Specially, $\frac{\partial n_0}{\partial t^{\dagger}}=-\sum_{i\notin\phi_0}d_i^{\theta}$, meaning the marginal rate of queue growth due to signal switch point $t^{\dagger}$ for the current approach, while $\frac{\partial w^{\theta}(\mathbf{n_0},T)}{\partial n_0}\approx\frac{1}{\sum_{i\in\phi_0}d_i^{\theta}}\times\sum_{i\in\phi_k:\phi_k\in\Phi}q_i$, in which $\frac{1}{\sum_{i\in\phi_0}d_i^{\theta}}$ represents the average discharge time for this incremental vehicle in the current phase, meaning that each vehicle in the current phase increases all queued vehicles in all subsequent approaches by an additional time $\frac{1}{\sum_{i\in\phi_0}d_i^{\theta}}$. This above relationship is easier to understand if one treats n_0 as the discharged vehicles in the current approach. An alternative interpretation, if the current signal is switched by a small amount of time earlier Δ, all queued vehicles in the subsequent approaches will wait approximately for less amount of time $\sum q_i^{\theta}\Delta$. Therefore the marginal effect on reduced waiting time for the subsequent phases is just $\sum_{\forall i:i\notin\phi_0}q_i^{\theta}(t_i)$. Regarding the second term on the right hand side of Equation 3.9, the rational is as follows. Earlier switch of signal by an amount of time Δ would make $\sum_{\forall i\in\phi_0}d_i^{\theta}(t^{\dagger}+)\Delta$ vehicles wait for another full signal cycle that they otherwise could have passed the intersection, which dictates a marginal effect represented by the second term in Equation 3.9. This explains Equation 3.9. The above arguments seem relatively more appropriate in medium to heavy traffic situations, which might justifies the impressive of the proposed algorithm during peak hours in the later numerical tests.

Equation 3.9 offers a perspective to argue for the convexity of $w^{\theta}(\mathbf{n}, t)$ in this approximation method. If one takes another derivative of both sides of Equation 3.9, one gets the second order derivative of $w^{\theta}(\mathbf{n_0}, T)$ over $t^{\dagger}$ as follows if the second term on the RHS is ignorable.

$$\frac{\partial^2 w^{\theta}(n_0, T)}{\partial (t^{\dagger})^2} \tag{3.10}$$

where

$$\xi_i = \begin{cases} \frac{\lambda_i(t_i)s_i}{s_i - \lambda_i(t_i)}, & \text{when } q_i \text{ is discharged within (min,max) of phase } i. \\ 0, & \text{otherwise.} \end{cases}$$

With slight notional abuse, here $\lambda_i(t_i)$ represents the vehicle arrival rate at the end of green phase of an interval t_i. The right side of the above equation represents the rate at which the queues are joined by arriving vehicles. Case I: if $t^{\dagger}$ is made earlier (e.g. larger), the queued vehicles for approach i would be reduced at a rate equal λ_i 'normally', the queue for approach i is cleared between the maximum and minimum green times. Case II: if excessive queues exist for approach i, or if the queue is too short and is cleared within the minimum green time, the right hand for this approach i of this subsequent cycle shall be zero. A brief argument for Case I is made here. Suppose the reference case, the time taken to clear queued vehicles in approach i is $t = \frac{\hat{q}_i^{\theta}(0)}{s_i - \lambda_i(t_i)}$, where $\hat{q}_i^{\theta}(0)$ is the observed queue at the start of the green signal. If $t^{\dagger}$ for the current phase is made earlier by a small time Δ, everything else unchanged (a strong assumption), the observed queue at approach i at the start of its green signal would be $\hat{q}_i^{\theta}(0) - \lambda_i(t_i)\Delta$. The new green time for approach i is therefore $t^* = \frac{\hat{q}_i^{\theta}(0) - \lambda_i(t_i)\Delta}{s_i - \lambda_i(t_i)}$. Therefore the change of the total cleared vehicles that had been queued due to the shift of $t^{\dagger}$ is $\frac{\lambda_i(t_i)s_i\Delta}{s_i - \lambda_i(t_i)}$. The rate at which the queued vehicles change due to $t^{\dagger}$ is therefore as in Equation 3.10.

Note that so far, this section has been discussing about $\frac{\partial^2 w^{\theta}(\mathbf{n_0}, T)}{\partial (t^{\dagger})^2}$ and $\frac{\partial w^{\theta}(\mathbf{n_0}, T)}{\partial (t^{\dagger})}$ based on the effect on one subsequent cycle only. If one is technically able to examine the delay effect beyond one cycle, the result shall be more accurate. In this study, the author only considers one cycle in the numerical method because the the author only has a general confidence on the immediate phases within the next cycle (but excluding the current phase, which could significantly contribute to suboptimal result). Similarly, the one cycle concept is also adopted in Gartner's study [28] [6] [29].

Equation 3.7 combined with Equation 3.9 gives rise to the following:

$$\begin{aligned}\sum_{\forall i} d_i^\theta(t^\dagger+) &< \frac{\sum_{\forall i: i \notin \phi_0} q_i^\theta}{C-(t^\dagger-T)} \\ &= \frac{\sum_{\forall i: i \notin \phi_0} q_i^\theta}{C-AR}\end{aligned} \tag{3.11}$$

Where **AR** is the all-red period due to signal switch. $t^\dagger - T$ represents the 'phase' right after signal switch, which by the definition is an All-Red phase. For situations without a clear All-Red period, readers may also develop an approximation in light of the logic here. The full cycle length C following the signal switch at $t^\dagger$ is estimated. The right hand side of Equation 3.11 is considered as intersection efficiency equivalent in the switch-to phase. Note that in Equation 3.9, q_i^θ only accounts for the phases in the current partial cycle starting with but excluding the current phase under green. It does not account for phases after a full cycle from now because the effect of this marginal change Δ in signal switch on queuing beyond one cycle is not clear: Δ may be too small to see a practical impact. The left side of Equation 3.11 is the discharge rate under the current phase because $d_i^\theta = 0$ for $\forall i \notin \phi_0$ at time $t^\dagger$.

The derivation above involves the end of horizon T. The choice of T value appears arbitrary. However, if one expands Equation 3.1 for T to be such that $(t^\dagger, T)$ includes all the phases in the coming cycle starting with but excluding the current phase, all the derivations above would remain similar, except that choice T would have a clearer explanation. Clearly, Equation 3.11 means that the average intersection efficiency in the remainder of current cycle, starting with the current phase, shall be larger than the current discharge rate in order to justify a signal switch. Conceptually, the choice of T shall be such that each phase overall maintains a similar service rate of vehicles.

Note that the result in Equation 3.11 may shed light on the queue clearance policy studied in literature such as [25] [26]. The queue clearance policy switches green signal after the queue is cleared for a current phase. Seemingly, this policy maximizes the intersection efficiency, but not always so. Grafton and Newell [31] examine this policy by assuming constant and continuous arriving and departure traffic at an intersection between two one-ways. They identify conditions under which the queue clearance policy is and is not optimal, respectively. The general results in this study are consistent to [31], and is more general for being applicable to the general heterogeneous traffic. The author conjecture that a queue clearance policy may be near optimal in most moderate situations when the intersection is roughly symmetric in terms of the lanes and discharge capacity between phases.

3.3 Basic Mechanism of Control Algorithm

With the condition of being a pivotal point $t^{\dagger}$ clarified earlier, it is ready to proposed an approximation optimal policy for an infinite time horizon. At current state $(\mathbf{n}, t)$ and green signal, the decision is whether to switch the green signal to the next phase while continuously satisfying the set of constraints Ω.

In the proposed algorithm below, N is the total number of phases that take turn to get the green signal by following a predefined sequence. The current green phase is indexed zero (e.g. ϕ_0). Let $\Delta T = C - AR$. Note that C shall be the cycle length for the cycle to come. It is exactly the cycle, but a length of time for the combination of phase intervals to get the largest possible switch-to efficiency, which will be estimated numerically.

Practically, Equation 3.11 becomes the following:

$$\varepsilon(\pi_0) < \varepsilon(\pi_1) \tag{3.12}$$

where $\varepsilon(\pi_0)$ and $\varepsilon(\pi_1)$ are numerical estimates of the current phase efficiency and the switch-to efficiency equivalent, respectively.

The current efficiency $\varepsilon(\pi_0)$ may be chosen such that $\varepsilon(\pi_0) = max_{\Delta t_0}\{\frac{D_i(t,t+\Delta t_0)}{\Delta t_0}\}$ while Δt_0 is the additional green time from the current decision point as allowed by the constraints in Ω.

Choice of Δt_0 is calculation of $\varepsilon(\pi_0)$ above is a heuristic method. One may examine the average intersection efficiency for up to a certain time length under the current green phase, for example 5 s, into the near future, by comparing the average throughput for each time length shorter than the limit, for example, 1 s, 2 s, 3 s, 4 s, 5 s, respectively. Within each time length, the expected arrivals is known and is divided by the according time length to get a current efficiency. As a special case, if a platoon of 10 vehicles arrive fairly uniformly between 3 s and 5 s from now, and if no vehicle arrives prior to time 3 s, $\varepsilon(\pi_0) = 2$ vehicles per second, assuming the preset time limit is 5 s and assuming the remaining green time allowed is more than five seconds till the end of the maximum green. In the following numerical test, 5 s is used to get $\varepsilon(\pi_0)$.

Next the switch-to efficiency is explained. $\Delta T = \sum_{i=1,2,\ldots,N-1} \Delta t_i + (N-1)L$, where N is the total number of phases. Here Δt_i is the estimated phase interval i. Donate the fixed sequence cycle as $(\phi_0, AR, \phi_1, AR, \phi_2, \ldots, \phi_{N-1}, AR)$, where $L = AR$. This set of notation repeats after signal switches.

$$\varepsilon(\pi_1) = max_{\Delta T}\{\frac{\sum_{i\in\phi_k:k=1,N-1} q_i^\theta}{\Delta T}\} \tag{3.13}$$

Here q_i^θ is the expected number of vehicles to be cleared in approach i during a *normally* optimized cycle starting from the current phase ϕ_0, which may include queued and free flow vehicles, or in other words, include both the currently observed in the queue $\hat{q}_i^\theta$ and the additional vehicles expected to be clear Δq_i^θ. It is important to explain the calculation of q_i^θ in the proposed algorithm later. Take phase ϕ_1 as an example. To facilitate understanding, assume only one approach for each phase here so that approach i corresponds to phase i. At time t, phase ϕ_1 has a queue of $\hat{q}_1^\theta$ vehicles. When the signal is turned green after a switch loss L, the green phase estimation for phase ϕ_1 in order to maintain a highest phase specific intersection efficiency is calculated to be as follows.

$$\Delta t_1 = max_{\Delta t_1\in[\alpha_1,\beta_1]}\Big\{\frac{q_1^\theta}{\Delta t_1 + L}\Big\} \tag{3.14}$$

Where q_i^θ may be equal to $\hat{q}_i^\theta + \Delta q_1^\theta(\Delta t_1 + L)$ with $\Delta q_1^\theta(\Delta t_1 + L)$ being the expected additional discharged vehicles during Δt_1 in phase 1 when phase 1 is expected to maintain its highest average discharge efficiency. Δt_1 satisfies the min/max green constraints. So, this explains the calculation

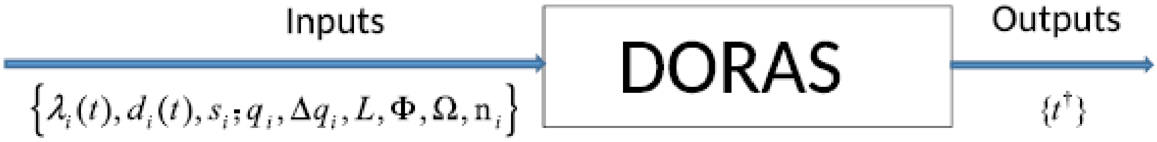

Inputs
$\left\{\lambda_i(t), d_i(t), s_i; q_i, \Delta q_i, L, \Phi, \Omega, \mathrm{n}_i\right\}$
DORAS
Outputs
$\{t^\dagger\}$

Figure 3.2: Algorithm DORAS Input-output Flow Chart.

of Δt_1.

In a similar fashion, the expected total vehicles to discharged and total discharge (green) time for each subsequent phase may be sequentially calculated. In a general case in which a phase has multiple approaches, i.e. $|\phi_k| > 1$, we have a general equation as follows.

$$\Delta t_k = max_{\Delta t_k \in [\alpha_k, \beta_k]} \left\{ \frac{\sum_{i \in \phi_k} q_i^\theta}{\Delta t_k + L} \right\}, k \leq N - 1. \tag{3.15}$$

In Equation 3.15, $\sum_{i \in \phi_k} q_i^\theta$ is the number of vehicles discharged during the time Δt_k for phase k. Those discharged vehicles are $\hat{q}_i$ and additional ones that have arrived during $(\sum_{i=1,2,\ldots,k} \Delta t_i + kL)$ (the current active phase has an index 0). Note that the green time starts for phase $k (k \geq 1)$ at time $t - \sum_{j=1,k-1} \Delta t_j - kL$, at which time point, the beginning queue for phase k needs to be predicted using the known arrival processes. Last but likely equally important, there is room to further fine tune the process to get the 'expected' green time for each subsequent phase so that the entire switch-to efficiency equivalent ε_1 may be maximized and be compared with current intersection efficiency ε_0. That is, $max \sum_{\{\Delta t_1, \Delta t_2, \ldots, \Delta t_{N-1}\}} \left\{ \frac{\sum_{i=1,N-1} q_i^\theta(\Delta t_i)}{\sum_{i=1,N-1} \Delta t_i + (N-1)L} \right\}$, which makes the proposed algorithm inherently share some similarity with the logic of OPAC III.

3.3.1 Dynamic, Optimal, Real-time Algorithm for Signals (DORAS)

The proposed algorithm DORAS is presented in this section, which takes inputs and generates outputs as illustrated in Figure 3.2.

Set incremental time step Δ. Along the time clock, after the minimum green time is reached until the green interval maxes out, at each Δ, the currently active green signal is examined for possible switch by calling the algorithm DORAS below. Note that Δ may be determined empirically such as set to 1.0 s. In the numerical test, 1.0 s is used.

Opposite to before, from now on, the time clock is assumed to increase in the presentation of DORAS. For simplicity, a function named GreenEst($\cdot$) is used in the algorithm DORAS to estimate the green time Δt_k needed for each subsequent phase k. Treating the next phase after the current one as phase 1, GreenEst($\cdot$) is explained as follows.

Algorithm GreenEst($\cdot$)

```
Initialization. Set $k = 1$.
while $k$ is not the last phase in the cycle do
  if $s_i\alpha_k \geq \hat{q}_i + D_i(\sum_{j=1,2,...,k-1} \Delta t_j + k * L + \alpha_k), \forall i \in \phi_k$ then
    $\Delta t_k = \alpha_k$, which means green time is set to the minimum when little queue is present.
  else if $s_i\beta_k \leq \hat{q}_i + D_i(\sum_{j=1,2,...,k-1} \Delta t_j + k * L + \beta_k), \forall i \in \phi_k$ then
    $\Delta t_k = \beta_k$, which means heavy traffic maximizes out the green time.
  else
    Use Equation 3.15 to determine the green interval for phase $k$.
  end if
  $i = k + 1$;
end while
```

In the above function, GreenEst($\cdot$) may be easier to be understood to new reader by assuming only one approach in each phase.

DORAS algorithm applies only when the current green phase has exceeded it minimum green time. There is no need to run DORAS during the all-red interval or when the current green time is less than the minimum duration. DORAS is explained in details below.

Algorithm DORAS

```
Step 1: Initialization. Updated $\hat{q}_i, \forall i$.
Step 2: Run GreenEst($\cdot$) to update $\Delta t_k$ for each phase $k \leq 1$.
Step 3: Update vehicles $\sum_{i \in \phi_k} q_i$ that will haven been discharged for each phase $k$, $1 \leq k \leq N-1$.
Step 4: Calculate the next phase equivalent efficiency factor $\varepsilon_1$ by following Equation3.13.
Step 5: Calculate the current phase efficiency factor $\varepsilon_0$ as explained earlier.
Step 6: Make switch decision according to the cases below:
- If $\varepsilon_1 > \varepsilon_0$, terminate the current phase and switch to all-red interval to transit to the next phase.
- If If $\varepsilon_1 \leq \varepsilon_0$,do not switch signal.
```

3.3.2 Existing-queue Based Heuristic DORAS-Q

One way feel too much of a requirement to know vehicle arrivals to implement DORAS. In fact, implementation of DORAS only requires knowledge of vehicle arrivals for the current phase up to a certain limited time length. In this simulation, 5 s is used. For major arteries, one may be able to easily extend it up to 10-20 s. For subsequent phases, only some ability to predict total arrivals for a certain time period is required. One way propose conservative or aggressive ways for the prediction though. Additionally, the queue lengths as required for DORAS may be estimated approximately in practice by counting in and out flows of approaches as facilitated by video imaging techniques. However, the requirement of predicting vehicle arrivals for each of the subsequent phases within the length of about a full signal cycle may still pose a practical challenge to many practitioners. To further simplify the implementation, another algorithm is proposed that only uses the existing queues and only requires to know traffic arrivals for the current approach for up to five seconds and to know the average historical arrival rates for other phases. This method is identical to DORAS except that it uses $\hat{q}_i^{\theta}$ to proportionally predict q_i^{θ} to calculate ε_1, which is shown in Equation 3.16.

$$q_i^\theta = \hat{q}_i^\theta + \frac{\hat{q}_i^\theta}{t_p} \times t_c = \hat{q}_i^\theta\Big(1 + \frac{t_c}{t_p}\Big) \tag{3.16}$$

where t_p is the passed time within the past cycle that has contributed to the formation of $\hat{q}_i^\theta$; $\hat{q}_i^\theta$ is expected to continue to grow until phase i if the signal switch now, and t_c represents this time period.

In predicting q_i with $\hat{q}_i^\theta$, we need to know the passed time within the past cycle that has contributed to the formation of $\hat{q}_i^\theta$, which gives rise to the average queue growth rate of $\hat{q}_i^\theta$. At this rate, $\hat{q}_i^\theta$ is expected to continue to grow until phase i if the signal switches now. This 'expected' duration in which the queue continues to grow is estimated by utilizing the average phase duration in the past. This method is termed as DORAS-Q. DORAS-Q is much less data demanding but does not require knowledge of the existing queues.

Worthy of a notion is that Dunne and Potts [25, 26] first studied a queue clearance policy, in which the green signal switches when the existing queue is cleared provided that the green interval has not reached it present maximum. [25] studied from the stability point of view while [26] examined from the performance perspective. The proposed method in this dissertation may be considered an extension to the line of literature on queue clearance. But the results often dictates a signal switch even if the current queue (likely from a minor approach) has not been cleared, a way consistent to the special analysis in [31].

3.3.3 An Example Parsing Actuated Control

In this section, a simple example is used to illustrate the algorithm DORAS vs. actuated control, which is show in Figure 3.3. The effective green loss is 3 s for each signal switch. It takes 6, 4, and 6 s respectively to clear the vehicle queues in phase 1 through phase 3 hypothetically. The minimum green time for each phase is 4 s and max 15 s. To be simple for presentation, no vehicles are assumed to join the queues for phase 1 through phase 3. The current phase $\phi_[0]$ has cleared all queued vehicles and has two arriving vehicles in 1.5 and 2.0 s respectively as their headway. The decision is whether to switch the green signal to phase 1. By information, it takes 4 s to clear 2

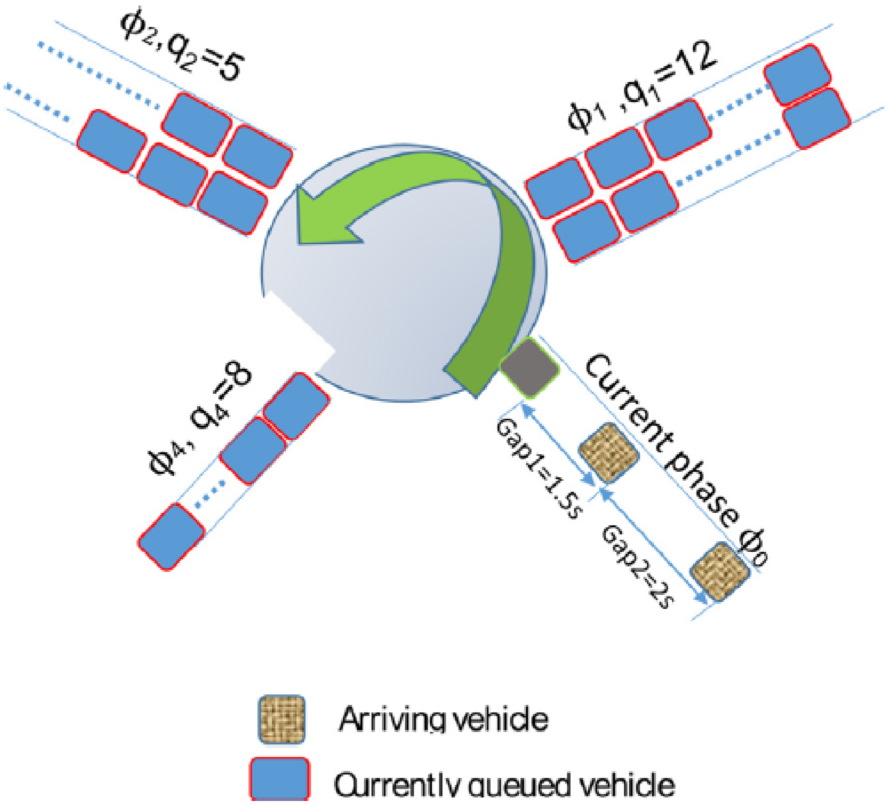
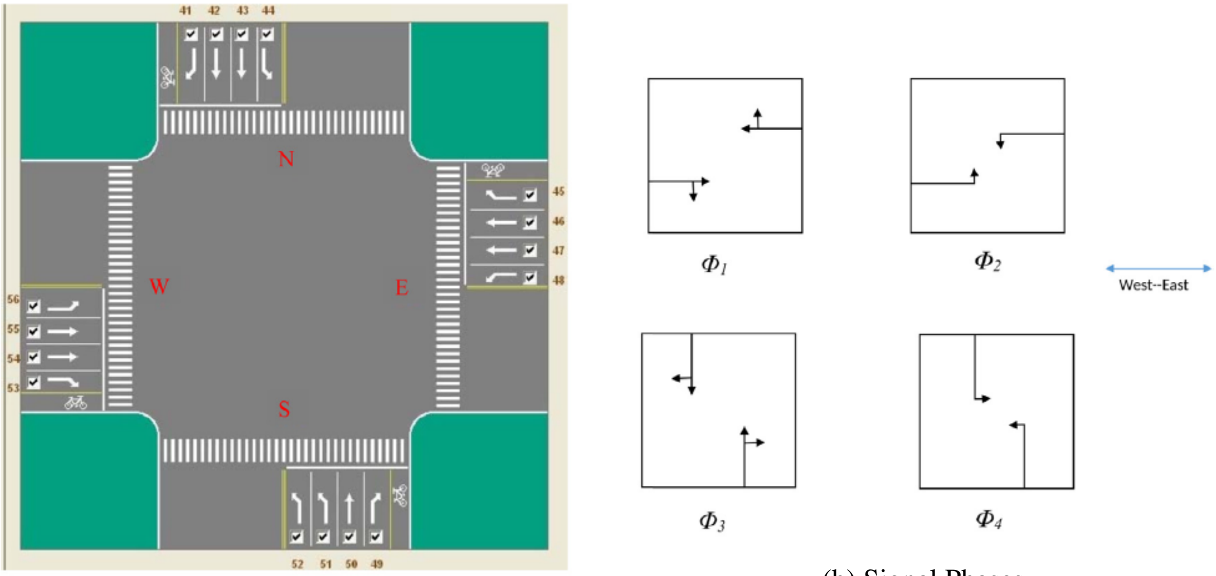


Figure 3.3: An Example for DORAS versus. Actuated Control.

queued vehicles in the current phase.

If the typical actuated signal control is set for the current phase, for example, with a critical gap of 2 s, the signal would wait till both arriving vehicles to have passed through the intersection. However, according to DORAS, $\varepsilon_1 = \frac{q_1+q_2+q_3}{3L+t_1+t_2+t_3} = \frac{12+5+8}{3\times3+6+4+6} = 1$ vehicle per second. The largest possible $\varepsilon_0 = \frac{1}{1.5} = \frac{2}{3}$ vehicle per second. Obviously, DORAS would suggest a signal switch at the current decision point and would subsequently suggest a switch of signal after clearance of queued vehicles for each phase. The total vehicle waiting time saved compared to the actuated control due to DORAS is 117.5 out of a total of 371.5 vehicle seconds, a 30 % saving.

Note that Figure 3.3 is a fairly normal, light traffic traffic example, considering that the queues are for the entire cycle time, and that each phase might serve several approaches, and each approach having multiple lanes. Therefore, the example here may imply only 2-3 vehicles in queues for



(b) Signal Phases

(a) Geometric Layout

Figure 3.4: Intersection for Simulation.

each lane. A critical gap of 2 s used in actuated control implies, according to DORAS, a consistent switch-to efficiency $\varepsilon_1 = 1800$ vph. A headway in the current phase smaller than the critical gap implies a current intersection efficiency $\varepsilon_0 \geq \varepsilon_1$, which justifies extension of green phase. In fact, the switch-to efficiency ε_1 varies with traffic, and is not a constant. Therefore, a fixed critical gap is usually suboptimal in the criterion of DORAS.

3.4 Numerical Tests

A real intersection in an urban area of several million people is used for assessing the efficiency of the proposed algorithm. The intersection layout is illustrated in Figure 3.4a, in which each lane is labeled by an index number. There are four phases adopted in the simulation as illustrated in Figure3.4b. The author simulate and compare four controls: fully actuated, DORAS, DORAS-Q, and OPAC III. OPAC III is selected because it represents a well established method in literature as elaborated in [28]. OPAC III conducts myopic optimization within a horizon of about a signal cycle with optimality and optimization horizon of each time similar to those of DORAS. They all repeat on a rolling horizon.

This intersection has vehicle detectors for each approach. The actual vehicle arriving processes are recorded through the detectors. Arrival processes recorded in history is used as input to the simulation. Traffic data from both peak hours and off peak hours are selected. Traffic volumes are summarized in Table 3.1and Table 3.2 on two days. As shown, the east-west direction is the major corridor while the north-south direction is relatively minor. The simulation setting includes: a 10 s minimum green time for each phase mainly to consider pedestrian crossing need, a 50 s maximum green time for phase 1 and 2, and 35 s for phase 3 and 4. For the actuated control, a 2 s critical gap is set up by following the conventional method. Other setups include a saturation flow rate of two seconds for each vehicle on each lane and a point queue policy. These parameters are generally adopted from field implementation, and remain consistent in the simulation. In the setup for OPAC III, 60 s is used as the horizon for each optimization, which is roughly the average cycle length.

Table 3.1: Traffic Count in Selected Hours on Day One in Cars.

Time period	Total	Left-turn/Through traffic volume			
		Northbound	Southbound	Eastbound	Westbound
7:30-8:30 AM	4429	358/360	99/538	289/1212	269/1304
4:30-5:30 AM	3430	260/272	112/412	194/1071	168/941
1:00-2:00 PM	2667	235/200	67/249	130/852	146/788
12:00-1:00 AM	454	45/31	10/51	14/150	21/132

Table 3.3 and Table 3.4 as equivalent illustrated by Figure 3.5 show the intersection performance via simulation by using two day actual traffic arrival data. DORAS, DORAS-Q, and OPAC III all consistently and significantly outperform the fully actuated control. Compared to the fully actuated control, DORAS and DORAS-Q may be reduced the average delay up to about 20% in

Table 3.2: Traffic Count in Selected Hours on Day Two in Cars.

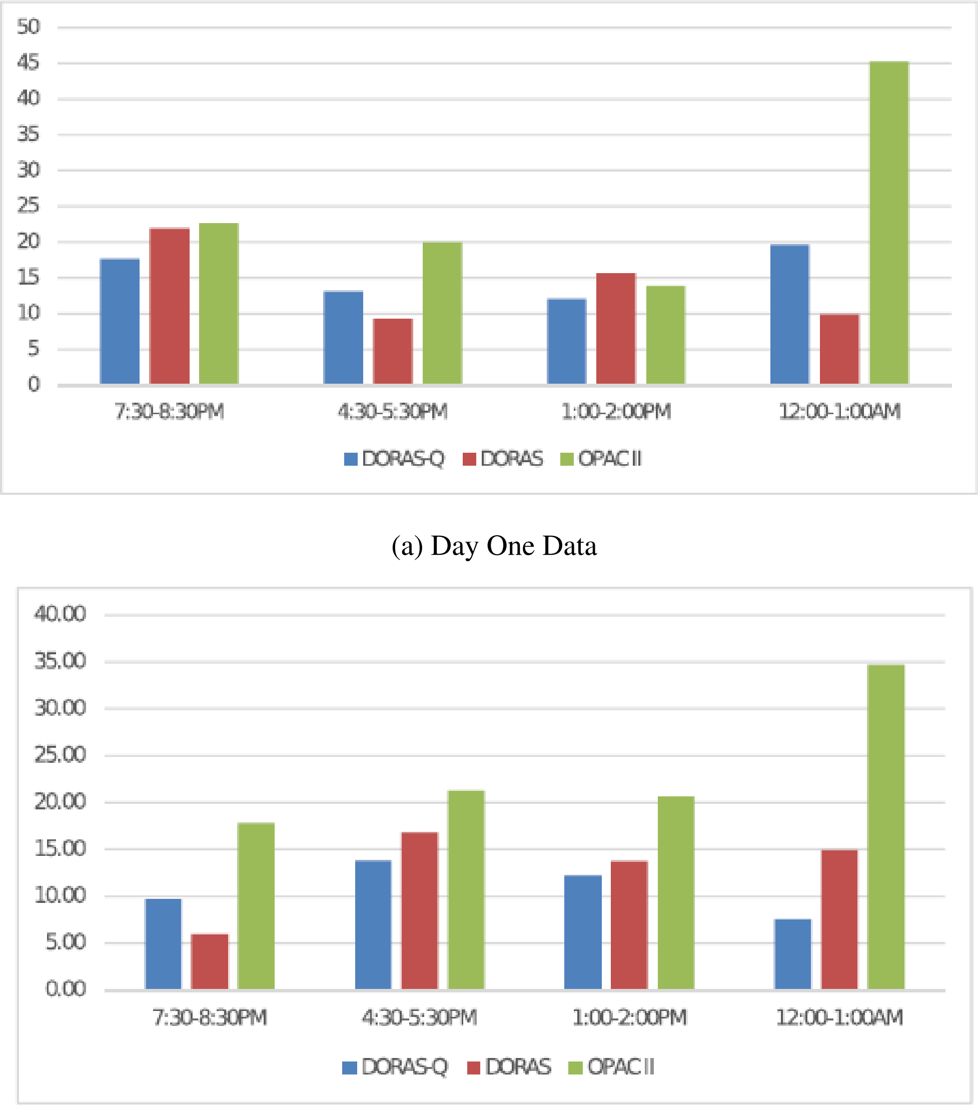
Time period	Total	Left-turn/Through traffic volume			
		Northbound	Southbound	Eastbound	Westbound
7:30-8:30 AM	4369	338/345	75/493	252/1278	292/1296
4:30-5:30 AM	4399	330/396	146/588	313/1390	215/1021
1:00-2:00 PM	2802	184/213	77/261	159/964	144/800
12:00-1:00 AM	438	49/39	4/49	27/127	23/120

the peak hour. DORAS overall appears slightly better than DORAS-Q, but DORAS-Q is much less data demanding and is the simplest to implement among all the three. OPAC III performs the best among all of them, especially during midnight when the traffic is very sparse.

Table 3.3: Day One Vehicle Delay Comparison in Seconds.

Time period	Actuated	DORAS	DORAS-Q	OPAC III
7:30-8:30 AM	20.24	17.36	18.17	17.23
4:30-5:30 AM	17.73	14.95	15.75	15.28
1:00-2:00 PM	16.66	15.12	14.47	13.33
12:00-1:00 AM	14.16	12.76	11.39	7.75

OPAC III has an impressive performance in the simulation. OPAC III out looks into a cycle length for traffic arrivals for all approaches including the current one. In contrast, DORAS only looks over a period of 4-5 s for the current signal. In this regard, DORAS appears more myopic



(a) Day One Data

(b) Day Two Data

Figure 3.5: Vehicle Delay Reduction in Percentage Compared to Fully Actuated.

Table 3.4: Day Two Vehicle Delay Comparison in Seconds.

Time period	Actuated	DORAS	DORAS-Q	OPAC III
7:30-8:30 AM	20.36	19.15	18.39	16.73
4:30-5:30 AM	21.73	18.08	19.74	17.10
1:00-2:00 PM	16.31	14.06	14.33	12.94
12:00-1:00 AM	13.94	11.86	12.89	9.10

than OPAC III. However, OPAC III resorts to an enumeration method that cost significantly computational time. For each of the four select peak hours on the two days, DORAS takes less than 30 s to complete the hourly simulation while OPAC III takes an average of about 1970 s, 60-70 times more than DORAS. The time difference is large due to the control algorithms.

Table 3.5: Day Two Intersection Cycles under Different Signal Control in Seconds.

Time period	Actuated	DORAS	DORAS-Q	OPAC III
7:30-8:30 AM	66.47	61.90	66.46	65.05
4:30-5:30 AM	56.05	57.00	61.72	62.63
1:00-2:00 PM	52.37	54.80	57.48	63.79
12:00-1:00 AM	48.22	48.74	62.07	68.42

Table 3.5 indicates that the improved controls tend to have a slightly longer cycle length. Compared with actuated, the obtained results also show that all other three controls have consistently and significantly reduced the number of vehicle stops, which shows consistency between two often

mentioned objective: minimizing vehicle waiting time and vehicle stops at the intersection.

Nevertheless, in order to make the study more reasonable, fixed-time control is also tested using HCS 7 software package, and the results are shown in Table 3.6 below. Different to other methods, HCS 7 optimizes the intersection measurement in terms of overall delay by testing varying cycle lengths. The minimum delay result after multiple runs is generated for each traffic volume case during different time periods. It clearly shows that fixed-time control performs much worse than any other methods mentioned earlier except for the midnight period, in which the results are much closer to those of other methods.

Table 3.6: Vehicle Delay under Fixed-time Control in Seconds.

	Day One	Day Two
7:30-8:30 AM	63.9	60.6
4:30-5:30 PM	35.2	63.9
1:00-2:00 PM	28.4	29.1
12:00-1:00 AM	15.8	15.7

DORAS-Q uses almost only the queuing data at approaches along with limited statistic such as average phase lengths. However, when the same partial data is applied to OPAC III, OPAC III could perform much worse than even the actuated in the simulation. In summary, if data is available in a longer term as evidenced by the inter-vehicle communication technologies, OPAC III appears superior. For shorter terms, DORAS-Q appears advantages. However, the performance of DORAS and DORAS-Q is conditional on how to approximate the salvage effect. If between approximations to the salvage effect are developed, their relative performance may be further improved.

3.5 Results Discussion

This chapter proposes a simple and efficient algorithm DORAS, which is based on projected vehicle clearance for each phase. An intersection control is simulated by using actual intersection traffic to show the efficiency of the proposed algorithm DORAS. Tests show that DORAS significantly reduces delay compared with fully actuated control. Another easy-to-implement, queue based control called DORAS-Q is also proposed, which requires little advance information except for the existing queues. DORAS-Q also significantly reduces delay of vehicles. All these controls are compared with an established one in literature, OPAC III. Tests shoe that OPAC III consistently outperforms DORAS and DORAS-Q, but is computationally much more costly. In addition, if OPAC III is subject to the data supply as for DORAS-Q, OPAC III performs much worse.

DORAS and DORAS-Q are fairly flexible for implementation. They can flexibly accommodate constraints such as the mini- mum/maximum green time and phase sequence requirement. In addition, they can conduct differential treatment of vehicles by assigning varying weight factors to vehicles such as buses. The optimal intersection control in turn highlights a social implication of intersection geometric design. Geometrically asymmetric intersections almost always mean that ROW for minor streets implies loss of intersection efficiency or increased waiting time for vehicles along the major approaches. This social burden (or cost) due to intersection with a minor street can be relieved by raising the number of pass-through lanes for the minor streets at the intersection, in which case, a queue clearance policy would likely be optimal.

4. COORDINATED SIGNAL CONTROL ALGORITHMS: A NEW PERSPECTIVE

Network signal control is different from isolated intersection control and needs to take several intersections into account. Signal coordination is so far the most widely used method for network signal control, especially for corridors in downtown areas with short signal distance. The basic idea of coordination is to let vehicles pass through several intersections without stopping, which could reduce waiting time, increase average travel speed and improve overall safety.

In this chapter, different arterial coordinated control algorithms are compared and tested in a same simulation framework. Four different methods/algorithms are discussed including PASSER V, DORAS-Q, Max Pressure (MP), and an mixed algorithm between DORAS-Q and MP. The details of those algorithms are introduced first, followed by an description of the simulation network. The simulation results generated from each algorithm are compared and investigated at last to further show the advantages and limitations of each algorithm.

4.1 Background of Coordinated Signal Control

4.1.1 Pre-timed Coordination Signal Control

Much of the literature about signal coordination is about the method applied to a single corridor. In the corridor signal coordination, uniform signal cycle length is a must according to the literature. The major setup includes cycle split and also offset along the corridor. The signal timing plan are designed to maximize the width of green interval in both directions along the arterial in which vehicles may travel through the corridor without having to stop. Although maximizing the bandwidth is only a surrogate performance measure, it has been shown to provide significant benefits in increased travel speeds and smoothness of traffic flow. In general, such signal systems operate best when the main-street flow is predominantly through traffic and predominantly large than side streets.

Bandwidth maximization has seen a relatively rich literature such as Morgan and Little [58], Little [59],Gartner and Stamatiadis [60], Mireault [61], Chaudhary et al. [62], Pillai et al. [63].

Accordingly, a number of software packages were developed to implement this method, including MAXBADN [64], PASSER series [65–68], and MULTIBAND [69]. Since the PASSER software package is used in this paper later to optimize signal timings for a signalized arterial network, the details are introduced as follows.

PASSER II [65], which was the original version, was first developed by the Texas Deportment of Transportation (TxDOT) in the earlier 1990s. The heuristic signal timing optimization model of PASSER II is based on a graphical technique [70]. PASSER II is able to select the plan that maximizes progression efficiency, a unitless quantity obtained by diving the progression band by the cycle length. It appears to be one of the most computationally efficient programs because of its simplicity. It can also perform exhaustive searches over the range of cycle length provided by the user. The process starts with calculating splits using Webster's method. Then, it applies a hill-climbing approach and adjusts splits to minimize delay. Finally, it applies its bandwidth optimization algorithm using the pre-calculated splits as input to that model. At the optimization stage, it can find the cycle length, offsets, and phase sequence that produce maximum two-way progression. Later, PASSER V [68] revised the algorithm by adjusting the method to calculate splits. It also solved the early termination problem by applying algorithm for both directions, therefore increasing the ability to find better solutions.

4.1.2 Actuated Coordinated Signal Control

Actuated coordinated signals offer additional flexibility in comparison to pre-timed coordinated signals. They can response to cycle-by-cycle variations in traffic volumes to better assign green time to each phase, thus reducing unnecessary delays and stops. However, since most signal timing optimization software are designed for fixed-time signals, the optimized control parameters do not always work efficiently for coordinated arterials. In addition, we should pay more attention to determine appropriate signal settings, particularly offsets due to the fact that the start of green of the sync phases is not fixed.

In actuated coordinated systems, variations in green time due to phase actuation often leads to the so-called "early return to green" problem, which is discussed in several studies (e.g. Jovanis and Gregor [71]; Ficklin [72]). This "early return to green" occurs when the sum total of the time required by the non-coordinated phases is less than the sum total of the vehicle splits coded for the phases. While this may reduce delay at the first intersection, it may increase system delay because of inefficient flow at downstream intersections.

Among adaptive network signal control algorithms, the max pressure (MP) algorithm developed by Varaiya [73] and the real-time traffic signal control system (RHODES) developed by Mirchandani and Head [74] are very representative. In what follows, we briefly introduce these two methods.

Varaiya [73] introduced the control algorithm based on the max pressure principal for a signalized network with multiple intersections. Max pressure principle is a fluid theory that dictates fluid flow along the direction that the fluid pressure decrease. In network intersection, the "pressure" is represented by the intersection queues. At each intersection, MP selects a stage that depends only on the queues adjacent to the intersection. MP then maximize the throughput of the network based on the following equation:

$$\gamma(S)(X) = \sum_{l,m} c(l,m)w(l,m)(X)S(l,m) = \sum_{l,m:S(l,m)=1} c(l,m)w(l,m)(X) \tag{4.1}$$

In Equation 4.1, $\gamma(\cdot)$ is the measure of pressure for a particular signal control. X is a vector for intersection queues on the network. $c(l,m)$ is the saturation flow rate for the movement of traffic form link l to link m where two links are connected at the intersection under control. $w(l,m)$ is the weight assigned to a particular movement. In particular, $w(l,m) = x(l,m) - \sum_{p \in Out_m} r(m,p)x(m,p)$. Here $r(m,p)$ is the turning ratio of traffic from link m onto a connected, downstream link p. $\sum_{p \in Out_m} r(m,p)x(m,p)$ is the average downstream queue. Out_m is the set of downstream links from link m. $S(l,m)$ is signal indicator, which is 1 when signal is activated for the movement of traffic from link l to link m; and it is 0 otherwise.

This method has several advantages. It does not require knowledge of OD traffic, or traffic arrival processes. It only requires to know queues as well as traffic split ratios at intersections.

Under the condition that intersection are under saturated, a fixed time control policy can ensure network stability. However, this method is not optimal in the case of isolated intersection control.

RHODES is another signal control method that is able to do adaptive signal control for signalized network. RHODES uses a model called REALBAND [75] to optimize the movement of observed platoons in the network. REALBAND attempts to form progression bands based on actual observed platoons in the network. In general, any delay and/or stops based measure of performance may be optimized.

The basic logic of REALBAND is described as follows. It defines platoons from observed detector as a flow density above a pre-specified level for some length of time. Each platoon is characterized in terms of size (number of vehicles) and speed. Typically, RHODES uses a 20-40 s rolling horizon to predict arrivals and queues at each intersection based on upstream detector data. However, at network flow control level, it will use a 200-300 s rolling horizon to predict platoons.

A decision tree is built later where each branch of the tree represents one possible of a conflict, meaning two platoons request opposing signal phases the same time. Then REALBAND evaluates the performance for each branch of the decision tree to further find a path with best-estimated performance using APRES-NET model [76]. APRES-NET model is a simplified traffic simulation model that propagates platoons instead of vehicles through a subnetwork of intersections. It can used not only to evaluate objective function, but also to estimate network wide performance based on network control logic.

The REALBAND decision are used as constraints to the intersection control logic (COP) when deciding the phase durations for every intersections using the rolling horizon optimization.

4.2 Network Control Alternatives for Comparison

Before introducing the proposed coordinated signal control algorithm, we first describe several existing methods that are compared later with our proposed algorithm. MP method is adjusted in order to better fit the study framework of this paper. Coordination control usually intends to minimize the overall delay/stops or maximize the efficiency of a network system by applying varying control policy to each individual intersection. However, even we can have the optimal control policy for each individual intersection, the system may still not be able to achieve the best performance in terms of the chosen system measures. In order to find the best way to achieve systemic optimality, different categories of control algorithms are compared in this paper, including the isolated intersection based method DORAS-Q [77], arterial based method PASSER V, and system based method Max Pressure. Some of the notations that were used to develop the DORAS are listed as follows, which all have the same definition as before.

Notations

θ control policy that determines signal switch from one approach to another during the entire control horizon. With slight notional abuse, $\theta(t)$ is used for the specific signal control at time t.

Φ set of phases. N is the total number of phase in Φ. The phases are indexed from 0 to $N-1$ sequentially according to the order in a signal cycle with 0 being the current one with green signal. The index is therefore on a rotational basis.

ϕ_k set of approaches on phase $k \in \Phi$. $k = 0$ means current approach with ROW.

$D_i^\theta(t_1, t_2)$ total arrivals of vehicles from approach i during time interval (t_1, t_2) under policy θ. $D_i(\Delta t)$ means total arrivals in the past period Δt, which slight notional abuse.

$\hat{q}_i$ currently observed queue length for approach i at the decision point.

Δq_i expected additional vehicles to be discharged beyond $\hat{q}_i$ for approach i during a normally run phasing in the current cycle forward.

q_i the total number of vehicles discharged for approach i during the current signal cycle that starts from the current phase, $q_i = \hat{q}_i + \Delta q_i$.

L green time loss due to signal switch, referred to as all-red interval in this paper.

ε_0 current intersection efficiency under the current green phase.

ε_1 next (or, switch-to) phase intersection efficiency equivalent.

Note that the term "phase" used in this paper refers to the right-of-way, yellow change, and red clearance (all-red) intervals in a cycle that are assigned to an independent traffic movement or combination of traffic movements. Yellow interval and red clearance (all-red) interval combined is called amber interval.

4.2.1 PASSER V

PASSER V is the latest version in the PASSER series of programs developed by Texas A&M Transportation Institution (TTI) [68] for timing arterials and signalized diamond interchanges. To achieve arterial/corridor signal coordination, PASSER V develops signal timing plans to maximize progression. It is a fixed timing system, not for real-time, traffic responsive control. It develops a plan using offline simulation and apply to real control. It performs exhaustive cycle-length search in the user-selected range and maximizes bandwidth efficiency for each cycle using the interference minimization algorithm. The details are shown as follows.

First of all, PASSER V calculates preliminary splits and adjust these splits in order to minimize intersection delay. The pre-calculated splits are then input to the bandwidth optimization algorithm. For bandwidth optimization, PASSER V starts by selecting a cycle and calculating perfect one-way progression in the A (arbitrarily selected) direction. Then it minimizes band interference in the B (opposite) direction by adjusting phasing sequences and offsets. The maximum total band calculated by the program is as follows:

$$
\begin{aligned}
\text{Total Band} &= G_A + G_B - I \qquad (4.2)\\
\text{where: } G_A &= \text{least green in A-direction, in seconds}\\
G_B &= \text{least green in B-direction, in seconds}\\
I &= \text{minimum possible band interference, in seconds}
\end{aligned}
$$

After achieving the best band (minimum inference) in the B direction, the programs adjusts the two bands according to user-desired options for directional priority.Finally, the program calculates delays, bandwidth efficiency, and attainability. Efficiency and attainability measure how good a bandwidth solution is. Efficiency for a direction is the percent of a cycle used for progression. Attainability is the percent of bandwidth in a direction in related to the minimum green split in that direction. The following formulas are used to calculate the combined efficiency and attainability for the two arterial directions:

$$\text{Progression Efficiency (\%)} = \frac{\text{ArterialBand}_A + \text{ArterialBand}_B}{2 \times \text{Cycle Length}} \times 100 \tag{4.3}$$

$$\text{Progression Attainability (\%)} = \frac{\text{ArterialBand}_A + \text{ArterialBand}_B}{\text{Min.Green}_A + \text{Min.Green}_B} \times 100 \tag{4.4}$$

The readers should note that while bandwidth generally increase with an increase in cycle length, efficiency may increase, decrease or remain constant. Thus, it is desirable to select a solution that provides the maximum efficiency and an attainability of 100%.

4.2.2 DORAS-Q

DORAS-Q is an real-time, traffic responsive control applied to isolated intersections [77]. It is a deviant from DORAS by relieving the requirement of known granular traffic arrivals to the intersection. DORAS-Q uses the existing queues and only require to know traffic arrivals for the current approach for up to 5 seconds and to know the average historical arrival rates for other phases. This method is identical to DORAS except that it uses the current queue observed at the moment $\hat{q}_i^\theta$ to proportionately predict the theoretically needed total vehicle discharged q_i^θ to calculate ϵ_1, where ϵ_1 is the next discharge efficiency (e.g. switch-to efficiency). In predicting q_i^θ with $\hat{q}_i^\theta$, we need to know the passed time within the past cycle that has contributed to the formulation of $\hat{q}_i^\theta$, which gives rise to the average queue growth rate for $\hat{q}_i^\theta$. At this rate, $\hat{q}_i^\theta$ is expected to continue to grow until phase ϕ_k ($i \in \phi_k$) if the signal switches now. This "expected" duration in which the queue continues to grow is estimated by utilizing the average phase duration in the past. DORAS-Q is much less data demanding, but does require knowledge of the existing queues. DORAS-Q is tested to perform well for isolated intersections. The equation to calculate q_i^θ (assuming$i \in \phi_k$ and total number of phases is N, and that ϕ_k is the switch-to phase) is as follows:

$$
\begin{aligned}
q_i^\theta &= \hat{q}_i^\theta + \frac{\hat{q}_i^\theta}{t_p} \times t_c \\
&= \hat{q}_i^\theta (1 + \frac{t_c}{t_p}) && (4.5) \\
t_p &= \sum_{j=1,\ldots,k-1,k+1,\ldots N-1} \Delta t_j && (4.6) \\
t_c &= \sum_{j=1}^{k} \overline{\Delta t_j} && (4.7)
\end{aligned}
$$

where: t_p = passed time within past cycle that contributed to the formation of $\hat{q}_i^\theta$;

t_c = expected time from current point to phase ϕ_k if the signal switches now;

Δt_j = phase interval of phase j in the past cycle;

$\overline{\Delta t_j}$ = average phase interval of phase j;

DORAS compares current phase efficiency (ϵ_0) with the switch-to efficiency (ϵ_1) to decide whether terminate the current phase or not at each moment of signal decision. The equation to calculate the current efficiency (ϵ_0) is shown as Equation 4.8, as explicated in [77].

$$
\epsilon_0 = max_{\Delta t_o} \frac{\sum_{i \in \phi_o} D_i(t, t + \Delta t_0)}{\Delta t_0} \tag{4.8}
$$

The equation to calculate the switch-to efficiency ϵ_1 is similar to that of DORAS as follows:

$$
\epsilon_1 = max_{\Delta T} \frac{\sum_{i \in \theta_k : k=1,N-1} q_i^\theta}{\Delta T} \tag{4.9}
$$

where $\Delta T = \sum_{i \in \theta_k : k=1,N-1} \overline{\Delta t_i} + (N-1)L$, and L is the all-red interval including the lost effective green due to signal turns.

4.2.3 Max Pressure (MP)

Varaiya proposes the max pressure algorithm (MP) [73]. MP selects the network signal control matrix with the maximum pressure at every state. The pressure of a signal control matrix is the sum of the saturation flows multiplied by the weights of all the approaches that have the right of way. The weight of an approach is the queue length at the current approach minus the average queue length at the downstream approaches, which is calculated based on the turning percentages. Queue lengths are represented by the number of queued vehicles. For example, if the queue length at current approach is 20, while the right, through, left-turn percentage are 15%, 60%, 25% , respectively, the according queue lengths for the three downstream approaches are 10, 15, 5, respectively. Then the weight of the current approach is calculated as follows: $20 - (10 \times 15\% + 15 \times 60\% + 5 \times 25\%) = 8.25$.

In order for a coordinated control along the corridor, a modified version of the MP algorithm is introduced as follows.

Step 1. For each intersection n in each time step T, first check maximum green time requirement. If the current green phase is less than the maximum green, assign the weight w_{n0} for current phase. Otherwise, terminate the current green interval and switch to amber interval to transit to next phase.

$$w_{n0} = \sum_i q_i - \sum_i \sum_j p_{ij} q_j \tag{4.10}$$

where i is the approach index for current phase; j is the index of corresponding downstream approach; p_{ij} is the percent of volume goes from current approach i to downstream approach j; q_i is the current queue for approach i.

Step 2. Calculate the switch-to weight w_{n1}.

$$w_{n1} = \sum_l q_l - \sum_l \sum_m p_{lm} q_m \tag{4.11}$$

where l is the approach index for switch-to phase; m is the index of corresponding downstream approach; p_{lm} is the percent of volume goes from current approach l to downstream approach m; q_l is the current queue for approach l.

Step 3. Make switch decision according to the cases below:

- If $w_{n0}s_{n0} \geq w_{n1}s_{n1}$, do not switch signal;

- Otherwise, terminate current green interval and switch to amber interval to transit to next phase.

where s_{n0} is the saturation flow for current phase and s_{n1} is the saturation flow for switch-to phase. In our calculation, $s = 4000$ veh/hr for left-turn phase while $s = 8000$ veh/hr for through phase.

Step 4. Move on to next time step. $T = T + 1$ and return to Step 1.

Repeat the above algorithm until all the intersections in the system are covered. Note that the original max pressure Equation 4.1 maximizes the network pressure in terms of the sum of the saturation flows multiplied by the weights. However, since saturation flows are always positive, we eliminate it in the modified version to make it more straight-forward. Based on this setup, the pressure maximization equation simply becomes the total weight of every approach in current phase. Such modification results in less computation, since control decision is made for each individual intersection instead of solving a linear programming problem to obtain the decision variable vector (X in Equation 4.1).

4.2.4 Actuated Control

In addition to the three algorithms mentioned earlier, the traditional actuated control method is tested as well in this paper as a locally adaptive control without far sight information. We use fully-actuated control for each intersection in the simulation network, meaning that loop detectors are implemented on all approaches. The detector is placed 200 ft before the stop line for through lane, while the distance becomes 150 ft for left-turn lane considering lower travel speed. The details about signal setting (min/max green, green interval extension) are described in later chapter.

4.3 Basic Mechanism of Proposed Algorithm

Either DORAS or DORAS-Q for now focuses on a single intersection signal control optimization by investigating the current and switch-to efficiency of a particular intersection at each moment of signal decision. However, it does not specifically consider the effects between upstream and downstream intersections as MP algorithm does, which may cause queue spillback at downstream intersections, except for the fact that input traffic flow to an approach is the output traffic flow from its upstream intersection, which implies an implicit consideration of intersection connection. As mentioned earlier, MP algorithm explicitly considers the difference between the upstream and downstream queue length, which has an effect of preventing from pushing vehicle waiting time down to downstream intersections. In light of the network effect of a current intersection signal policy (e.g. mainly in pushing the existing queue one intersection down), a mixed algorithm of both DORAS-Q and MP is proposed in this paper, which is called MADM later in this paper. Similar to DORAS/DORAS-Q, the following algorithm applies only when the current phase has exceeded its minimum green time but is still within the maximum green time. The details are presented as follows:

Step 1. At each time step T, first check maximum green time requirement. If the current green phase is less than the maximum green, calculate the decision variable $\xi_i(x)$ for current phase. i is the index number for each individual intersection in the network with a total of n intersections, $i \in [1, n]$. Decision variable x can be 0 or 1, 0 being for current phase and 1 for the next (switch-to phase).

Step 2. Calculate the decision variable value for current phase.

$$\xi_i(0) = w_{i0} s_{i0} \epsilon_{i0}^{1} \epsilon_{i1}^{0} \tag{4.12}$$

where w_{i0} is the weight of current phase obtained from MP algorithm for intersection i; s_{i0} is the saturation flow rate for current phase; ϵ_{i0} is the current phase efficiency obtained from DORAS or DORAS-Q, while ϵ_{i1} is the switch-to efficiency calculated by DORAS-Q.

Step 3. Calculation the decision variable value for switch-to phase.

$$\xi_i(1) = w_{i1} s_{i1} \epsilon_{i0}^0 \epsilon_{i1}^1 \tag{4.13}$$

Step 4. Make switch decision according to the cases below for each individual intersection i:

- If $\xi_i(0) \geq \xi_i(1)$, continue the current green interval;

-If $\xi_i(0) < \xi_i(1)$, terminate current green interval and switch to amber interval, then transit to next phase.

Step 5. Move the time clock forward by one time step. $T = T + 1$ and return to Step 1.

Note that Equation 4.12 can also be written as a matrix format in terms of vector **x**, in which case the equation looks similar to the original and complete network control matrix of MP method (Equation 4.1). Similar to the modified version of MP algorithm, both ways of presentation are equivalent in terms of finding the maximum network efficiency. However, it is easier to solve to find the optimal solution for the system by solving each intersection individually. Since DORAS/DORAS-Q in previous paper is also developed for individual intersection, readers can follow the process and idea of MADM quickly.

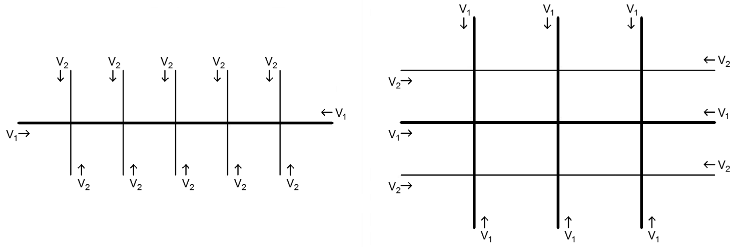
4.4 Numerical Tests

we have conducted numerical tests on two networks: a single corridor and signals on networked grid network. The reason for separately testing on single corridor is because single corridors are often the major means dealing with urban traffic and established methods are developed for it.

VISSIM is a microscopic traffic simulation package that tracks every vehicle and allows real time control of signals. it is used as the simulation platform for all the algorithms except PASSER V while PASSER V itself is a separate, independent package to use. DORAS-Q, MP, and MADM are all coded in C++ to implement through the interface provided by VISSIM during the simulation.

The two networks generated for the numerical tests are representative. The one with a dominating single corridor as Figure 4.1a supposedly should work well for the popular signal coordination packages, particular in the case in which traffic intensity (the rate in the Poisson process) is stable over the time of simulation. Figure 4.1a has the minor street crossings spaced by 2100, 3200, 2100 and 3200 feet apart eastbound along the corridor with a corridor travel speed of 50 mph. The normal travel times traversing the two distances are about 29s and 44s, respectively, roughly about half a signal cycle or slightly less. The grid network is built as Figure 4.1b to deliberately not have singular corridors dominating the network so that the case may represent the other extreme of traffic network. In Figure 4.1b, the roadway spacing from east to west is 2035 and 3130 feet respectively, equivalent to 27s and 43s of travel time, and is 1160 and 1180 feet respectively from north to south, equivalent to 16s each. In the east-west direction, the travel time between intersections is roughly half of the intersection signal cycle while in the north-south direction, the travel time is much shorter, like only equal to the duration of a phase, a quarter of a signal cycle. Both Figure 4.1a and Figure 4.1b are representative of reasonable dense urban streets. In addition, each road segment is a four lane street with two lanes in each direction. At each intersection, there is a left-turn bay, two through lanes with the right lane also handling right-turning traffic. In reality, sometimes, there are singular dominating corridors that can be easily identified while at other times, the roadway traffic make it hard to find the dominating singular corridors, several crossing traffic all being likely major ones.

All the four algorithms are tested in both the arterial and grid network as shown in Figure 4.1. The arterial case consists of one major arterial with five minor road crossings. The network has a total of six arterial roads, where four of them are major arterial roads (bold lines) with higher traffic volumes. The rest two are minor arterial roads with lower volumes. The directional volume on major arterial roads is V_1 and the volume on minor arterial is V_2. The volumes are shown in Table 4.3. Three different volume cases: low, medium, and high, are tested for each control. The specific scenarios and traffic volumes are shown as in Figure 4.1. As can be seen, the two-way volume for the major arterial road is the sum of the two one-way traffic in opposite directions. Note that the directional traffic is set to be balanced, which means that on each arterial, the itinerary volume is balanced in opposite directions. For example, for the east and west arterial, the east bound traffic originated at the west beginning of the arterial equals the west bound traffic originated at the east end of the east-west arterial road. It is similar with other arterial roads. The right-turn, through, and left-turn percentage of each approach are set as a constant across intersections. In this paper, these three percentages are 15%, 60%, and 25%, respectively, to reflect a reasonable observation. Other splits can also be easily set up, but we do not believe would have a significant impact on the performance tested.



(a) A Single Arterial (b) Grid Network

Figure 4.1: Roadway Network Configuration for Simulation.

VISSIM generates traffic by user defined volumes according to a predefined Poisson process. The volume set above serves to get the traffic intensity. The headway follows exponential distribution. The stochastic functions in VISSIM run by taking in different seed values at different runs. With this setup, and with a given constant traffic intensity on the network as Figure 4.1a, one would expect the corridor coordination package would outperform other algorithms. But the test results show surprisingly to the contrary.

The Component Object Model (COM) in VISSIM gives access to data and function contained in other programs. Data contained in the software can be accessed via COM interface using VISSIM as an automatic server. In this study, VISSIM COM interface is used to achieve the goal of controlling the signal timing plan of each intersection in the simulation network by the given algorithms that coded with C++ language.

The signal settings are simple as in Table 4.1 for both networks. The amber interval in the table represents the time between two consecutive phases to clear the intersection, which refers to the total of yellow and all-red interval. For the actuated control, the green extension time is 3 seconds, meaning that the current green interval is extended by 3 seconds if the advance loop detector is actuated by an arriving vehicle subject to the maximum green time limit.

Table 4.1: Network Signal Settings.

	Phase Index	Min Green (s)	Max Green (s)	All Red (s)
East-West Through	1	5	20	3
East-West Left Turn	2	5	20	3
North-South Through	3	5	20	3
North-South Left Turn	4	5	20	3

In Table 4.1, the phase index indicates the sequence of phases within the signal cycle. As mentioned earlier, the algorithm that PASSER V applies can calculate green splits based on different cycle lengths. PASSER V generally runs the simulation on varying cycle lengths within a range (e.g. 40 sec to 90 sec) at a certain increment (e.g. 5 sec). The parameter for increment can be modified by the user. The cycle length that produces the highest efficiency and attainability is selected as the best option. In this paper, a $[40, 90]$ range is used with an increment of 5 seconds for PASSER V to conduct the exhaustive search. By doing so, we intend to give PASSER V an opportunity to perform well.

4.4.1 Numerical Test on a Single Corridor

Signal coordination on a single corridor as Figure 4.1a is likely the most successful practice, or at least the most noticed, regarding network control. Its goal is twofold. One is for vehicle progression along the corridor without stop (mainly via offset); and the other is for as many as possible such vehicles to pass (via green band maximization). Arguably, established coordination algorithms perform better along a single corridor than on a network. Therefore, it constitutes a distinct case for performance test.

The test corridor has 5 intersections, which we believe is large enough to serve the purpose. Three traffic situations are used for the test as indicated in Table 4.3. They represent high, medium and low traffic situations respectively. VISSIM uses a default saturation flow rate of 2000 vph per lane under scenario 99-3 of Wiedemann 99 Car Following Model. The scenario describes a 2-lane road, with passenger car of 80 km/h and without heavy goods vehicle (HGV). The $\frac{v}{s}$ ratio of intersections for Figure 4.1a are shown in Table 4.2: .

Table 4.2: $\frac{v}{s}$ Ratio of Intersections in Figure 4.1a.

Volume	Intersection 1	Intersection 2	Intersection 3	Intersection 4	Intersection 5
Low	0.22	0.18	0.16	0.18	0.22
Medium	0.44	0.39	0.36	0.39	0.44
High	0.72	0.63	0.58	0.63	0.72

The reason why we use $\frac{v}{s}$ instead of $\frac{v}{c}$ is that there's no fixed cycle length for DORAS-Q, MP and MADM, where c is the cycle length. Since the total maximum cycle length is 80 seconds, the minimum capacity loss due to green loss shouble be $\frac{12}{80+12} = 0.13$. So if we converted the $\frac{v}{s}$ of high volume case to the traditional $\frac{v}{c}$, we have $\frac{v}{c} \approx \frac{0.7}{1-0.13} \approx 0.80$.

Each volume case is tested five times under different random simulation seeds. Simulation time for each is 3600 s in VISSIM, and the average delay and average stop per vehicle of these five runs is recorded. The simulation results are summarized in Table 4.4 and 4.5, Figure 4.2 and 4.3, respectively.

Table 4.3: Traffic Volume Used in Arterial Case.

Volume	On Major Arterial, V$_1(veh/h)$	On Minor Arterial,V$_2(veh/h)$
Low	500	200
Medium	900	500
High	1500	800

As shown in the results above, MADM appears the best option overall among all the five algorithms. For low to medium volume cases, DORAS-Q, MP, MADM and Actuated control all

Table 4.4: Corridor Case: Average Vehicle Delay in Seconds.

Volume	DORAS-Q	MP	MADM	PASSER V with actuation
Low	14.36	17.05	13.22	18.90
Medium	17.83	19.44	16.59	25.18
High	34.41	31.32	26.31	33.67

Table 4.5: Corridor Case: Average Vehicle Stop.

Volume	DORAS-Q	MP	MADM	PASSER V with actuation
Low	0.55	0.57	0.50	0.52
Medium	0.61	0.65	0.58	0.65
High	1.22	0.95	0.76	0.78

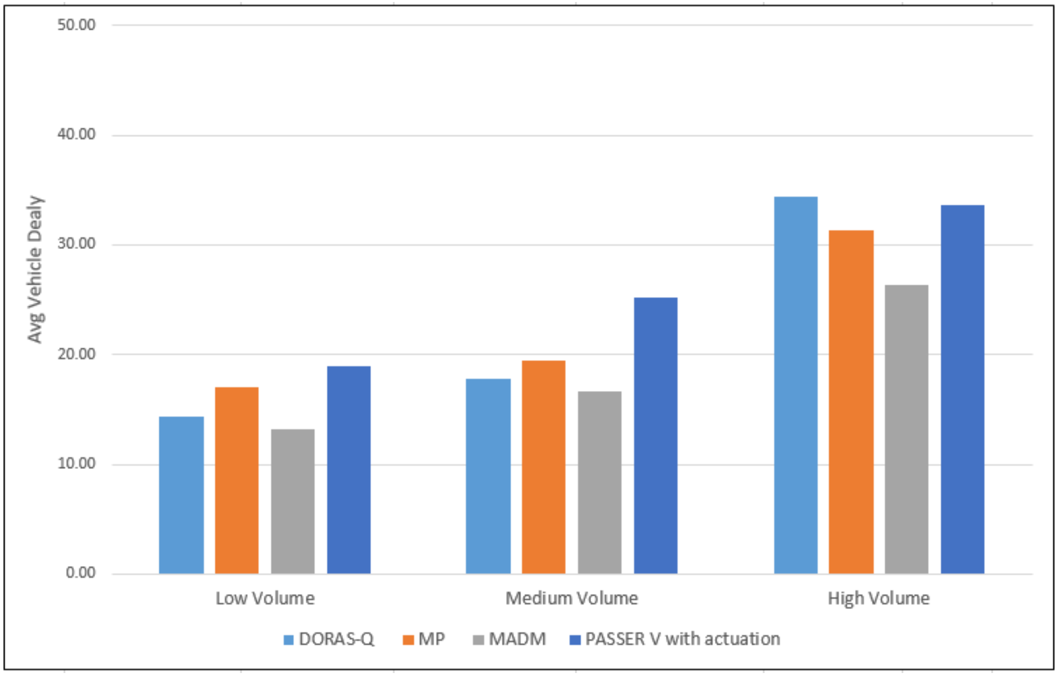


Figure 4.2: Average Vehicle Delay Through Arterial in Seconds.

have similar performances with trivial difference. In the high volume case, MADM performances noticeably better than the rest in terms of delay. In terms of vehicle stops, DORAS-Q has a comparably good performance in all the three traffic scenarios with MADM. The following analysis might help explain the results. First, PASSER V is a fixed timing system that does not respond to

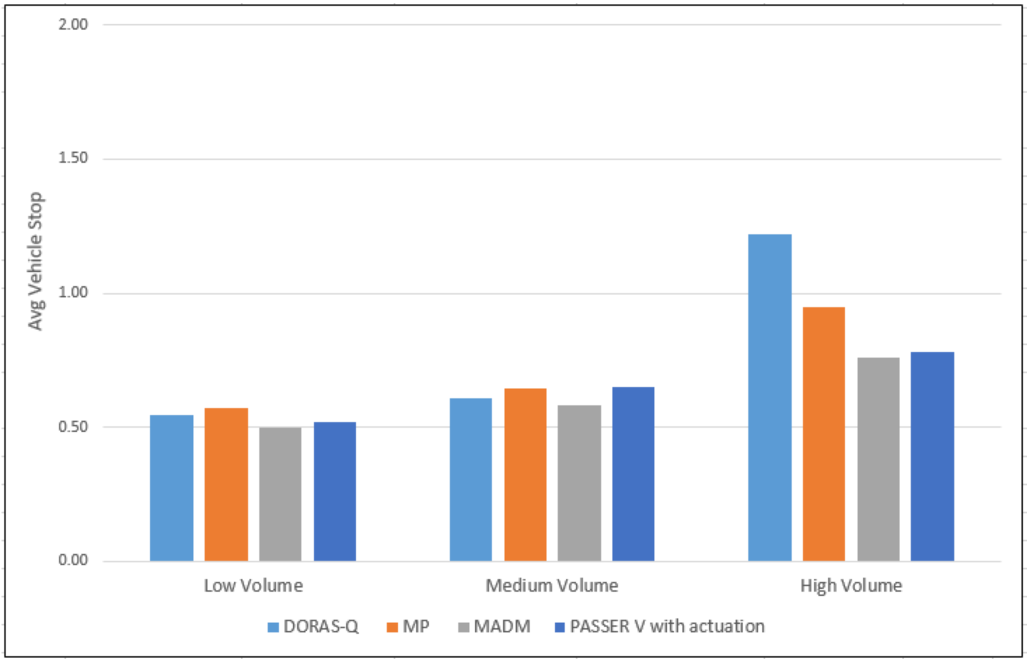


Figure 4.3: Average Vehicle Stop Through Arterial.

traffic so that it does not perform as well as other traffic responsive schemes; Second, DORAS-Q makes intersections efficient discharging queued vehicles with a certain capacity to outlook into arriving traffic. Note that the test network (e.g. corridor in this case) does not generate or end traffic within it, but only carries through traffic. Due to the special progression pattern in the form of platoons, the current phase, when its queue is cleared, has the current efficiency quickly dropping to zero to allow the switch of signal to the next phase. Therefore, DORAS-Q likely demonstrates the typical queue clearance process under heavy traffic. If every intersection performs most efficiently in vehicle discharging, the network performs efficiently in general because the total number of intersection crossings is a deterministic value when the network itinerary traffic is known. The traffic data show that the network is not saturated by traffic as indicated by the calculated V/C ratios. In these tested scenarios, queues at each intersection are generally able to be discharged within the corresponding phases with few exceptions.

Last, PASSER V does not perform as well as expected. The likely reason is that PASSER V optimizes for a fixed-timing plan. It only maximizes the bandwidth to accommodate the most vehicles along the corridor for best progression efficiency. Thus the phase plan and cycle length does not adjust with situational traffic.

4.4.2 Numerical Test on Grid Network

On this network as in Figure 4.1b, each volume case is tested five times under different simulation seeds in VISSIM with a total simulation time of 3600 s, and the average delay of these five runs is recorded. The test network has 9 intersections. Three traffic situations are used for the test as indicated in Table 4.7. The $\frac{v}{s}$ ratios for the intersections under the three traffic scenarios are shown in Table 4.6. The performance summary from the simulation is presented in Table 4.8 and equivalently in Figure 4.4.

Table 4.6: $\frac{v}{s}$ Ratio of Intersections in Figure 4.1b.

		Intersection								
		1	2	3	4	5	6	7	8	9
Volume	Low	0.27	0.25	0.27	0.29	0.28	0.29	0.27	0.25	0.27
	Medium	0.50	0.48	0.50	0.53	0.51	0.53	0.51	0.48	0.51
	High	0.84	0.80	0.83	0.88	0.85	0.88	0.84	0.80	0.84

Similar to corridor case, if we converted the $\frac{v}{s}$ of high volume case to the traditional $\frac{v}{c}$, we have $\frac{v}{c} \approx \frac{0.85}{1-0.13} \approx 0.98$, which is equally over-saturation condition.

Table 4.7: Traffic Volume Used in Network Case.

Volume	On Major Corridor, V$_1(veh/h)$	On Minor Corridor,V$_2(veh/h)$
Low	500	200
Medium	900	500
High	1500	800

Consistent to the results on a single arterial road as for Figure 4.1a, MADM performs the best among all the five algorithms in terms of delay, except for the low volume case in which MADM

Table 4.8: Network Case: Average Vehicle Delay in Seconds.

Volume	DORAS-Q	MP	MADM	PASSER V with actuation
Low	14.70	15.92	14.68	19.77
Medium	18.25	18.09	18.60	31.80
High	50.01	34.01	37.52	50.40

Table 4.9: Network Case: Average Vehicle Stop.

Volume	DORAS-Q	MP	MADM	PASSER V with actuation
Low	0.57	0.56	0.57	0.61
Medium	0.63	0.64	0.66	0.71
High	1.94	1.04	1.00	1.17

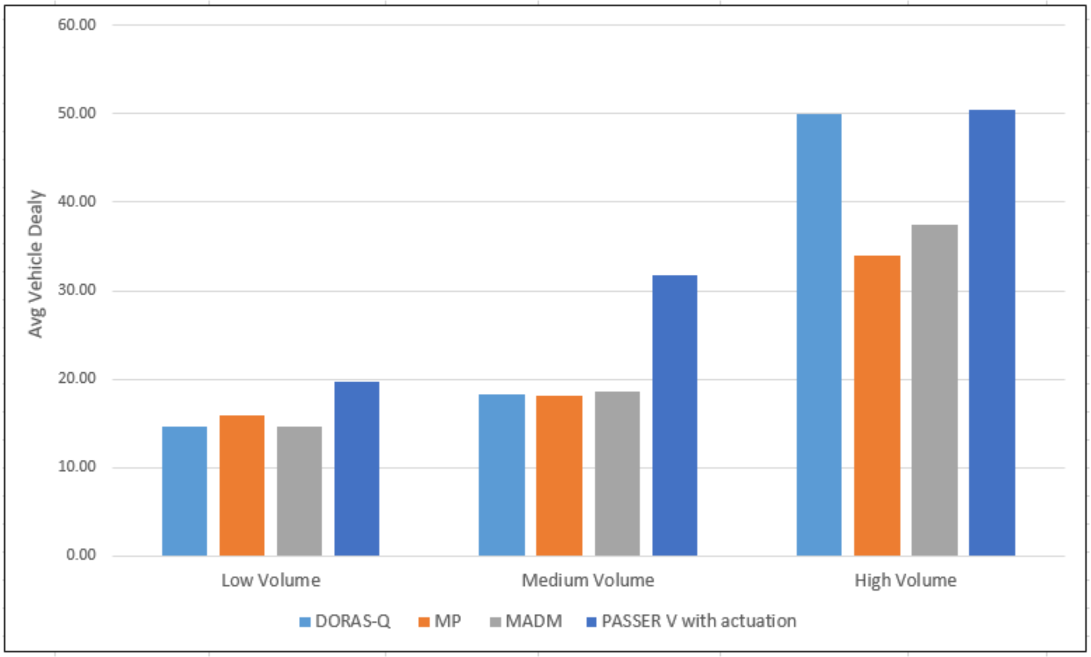


Figure 4.4: Average Vehicle Delay on Network in Seconds.

is second to DORAS-Q with an trivial margin. For low volume case, downstream queue length is not crucial and all the queued vehicles can be cleared in almost every phase/cycle. It is therefore not needed to check downstream queueing situation when deciding signal switch, meaning that it's good enough to let intersections function as isolated, which makes DORAS-Q the best or

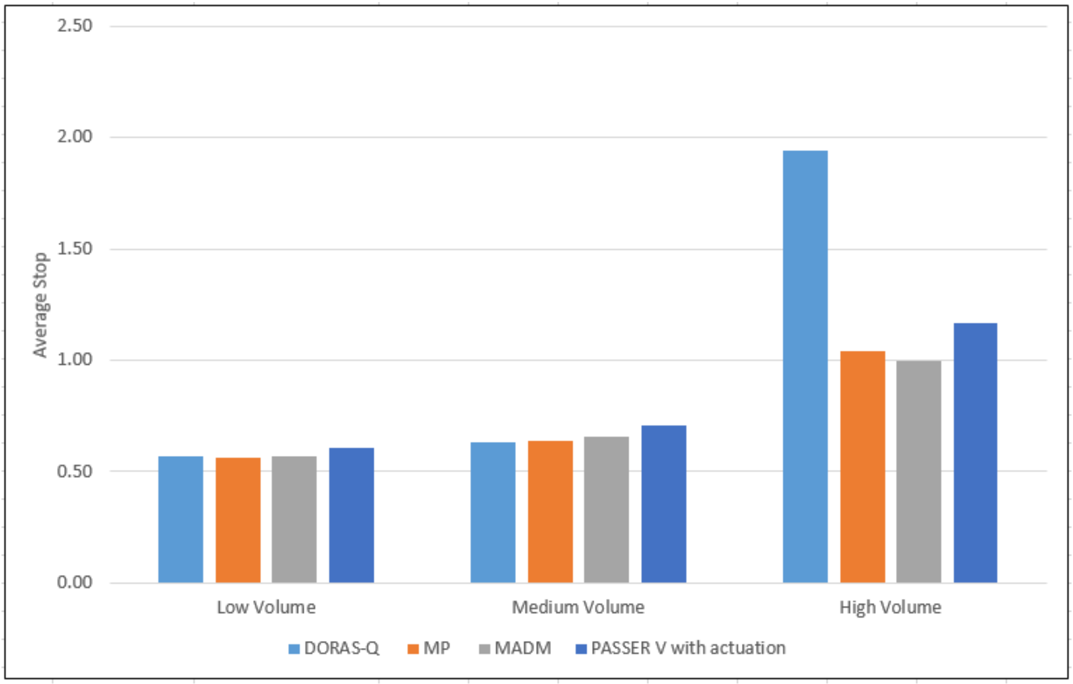


Figure 4.5: Average Vehicle Stops on Network.

very near best option. When traffic volume increases to medium level, DORAS-Q becomes less advantageous while MP and MADM are gradually outperforming and are very close to each other in terms of average delay. However, when the traffic volume further increases, MADM performs much better than all the rest methods. The possible explanation is that when downstream queues are large (weight $w < 0$), MADM will terminate current phase and switch to next phase even when the current intersection efficiency is higher than the switch-to efficiency. In doing so, MADM equivalently considers the network effect by avoiding pushing vehicle delay down to the next intersection. This case would probably cause the signal to switch more frequently than DORAS-Q and MP and would also increase the *network* discharge efficiency. To reiterate the idea of *network* efficiency, even if a few more vehicles are released during the current green phase if the signal does not switch, those vehicles would still need to wait at downstream intersection by joining the existing long queue there. So switching to next phase may benefit the entire system by releasing the vehicle at subsequent phase earlier if there is a larger *network* effect there.

Another finding is that actuated control method shows competitive performance especially when volume is high. The reason may be that it has the capacity of extending the green light to serving arriving platoons. In the tested network, close arrivals are almost always platoons, which makes the critical gap of the actuated control work very well. Between MP and MADM, MP does not consider intersection efficiency and therefore understandably under performs MADM, especially under heavy traffic.

Last, PASSER V performs the worst among all the algorithms in terms of the average delay. There are two possible reasons for this. The first is that PASSER V focuses on analyzing single arterial with multiple intersections. Even if multiple single arterial roads are identified and optimized by the bandwidth maximization method, it does not necessary optimize the entire system performance. Importantly, the default interference minimization algorithm in PASSER V only maximizes the green bandwidth to achieve the best progression efficiency, causing extra delay to vehicles from crossing roads or at other intersections.

4.5 Conclusions and future work

This paper builds on the previous work by Wang, Cao and Wang [77] about real-time optimal control algorithm for isolated intersections (DORAS) to propose a method for the real-time network traffic control problem. Network traffic signal control is a challenging topic without much satisfying theoretical foundation built so far. When developing signal control method for arterial network, how to include the impact from upstream and downstream intersections becomes a major challenge.

As an advantage, the max pressure method (MP) is able to consider the downstream and current intersection queuing situation to account for the network effect to some extent. On the other hand, the developed method DORAS considers intersection efficiency. Therefore, our proposed method combines both. Specifically, this paper proposes a mixed algorithm combining both the Max Pressure and DORAS algorithm.

Tests show that the mixed algorithm (MADM) is able to reduce the delay of vehicles in the network significantly compared with DORAS, MP, PASSER, or fully actuated control. There is no big difference between every control algorithm when traffic volume is low except for PASSER.

However, when traffic volume becomes larger, MADM outperforms other algorithms consistently. In addition, MADM does not require more traffic information than DORAS-Q, and is easy to be updated from DORAS or DORAS-Q.

Both DORAS/DORAS-Q and MP method have their own disadvantages. DORAS/DORAS-Q only focuses on isolated intersection level without considering the relationship between consecutive intersections. MP, however, only considers the current queue lengths but exclude the impact of future traffic arrivals. All of these shortcomings result in a weak network signal control policy. MADM therefore attempts to overcome those weakness from both local and systemic levels in a roadway network to achieve better outcomes, which are eventually proved in this paper.

Future work includes testing MADM in a real arterial roadway network to future prove the effectiveness and flexibility of the method, controlling computationally cost when dealing with larger network systems, and comparing MADM with more other existing arterial or network control methods.

5. LEFT-TURN SPILLBACK ESTIMATION: A SPECIAL SCHEME STUDY*

Different from the general optimal control studied earlier, in this chapter we study a special control application that is related to a present practice. This chapter sets foot on innovations to existing practices while the earlier chapters more about potential, transformation future theories.

In a just published paper by Cao *et al* [78], a left-turn spillback estimation algorithm is proposed and is included in this thesis. This algorithm has the potential to be implemented together withe the proposed algorithms for isolated intersection and signalized arterial network in the future.

Intersection queue length estimation is important to signal timing. Some intuitive observation tells that signal timing is about reducing vehicle queuing at the intersection. A better knowledge of the intersection queuing enables development of a better signal timing. In a special circumstance where left-turn vehicles often block through traffic when they exceed the left-turn bay capacity, the left-turn vehicular queue estimation is of particular significance. This is because blockage of through traffic undermines the intersection efficiency and makes the signal timing sub-optimal. This part of the thesis specially deals with the left-turn vehicular queue as well as the probability that this left-turn queue spillback obstructs through traffic movement.

This thesis position the study in a context of connected vehicles. Connected Vehicles is a promising new technology built on short range wireless communication [39]. The application of connected vehicles allows communication between vehicles and between vehicles and the infrastructure, and therefore shows a promise to overcome the limitations of fixed location detectors by having more longitudinal vehicle and traffic information. This additional longitudinal traffic information offers a new opportunity to left-turn traffic treatment and therefore, the entire intersection signal timing. This is because the left-turn vehicles have to stop on the through lane waiting to enter the left-turn bay, which is called left-turn vehicle spillback, when the left-turn volume exceeds left-turn bay's capacity. Left-turn spillback blocks through movement and reduce the through capacity

*Reprinted with permission from "Left-Turn Spillback Probability Estimation in a Connected Vehicle Environment"by X. Cao, J. Jiao, Y. Zhang, and X. Wang, 2019. *Transportation Research Record*, vol. 2673, pp. 753-761, Copyright 2019 by SAGE Publishing.

at intersections. Hence the study of left-turn spillback is desired for better intersection operation or signal timing. This study proposes a probabilistic model for the left-turn spillback estimation by first starting with queue length estimation from connected vehicle applications, and presents a demonstration on how left-turn spillback estimation can help signal timing improvement.

5.1 Queue Length Estimation

This section is to estimate the total queue length at the intersection (Figure 5.1) if locations of connected vehicles in the queue are all known. The total queue length here refers to the total number of queued vehicles in both left-turn bay and the adjacent through lane. Technically, the goal is to determine the expected value of the total queue length given the locations of connected vehicles. The geometry of this approach is a through lane along with a left-turn bay, which has a length of 4 vehicles.

It is assumed that the distance of connected vehicle from the stop line can be measured by location tracking technologies. The location of vehicles in queue will be measured by the number of vehicles to the stop line. The arrivals are assumed to follow a Poisson distribution. Notations used in this paper are introduced as follows.

Notations

p penetration rate.

λ average arrival rate.

N_{LC} location of the last connected vehicle in the queue in the through lane, in number of vehicles; [As in Figure 5.1(b) later, in the presence of spillback, N_{LC} is the sum of N_B and the queued vehicles after the left-turn bay in the through lane until last connected vehicle; in the absence of spillback, N_{LC} is the total number of the queued vehicles in the through lane until the last connected vehicle;]

N location of the last queued vehicle in the through lane from the stop line, in number of vehicles; [As in Figure 5.1(b) later, in the case of spillback, N is the sum of N_B and the queued vehicles after the left-turn bay in the through lane; in the absence of spillback, N is the total number of the queued vehicles in the through lane;]

N' total number of queued vehicles in both left-turn bay and adjacent through lane.

N_B bay storage length in number of vehicles.

ρ_{LT} left-turn vehicle percentage among vehicles coming in the approach of interest.

N_{LT} number of left-turn vehicles ($N_{LT} = N' \times \rho_{LT}$).

N_{TH} number of through vehicles ($N_{TH} = N' - N_{LT}$).

N can be calculated based on Comert and Cetin's method [45] using the Equation 5.1 below:

$$E(N) = \sum_{n=N_{LC}}^{\infty} \frac{n[(1-p)\lambda]^n}{n! \sum_{k=N_{LC}}^{\infty} \frac{[(1-p)\lambda]^k}{k!}} \tag{5.1}$$

As shown in Figure 5.1, the final status before the through green phase starts should either be 5.1(a) or 5.1(b). All the parameters (N, N_{LC}, N_B) in this figure are for the number of vehicles. Here, N_B is equal to 4, meaning that the bay can store 4 vehicles.

In Figure 5.1(a), N is the expected number of queued vehicles in the adjacent through lane. Similarly, in Figure 5.1(b), if we switch the left-turn queued vehicles with those in the adjacent through lane before the dashed line, the graph will become just like 5.1(a). Clearly, the expected total queue length is the sum of N and the number of vehicles in left-turn bay, which can be any number within $[0, N_B]$. In order to simplify the calculation, we assume the probability of any number of vehicles within $[0, N_B]$ in the left-turn bay to be equal. The equation to find N' is shown below:

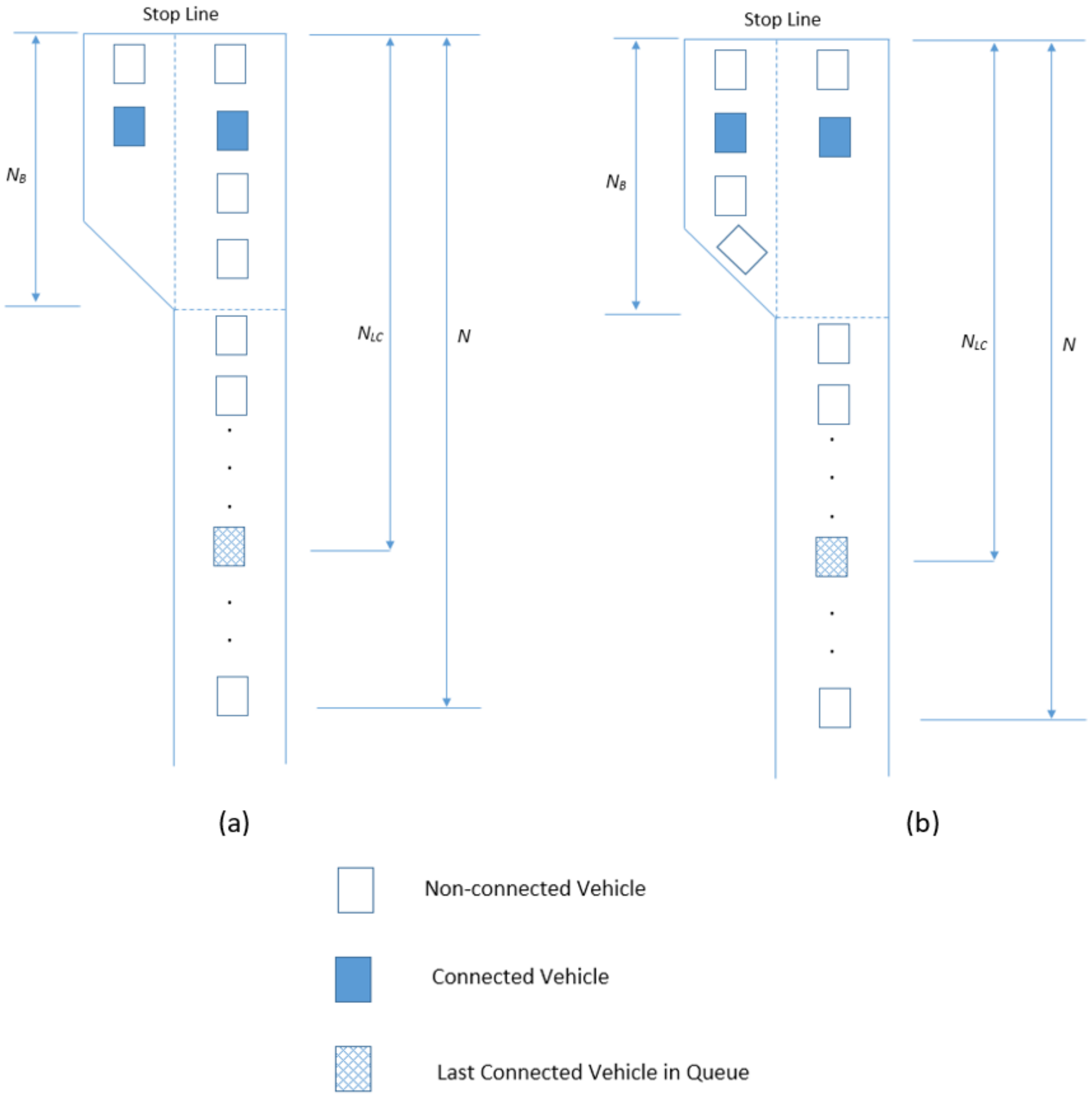


Figure 5.1: Layout of Lane Approach with Left-turn Bay ($N_B = 4$).

$$E(N') = \frac{\sum_{i=o}^{N_B} \sum_{n=N_{LC}+i}^{\infty} \frac{n[(1-p)\lambda]^n}{n! \sum_{k=N_{LC}+i}^{\infty} \frac{[(1-p)\lambda]^k}{k!}}}{N_B} \tag{5.2}$$

In addition to the situation shown in Figure 5.1, it is possible that the last connected vehicle is queued in the left-turn bay. There are two situations in which this can happen. First, the arriving volume is low from the approach of interest, implying a low probability of left-turn spillback. The other one is that the connected vehicle penetration rate is very low, and that we only know the locations of very few vehicles. It is difficult to estimate the total number of queued vehicles given such limited information, unless the total number of queued vehicles are large enough (see the results in Figure 2). Since this study does not focus on examining the impact of penetration rate, the last connected vehicle is assumed always in the adjacent through lane in the following sections. This assumption is reasonable especially in the simulation because the arriving traffic volume is set large enough to generate enough cases of left-turn spillback. In these simulations, the probability of having the last connected vehicle in the through lane is relatively high.

5.2 Spillback Estimation

After the expected total number of arrivals is obtained, we are able to find the possibility that left-turn spillback happens during the green phase. As mentioned earlier, spillback happens when at least one through vehicle is blocked and cannot move through. A blockage of the through movement also means that there are at least $N_B + 1$ left-turn vehicles before the first blocked through vehicle. Here the probability of having a left-turn spillback is defined as P_{sb} . Since the left-turn bay capacity is N_{LT} , we have the following equation:

$$P_{sb} = \frac{\sum_{i=N_B+1}^{N_{LT}} \binom{N_{TH}-1+i}{i}}{\binom{N'}{N_{LT}}} \tag{5.3}$$

In Equation 5.3, the numerator is the number of combinations, each of which has a spillback, while the denominator is the total number of combinations for all the queued vehicles, with and without spillback.

5.3 Analysis of Spillback Probability, Left-turn Percentage and Bay Length

To evaluate the accuracy and reliability of the spillback estimation, the computed result is compared with simulation results from VISSIM. VISSIM simulation results are regarded as the real-world scenarios for the evaluation purpose. First, VISSIM runs the simulations and produces files of vehicle trajectory data. Second, Matlab is used to process the output files by investigating queue conditions at the end of every red phase of the interest approach and check whether there is spillback or not. Next the probability of left-turn spillback is calculated by dividing the number of cycles with spillback by the total number of cycles in the simulation. The results from the proposed estimation are compared with those from the simulations.

5.3.1 Simulation Setup

An illustrative, isolated intersection is set up for the left-turn spillback test in VISSIM. The layout of the intersection is seen in Figure 5.2. The intersection has four legs with a left-turn bay for each approach. Because the intersection is meant to test left-turn spillback, only one approach is investigated. In this case, we select the eastbound approach for study. It should be noted that the proposed estimation method can be applied to intersections with multiple through lanes. Since only queues of left-turn bay and adjacent through lane will be used in our model, the number of through lanes does not impact the results. For simplicity, during the simulation only one through lane is set up for each approach. We decide that the volume generated onto the approach of study is 900 veh/hr in order to ensure a reasonable level of spillback probability. Low volumes spillback rarely occurs. The signal timing of the intersection is obtained from SYNCHRO 9 optimization with the same intersection settings. There are four phases for this intersection, which are east-west though, east-west left-turn, north-south though, and north-south left-turn. We use actuated control in the simulation. The detailed setting includes: a 2s critical gap, a 15s minimum green time for though phases, a 35s maximum green time for through phases, a 10s minimum green time for left-turn phases, and a 25s maximum green time for left-turn phases.

VISSIM has randomness in every cycle of simulation. Although the volume is set to be 900

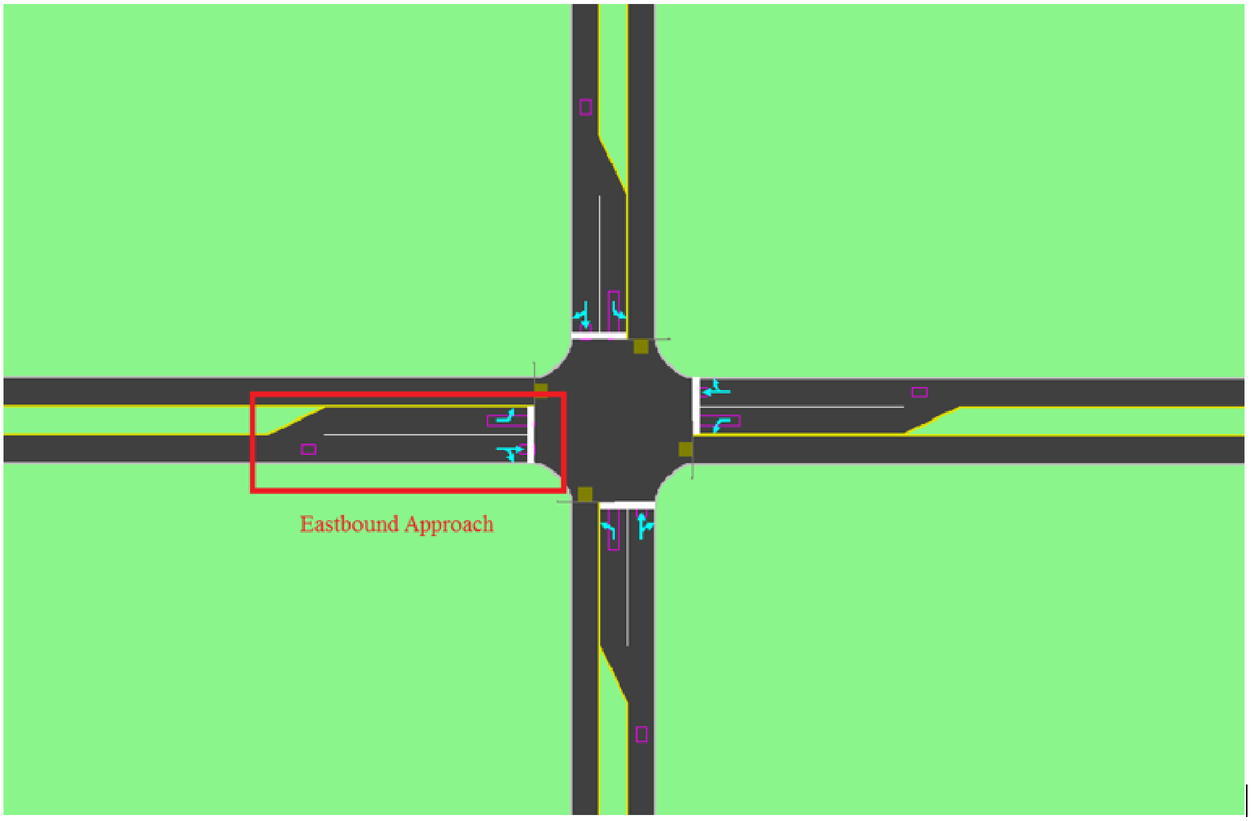


Figure 5.2: Layout of Intersection for Left-turn Spillback Simulation.

veh/hr for an approach, the actual number of vehicles arrived at the intersection in each cycle varies randomly. This is also true for left-turn percentage: as in one cycle it can either be greater or smaller than the pre-defined value of turning percentage. For example, a setting of 50% left-turn percentage may have actually 20% or 70% left-turn vehicles in a specific cycle. This nature of VISSIM reflects real world scenarios, but is an issue for a fair comparison with the calculated result. The proposed left-turn estimation method is based on knowledge of left-turn percentage of the queue at the end of each cycle. Hence, to obtain the accurate left-turn percentage at the end of each cycle, the simulation results are collected by classifying the queuing condition of each cycle by specific characteristics and then aggregating cycles with the same characteristics together. The results of VISSIM simulation are classified into two categories: the total number of queued vehicles during the red phase and the total number of left-turn vehicles queued during the red phase. For the total number of vehicles queued during the red phase, the number 18, 19, 20 and 21 occurs most in the aggregated results, and are considered to best represent the current volume setting. The total number of left-turn vehicles in queue has to be greater than the bay length (represented by the number of vehicles that the bay can accommodate) for spillbacks to occur. For a bay length of 4 vehicles for example, the total number of left-turn vehicles has to be at least 5. On the other hand, when the number of left-turn vehicles is larger than 11 the spillback probability is almost 1. This gives a range of 5 to 11 left-turn vehicles in the queue. The number of total queued left-turn vehicles corresponds to different left-turn percentages during a cycle. Hence the results are classified into categories according to the specific number of total queued vehicles and the specific number of total queued left-turn vehicles. The probability of spillback is calculated in each category.

To account for different left-turning percentages and to include enough samples, three levels, 35%, 50% and 70%, representing medium, high and very high value of left-turn percentages, are selected as inputs in VISSIM. Each level of left-turn percentage is run in VISSIM for 50 times with 50 random seeds for a total of 500,000 seconds. All the simulation results are then aggregated and further classified according to the categories defined previously. To investigate the effect of the left-turn bay length, the storage length in VISSIM is modified in order to accommodate 3, 4, 5, 6 and 7 vehicles. The same simulation process is then applied to scenarios with different bay lengths.

5.3.2 Simulation Results

Figure 5.3 shows the relationship between the estimated left-turn spillback probability and actual results obtained from simulation. Four different values of queue lengths are selected in order to test the proposed method under different volumes. The reason is that as mentioned previously, even the volume is set to be 900 vph in VISSIM, the actual number of arrivals during each cycle can still be quite different according to the software characteristics of VISSIM. In order to study the accuracy of the proposed method under different volumes, N' can be used as a substitution of actual volume. In order to study the effect of left-turn ratio, different numbers of left-turn vehicles should be examined as well. Since the number of left-turn vehicles needs to be an integer, it is straight-forward to just use the number instead of a fixed ratio.

Based on the graphs, it is obvious that the actual left-turn spillback probability (e.g. in the simulation) shows consistency with that of the estimated probability obtained from the proposed model. In most cases, the model estimated probability is slightly greater than the actual. The reason can be explained that our model considers all the spillback combinations. However, it appears that the simulation results do not cover all those cases in the finite simulation time. The actual situation as in the simulation is more complicated. For example, some left-turn drivers in the through lane may change their mind to go through at last, which can impact the real spillback probability calculation.

In actual situations, different intersections have different bay lengths. Figure 5.4 shows the spillback probability estimation compared with simulation results. For the purpose of illustration, the total number of arrivals is set to be 18 while 8 of them are left-turn vehicles. From the Figure 5.4, it is easy to recognize that the proposed method works better when the bay length is small, which also means that left-turn spillback is more likely to happen. Longer bay length results in less accurate estimations. The reason might be that when left-turn spillback happens only few

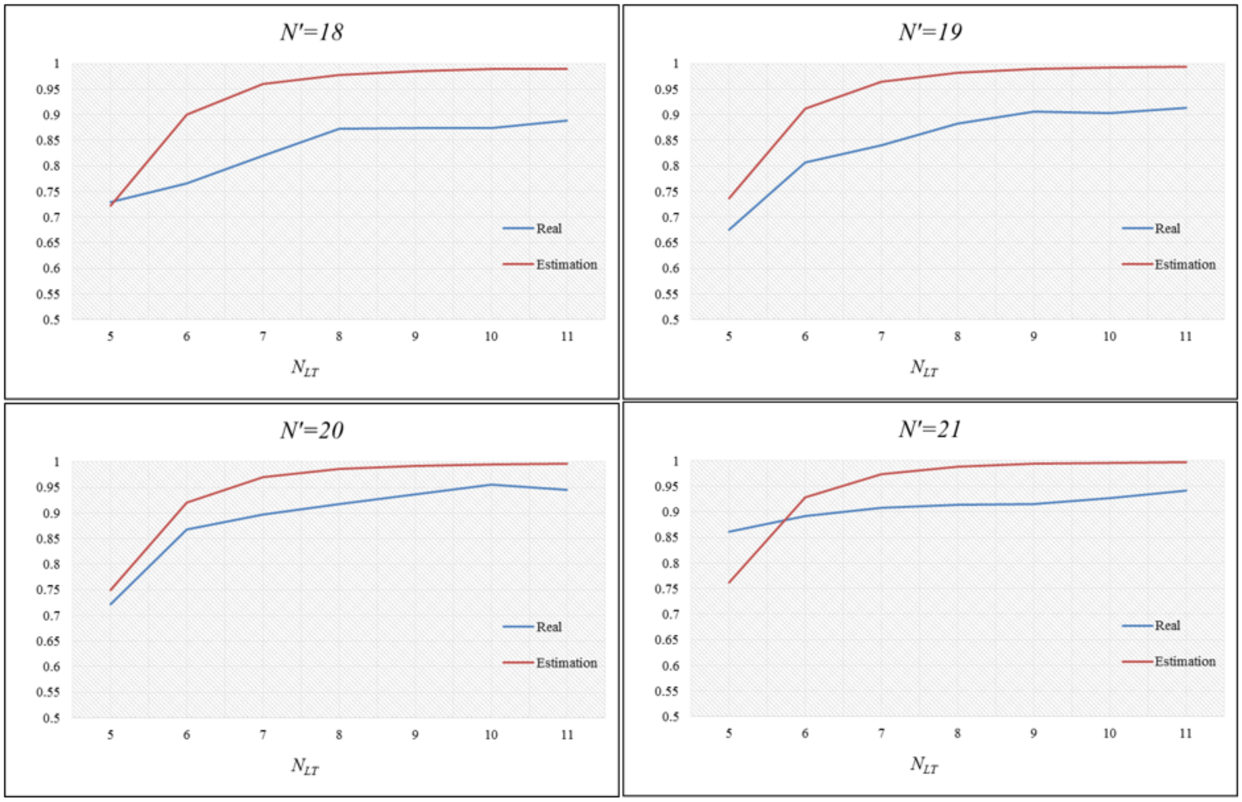


Figure 5.3: Estimated Left-turn Spillback Probability versus. Real Probability for Different N' and N_{LT} .

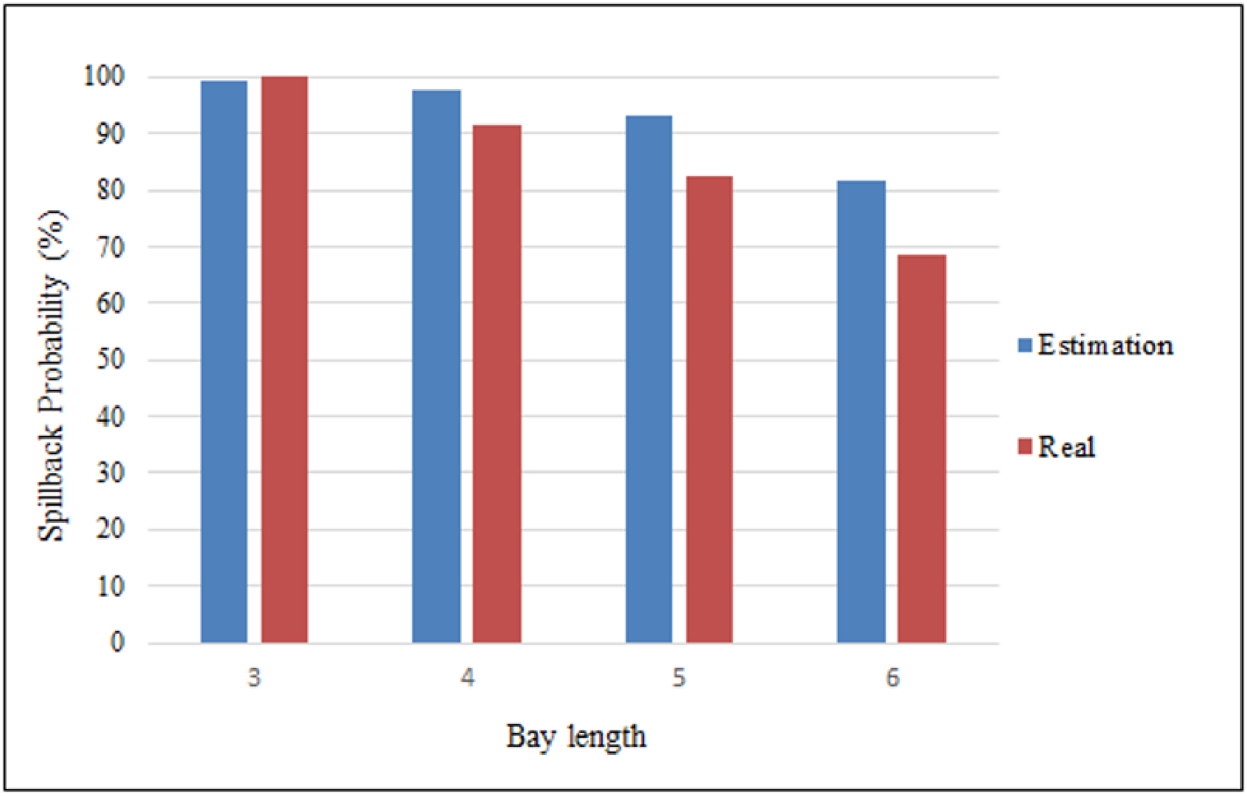


Figure 5.4: Estimated Left-turn Spillback Probability versus. Real Probability for Different N_B.

times during a certain time interval, it is more difficult to predict accurately the occurrences and the probability of the spillback. This finding can also help assess whether a left-turn bay is long enough to keep the spillback probability below a threshold in peak hours. Bay extension would be necessary if the spillback probability is too high, especially during peak hour.

5.4 Improving Signal Control Using Spillback Estimates

As mentioned previously, the proposed left-turn spillback estimation method can be used to improve signal control. A demonstration of the implementation is presented in this section. A real world case in College Station, Texas with two consecutive intersections is selected for the demonstration. Figure 5.5 shows that Southwest Parkway intersects with Anderson Street to form the upstream intersection and intersects with Texas Avenue to form the downstream intersection. Left-turn spillback happens frequently at the eastbound approach of the downstream intersection, especially during peak hours, as vehicles turn to the major arterial, Texas Avenue. This results in

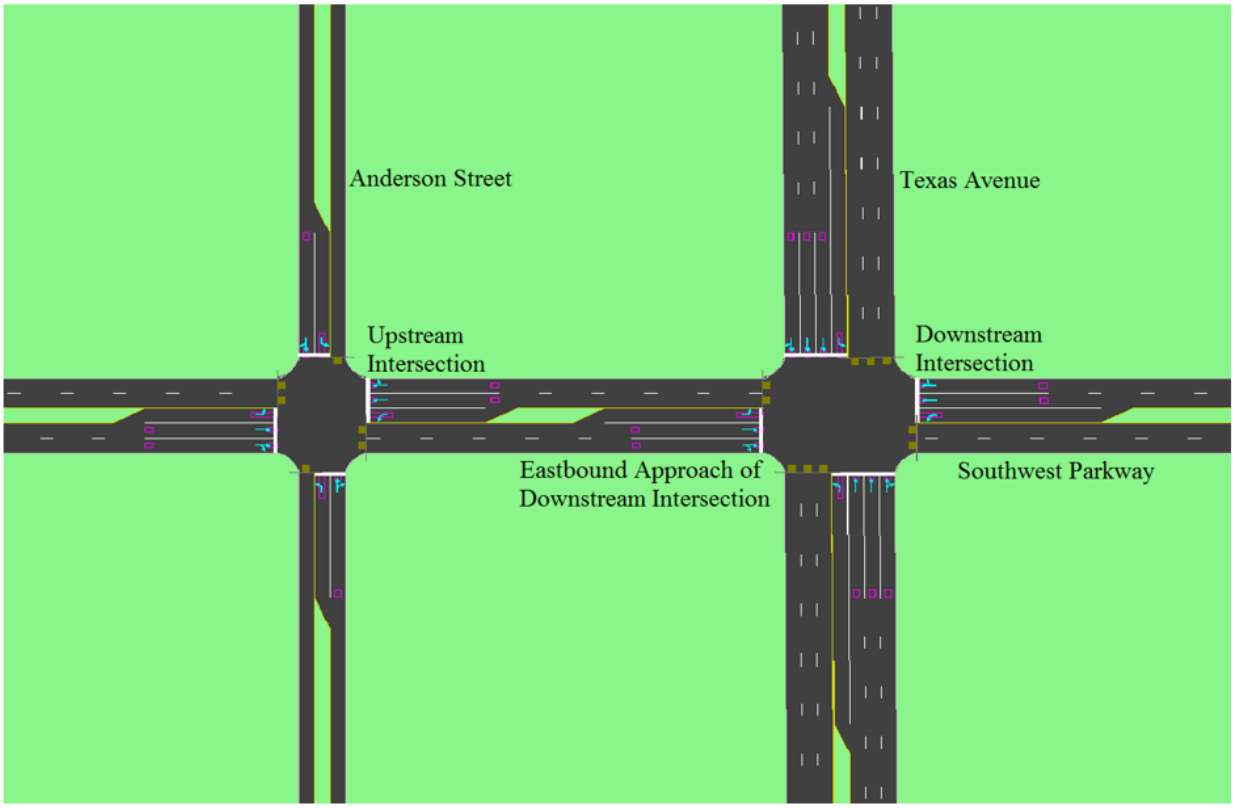


Figure 5.5: Layout of Two Consecutive Intersections for Signal Control Simulation.

long queues and large delays at the downstream intersection, and it is desirable to adjust signal timing strategy in order to improve the traffic conditions. Hence this downstream intersection is the major intersection to examine.

Real volume data and signal timing are input into VISSIM simulation for comparison. The simulation is run for 3600 seconds. The input volume is set to be 1000 vph for the eastbound traffic at the upstream intersection, resulting an average volume of 900 at the downstream intersection on Texas Avenue. Due to the randomness, the actual arrivals in each cycle vary. An observation shows about 30% of the cycles having spillback.

There are two methods for signal timing adjustment using the left-turn traffic spillback information. The first is to adjust the timing of the downstream intersection directly whenever spillback is detected. The second is to reduce the incoming traffic of the upstream intersection after spillback is detected at the downstream intersection. Taking into account that the downstream intersection involves a major arterial (e.g. the Texas Avenue), this section only demonstrates the second method, which adjusts the upstream intersection signal timing to reduce the incoming traffic. The arrivals on the major arterial will not be affected by the signal timing adjustment. Since the volume on Anderson Street is relatively low, most of the incoming traffic to the downstream intersection comes from the eastbound approach of the upstream intersection. Hence adjusting the through phase green of the eastbound approach alone is able to satisfy the need of downstream volume reduction. The proposed method calculates every second the probability of a spillback. The upstream intersection signals are adjusted immediately once the calculated spillback probability exceeds a certain threshold. The detailed signal control strategy is summarized as following:

Simulation clock starts at time $i = 0$

- Step 1: Given $p, \lambda, N_{LC}, \rho_{LT}$. Calculate P_{sb} at time i. If $P_{sb} \geq \alpha$, go to Step 2. Otherwise, go to Step 3.
- Step 2: If through phase is green at upstream intersection, switch to next phase immediately, $i = i + t$. Go to Step 1; Otherwise, go to Step 3.
- Step 3: If $i \leq T$, $i = i + 1$, go to Step 1; Otherwise, simulation ends.

p is set up as 0.1 and 0.9 in the test for the purpose of comparison. N_{LC} is assumed to be the last connected vehicle in the through lane adjacent to the left-turn bay on eastbound approach of downstream intersection. α is the threshold that is used to decide if the spillback probability is too high, as described earlier. t represents travel time from the upstream intersection to downstream intersection based on the distance divided by the speed limit. Those two parameters could be chosen based on real situation or practical judgment. In this thesis, α is set to be 50$\%$ while t equals to 20 seconds. T is the total simulation period, which is 3600 seconds as mentioned before for each run. 20 runs in total are conducted.

ρ_{LT} is set up in a different way since it can vary from cycle to cycle. However, based on the given connected vehicle information, the total number of connected vehicles in the through traffic and total number of left-turn vehicles before the last connected vehicle are both available as well.

As a result, the estimated left-turn ratio is obtained. Note that in calculating the left-turn ratio, only the vehicles in the adjacent through lane and left-turn bay are used. For example, if there are total 10 through connected vehicles and 5 left-turn connected vehicles,$\rho_{LT} = \frac{5}{5+10} = 33.3\%$.

If the upstream through phase is switched to the next phase, P_{sb} will not be assessed for the succeeding 20 sec. The reason is that the adjusted incoming volume needs this travel time to arrive at the downstream intersection to actually have an impact on the spillback probability. So, during this period, no judgment is needed.

The results of the simulations are presented in Figure 5.6. VISSIM provides results on average vehicle delay (e.g. delay per vehicle) for each movement of the two intersections. The bars under unadjusted category are for results from the original signal timing and the bars under adjusted category represent results from the adjusted signal timing based on the proposed method. The first group in Figure 5.6 represents the average delay for all the vehicles entering the two intersections. The second group of bars represents the average delay of vehicles on the eastbound approach of downstream intersection. The last group represents the average delay of only left-turn vehicles on the eastbound approach of the downstream intersection.

5.5 Conclusion and Discussion

This chapter focuses on estimating left-turn vehicle spillback probability. It starts with the total queue length estimation for vehicles on a through lane with a left-turn bay using location of the last connected vehicle. The results show that total queue length estimation is more accurate if the connected vehicle penetration rate p is larger. Furthermore, a larger N_{LC} can also improve estimation when p is small.

An equation is proposed to calculate the probability of left-turn spillback, which is tested using VISSIM traffic simulation. Several cases are compared in order to access the accuracy of the proposed equation with varying traffic volume, left-turn ratio, and bay length. The results show that the proposed equation is able to provide fairly accurate spillback probability, especially at large left-turn ratios. The equation works better for shorter bay length, while the overall result for different bay lengths is still very convincing.

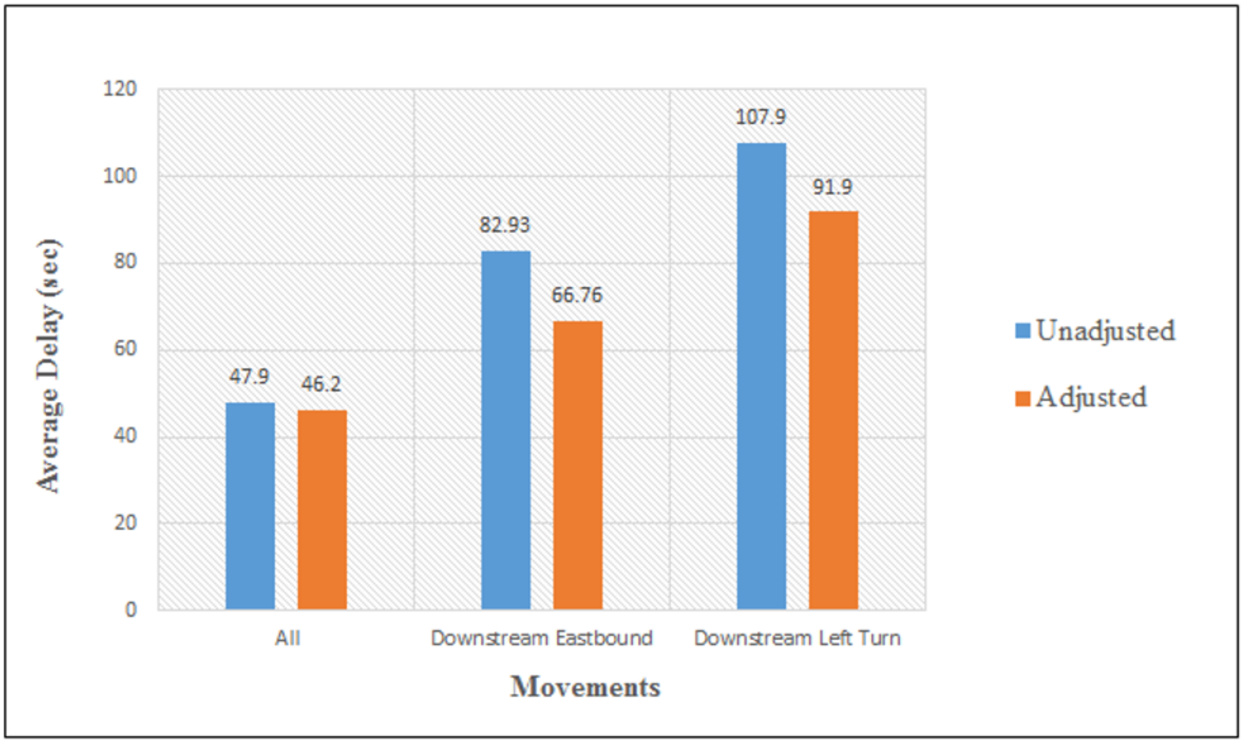


Figure 5.6: Average Vehicle Delay Comparison.

Finally, an improved signal control in a connected vehicle environment is demonstrated based on the spillback probability obtained from the proposed equation. The main idea is to adjust upstream signal phase to decrease incoming volume to the major intersection, which has high spillback probability. The results suggest that such a strategy is very effective in reducing potential left-turn spillback without causing extra delay to the whole system. The average delay decreases 20% for all the vehicles going from upstream intersection to downstream intersection. If left-turn vehicles are considered independently, the decrease is about 15%. Overall, the proposed spillback treatment strategy is considered to be effective.

Future work includes relaxing the requirement of last connected vehicle in estimating left-turn spillback. Such left-turn spillback algorithm is also available to be combined with the previous proposed real-time signal control methods for both isolated intersection and arterial network, which will be developed in the future study.

6. CONCLUSION

This dissertation attempted to study the optimal policy for both isolated and coordinated real-time signal control with complete traffic information. The optimal policy was defined as one with a series of signal switch points of time that keep the intersection vehicle waiting time to its minimum.

For the isolated intersection case, a simple structure to interpret and intuitively understandable, the optimal condition implied was that signal control shall maximize the intersection efficiency, which deepened the understanding of the queue clearance policy that was well studied in literature. The proposed algorithm DORAS, which was based on projected vehicle clearance for each phase, was proved to be able to significantly reduce delay compared with fully actuated control. Another simpler version called DORAS-Q was also proposed, which required much less traffic information than DORAS but still provided obvious reduction in vehicle waiting time. Although both algorithms were not superior compared with OPAC III, they required much less time in computing and appeared to be more efficient, which could save lots of labor and time in practical implementation.

DORAS and DORAS-Q both were able to accommodate constraints such as minimum/maximum green time and phase sequence requirement. In addition, they were able to conduct differential treatment of other parameters, such as vehicle types (trucks & buses) and lane types (left-turn & u-turn), by assigning varying weight factors and parameters.

Secondly, this study investigated the case of coordinated signal control on the network. The network control algorithms were proposed in order to minimize the average vehicle waiting time of the whole system. Two different ways of treating network signal control were considered. The first one was to simply apply the control algorithms that are developed for isolated intersection (DORAS-Q and actuated control) to every intersection in the network. The second way took the relationship between upstream and downstream intersection into account by using weighted algorithms (MP and MADM). Varying volume cases were investigated using different control algorithm, including the traditional bandwidth maximization method. Overall, the proposed algorithm MADM performed the best especially under high volume case. DORAS-Q and MADM were very close to each other under low and medium volume cases, which appeared to be reasonable since the impact from downstream intersection to upstream intersection is not significant. Although DORAS-Q and MP were able to compete with MADM under low and medium cases, they performed much worse than MADM when traffic volume was high with about 36% increase in average delay. When traffic volume was high, DORAS-Q and MP switched the signal of each intersection in different speeds, while MP was much faster than DORAS-Q. However, MADM was able to achieve a much more balanced switching speed than the other two, which in turn generated a better control policy to minimize the delay. In addition, the traditional bandwidth maximize method performed much worse than the rest, which showed its weakness in handling arterial network rather than a single arterial.

The last part of this dissertation studied the left-turn spillback problem. An estimating equation was proposed to calculate the probability of left-turn spillback based on probability theory. The simulation results showed that the propose equation was able to provide fairly accurate spillback probability, especially at large left-turn ratios. Moreover, an improved signal control algorithm in a connected vehicle environment was developed, which was based on the spillback probability obtained from the proposed equation. The simulation results indicated that the algorithm was very effective in reducing potential left-turn spillback without causing extra delay to the whole system. This left-turn spillback treatment can also be added to DORAS or MADM if the spillback problem exists in an isolated intersection or arterial network.

Future work includes further improving the estimation of the marginal effect from signal switch, deriving methods to estimating the future arrivals for each approach more accurately to a longer time period, improving the coordinated control algorithm by not only consider the queue condition but also other parameters (platoons,speed limits, etc.). In addition, both the isolated intersection signal control and coordinated network signal control still need to be tested in practice to see if there is any other limitations.

REFERENCES

[1] D. Schrank, T. Lomax, and B. Eisele, "2012 urban mobility report," *Texas Transportation Institute*, pp. 1–57, 2011.

[2] C. Systematics, "Traffic congestion and reliability: Linking solutions to problems," tech. rep., United States. Federal Highway Administration, 2004.

[3] E. A. Mueller, "Aspects of the history of traffic signals," *IEEE Transactions on Vehicular Technology*, vol. 19, no. 1, pp. 6–17, 1970.

[4] C. McShane, "The origins and globalization of traffic control signals," *Journal of Urban History*, vol. 25, no. 3, pp. 379–404, 1999.

[5] N. J. Goodall, B. L. Smith, and B. Park, "Traffic signal control with connected vehicles," *Transportation Research Record*, vol. 2381, no. 1, pp. 65–72, 2013.

[6] N. H. Gartner, *OPAC: A demand-responsive strategy for traffic signal control*. No. 906, 1983.

[7] F. Orcutt, "Primer for traffic selection," *Public Works*, vol. 106, no. 3, 1975.

[8] F. V. Webster, "Traffic signal settings," tech. rep., 1958.

[9] Y. Zhao and Z. Tian, "An overview of the usage of adaptive signal control system in the united states of america," in *Applied Mechanics and Materials*, vol. 178, pp. 2591–2598, Trans Tech Publ, 2012.

[10] P. T. Martin, J. Perrin, B. R. Chilukuri, C. Jhaveri, Y. Feng, *et al.*, "Adaptive signal control ii," tech. rep., University of Utah. Dept. of Civil and Environmental Engineering, 2003.

[11] S. Goel, S. F. Bush, and C. Gershenson, "Self-organization in traffic lights: Evolution of signal control with advances in sensors and communications," *arXiv preprint arXiv:1708.07188*, 2017.

[12] G. Newell and E. Osuna, “Properties of vehicle-actuated signals: Ii. two-way streets,” *Transportation Science*, vol. 3, no. 2, pp. 99–125, 1969.

[13] R. Cowan, “An improved model for signalised intersections with vehicle-actuated control,” *Journal of Applied Probability*, vol. 15, no. 2, pp. 384–396, 1978.

[14] M. C. Dunne, “Traffic delay at a signalized intersection with binomial arrivals,” *Transportation Science*, vol. 1, no. 1, pp. 24–31, 1967.

[15] R. Akcelik, “Estimation of green times and cycle time for vehicle-actuated,” 1994.

[16] F.-B. Lin, “Estimation of average phase durations for full-actuated signals,” *Transportation Research Record*, vol. 881, no. 881, pp. 65–727, 1982.

[17] G. F. Newell, “Theory of highway traffic signals,” 1989.

[18] X. B. Wang, K. Yin, and H. Liu, “Vehicle actuated signal performance under general traffic at an isolated intersection,” *Transportation research part C: emerging technologies*, vol. 95, pp. 582–598, 2018.

[19] A. J. Miller, “A computer control system for traffic networks.,” 1963.

[20] D. Ross, R. Sandys, and J. Schlaefli, “A computer control scheme for critical-intersection control in an urban network,” *Transportation Science*, vol. 5, no. 2, pp. 141–160, 1971.

[21] A. Stevanovic, *Adaptive traffic control systems: domestic and foreign state of practice*. No. Project 20-5 (Topic 40-03), 2010.

[22] N. H. Gartner, P. J. Tarnoff, and C. M. Andrews, “Evaluation of optimized policies for adaptive control strategy,” *Transportation Research Record*, no. 1324, 1991.

[23] K. L. Head, P. B. Mirchandani, and D. Sheppard, *Hierarchical framework for real-time traffic control*. No. 1360, 1992.

[24] K. L. Head, P. B. Mirchandanai, S. Shelby, *et al.*, *The RHODES prototype: a description and some results*. Transportation Research Board Washington DC, USA, 1998.

[25] M. C. Dunne and R. B. Potts, “Algorithm for traffic control,” *Operations Research*, vol. 12, no. 6, pp. 870–881, 1964.

[26] M. Dunne and R. Potts, “Analysis of a computer control of an isolated intersection. in vehicular traffic science,” in *Proceedings of the Third International Symposium on the Theory of Traffic FlowOperations Research Society of America*, 1967.

[27] J. Van Zijverden and H. Kwakernaak, “A new approach to traffic-actuated computer control of intersections,” *Proc. Fourth Internal. Sym-posium on the Theory of Traffic Flow, Karlsruhe*, 1968.

[28] N. H. Gartner, “Demand-responsive decentralized urban traffic control. part i: Single-intersection policies,” tech. rep., 1982.

[29] N. H. Gartner, “Development and implementation of an adaptive control strategy in a traffic signal network: the virtual-fixed-cycle approach,” in *Transportation and Traffic Theory in the 21st Century: Proceedings of the 15th International Symposium on Transportation and Traffic Theory, Adelaide, Australia, 16-18 July 2002*, pp. 137–155, Emerald Group Publishing Limited, 2002.

[30] S. Sen and K. L. Head, “Controlled optimization of phases at an intersection,” *Transportation science*, vol. 31, no. 1, pp. 5–17, 1997.

[31] R. Grafton and G. F. Newell, “Optimal policies for the control of an undersaturated intersection. in vehicular traffic science,” in *Proceedings of the Third International Symposium on the the Theory of Traffic FlowOperations Research Society of America*, 1967.

[32] Q. Huang and R. Miller, “Reliable wireless traffic signal protocols for smart intersections,” in *ITS America Annual Meeting*, 2004.

[33] A. Olia, H. Abdelgawad, B. Abdulhai, and S. N. Razavi, “Assessing the potential impacts of connected vehicles: mobility, environmental, and safety perspectives,” *Journal of Intelligent Transportation Systems*, vol. 20, no. 3, pp. 229–243, 2016.

[34] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, “A real-time adaptive signal control in a connected vehicle environment,” *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 460–473, 2015.

[35] J. Wu, A. Abbas-Turki, A. Correia, and A. El Moudni, “Discrete intersection signal control,” in *2007 IEEE International Conference on Service Operations and Logistics, and Informatics*, pp. 1–6, IEEE, 2007.

[36] J. Cheng, W. Wu, J. Cao, and K. Li, “Fuzzy group-based intersection control via vehicular networks for smart transportations,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 751–758, 2017.

[37] M. Ahmane, A. Abbas-Turki, F. Perronnet, J. Wu, A. El Moudni, J. Buisson, and R. Zeo, “Modeling and controlling an isolated urban intersection based on cooperative vehicles,” *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 44–62, 2013.

[38] K. Tiaprasert, Y. Zhang, X. B. Wang, and X. Zeng, “Queue length estimation using connected vehicle technology for adaptive signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2129–2140, 2015.

[39] S. I. Guler, M. Menendez, and L. Meier, “Using connected vehicle technology to improve the efficiency of intersections,” *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121–131, 2014.

[40] R. W. Denney Jr, E. Curtis, and P. Olson, “The national traffic signal report card,” *ITE Journal*, vol. 82, no. 6, pp. 22–26, 2012.

[41] W. S. Homburger, L. E. Keefer, and W. R. Mcgrath, *Transportation and traffic engineering handbook*. 1982.

[42] A. Sharma, D. M. Bullock, and J. A. Bonneson, "Input–output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections," *Transportation Research Record*, vol. 2035, no. 1, pp. 69–80, 2007.

[43] H. X. Liu, X. Wu, W. Ma, and H. Hu, "Real-time queue length estimation for congested signalized intersections," *Transportation research part C: emerging technologies*, vol. 17, no. 4, pp. 412–427, 2009.

[44] J. Argote, E. Christofa, Y. Xuan, and A. Skabardonis, "Estimation of measures of effectiveness based on connected vehicle data," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1767–1772, IEEE, 2011.

[45] G. Comert and M. Cetin, "Queue length estimation from probe vehicle location and the impacts of sample size," *European Journal of Operational Research*, vol. 197, no. 1, pp. 196–202, 2009.

[46] G. Comert and M. Cetin, "Analytical evaluation of the error in queue length estimation at traffic signals from probe vehicle data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 563–573, 2011.

[47] J.-Q. Li, K. Zhou, S. E. Shladover, and A. Skabardonis, "Estimating queue length under connected vehicle technology: Using probe vehicle, loop detector, and fused data," *Transportation Research Record*, vol. 2356, no. 1, pp. 17–22, 2013.

[48] Y. Cheng, X. Qin, J. Jin, and B. Ran, "An exploratory shockwave approach for signalized intersection performance measurements using probe trajectories," in *89th Annual Meeting of Transportation Research Board*, 2010.

[49] B. E. Badillo, H. Rakha, T. W. Rioux, and M. Abrams, "Queue length estimation using conventional vehicle detector and probe vehicle data," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 1674–1681, IEEE, 2012.

[50] C. J. Messer and D. B. Fambro, *Effects of signal phasing and length of left-turn bay on capacity*. No. 644, 1977.

[51] K. Yin, Y. Zhang, and B. X. Wang, "Analytical models for protected plus permitted left-turn capacity at signalized intersection with heavy traffic," *Transportation Research Record*, vol. 2192, no. 1, pp. 177–184, 2010.

[52] Y. Zhang and J. Tong, "Modeling left-turn blockage and capacity at signalized intersection with short left-turn bay," *Transportation Research Record*, vol. 2071, no. 1, pp. 71–76, 2008.

[53] P. Wang, S. L. Jones, S. Gurupackiam, and L. Wang, "Novel cell transmission model–based simulations of left-turn blockages inside intersections," *Transportation Research Record*, vol. 2390, no. 1, pp. 60–67, 2013.

[54] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE internet of things journal*, vol. 1, no. 4, pp. 289–299, 2014.

[55] Y. Feng, M. Zamanipour, K. L. Head, and S. Khoshmagham, "Connected vehicle–based adaptive signal control and applications," *Transportation Research Record*, vol. 2558, no. 1, pp. 11–19, 2016.

[56] I. H. Zohdy and H. A. Rakha, "Intersection management via vehicle connectivity: The intersection cooperative adaptive cruise control system concept," *Journal of Intelligent Transportation Systems*, vol. 20, no. 1, pp. 17–32, 2016.

[57] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.

[58] J. T. Morgan and J. D. Little, “Synchronizing traffic signals for maximal bandwidth,” *Operations Research*, vol. 12, no. 6, pp. 896–912, 1964.

[59] J. D. Little, “The synchronization of traffic signals by mixed-integer linear programming,” *Operations Research*, vol. 14, no. 4, pp. 568–594, 1966.

[60] N. H. Gartner and C. Stamatiadis, “Arterial-based control of traffic flow in urban grid networks,” *Mathematical and computer modelling*, vol. 35, no. 5-6, pp. 657–671, 2002.

[61] P. Mireault, “A branch-and-bound algorithm for the traffic signal synchronization problem with variable speed,” *TRSTAN I, Montreal., Canada*, 1991.

[62] N. A. Chaudhary, A. Pinnoi, and C. J. Messer, *Proposed enhancements to MAXBAND 86 program*. No. 1324, 1991.

[63] R. S. Pillai, A. K. Rathi, and S. Cohen, “A restricted branch and bound approach for setting the left turn phase sequences in signalized networks,” tech. rep., Oak Ridge National Lab., TN (United States), 1994.

[64] J. D. Little, M. D. Kelson, and N. H. Gartner, “Maxband: A versatile program for setting signals on arteries and triangular networks,” 1981.

[65] E. Chang and C. J. Messer, “Arterial signal timing optimization using passer ii-90-program user’s manual,” 1991.

[66] S. Venglar, P. Koonce, and T. Urbanik II, “Passer tm iii-98 application and userâĂŹs guide,” 1998.

[67] N. A. Chaudhary and C. J. Messer, “Passer iv-94, version 1.0, user/reference manual,” 1993.

[68] N. A. Chaudhary, V. Kovvali, C. Chu, J. Kim, and S. Alam, “Software for timing signalized arterials,” tech. rep., 2002.

[69] N. H. Gartner, S. F. Assmann, F. Lasaga, and D. L. Hous, “Multiband–a variable-bandwidth arterial progression scheme,” *Transportation Research Record*, no. 1287, 1990.

[70] C. J. Messer, R. H. Whitson, C. L. Dudek, and E. J. Romano, *A variable-sequence multiphase progression optimization program.* No. 445, 1973.

[71] P. P. Jovanis and J. A. Gregor, “Coordination of actuated arterial traffic signal systems,” *Journal of Transportation Engineering*, vol. 112, no. 4, pp. 416–432, 1986.

[72] N. C. Ficklin, “For and against semi-actuated traffic signal control,” *Traffic Engineering, Inst Traffic Engr*, vol. 43, no. 6, 1973.

[73] P. Varaiya, “Max pressure control of a network of signalized intersections,” *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[74] P. Mirchandani and L. Head, “A real-time traffic signal control system: architecture, algorithms, and analysis,” *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.

[75] P. DELL and B. MIRCHANDANI, “Realband: An approach for real-time coordination of traffic flows on networks,” 1995.

[76] P. DellâĂŹOlmo and P. B. Mirchandani, “A model for real-time traffic coordination using simulation based optimization,” in *Advanced Methods in Transportation Analysis*, pp. 525–546, Springer, 1996.

[77] X. B. Wang, X. Cao, and C. Wang, “Dynamic optimal real-time algorithm for signals (doras): Case of isolated roadway intersections,” *Transportation research part B: methodological*, vol. 106, pp. 433–446, 2017.

[78] X. Cao, J. Jiao, Y. Zhang, and X. Wang, “Left-turn spillback probability estimation in a connected vehicle environment,” *Transportation Research Record*, p. 0361198119837178, 2018.

APPENDIX A

SELECTED CODING FOR MADM

```
#import "C:\Program Files\PTV Vision\PTV Vissim 10\Exe\VISSIM100.
   exe" rename_namespace ("VISSIMLIB")
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
#include <vector>
#include <math.h>
using namespace std;
using std::cout;
using std::endl;
int main(int argc, char* argv[])
{
  std::ofstream myfile;
  myfile.open("MADM-High.csv");
  for (int i = 10; i <= 50; i = i + 10)
  {
    CoInitialize(NULL);
    {
      VISSIMLIB::IVissimPtr Vissim;
      Vissim.CreateInstance("Vissim.Vissim");
```

```
bstr_t Path_of_network = "C:\\Users\\caoxiaowei.AUTH\\
   Documents\\Simulation Network\\";
bstr_t Filename = Path_of_network + "network.inpx";
//cout << Filename << endl;
//cout << Filename << endl;
cout << "Current Seed is: " << i << endl;
bool flag_read_additionally = false;
Vissim->LoadNet(Filename, flag_read_additionally);
Filename = Path_of_network + "network.layx";
Vissim->LoadLayout(Filename);



double End_of_simulation = 7200;
Vissim->Simulation->PutAttValue("SimPeriod",
   End_of_simulation);

int Random_Seed = i;
Vissim->Simulation->PutAttValue("RandSeed", Random_Seed);

double Sim_break_at = 0;

double current_phase_green1 = 0, current_phase_green2 = 0,
   current_phase_green3 = 0, current_phase_green4 = 0,
   current_phase_green5 = 0, current_phase_green6 = 0,
   current_phase_green7 = 0, current_phase_green8 = 0,
   current_phase_green9 = 0;
```

```
double minGreen = 5;

double maxGreen = 20;

bstr_t red = "RED";
bstr_t green = "GREEN";
bstr_t amber = "AMBER";

int phase1 = 0, phase2 = 0, phase3 = 0, phase4 = 0, phase5
   = 0, phase6 = 0, phase7 = 0, phase8 = 0, phase9 = 0;
int Amber1 = 0, Amber2 = 0, Amber3 = 0, Amber4 = 0, Amber5
   = 0, Amber6 = 0, Amber7 = 0, Amber8 = 0, Amber9 = 0;

double phase_length11 = 0, phase_length12 = 0,
   phase_length13 = 0, phase_length14 = 0;
double phase_length21 = 0, phase_length22 = 0,
   phase_length23 = 0, phase_length24 = 0;
double phase_length31 = 0, phase_length32 = 0,
   phase_length33 = 0, phase_length34 = 0;
double phase_length41 = 0, phase_length42 = 0,
   phase_length43 = 0, phase_length44 = 0;
double phase_length51 = 0, phase_length52 = 0,
   phase_length53 = 0, phase_length54 = 0;
double phase_length61 = 0, phase_length62 = 0,
   phase_length63 = 0, phase_length64 = 0;
double phase_length71 = 0, phase_length72 = 0,
   phase_length73 = 0, phase_length74 = 0;
```

```
double phase_length81 = 0, phase_length82 = 0,
   phase_length83 = 0, phase_length84 = 0;
double phase_length91 = 0, phase_length92 = 0,
   phase_length93 = 0, phase_length94 = 0;

int total_phase_number11 = 0, total_phase_number12 = 0,
   total_phase_number13 = 0, total_phase_number14 = 0;
int total_phase_number21 = 0, total_phase_number22 = 0,
   total_phase_number23 = 0, total_phase_number24 = 0;
int total_phase_number31 = 0, total_phase_number32 = 0,
   total_phase_number33 = 0, total_phase_number34 = 0;
int total_phase_number41 = 0, total_phase_number42 = 0,
   total_phase_number43 = 0, total_phase_number44 = 0;
int total_phase_number51 = 0, total_phase_number52 = 0,
   total_phase_number53 = 0, total_phase_number54 = 0;
int total_phase_number61 = 0, total_phase_number62 = 0,
   total_phase_number63 = 0, total_phase_number64 = 0;
int total_phase_number71 = 0, total_phase_number72 = 0,
   total_phase_number73 = 0, total_phase_number74 = 0;
int total_phase_number81 = 0, total_phase_number82 = 0,
   total_phase_number83 = 0, total_phase_number84 = 0;
int total_phase_number91 = 0, total_phase_number92 = 0,
   total_phase_number93 = 0, total_phase_number94 = 0;
double average_phase11 = 0, average_phase12 = 0,
   average_phase13 = 0, average_phase14 = 0;
double average_phase21 = 0, average_phase22 = 0,
   average_phase23 = 0, average_phase24 = 0;
```

```
double average_phase31 = 0, average_phase32 = 0,
   average_phase33 = 0, average_phase34 = 0;
double average_phase41 = 0, average_phase42 = 0,
   average_phase43 = 0, average_phase44 = 0;
double average_phase51 = 0, average_phase52 = 0,
   average_phase53 = 0, average_phase54 = 0;
double average_phase61 = 0, average_phase62 = 0,
   average_phase63 = 0, average_phase64 = 0;
double average_phase71 = 0, average_phase72 = 0,
   average_phase73 = 0, average_phase74 = 0;
double average_phase81 = 0, average_phase82 = 0,
   average_phase83 = 0, average_phase84 = 0;
double average_phase91 = 0, average_phase92 = 0,
   average_phase93 = 0, average_phase94 = 0;

double average_cycle1 = 0, average_cycle2 = 0,
   average_cycle3 = 0, average_cycle4 = 0, average_cycle5 =
    0, average_cycle6 = 0, average_cycle7 = 0,
   average_cycle8 = 0, average_cycle9 = 0;
double e10 = 0, e20 = 0, e30 = 0, e40 = 0, e50 = 0, e60 =
   0, e70 = 0, e80 = 0, e90 = 0;
double e11 = 0, e21 = 0, e31 = 0, e41 = 0, e51 = 0, e61 =
   0, e71 = 0, e81 = 0, e91 = 0;
double w10 = 0, w20 = 0, w30 = 0, w40 = 0, w50 = 0, w60 =
   0, w70 = 0, w80 = 0, w90 = 0;
double w11 = 0, w21 = 0, w31 = 0, w41 = 0, w51 = 0, w61 =
   0, w71 = 0, w81 = 0, w91 = 0;
```

```
VARIANT VehicleVar;
VISSIMLIB::IVehiclePtr Vehicle;
int veh_number;
int veh_type;
double veh_speed;
double veh_position;
bstr_t veh_linklane;

while (Sim_break_at < End_of_simulation - 1)
{
  Sim_break_at = Sim_break_at + 1;

  Vissim->Simulation->PutAttValue("SimBreakAt",
     Sim_break_at);

  Vissim->Simulation->RunContinuous();


  //cout << "SimBreakAt: " << Sim_break_at << "; Current
     Green; " << current_phase_green1 << endl;
  vector<double> queue;
  for (int i = 1; i <= 72; i++)
    queue.push_back(Vissim->Net->QueueCounters->
       GetItemByKey(i)->GetAttValue("QLen(Current, Last)"))
       ;
```

```
//Intersection 1
//check the current phase
int SH_number = 1; // SH = SignalHead
bstr_t EW_TH = Vissim->Net->SignalHeads->GetItemByKey(
    SH_number)->GetAttValue("SigState");
SH_number = 5; // SH = SignalHead
bstr_t EW_LT = Vissim->Net->SignalHeads->GetItemByKey(
    SH_number)->GetAttValue("SigState");
SH_number = 7; // SH = SignalHead
bstr_t NS_TH = Vissim->Net->SignalHeads->GetItemByKey(
    SH_number)->GetAttValue("SigState");
SH_number = 12; // SH = SignalHead
bstr_t NS_LT = Vissim->Net->SignalHeads->GetItemByKey(
    SH_number)->GetAttValue("SigState");




if (EW_TH == green  && EW_LT == red && NS_TH == red &&
    NS_LT == red)
{
  phase1 = 1;
  w10 = queue[8] * 2 + queue[12] * 2 - (queue[16] * 2 +
      queue[17])*0.8 - (queue[38] * 2 + queue[39])*0.2 - (
    queue[5] * 2 + queue[6])*0.8 - (queue[26] * 2 +
    queue[27])*0.2;
w10 = w10 / 20 * 2;
w11 = queue[9] + queue[13] - (queue[26] * 2 + queue
    [27]) - (queue[38] * 2 + queue[39]);
w11 = w11 / 20;


//calculate the current intersection efficiency.
int QC_number = 9;

double Q1 = Vissim->Net->QueueCounters->GetItemByKey(
    QC_number)->GetAttValue("QLen(Current, Last)");


QC_number = 13;
double Q2 = Vissim->Net->QueueCounters->GetItemByKey(
    QC_number)->GetAttValue("QLen(Current, Last)");

e10 = 0;
e11 = 0;


double Q11 = 0, Q12 = 0, Q13 = 0, Q14 = 0, Q21 = 0, Q22
     = 0, Q23 = 0, Q24 = 0;
```

```
if (Q1 > 0)
{
  e10 = e10 + 1;
}
else
{
  VISSIMLIB::IIteratorPtr Vehicles_Iterator = Vissim->
     Net->Vehicles->GetIterator();
  while (Vehicles_Iterator->GetValid())
  {
    Vehicle = Vehicles_Iterator->GetItem();
    veh_number = Vehicle->GetAttValue("No");
    veh_type = Vehicle->GetAttValue("VehType");
    veh_speed = Vehicle->GetAttValue("Speed");
    veh_position = Vehicle->GetAttValue("Pos");
    veh_linklane = Vehicle->GetAttValue("Lane");
    bstr_t link1 = "17-1";
    bstr_t link2 = "17-2";
    bstr_t link3 = "19-1";
    bstr_t link4 = "19-2";
    //1s
    if (veh_linklane == link1 || veh_linklane == link2)
    {

      //Q1
      if (753 - veh_position > 0 && 753 - veh_position
         <= 40.18)
```

```
{
  Q11 = Q11 + 1;
  Q12 = Q12 + 1;
  Q13 = Q13 + 1;
  Q14 = Q14 + 1;
}

if (753 - veh_position > 40.18 && 753 -
   veh_position <= 68.88)

{
  Q12 = Q12 + 1;
  Q13 = Q13 + 1;
  Q14 = Q14 + 1;
}

if (753 - veh_position > 68.88 && 753 -
   veh_position <= 86.1)

{
  Q13 = Q13 + 1;
  Q14 = Q14 + 1;
}

if (753 - veh_position > 86.1 && 753 -
   veh_position <= 91.84)
```

```
      {
        Q14 = Q14 + 1;
      }
    }
    Vehicles_Iterator->Next();
  }
  double a = Q11 * 1;
  double b = Q12 * 0.5;
  double c = Q13 * 0.33;
  double d = Q14 * 0.25;
  e10 = max(a, e10);
  e10 = max(b, e10);
  e10 = max(c, e10);
  e10 = max(d, e10);

}

if (Q2 > 0)
{
  e10 = e10 + 1;
}
else
{
  VISSIMLIB::IIteratorPtr Vehicles_Iterator = Vissim->
     Net->Vehicles->GetIterator();
  while (Vehicles_Iterator->GetValid())
  {
```

```
Vehicle = Vehicles_Iterator->GetItem();
veh_number = Vehicle->GetAttValue("No");
veh_type = Vehicle->GetAttValue("VehType");
veh_speed = Vehicle->GetAttValue("Speed");
veh_position = Vehicle->GetAttValue("Pos");
veh_linklane = Vehicle->GetAttValue("Lane");
bstr_t link1 = "17-1";
bstr_t link2 = "17-2";
bstr_t link3 = "19-1";
bstr_t link4 = "19-2";
if (veh_linklane == link3 || veh_linklane == link4)
{
  if (753 - veh_position > 0 && 753 - veh_position
      <= 40.18)
  {
    Q21 = Q21 + 1;
    Q22 = Q22 + 1;
    Q23 = Q23 + 1;
    Q24 = Q24 + 1;
  }

  if (753 - veh_position > 40.18 && 753 -
      veh_position <= 68.88)

  {
    Q22 = Q22 + 1;
    Q23 = Q23 + 1;
```

```
      Q24 = Q24 + 1;
    }

    if (753 - veh_position > 68.88 && 753 -
       veh_position <= 86.1)

    {
      Q23 = Q23 + 1;
      Q24 = Q24 + 1;
    }

    if (753 - veh_position > 86.1 && 753 -
       veh_position <= 91.84)

    {
      Q24 = Q24 + 1;
    }

  }
  Vehicles_Iterator->Next();
}

double a = Q21 * 1;
double b = Q22 * 0.5;
double c = Q23 * 0.33;
double d = Q24 * 0.25;
double e = 0;
```

```
  e = max(a, e);
  e = max(b, e);
  e = max(c, e);
  e = max(d, e);
  e10 = e10 + e;

}

//cout << "Intersection 1 EW-TH Current Efficiency:" <<
    ": " << e0 << endl;

//calculate the switch-to intersection efficiency.

double Q3 = Vissim->Net->QueueCounters->GetItemByKey
   (10)->GetAttValue("QLen(Current, Last)");
Q3 = Q3 / 20;
double Q4 = Vissim->Net->QueueCounters->GetItemByKey
   (14)->GetAttValue("QLen(Current, Last)");
Q4 = Q4 / 20;
double Q5 = Vissim->Net->QueueCounters->GetItemByKey
   (11)->GetAttValue("QLen(Current, Last)");
Q5 = Q5 * 2 / 20;
double Q6 = Vissim->Net->QueueCounters->GetItemByKey
   (15)->GetAttValue("QLen(Current, Last)");
Q6 = Q6 * 2 / 20;
double Q7 = Vissim->Net->QueueCounters->GetItemByKey
   (12)->GetAttValue("QLen(Current, Last)");
  Q7 = Q7 / 20;
  double Q8 = Vissim->Net->QueueCounters->GetItemByKey
      (16)->GetAttValue("QLen(Current, Last)");
  Q8 = Q8 / 20;
  average_phase12 = phase_length12 / (
      total_phase_number12 + 0.0001) + 0.0001;
  average_phase13 = phase_length13 / (
      total_phase_number13 + 0.0001) + 0.0001;
  average_phase14 = phase_length14 / (
      total_phase_number14 + 0.0001) + 0.0001;

  double e11 = ((Q3 + Q4)*(1 + (3 + average_phase12) / (
      average_phase13 + average_phase14 +
      current_phase_green1 + 9)) + (Q5 + Q6) * (1 + (
      average_phase12 + average_phase13 + 6) / (
      average_phase14 + current_phase_green1 +
      average_phase12 + 9)) + (Q7 + Q8)*(1 + (
      average_phase12 + average_phase13 + average_phase14
      + 9) / (current_phase_green1 + average_phase12 +
      average_phase13 + 9))) / (average_phase12 +
      average_phase13 + average_phase14 + 9);
}
```