Title
Data-Driven Approaches for Robust Signal Plans in Urban Transportation Networks

Author
Amini, Zahra

Publication Date
2018

Peer reviewed|Thesis/dissertation

Data-Driven Approaches for Robust Signal Plans in Urban Transportation Networks

By

Zahra Amini

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering- Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in Charge:

Professor Alexander Skabardonis, Chair
Professor Michael Cassidy
Professor Pravin Varaiya

Spring 2018

Abstract

Data-Driven Approaches for Robust Signal Plans in Urban Transportation Networks

by

Zahra Amini

Doctor of Philosophy in Civil and Environmental Engineering

University of California, Berkeley

Professor Alexander Skabardonis, Chair

In urban transportation networks with signalized intersections a robust pre-timed signal plan is a practical alternative to adaptive control strategies, since it has less complexity and an easier implementation process. Recent advances in technology are making data collection at traffic signals economical and data-driven approaches are likely to benefit from the large traffic data. Data-driven approaches are necessary for designing robust timing plans that can satisfy rapid traffic volume fluctuation and demand growth. This dissertation introduces four data-driven approaches for studying and improving traffic conditions at signalized intersections.

Firstly, I discuss the development and testing of two algorithms for checking the quality of traffic data and for estimating performance measures at intersections. The first of these algorithms estimates the systematic error of the detector data at signalized intersections by using flow conservation. According to the ground truth data from a real-world network, the algorithm can reduce the error in the data up to 25%. The second algorithm helps in estimating intersection performance measures in real-time by measuring the number of the vehicles in each approach using high resolution(HR) data.

An offset optimization algorithm was developed to adjust signal offsets so as to improve the delay in the system. The performance of three real-world networks using the offsets obtained by the algorithm and those obtained from the widely used Synchro optimization tool, are compared using the VISSIM microscopic simulation model. Simulation results show up to a 30% reduction in the average number of stops and total delay that vehicles experience along the major routes when using the proposed algorithms' optimized offsets. The fourth algorithm estimates the appropriate switching time between designed timing plans during the day based on the traffic profile of the intersection by using the K-means clustering method.

In conclusion, these four algorithms extract useful information from HR data about traffic at signalized intersections. Moreover, the algorithms assist in designing robust timing plans for satisfying demand fluctuations at signalized intersection. Lastly, simulation results from real-world networks illustrate the significant improvements that the proposed data-driven approaches can make in the control systems at urban transportation networks.

To my parents: Nahid & Mohammad

Table of Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Tables

Chapter 2

Chapter 3

Chapter 4

Acknowledgments

First and foremost, I would like to thank my PhD advisor, Professor Alexander Skabardonis. I learned a lot from his knowledge, experience, and perspective during my PhD and I would always be grateful for his support and great advices. Also, I would like to thank Professor Michael Cassidy for introducing me to Transportation Engineering discipline when I was an undergraduate student in UC Berkeley, and his kind and effective guidance as a member of my PhD dissertation committee.

During my PhD, I was honored to learn from Professor Pravin Varaiya, one of the most distinguished Professors in the world. His vision, wisdom, and grandeur dramatically influenced my academic and personal life and helped me to improve my motivation and self-esteem.

Over the past years, I had the privilege of collaborating with Professor Ramtin Pedarsani, Professor Weihua Gu, and Dr. Christopher Flores. I was very fortunate to meet and work with many amazing and knowledgeable people like Dr. Alexander A. Kurzhanskiy and Dr. Michael Mauch. Also, I would like to thank Chris Chin and Jasper Lee, who I mentored during my PhD. Chris and Jasper are very smart students and I can see a bright future ahead of them. During the last year of my PhD I had the opportunity of working with Professor Samuel Coogan, who I learned a lot from and became one of my academic role models. Professor Coogan is a great example of a patient, motivated, and supportive advisor.

Although PhD life has ups and downs and sometimes can get tough, I still feel I enjoyed every second of my Berkeley life because of the wonderful and caring friends like Nazanin Pajoom, Sajjad Moazeni, Faraz Tavakoli, and Sahand Golnarian. I also would like to thank Dorsa Sadigh, Delagah Dadbeh, and Negar Zahedimehr, for always being there for me and never giving up on me. Their friendship helped me to step beyond my mental boundaries and see the beauty of life from many other possible aspects. Last but not least, I would like to thank Sina Akhbari for bringing joy and glory to my life and supporting me non-stop.

All of the achievements that I had and will have in my life is because of my great family who supported me and always had my back. My special thanks goes to my sister, Laleh Amini, for being my best friend and having the kindest heart in the world, Rouzbeh Amini, the funniest and the most supportive brother, Hamed Banimostafai, for being a warm and good-natured friend, and Shirin Dilmaghani, for her great, kind, and generous heart. My final thanks goes to my parents: Nahid Yousefian, my childhood role model and the most devoted and compassionate mother, Mohammad Amini, for his unconditional love and support and for being the greatest father.

Chapter 1. Introduction

1.1 Motivation

Considerable attention has been given to new approaches for improving the urban transportation system because of limited funding and environmental concerns over constructing new highways. One promising approach is the implementation of advanced signal control strategies along arterials. This would reduce delays and stops at traffic signals and cut fuel consumption. In many instances, these arterial facilities also serve as reliever routes for congested freeways especially under incident conditions. Thus, arterial efficient operation could help the traffic performance along the entire travel corridor.

An assessment of the state of practice in signal management on urban networks indicates that on average intersections are poorly managed, according to the National Transportation Operations Coalition (NTOC) 2012 signal systems report card [1]. The main reason for the poor performance is the lack of systematic data collection to estimate performance measures and the unavailability of simple control strategies that are responsive to real-time changes in traffic patterns. Adaptive traffic control systems provide greater flexibility in real-world cases with varying demand by updating the information based on real-time data from detectors [2]. Several adaptive control strategies have been developed, but their implementation is limited because of their complexity and their extensive and costly data requirements [3]. A robust pre-timed signal plan is one alternative to adaptive control system [2].

High Resolution (HR) data at signalized intersections refers to continuous and simultaneous acquisition of detector data and signal status. Moreover, recent advances in technology are making HR data collection at traffic signals economical [4,5]. HR data presents a game-changing opportunity in traffic management. These data enable us to design a robust pre-timed signal plan that responds to the traffic demand fluctuation without the complexity and expenses of adaptive control systems [6]. Furthermore, these data can be complemented with measurements of aggregate performance, such as travel times and origin-destination patterns, derived from vehicle-based records of GPS traces or Bluetooth and WiFi addresses [7].

1.1 Objective

The goal of the research is to develop data-driven approaches for designing robust pre-timed signal plans, which increase throughput in urban transportation network and reduce vehicle delay and GHG emissions as well. The Dissertation focuses is as follows:

- **Data Quality:** I will present a data filtering algorithm that reduces the systematic error in detector data.

- **Performance Measures:** Estimating different performance measures for a signalized intersection helps to evaluate the performance of the control strategy and makes it possible to improve the system. I will introduce an algorithm that estimates the number of vehicles in each approach to an intersection in real-time by using HR data.

- **Control Parameters:** In signalized networks with coordinated signals, optimum signal offsets minimize the vehicle delay and number of stops in the network. In this dissertation, I develop an offset optimization algorithm and compare it with current popular approaches.

- **Control Plan Switching Time:** After designing robust timing plans, it is important to estimate appropriate activation intervals for each control plan during the day. I will show how to use traffic profile data from an intersection to estimate the most accurate and suitable switching time.

1.3 Contribution

This dissertation introduces new ways to benefit from HR traffic data at signalized intersections and illustrates the impact of the data on the management of traffic in an arterial. Contribution of my dissertation includes analysis of HR data, estimating performance measures at traffic signals, optimizing the offset of signals in urban networks, and developing simple robust timing plan for addressing the variability in traffic demand. The methodology consists of empirical data collection, filtering, and analysis and analytical modeling. It builds on recent and ongoing works by [7], [8], [9], and [10] on algorithms for estimating performance measures based on HR and emerging data sources and on the development of control strategies for traffic signals.

1.4 Organization

The rest of this dissertation is organized as shown in Figure 1.1 flow-chart. Chapter 2 is about step 1 and step 2 of the flow chart. This chapter explains the HR data collection process, and then describes a data filtering algorithm along with some results from a case study. Chapter 3 contributes to the third step of the flow-chart, which is evaluating the performance of current control system in test network, by developing an algorithm for estimating vehicle accumulation in each approach at the intersection for any given time during the cycle. Step 4 of the flow chart is estimating new control parameters and Chapter 4 contributes to this step by estimating optimum signal offsets, which is one of the signal control parameters. The same chapter presents simulation results from implementing the offset optimization algorithm in a simulator, which covers step 6 of the flow chart as well. Chapter 5 studies the fifth step of the flow-chart, which is estimating the activation time for each timing plan during the day. Lastly, Chapter 6 summarizes all the previous chapters and introduces new directions for future studies.

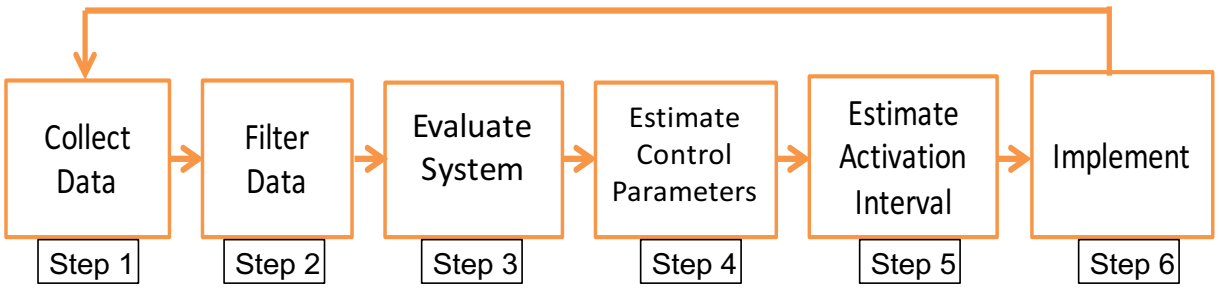


Figure 1. 1 Process of designing and applying robust timing plan in a signalized intersection

Chapter 2. HR Data Collection and Filtering

2.1 Introduction

As shown in Figure 1.1, data are the main requirement for a data-driven system. Therefore, I collaborated with Sensys Networks, Inc., and thus gained access to their database. They collect High Resolution(HR) data in real time from a large number of intersections across the nation by using magnetic field sensors.

Magnetic sensors measure changes in the earth's magnetic field induced by a vehicle, then processes the measurements to detect the vehicle, and transfers the processed data via radio to the Access Point (AP) [11]. The AP combines data from the multiple detectors into information for the local controller or the Traffic Management Center (TMC). A HR system called SAMS (Safety and Mobility System) detects and records the lane, speed, signal phase and time when each vehicle enters and leaves the intersection based on the available information in the AP [8], [12].

Figure 2.1 shows the schematic of the detector layout for a signalized intersection in the city of Danville, CA, equipped with SAMS (Safety and Mobility System). In this intersection, each approach has a stop bar, along with advanced and departure detectors. There is a network monitoring card that provides the signal phase. The intersection is also equipped with a PTZ camera, which allows the verification of the accuracy of the data collection process.

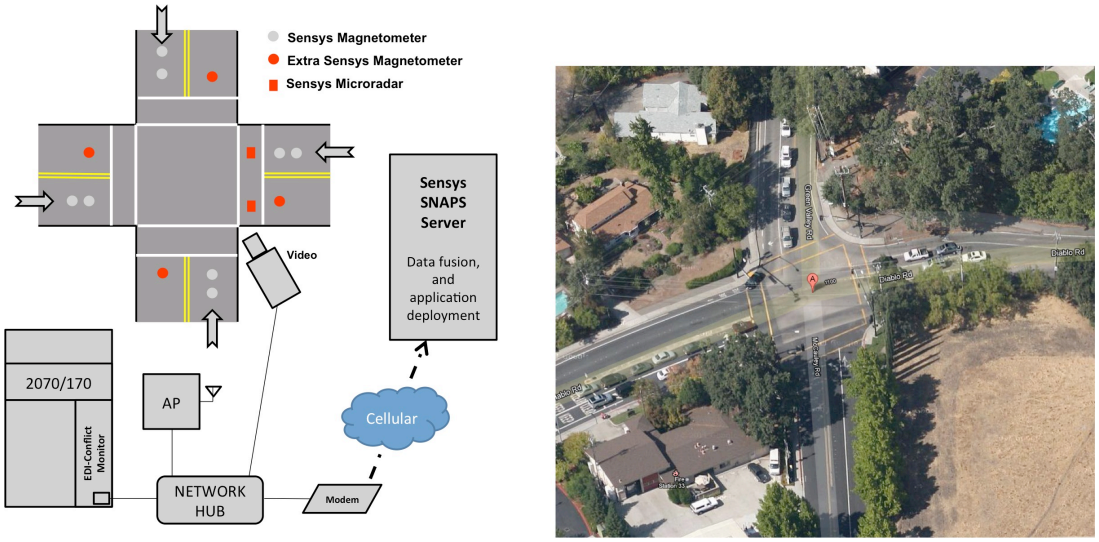


Figure 2. 1 Sample SAMS Instrumented Intersection, Danville CA

HR data can be beneficial only if they provide reliable and accurate information about the traffic conditions. According to studies, the failure of detectors and communication links causes error in detector data [13], [14]. These errors could be investigated by running a health inspection on the equipment and the system. Also, there are errors due to detector malfunction, which are challenging to investigate because the source of the error is not clear and the magnitude of the

error is usually small compared to the detected value. A study by [15] uses some microscopic approaches, which uses speed estimated by the dual loop detectors to investigate the error. Some studies use macroscopic-level approaches. For example [16] applies different threshold test on the data to improve the accuracy. PeMS (Caltrans Performance Measurement System) uses a data-filtering algorithm, which was developed by [17] to detect bad loop detectors from their output and imputes the missing data from neighboring good loops. Moreover, [18] uses Fourier transform-based techniques to detect the data errors dues to malfunctioning detectors. On a different approach, [19] and [20] use a neural network model and genetic algorithms to address detector issues. In addition, [21] studies forecasting techniques such as historical average, nearest-neighbor algorithm, and autoregressive integrated moving average models for data correction and substitution.

In this chapter I use flow conservation to estimate the systematic error of the detectors; i.e. at any time the overall number of the vehicles that enter the intersection is equal to the sum of the vehicles that exit the intersection. My approach has several advantages over the other existing methods. Most importantly, it can be used in real time and it is possible to include data-filtering algorithm in the data collection process. Moreover, it is easy to modify the algorithm and apply it on intersections with different detector systems and geometries. Finally, in addition to estimating the systematic error, it can substitute for missing value by using available data.

2.2 Algorithm Description

In an isolated intersection with 4 legs there are 4 inputs, 4 outputs, and the flows of 12 movements as shown in Figure 2.2. The same figure also shows the location of the detectors. According to Figure 2.2 advance detectors are located at the upstream of the intersection and they measure the input flow. The output flow is measured by the departure detectors, which are located in the departure lanes, downstream of the intersection. Stop-bar detectors, which are located at the stop line, count the number of vehicles in each movement (right, left, through). Thus, we will have 20 values in total and each value will have a different amount of error.

There are multiple sources of error in traffic data. Some errors are due to a vehicle's movement, such as a lane change and driving over the lane. Others are due to environmental conditions such as rain. Accidents can also cause error in traffic data and interrupt the data collection process. Some detectors miscount vehicles with unusual length or when the vehicles are traveling too close to each other. In addition, there are systematic error, which is constant and independent from outside environment. For example, a detector always over count number of the vehicles by 10%, that would be the systematic error of the detector. Using different detectors to count the same vehicle improves the accuracy of the detected value. Figure 2.2 shows an example where each vehicle has been counted 3 times and each measured value is equal to the true value plus some error. The data filtering algorithm goal is to minimize the summation of all the error terms.

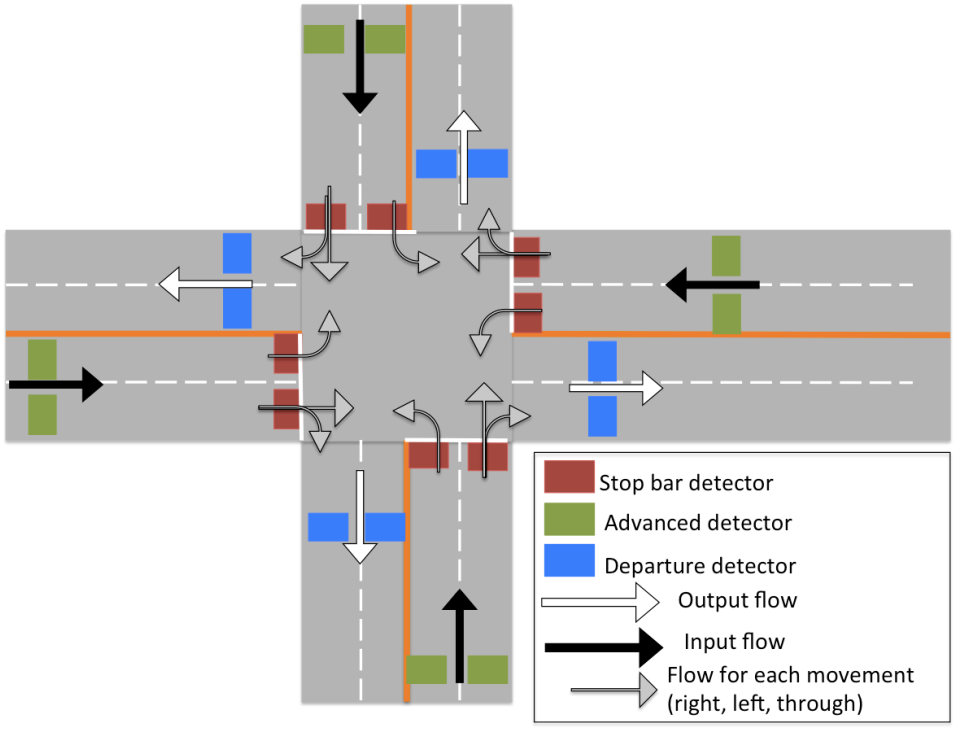


Figure 2. 2 A signalized intersection with 4 legs

As shown in Equation 2.1, my problem formulation assumes that detected flow value, $\hat{\boldsymbol{f}}_{\boldsymbol{i}}$, equals to the imputed flow value, $\boldsymbol{f}_{\boldsymbol{i}}$, plus the error term, $\boldsymbol{\varepsilon}_{\boldsymbol{i}}$. Therefore, a negative error term means the detected value is less than the actual value and the detector is undercounting. A positive error term implies that the detector is over counting.

$$\hat{f}_i = f_i + \varepsilon_i \quad for\ all\ i \tag{2.1}$$

The estimated value, f_i, is not necessarily equal to the actual value because ε_i is only the systematic error, which is one of the source of error in the data. However, my claim is that the imputed value would be closer to the actual value compared to the detected value. In the next section I test this approach on a real-world intersection, where the ground-truth vehicle count collected by observers is available.

The cost function, C, in Equation 2.2, is corresponding to the example intersection in Figure 2.2 includes 20 error terms, one for each detected movement. In the cost function, C is equal the summation of the ε_i^2 values. I use the squared of the error terms to include both positive and negative errors and make sure that they do not cancel each other. Also, the squared error is an appropriate fit for our cost function because it is differentiable. Moreover, each ε term has a corresponding ω_i value, which is the weight of each ε_i term and its value depends on the level of accuracy of $\hat{f}_i$. In a normal condition, ω_i would be same for all the variables, but it is possible that we have more confident in one or some detectors' performance than others. For example, it is reasonable to say that advanced detectors have a better accuracy in their counting process than stop-bar detectors, because of the lane change and stops that happen at the stop lines. Based on

available information about the detectors and previous studies we can assign an appropriate value for ω_i.

$$C = \sum_i \omega_i * \varepsilon_i^2 \quad (2.2)$$

Also, we have a set of constraints based on the flow conservation rule. The input flow in each leg must equal the sum of turn-movement flows in the same leg. Also, output flow in each leg must equal the sum of turn-movement flows that are feeding the output direction. Finally, all the detected and estimated flow values, $\hat{f}_i$and f_i, should be equal or greater than zero.

Ultimately, the detected flow values are given and I estimate the errors by minimizing C, which is a convex function, subject to linear constraints.

After formulating the cost function, assigning appropriate ω values, and inputting the detected flow values I run the optimization problem. This process can be done in real time as well. After minimizing C, I get the ε_i^2 for each $\hat{f}_i$ value, then I can estimate, f_i which is my imputed flow value. The time-interval for running the optimization needs to be long enough to minimize the impact of the incomplete trips, but at the same time it should be small enough so the traffic volume is uniform during each time step.

2.3 Algorithm Evaluation

In this section, I use the flow conservation to estimate each detector's systematic error at the intersection of Montrose Rd. and Tildenwood Dr., in Montgomery County. Figure 2.3 shows the schematic of the detector layout for the signalized intersection. The intersection has four approaches (legs) and there are three movements per approach. As shown in Figure 2.3, in each leg there are 3 different types of detectors and each one counts vehicles independently. Advance detectors (located approximately 200-300 ft upstream from the stop-line) record the input values, which is shown by $f_{i,n}$ where *n* is the leg number. The output value, $f_{o,n}$ is recorded by departure detectors. On the same figure we can see the flows from the turn movements denoted by $f_{(n,m)}$, where left, right, and through movements are respectively m=1,2, and 3. The intuitively the approach is to minimize the error corresponding to each variable. We have 4 inputs, 4 outputs, and 12 turn movement variables, which gives 20 variables in total. The measured value, $\hat{f}$, equals the estimated value, f, plus the error term ε (Equations 2.3-2.5).

$$\hat{f}_{(n,m)} = f_{(n,m)} + \varepsilon_{f_{(n,m)}} \quad for\ all\ (n,m), n = 1,2,3,4\ and\ m = 1,2,3 \quad (2.3)$$

$$\hat{f}_{i,n} = f_{i,n} + \varepsilon_{f_{i,n}} \quad for\ all\ n = 1,2,3,4 \quad (2.4)$$

$$\hat{f}_{o,n} = f_{o,n} + \varepsilon_{f_{o,n}} \quad for\ all\ n = 1,2,3,4 \quad (2.5)$$

The cost function in Equation 2.6 includes the square of all the error terms multiplied by the weights, α_n, β_n, and $\gamma_{(n,m)}$. Flow values should be positive as well (Equations 2.7-2.9). Also, I have a set of constraints based on the flow conservation rule, which input counts equal the turn movement counts of that approach and output counts equal sum of all the movements that enter that approach (Equations 2.10-2.14). For example, in Equation 2.11 output flow in leg 1 equals

the right movement flow from leg 3 plus the left movement flow from leg 4 and the through movement from leg 1.

$$\min_{f_{i,n},f_{o,n},f_{(l,m)}} \sum_{n=1}^{4} \alpha_n \varepsilon_{f_{i,n}}^2 + \sum_{n=1}^{4} \beta_n \varepsilon_{f_{o,n}}^2 + \sum_{n=1}^{4} \sum_{m=1}^{3} \gamma_{(n,m)} \varepsilon_{f_{(n,m)}}^2 \quad (2.6)$$

$$s.t. 0 \leq f_{n,m} \; for \; all \; (n,m) \quad (2.7)$$

$$0 \leq f_{i,n} \; for \; all \; n \quad (2.8)$$

$$0 \leq f_{o,n} \; for \; all \; n \quad (2.9)$$

$$f_{i,n} = \sum_{m=1}^{3} f_{(n,m)} \quad for \; n = 1,2,3,4 \quad (2.10)$$

$$f_{o,1} = f_{(2,3)} + f_{(3,2)} + f_{(4,1)} \quad (2.11)$$

$$f_{o,2} = f_{(1,3)} + f_{(4,2)} + f_{(3,1)} \quad (2.12)$$

$$f_{o,3} = f_{(4,3)} + f_{(2,2)} + f_{(1,1)} \quad (2.13)$$

$$f_{o,4} = f_{(3,3)} + f_{(1,2)} + f_{(2,1)} \quad (2.14)$$

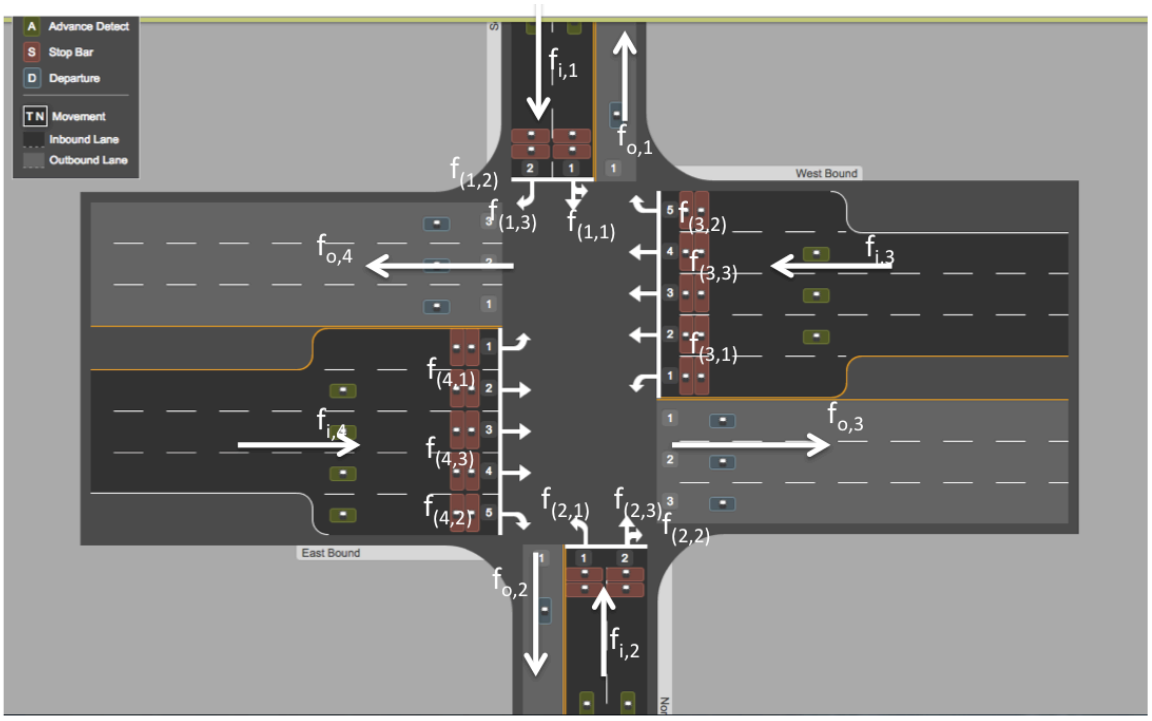


Figure 2. 3 Intersection of Montrose Rd. and Tildenwood Dr. layout

I used the data-filtering algorithm to estimate the systematic error in 15-minute vehicle counts detector data from 6AM until 7PM on June 14, 2016. I put all the weights equal to 1 and assumed all detectors are in equal condition and have the same level of accuracy. For the same intersection and the same day, I have the ground truth turn-movement flows collected by observers over a 13-hour interval (6AM to 7PM) for every 5 minutes. I aggregated the data ground truth data to get the vehicle count for 15-minute interval, and then compared it with the turn movement count from detectors and imputed values estimated by data-filtering algorithm. The first column of Table 2.1 shows different movements in each leg and it separate the major movements from minor ones. Major movements have very high traffic demand, while the traffic

volume in minor approaches are minimal. The second column shows the difference between the true and detected value divided by the true value in percentage and the last column is the difference between the true and imputed value in percentage. For Table 2.1 I estimated the error for each 15-minute time interval and then found the average error value over 13-hour.

Based on the Table 2.1 results in the major movements the data-filtering algorithm reduces the systematic error up to 25%, $(100*(4-3)/4=25\%)$, which is about 100s of vehicles per day or several vehicles per cycle. For the minor movements, there is no improvement, however the traffic volume is very small so the error is about few vehicles per day.

Movement		Error for true vs. detected (%)	Error for true vs. imputed (%)
Major Movements	Leg 3 through	4%	3%
	Leg 4 through	6%	5%
Minor Movements	Leg 3 Right	6%	9%
	Leg 3 Left	-43%	-35%
	Leg 4 Left	-41%	-11%
	Leg 4 Right	-10%	7%
	Leg 2 Left	6%	2%
	Leg 2 Right	5%	-20%
	Leg 1 Left	-8%	-37%
	Leg 1 Right	7%	-19%
	Leg 2 through	-75%	115%
	Leg 1 through	-50%	198%

Table 2. 1 Detected, imputed, and true values comparison

In addition, Figure 2.4 presents the cumulative vehicle counts over the 13-hour interval. In this figure, each curve presents the cumulative vehicle count for a different method. Blue curve is the estimated value from the algorithm, the red one is the detected values before we apply the algorithm and the green curve is the true value from the observers. It is clear that the algorithm is improving the detected value and making it closer to the true value.

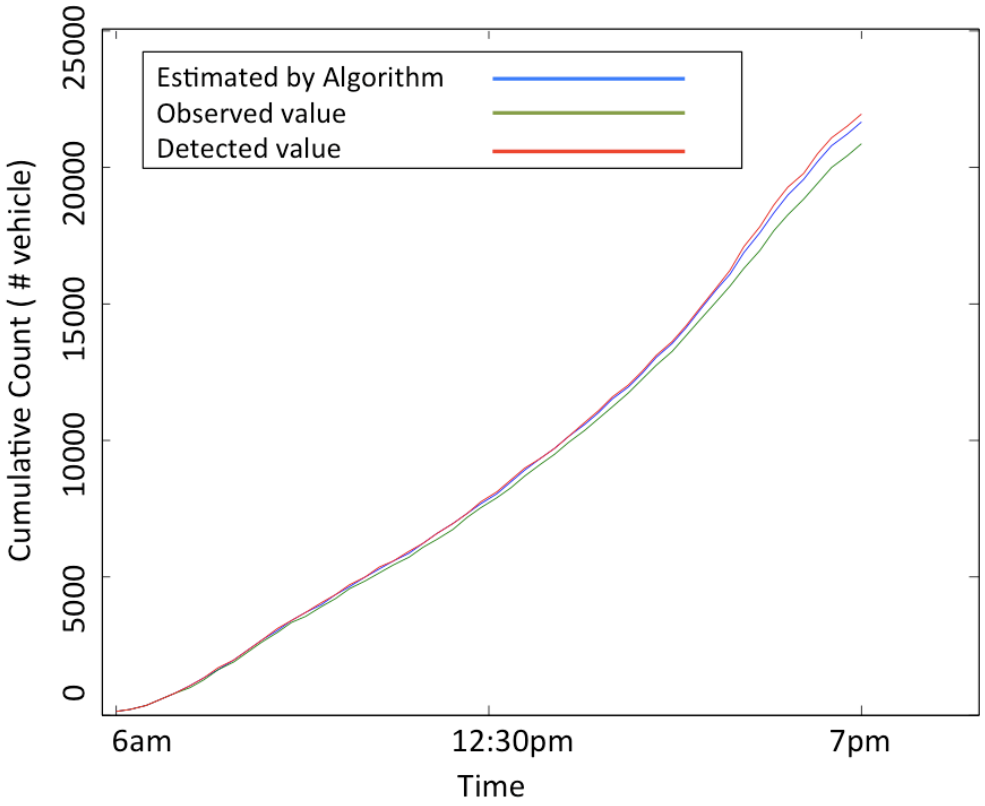


Figure 2. 4 Leg 3 through movement cumulative count curves

Another conclusion that we can make from Table 2.1 is that detected values from minor movements are closer to the true values compared to the imputed values. Therefore, I increase the weights of minor movements, which means detectors in minor movements are more accurate compared to the major movements. Figure 2.5 shows result after calibrating the algorithm. In Figure 2.5 the dark blue curve is the imputed value in a situation that all the weights are equal to 1 and the light blue curve is the imputed value in the calibrated scenario. When the algorithm is calibrated, the minor movement imputed value is improved as well.

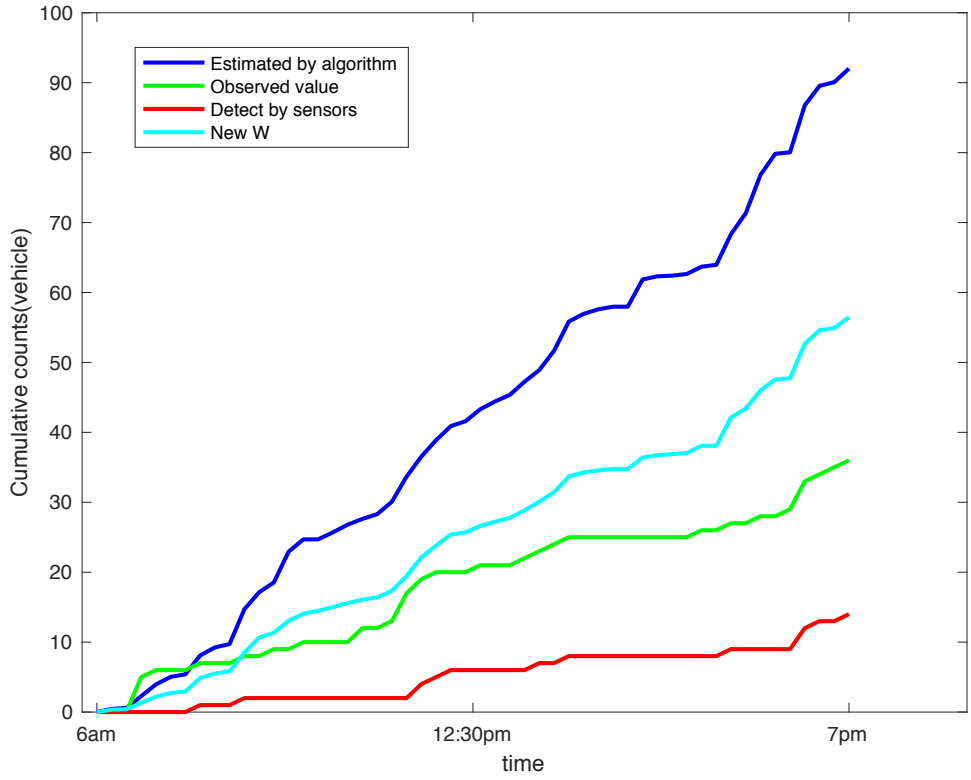


Figure 2. 5 Leg 2 through movement cumulative count curves

2.4 Discussion

HR data can provide useful information about the traffic condition. Having access to HR data enables us to improve the control system at urban networks by increasing the capacity at intersections. Therefore, the accuracy of these data sets is substantial. The data filtering algorithm is a fast and efficient way for estimating the systematic error of the detector data and amending the accuracy of the data set. The advantage of this algorithm over similar methods is that it works in real time and it can be a part of the data collection process. Comparing the imputed values to the ground-truth values show that the data filtering algorithm can reduce the error of the data point up to 25% in traffic approaches with high traffic volume. Data filtering algorithm could work for minor approaches as well.

Chapter 3. Signalized Intersection System Evaluation

There are different performance measures to evaluate the performance of the existing control system in the test network, including the delay, queue length, volume over capacity ratio, and many other parameters. In this Chapter I estimate different performance measures using HR data from a real-world signalized intersection and then I use stochastic gradient descent to estimate the vehicle accumulation in each approach at an intersection.

3.1 Performance Measures

Several measures of effectiveness (MOEs) have been proposed and used for evaluating traffic operations at intersections controlled by traffic signals. Table 3.1 displays the MOEs commonly proposed for the development and evaluation of traffic signal control algorithms. These MOEs vary depending on the operating environment (isolated intersections, arterials, grid networks), traffic conditions and patterns (e.g., congested-undersaturated vs. oversaturated conditions) and objectives/constraints (mobility, safety, environment). Field data collection most of these MOEs with conventional approaches is expensive and time consuming, which limits the collection of data to when major design or control improvements are implemented, and does not allow for systematic monitoring of traffic performance.

I. MOBILITY

Operating Environment	Operating Conditions	Performance Measure	Units	Comments
Intersection	Undersaturated	Average control delay	sec/veh	Difference free-flow travel time and actual travel time
		Max back of queue	#veh (ft/l)	Average and 95th % of the max extend of queue throughout the cycle
		Cycle failure	%	Proportion of cycles that queue failed to clear during green
		Green time utilization	%	Proportion of green utilized by traffic demand served by the phase
	Oversaturated	Throughput	#	# vehicles served at the intersection per time interval
Arterial/ Grid Network	Undersaturated	Average travel time	(min)	Average travel time for movements served by coordinated signal phases
		Average travel speed	(mph)	Average travel speed for movements served by coordinated signal phases
		Travel time variability	(min)	st deviation, 80th or 95th percentile of travel times served by coordinated phases
		# of stops/stop rate	#/(%)	Average # of stops (fraction of veh stopped) for movements served by coordinated phases
		Total delay	veh-hr	Delay of all vehicles served in the system
		% vehicles in the green	%	Proportion of platoon arriving during the green time per signal cycle
		Bandwidth efficiency	%	Proportion of the green through bandwidth to the signal cycle
		Attainability	%	Proportion of green bandwith to the min green time for the through phase
		Transit delay[1]	sec/bus	average delay to transit vehicles at traffic signals
		Acceleration noise	ft/sec^2	Standard deviation of veh accelerations
	Oversaturated	Throughput	#	# veh served
		Extend of queue	#/mi	Distance or # of street segments with queue spillback
		Congestion duration	hr	Duration of oversaturated conditions
II. SAFETY				
Intersection/ Arterial/ Grid Network	Undersaturated/ Oversaturated	# accidents per type	#/yr	# of accidents by severity and/or traffic movement (e.g., # left turn related accidents)
		Encroachment time (ET)	# conflicts	Surrogate conflict measure
		# RLR	#	# of red light running violators
		# vehicles in yellow	#/cycle	# vehicles in platoon arrive in the yellow clearance interval per signal cycle
III. ENVIRONMENTAL				
Intersection/ Arterial/ Grid Network	Undersaturated/ Oversaturated	Fuel Consumption	gal	Excess fuel consumption due to delay & stops
		HC/CO/NOx/CO2/PM	/gr/m,	Air pollutant emissions / concentrations
		Noise	[db]	Inceased noise level due to congestion

Table 3. 1 Performance Measures for Signal Systems [22]

In addition to the MOEs in Table 3.1, the following indicators of performance are commonly used to assess the quality of traffic operations at highway facilities:

- Level of Service (LOS) per the Highway Capacity Manual (HCM) [22]
- Volume/Capacity ratio (v/c)

Level of service (LOS) is a quality measure describing operational conditions within a traffic stream, generally in terms of such service measures as delay, freedom to maneuver, traffic interruptions, and comfort and convenience. Six LOS are defined for each type of facility. LOS A represents the best operating conditions, and LOS F the worst. The average delay (sec/veh) is

used to characterize the LOS at signalized intersections. Table 3.2 shows the relationship between the average delay per vehicle and the LOS.

Control Delay (s/veh)	LOS by Volume-to-Capacity Ratio[a]	
	≤1.0	>1.0
≤10	A	F
>10–20	B	F
>20–35	C	F
>35–55	D	F
>55–80	E	F
>80	F	F

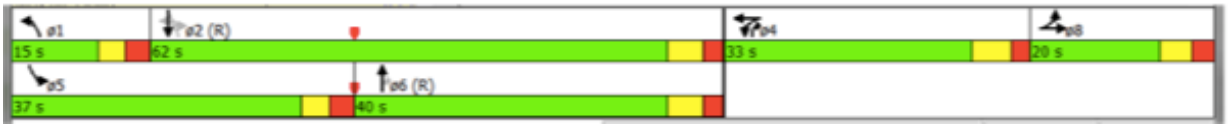
Note: [a] For approach-based and intersectionwide assessments, LOS is defined solely by control delay.

Table 3. 2 LOS table from Highway Capacity Manual

The (v/c) ratio indicates the level of congestion at an intersection approach or the entire intersection; v/c ratios greater than 1.00 indicate that the traffic demand is higher than capacity which results in oversaturated conditions with long delays, excessive queue lengths, and cycle failures.

Figure 3.1(a) shows a four-leg intersection located in in Beaufort, SC, which is equipped with Sensys Networks detectors. Figure 3.1(a) shows the layout of the intersection. Also, the signal operates as coordinated traffic actuated. The phasing and timing information are shown in Figure 3.1(b). There are leading left-turns on the main road (southbound and northbound) and split phasing on the cross streets. There are a total of 9 fixed-time timing plans for am, midday and pm peak on weekdays plus weekends, light traffic and saturated conditions. The signal settings for weekday conditions are shown in Figure 3.8(b). Fixed Control strategy changes according to the type of the day (weekday, Friday, and weekend) and time of the day. The HR data from advanced and stop-bar detector are available along with the control system information for Wednesday February 18, 2015 from 6:00am to 7:00pm.

(a) Intersection Geometrics



Time Period/Phase	1	2	3	4	5	6	7	8	Cycle Length
AM Peak (6AM-9AM)	15	40	N/A	39	15	40	N/A	16	110
Mid Day (9AM-2PM)	17	53	N/A	33	25	45	N/A	17	120
PM Peak (2PM-7PM)	15	62	N/A	33	37	40	N/A	20	130

(b) Signal Phase and Timing

Figure 3. 1 Test Intersection-Ribaurt Rd@Lady's Island Drive, Beaufort, SC

First performance measure is delay and according to Sensys Network's manual, its estimation process is as follows: after the light turns red, each time a vehicle crosses an advanced detector, that vehicle's time to join the queue is computed by calculating when the car would get to the stop-bar, traveling at the speed limit. This is done for each car that arrives during a red interval. Once the light turns green, the queue is assumed to disappear instantaneously. Thus, the average delay is computed using the time of arrivals of each vehicle to the queue, and the time when the light turns green. Since this approximation is weak, I suggest an alternative way for estimating delay in Section 3.2.

Figure 3.2 shows the average delay per vehicle per cycle for southbound through movement for one hour of the PM-peak interval (5PM-6PM). The same figure shows the average delays for the

same movement calculated using the analytical formula (3.1) from deterministic queuing (first term in the HCM delay estimation equation [22]):

$$d = \frac{C\left(1-\frac{g}{C}\right)^2}{2\left(1-\frac{g}{C}\times\frac{v}{c}\right)} \quad (3.1)$$

The HR data based delays are longer than the analytically estimated values, because the analytical method does not take into consideration the random delays (second term in the HCM signalized intersections delay equation).

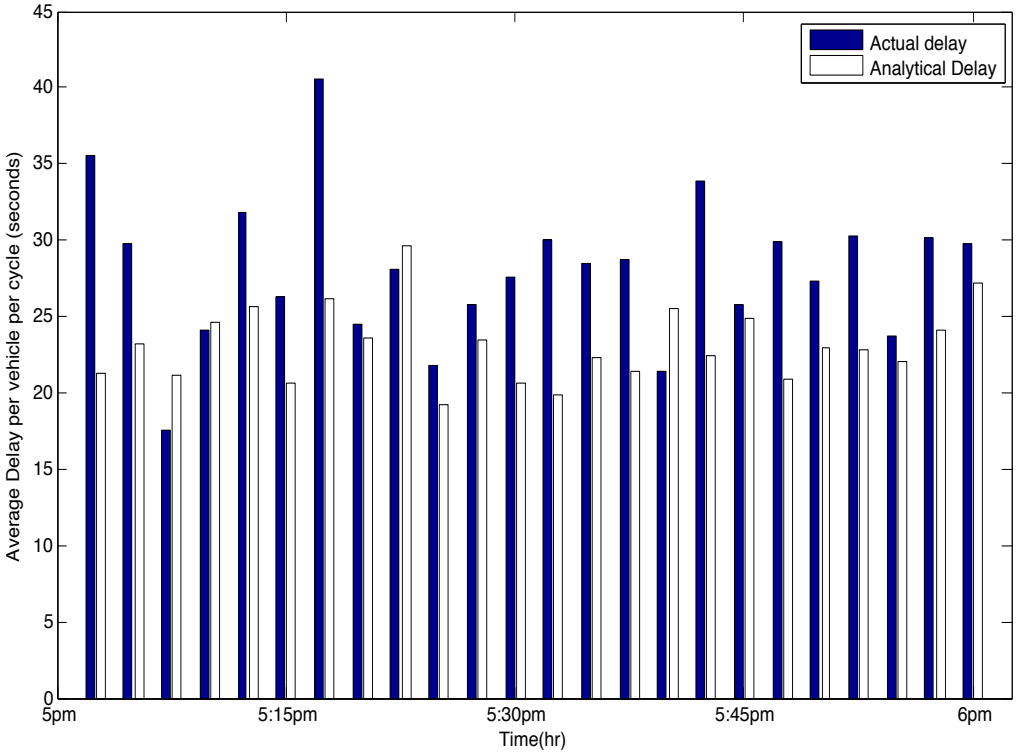


Figure 3. 2 Measured versus estimated delay for southbound through movement of Ribaurt Rd and Lady's Island Drive Intersection at 5PM-6PM

Figure 3.3 presents the Level of Service (LOS) on average delay curves for each phase, which estimated from detector data.
I use the average delay in second per vehicle per movement to estimate the LOS. Red lines on Figure 3.6 shows the limits for LOS values. According to Figure 3.3, phase 8, which covers all the movements in the northbound, has very high average delay compared to other phases. Other phases LOS are mostly at level B or C.

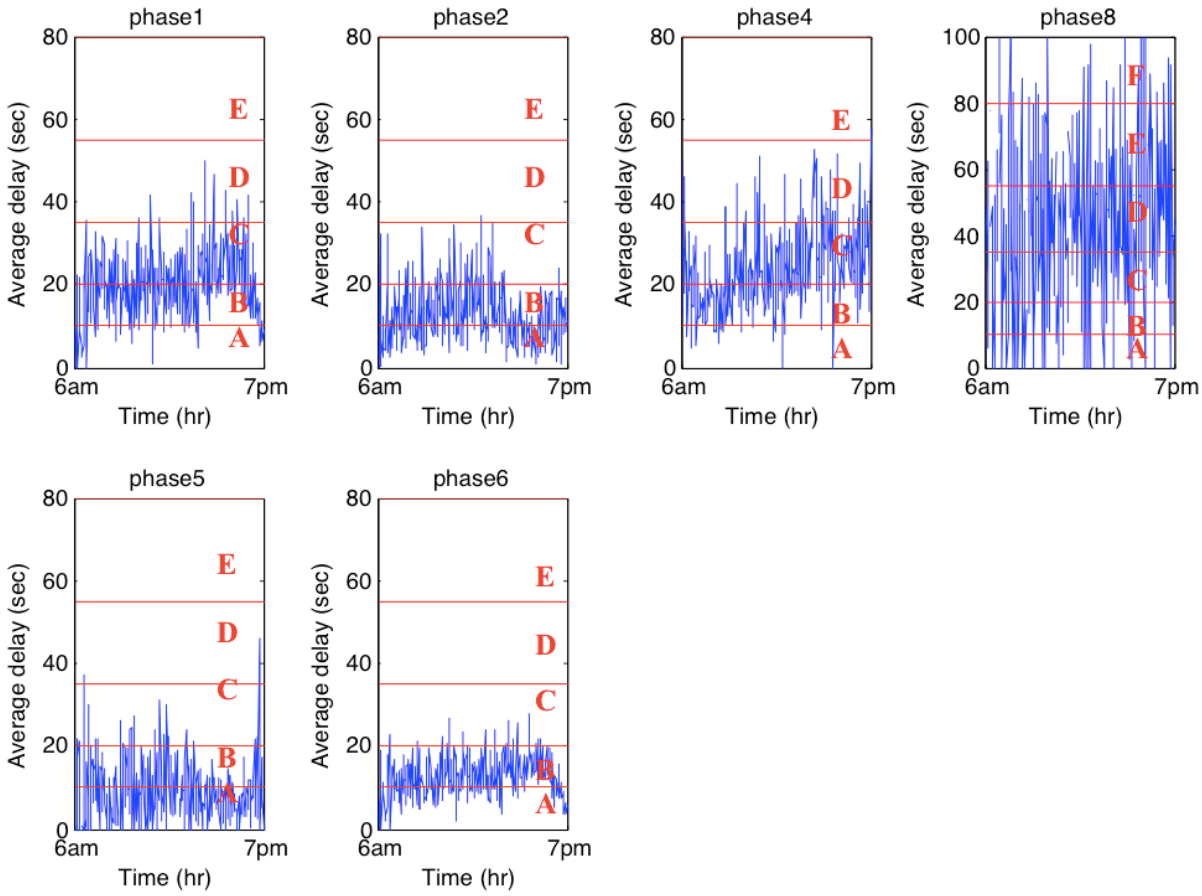


Figure 3. 3 Average delay with Level of Service (LOS) plots per cycle

In addition, there are signal timing data available that provide the green time per phase and the cycle length. Figure 3.4 shows the cycle length and the green time for the phases serving the left turns and through movements on southbound and northbound. Phase 2, which serves the through movement on southbound, has the longest green time. As shown in the Figure, the green time for Phase 6 which serves the through movement on northbound, is very close to phase 2.

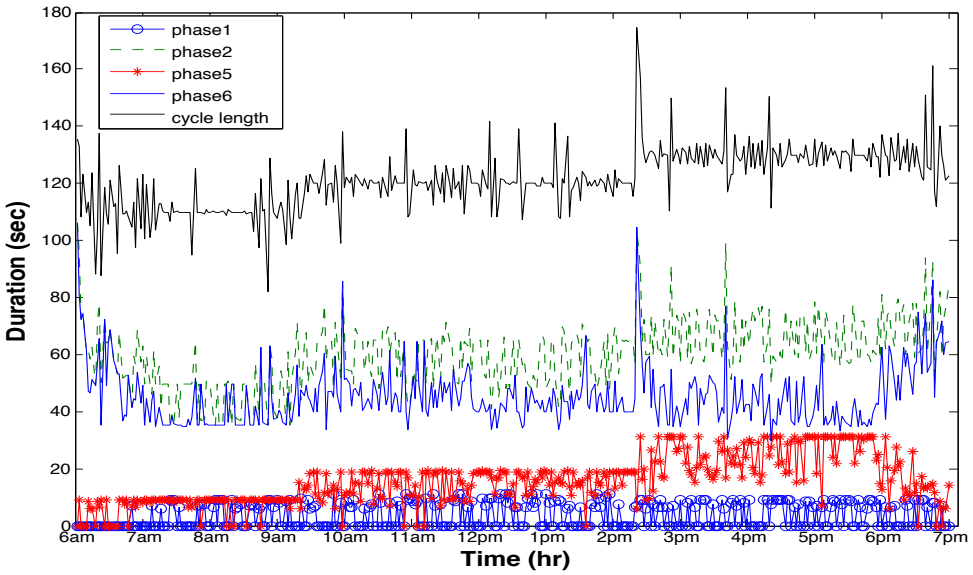


Figure 3. 4 Cycle length and green interval of phases that serve the northbound and southbound of the Ribaurt Rd and Lady's Island Drive Intersection for

The signal timing data also provide information about the wasted green time per phase. Wasted green time is the time interval that a particular phase remains green but no vehicles enter the intersection from that movement and there is an active call (request for service) by other phase(s). Figure 3.5 shows the frequency of the wasted green time for phase 4 (westbound) during the analysis period (6PM-7PM). It can be seen that in over 56% of the signal cycles there is a wasted green time of longer than 4 seconds (yellow time is about 4 seconds)., indicating the need for adjusting the existing signal settings.

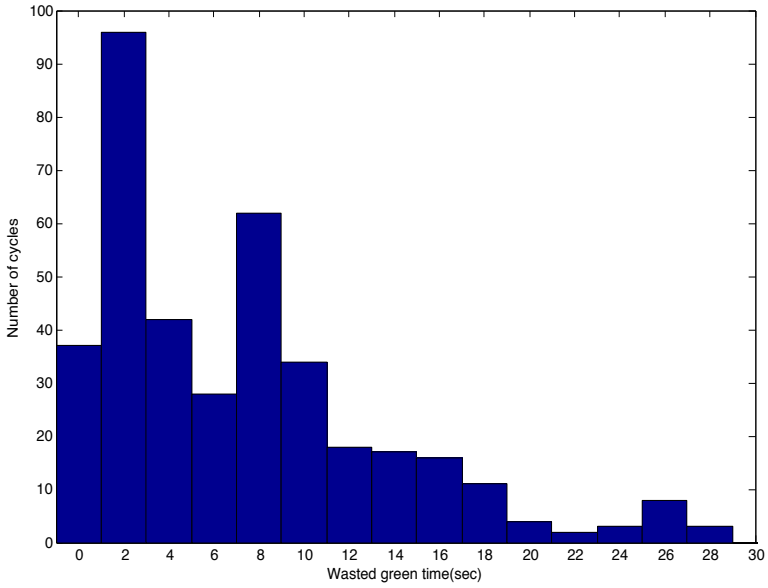


Figure 3. 5 Wasted green time for phase 4 (westbound movements)

3.2 Using Stochastic Gradient Descent for Estimating Vehicle Accumulation in Links

In this chapter I use an alternative method for estimating vehicle accumulation. I use HR data from the advanced detector and stop-bar detector to construct the arrival and departure curve. Next, I find the vehicle accumulation by finding the difference between the arrival and departure curve. On the other hand, the delay and queue length are easy to estimate by plotting the cumulative arrival and departure curve. [23], [24], and [25] have detailed explanation on estimating the queue length and delay from cumulative count curves. Although, in this dissertation I am only focusing on estimating the vehicle accumulation in real time for each approach by using real time HR data.

3.3 Problem Formulation

There are advanced detectors located at the beginning of the link and stop-bar detectors located at the end of the link, the link is only the road segment in between the two sets of detector. I use detector vehicle counts at the entrance to the link and at the exit of the link to construct a naive vehicle count estimate as the cumulative difference between the flows of vehicles at the entrance and the exit. But unknown biases in detector counts and random errors make this naive estimate useless, so estimation algorithms propose alternatives.

The approach is close to the naive estimator, corrected by compensation for the errors from biased and noisy advance and stop bar detector counts. An online learning algorithm based on stochastic gradient descent discovers the bias. The method provably learns the bias, and efficiently estimates the vehicle accumulation in the link with a theoretical guarantee under a certain condition on the detectors, namely, the stop bar detector reliably indicates when there is no vehicle in front of it. For the problem formulation time is continuous. Let $Q(t), t \geq 0$ be the vehicle accumulation in the link i.e. the number of vehicles between the advance and stop bar detector at time t.

The set $\{t|Q(t) > 0\}$ is a union of intervals called *busy periods* $(\tau_i, \bar{\tau}_i), i \geq 1; \tau_i$ is the beginning, $\bar{\tau}_i = \tau_i + T_i$ is the end, and T_i is the length of busy period i. More precisely, $Q(t) > 0$ if $t \in \cup_i (\tau_i, \bar{\tau}_i)$ and $Q(t) = 0$ if $t \notin \cup_i (\tau_i, \bar{\tau}_i)$. I assume that the stop bar detector indicates when $Q(t) = 0$, i.e. when $t \notin \cup_i (\tau_i, \bar{\tau}_i)$. $Q(t)$ evolves as

$$Q(t) = \begin{cases} A_n(t) - D_n(t) & t \in (\tau_n, \bar{\tau}_n), for\ some\ n \\ 0 & t \notin \cup_n (\tau_n, \bar{\tau}_n) \end{cases}, \tag{3.2}$$

in which $A_n(t), t \in [\tau_n, \bar{\tau}_n]$ and $D_n(t), t \in [\tau_n, \bar{\tau}_n]$ are the *cumulative* arrival and departure counts in the n-th busy period. That is, $A_n(t)$ vehicles entered the link and $D_n(t)$ vehicles departed the link during $[\tau_n, t]$. Note that arrivals during a non-busy period are immediately served, so a vehicle arriving during $t \notin \cup_n (\tau_n, \bar{\tau}_n)$ does not experience any delay.

Advance detectors at the entrance and stop bar detectors at the exit of the link measure $A_n(t)$ and $D_n(t)$possibly with some bias or independent noise. Denote by $\hat{A}_n(t), t \in [\tau_n, \bar{\tau}_n]$ the cumulative counts of the entrance detector, and by $\hat{D}_n(t), t \in [\tau_n, \bar{\tau}_n]$ the cumulative counts of the exit

detector. Because of detector noise and bias $\hat{A}_n(t)$ may not equal $A_n(t)$ and $\hat{D}_n(t)$ may not equal $D_n(t)$. Since the stop bar detector indicates when $Q(t) = 0$, a naive vehicle accumulation estimator is

$$\hat{Q}_{naive}(t) = \begin{cases} \hat{A}_n(t) - \hat{D}_n(t) & t \in (\tau_n, \bar{\tau}_n), for\ some\ n \\ 0 & t \notin \cup_n (\tau_n, \bar{\tau}_n) \end{cases}. \quad (3.3)$$

This naive estimator uses the information about when the link becomes empty only to reset its estimate to zero, and does not attempt to estimate any systematic counting error. Further, it may lead to negative estimates of the vehicle accumulation. The proposed estimation algorithm and its theoretical properties are based on the following model of the detector counting processes:

$$\hat{A}_n(t) - \hat{D}_n(t) = A_n(t) - D_n(t) + \int_{\tau_n}^{t} b(t)dt + Z_n(t - \tau_n), t \in [\tau_n, \bar{\tau}_n]. \quad (3.4)$$

Here $b(t)$ is the (possibly time-varying) systematic error or bias of the detectors' counting processes, and $Z_n(t)$ is a sequence of independent cumulative zero-mean noise random variables with $Z_n(0) = 0$. I assume that $E[Z_n^2(t)] \leq c_1 t$ for some finite constant $c_1 > 0$ for all n.

3.4 Algorithm Description

The estimation algorithm is based on stochastic gradient descent. I assume that $b(t) = b$. Let $\alpha_n, n \geq 1$, be the step-size (learning rate) of the algorithm, that is a positive decreasing sequence. The properties for the step size are standard for stochastic approximation [26]. The intuition is that the sum of the step sizes should be unbounded so that learning does not stop, while the sum of the squares of step sizes should be finite so that the cumulative error of estimation remains bounded. The estimate is designed to be $\hat{Q}(t) = \left[\hat{A}_n(t) - \hat{D}_n(t) - \varepsilon_n T_n\right]^+$ wherein ε_n is the correction term for busy period n, and $[x]^+ = \max(x, 0)$. ε_n is updated to learn the bias term b. Formally, the algorithm proceeds as follows.

1. Initialize $\varepsilon_0 = 0$ and $n = 0$.
2. If $t \notin \cup_n (\tau_n, \bar{\tau}_n)$, then $\hat{Q}(t)$=0.
3. If $t \in (\tau_n, \bar{\tau}_n)$ for some n, then $\hat{Q}(t) = \left[\hat{A}_n(t) - \hat{D}_n(t) - \varepsilon_n(t - \tau_n)\right]^+$.
4. Update the correction term at the end of the busy period: $\varepsilon_{n+1} \leftarrow \varepsilon_n + \alpha_n\left(\hat{A}_n(t) - \hat{D}_n(t) - \varepsilon_n T_n\right)$.
5. $n \leftarrow n + 1$. Repeat step 2-5.

I now provide some intuition for the algorithm, which tries to learn the bias b of the naive estimator in Equation 3.3. To find this bias adaptively, I consider a correction term $\varepsilon_n, n \geq 1$, that ideally should be close to b. I update ε_n based on stochastic gradient descent which tries to solve the following offline optimization problem: $min_\varepsilon f(\varepsilon) = \frac{1}{2}(b - \varepsilon)^2$. The solution to the optimization problem is obviously $\varepsilon^* = b$. If gradient descent is used to find the optimal solution, the update rule for ε would be $\varepsilon_{n+1} = \varepsilon_n - \alpha \frac{\partial}{\partial \varepsilon} f(\varepsilon) = \varepsilon_n + \alpha(b - \varepsilon)$, in which α is the step size. To find an algorithm based on knowing when the link is empty, I replace $b - \varepsilon$ by

its (scaled) unbiased estimator $\hat{A}(\bar{\tau}_n) - \hat{D}(\bar{\tau}_n) - \varepsilon_n T_n$.

3.5 Algorithm Evaluation

For simulation purposes, I assumed the following setting and parameters. Consider a single discrete-time link that has Poisson arrivals with rate λ vehicles per time slot. One can assume that each time slot in my simulation setup is 5 seconds. I set λ= 1.4 veh/ (5 sec) = 1008 veh/hr. I consider a cycle time of 12 time slots or 60 seconds with 6 time slots of green signal (30 seconds), and 6 time slots of red signal. The service time distribution in my simulation is deterministic with service rate μ = 0.6 veh/(sec) = 2160 veh/hr if the signal is green and μ = 0 if the signal is red.

There are 3 detectors at the link as shown in Figure 3.6. Detector A counts the number of vehicles that enter the link. Detector B counts the number of vehicles that exit the link, and detector C observes whether the link is empty or not. I evaluate the performance of the proposed algorithm when I assume that detector C is noiseless (since this is the key information I need for estimating the bias), detector A counts each arriving vehicle with probability 0.95 independently, and detector B counts each departing vehicle with probability 0.85 independently.

This model captures the two important properties of the model: 1) random noise since each vehicles is observed with some probability independently, and 2) time-invariant bias since the observation probabilities of detectors A and B are different but fixed, which creates a bias. Observe that with the explained choices of the parameters, I expect to see a bias ϵ = (0.95 - 0.85) λ = 0.14 vehicles per time slot. I choose the step size $\alpha_n = 0.02/n^{0.6}$. Figure 3.7 shows how the correction term ε_n converges to the bias ϵ = 0.14. One observes that after only around 30 busy periods the algorithm learns the bias and gets close to 0.14. The estimated vehicle accumulation for a period of 5000 seconds is shown in Figure 3.8.

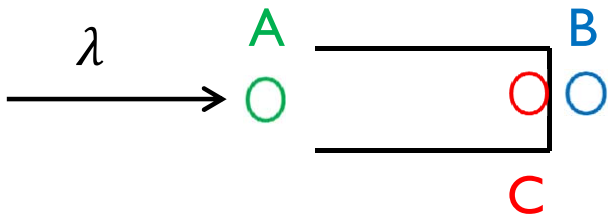


Figure 3. 6 Detector's layout for simulation

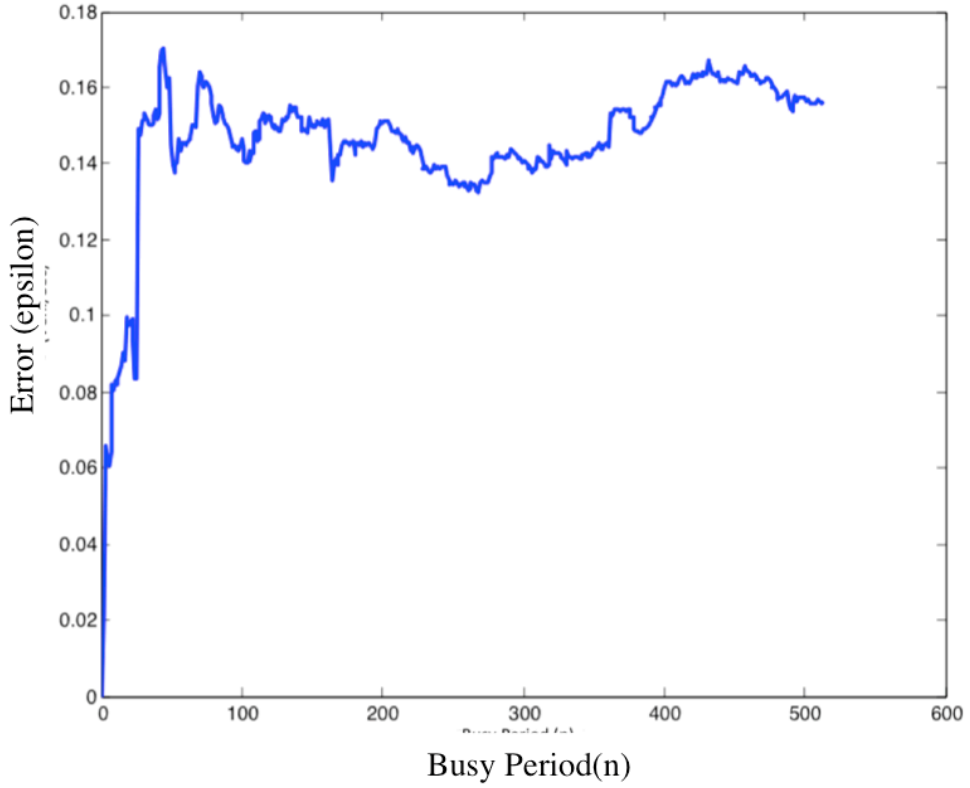


Figure 3. 7 Error convergences

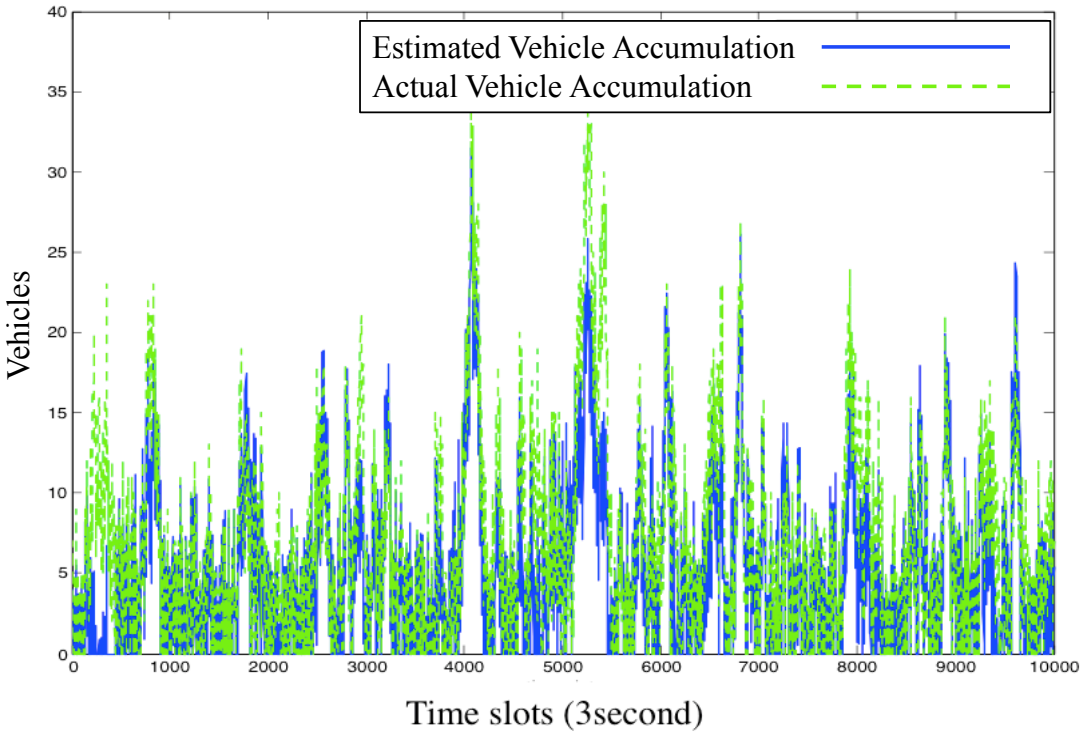


Figure 3. 8 Actual versus estimated vehicle accumulation

I tested this algorithm on the intersection presented in Figure 3.1. Since I do not have the technology to know the exact time that link is empty in this intersection, I define the following characteristics. If light is green for more than 1 time slot, which is 3 seconds, and no vehicle passes the stop-bar and advanced detector the link is empty. Of course, detecting whether the link is empty or not through the mentioned rule is not exact and does not completely match the theoretical model. Figure 3.9 shows the difference between the arrival and departure curve real-time for southbound through movement for 2-hour interval during the AM-peak. This information is useful for designing some control strategies such as Max-Pressure [27]. However, in this scenario, I am not able to verify the accuracy of the estimate because of the lack of ground-truth values.

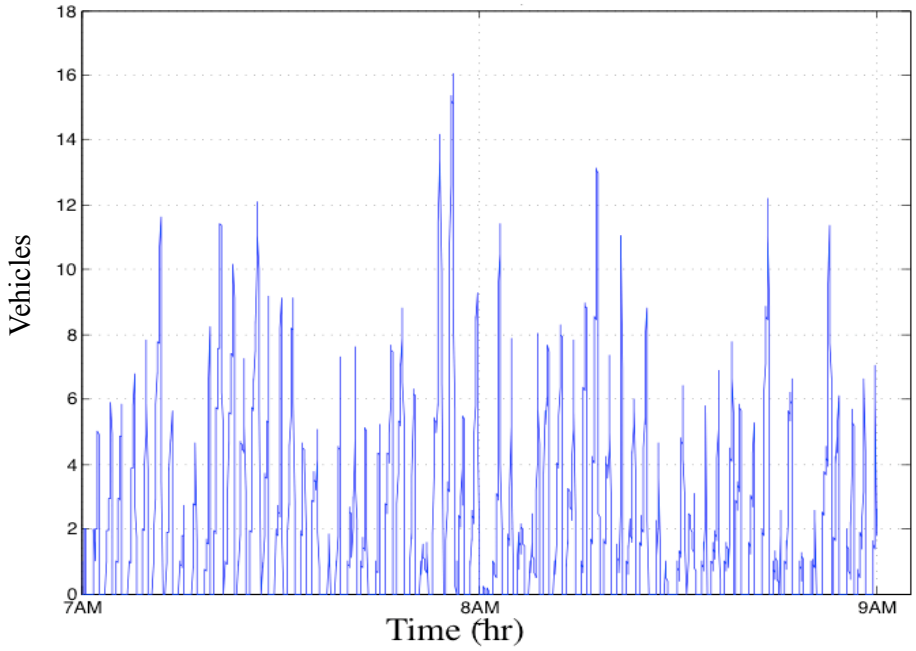


Figure 3. 9 Vehicle accumulation at southbound through movement for 7AM-9AM

3.6 Discussion and Future Research

I considered the problem of estimating the number of the vehicles behind red light at an intersection from noisy and biased vehicle count observations. I developed a real-time estimation algorithm based on stochastic gradient descent that provably learns detector bias, and estimates the arrival and departure curve with theoretical guarantee. I supported our theoretical contribution with simulations results and a detailed case study.

There are two immediate directions for future research:

- We assumed that the algorithm perfectly observes whether the link is empty or not. It would be interesting to investigate the performance of the algorithm with noisy observation of whether the link is empty, both theoretically and experimentally.

- One reason for estimating vehicle accumulation is to design efficient feedback control policies for the network. For example, the max-pressure algorithm [27] is known to be throughput-optimal, but it requires knowledge of the vehicle accumulation. An interesting question is to study the stability of the network with estimated vehicle accumulation that are asymptotically exact.

Chapter 4. Estimating Control Parameters at Signalized Network

Up to this point, the main focus of my work was on individual intersection and evaluation of single control system. Now, I want to extend my work to group of intersections and discuss designing control parameters and improving the control systems in urban networks.

Now that the performance of the current system was evaluated, according to Figure 1.1 flow-chart, the fourth step is to design new control parameters that can satisfy demand fluctuation and improve the performance of the system. The new data and performance measures are the input values and updated control parameters will be the output. There are many different algorithms, including the Signal Phase and Timing (SPaT) [28] algorithm, that estimates traffic signal control parameters. Algorithms, such as SPaT are normally focusing on the cycle time, green interval, and phase sequences, but there are still weaknesses in estimating the optimum offset values for arterial. Therefore, in this chapter I focus on estimating the optimum offset values for an urban network using a new method.

The last step in the flow-chart on Figure 1.1 is implementation. However, before applying any new method in the field and on real-world intersection, it is necessary to test its performance in simulation and report the weaknesses. Hence, I use the VISSIM microscopic traffic simulator to test developed algorithms on arterials and networks. I use Component Object Module (COM) program to communicate with VISSIM through MATLAB (Appendix A). The framework that is shown in appendix A could be used to collect data from simulator and update control parameters in urban networks in real-time. Such system is necessary for testing the performance of the robust timing plans, also it is beneficial for evaluating the impact of rapid update of the control system on the network's condition.

4.1 Introduction: Offset Optimization Algorithm

Traffic signal offsets specify the timing of a traffic light relative to adjacent signals. Offsets constitute the main parameter for coordinated traffic movement among multiple traffic signals. Optimizing the offsets in an urban network reduces the delay and the number of stops that vehicles experience. Existing offset optimization algorithms focus on two-way arterial roads. The papers [29] and [30] present algorithms for maximizing the green bandwidth, that is, the length of the time window in which a vehicle can travel along the entire road without being stopped by a red light. Traffic control software such as Synchro [31], TRANSYT [32], [33] optimize offset values by minimizing delay and number of the stops. Recently, advances in data collection technology led to methods for offset optimization using archived traffic data [34].

All of the above-cited methods assume sufficient storage capacity for links and therefore do not consider the risk of spill-back in which a segment of road between traffic lights is completely filled with vehicles so that upstream traffic cannot enter the link, even with a green light. However, spill-back is a critical condition that can arise in an urban network; [35] and [36] focus on detecting and modeling spill-back. In some cases, spill-back can be prevented by changing the control system setting [37]; another study [38] shows that severe congestion could be improved by dynamically adapting offset values.

The paper [39] introduced a new approach that formulates offset optimization as a quadratically constrained quadratic program amenable to convex relaxation. This approach has many advantages over previous studies. It considers all links in a network with arbitrary topology and is not restricted to a single arterial. Further, the approach is computationally efficient and used in [40] to optimize offsets for large networks with high traffic demand from multiple directions.

Next, I extend the algorithm in [39] to allow for links with finite storage capacity. By eliminating the assumption of infinite storage capacity, I get one step closer to improving the original model and make it more realistic. Moreover, with an example, I show how the result changes when the storage capacity constraints are active.

The rest of this chapter is organized as follows. Section 4.2 briefly describes the vehicles arrival and departure model and Section 4.3 explains the offset optimization algorithm from [39]. Section 4.4 presents the simulation results from three different networks. Section 4.5 introduces the storage capacity constraints and extends the algorithm from Section 4.2. Section 4.5 also presents the evaluation process of the optimization problem when the storage capacity constraints are active with an example. Finally, Section 4.6 and 4.7 present the conclusions and some ideas for future researches.

4.2 Problem Formulation

In this section, I briefly recall the traffic network model proposed in [39]. Consider a network with a set *S* of signalized intersections and a set *L* of *links*. Let $\sigma(l)$ denote the traffic signal at the head of the link *l* controlling the departure of vehicles from link *l*, and let $\tau(l)$ denote the signal at the tail of the link *l* controlling the arrivals of vehicles into link *l*. (Traffic in a link flows from its tail to its head.)

All signals have a common cycle time *T*, hence a common frequency $\omega = 2\pi/T$ rad/sec. The signals follow a fixed-time control. Relative to some global clock, each signal *s* has an offset value of $\theta_s = (0,2\pi]$ radians that represents the start time of the fixed control pattern of the intersection. This pattern has designated green interval for each movement that repeats every cycle and controls the vehicle flow. Therefore, each link $l \in L$ has a vehicle accumulation of $q_l \geq 0$ at time *t* equal to the difference between the cumulative arrivals of vehicles, $a_l\ (l)$, and departures, d_l ,

$$\dot{q}_l(t) = a_l(t) - d_l(t) \tag{4.1}$$

If exogenous arrivals into the network are periodic with period *T* and there is no spill-back, it is reasonable to assume the network is in periodic steady state so that all arrivals and departures are also periodic with period *T* [41]. I then approximate the arrival and departure processes in a link as sinusoids of appropriate amplitude and phase shift. To this end, for each entry link *l*, the arrival of vehicles into link *l* at signal $\sigma(l)$ is approximated as

$$\hat{a}_l(t) = A_l + \alpha_l \cos(\omega t - \varphi_l) \tag{4.2}$$

for constants $A_l, \alpha_l, \varphi_l \geq 0$ with $A_l \geq \alpha_l$. The constant A_l is the average arrival rate of vehicles into link *l*; α_l allows for periodic fluctuation in the arrival rate. For a non-entry link *l*, the arrival process is approximated by

$$\hat{a}_l(t) = A_l + \alpha_l \cos\left(\omega t - (\theta_{\tau(l)} + \varphi_l\right), \tag{4.3}$$

for

$$A_l = \sum_{i \in L} \beta_{li} A_i \,, \tag{4.4}$$

$$\alpha_l^2 = (\beta_{li} A_i \cos(\gamma_i))^2 + (\beta_{li} A_i \cos(\gamma_i))^2, \tag{4.5}$$

$$\varphi_l = \lambda_l + \arctan\left(\frac{\sum_{i \in L} \beta_{li} A_i \cos(\gamma_i)}{\sum_{i \in L} \beta_{li} A_i \sin(\gamma_i)}\right). \tag{4.6}$$

where λ_l denotes the travel time, in radians, of link *l* and β_{li} denotes the fraction of vehicles that are routed to link *l* upon exiting link *i*, which is given and fixed. The mid-point of the green interval in every cycle is specified by its offset $\gamma_i \in [0, 2\pi]$. The signal offsets at the tail and head of each link are respectively θ_τ and θ_σ.

Similarly, I approximate the departure process for both entry and non-entry links by

$$\hat{d}_l(t) = A_l \left(1 + \cos\left(\omega t - \left(\theta_{\sigma(l)} + \gamma_l\right)\right)\right) \tag{4.7}$$

where $(\theta_{\sigma(l)} + \gamma_l)$ is the actuation offset of link *l* as determined by the offset of signal σ_l at the head of link *l* and the green interval offset, γ_l, of link *l*.

From equation (4.4.1), (4.4.3), and (4.4.7) I formulate the approximating queueing process, $\hat{q}_l(t)$, as

$$\dot{\hat{q}}_l(t) = \hat{a}_l(t) - \hat{d}_l(t) = \alpha_l \cos\left(\omega t - \theta_{\tau(l)} - \varphi_l\right) - A_l \cos\left(\omega t - \theta_{\sigma(l)} - \gamma_l\right) = Q_l \cos(\omega t - \xi_l) \tag{4.8}$$

where

$$Q_l(\theta) = \sqrt{A_l^2 + \alpha_l^2 - 2A_l \alpha_l \cos\left(\left(\theta_{\tau(l)} + \varphi_l\right) - \left(\theta_{\sigma(l)} + \gamma_l\right)\right)}, \tag{4.9}$$

and ξ_l is a phase shift; I omit the explicit expression for ξ_l but note that it is easily computed.

Therefore, it follows that

$$\hat{q}_l(t) = \frac{Q_l}{\omega} \sin(\omega t - \xi_l) + B_l, \tag{4.10}$$

where B_l is the average vehicle accumulation on link *l*. Since $\hat{q}_l(t)$ cannot be negative, I conclude that $\frac{Q_l}{\omega} \leq B_l$.

The cost function is the total average vehicle accumulation in all links, $\sum_{l\in L} Q_l$. Since the first part $A_l^2 + \alpha_l^2$ of formula (4.9) is constant, instead of minimizing $\sum_{l\in L} Q_l$, I can maximize the negative part of the formula, $A_l\alpha_l \cos\left(\left(\theta_{\tau(l)} + \varphi_l\right) - \left(\theta_{\sigma(l)} + \gamma_l\right)\right)$ for all links and our objective becomes

$$\max_{\{\theta_s\}} \sum_{s\in S} \sum_{l\in L} A_l\alpha_l \cos\left(\theta_{\tau(l)} - \theta_{\sigma(l)} + \varphi_l - \gamma_l\right) \tag{4.11}$$

in which the offset values, θ_s, are the only decision variables and all other parameters are given by the green splits.

4.3 Algorithm Description

Problem 4.11 is non-convex. To solve it, [39] uses semi-definite relaxation. To this end, [39] first formulates the equivalent QCQP by defining $z=(x,y)$, where $x_s = \cos(\theta_s)$ and $y_s = \cos(\theta_s)$. Thus, the equivalent cost function becomes $z^T W z$, where

$$W_1[s,u] = \sum_{l\in L} A_l\alpha_l \cos(\varphi_l - \gamma_l) \tag{4.12}$$
$$W_2[s,u] = \sum_{l\in L} A_l\alpha_l \sin(\varphi_l - \gamma_l) \tag{4.13}$$

and

$$\underline{W} = \begin{bmatrix} W_1 & W_2 \\ -W_2 & W_1 \end{bmatrix}, W = \frac{1}{2}\left(\underline{W} + \underline{W}^T\right) \tag{4.14}$$

Also, I have the constraints $x_s^2 + y_s^2 = 1$ for all $s \in S$ let $E_s \in \mathbb{R}$ for $s \in S$ be given by

$$E_s[v,u] = \begin{cases} 1 & if\ u = v \\ 0 & otherwise \end{cases} \tag{4.15}$$

and define

$$M_s = \begin{bmatrix} E_s & 0 \\ 0 & E_s \end{bmatrix}. \tag{4.16}$$

Then the constraints $x_s^2 + y_s^2 = 1$ are equivalent to $z^T M_s z = 1$ As a result, the optimization problem is

$$\begin{aligned} &\max_{z\in\mathbb{R}^{2|S|+2}} z^T W z \\ &S.t \quad z^T M_s z = 1 \quad \forall s \in S \end{aligned} \tag{4.17}$$

The quadratic terms in 4.17 are of the form $z^T Qz$ with $Z = zz^T$. Further if the matrix Z is positive semi-definite and of rank 1, it can be decomposed as $Z = zz^T$. The advantage of this transformation is that is linear in the new variable Z.

In the original offset optimization algorithm, [39] relaxes the exact, non-convex QCQP into a convex Semi-Definite Programming (SDP) by removing the rank constraint and gets the following formulation

$$\begin{aligned} &\max_{z \in \mathbb{R}^{2|S|+2}} \boldsymbol{Tr}(WZ) \\ &S.t \quad \boldsymbol{Tr}(M_s Z) = 1 \quad \forall s \in S \\ &Z \geq 0 \end{aligned} \tag{4.18}$$

The solution to the relaxed convex SDP problem gives an upper bound on the value of the optimization problem. This upper bound is the optimum solution of the rank-constrained SDP problem if the solution matrix has rank 1. If the solution matrix has rank bigger than 1, [39] proposes vector decomposition as one of the options for obtaining the optimum solution. In this paper, I used vector decomposition to estimate the optimum solution as well.

4.4 Algorithm Simulation

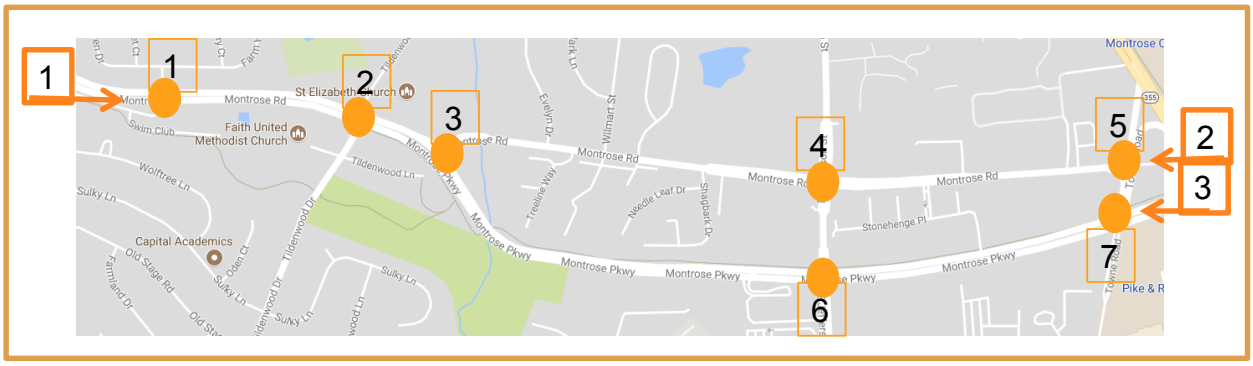
In this section, I test the performance of the offset optimization algorithm that was developed in [39] on three real world case study networks. Figure 4.1(a) shows a ten-intersection portion of San Pablo Ave. in Berkeley, California that serves as our first case study. The figure also indicates the input approaches that have the most traffic volume. The second and third case study networks are respectively shown in Figure 4.1(b) and Figure 4.1(c). Similar to the first network, Figure 4.1(b) and Figure 4.1(c) indicate the major inputs of the networks. These networks are equipped with detectors from Sensys Network Inc.. According to the detection data, major traffic flow into the Montrose Rd. Network from three directions and they flow into the Live Oak Ave Network from two directions. Figure 4.1(b) indicates that most traffic in the eastbound direction comes from input 1 and in the westbound direction comes from inputs 2 and 3.

In the San Pablo Ave Network, there are 2 through lanes in each direction and left turn lanes (1 lane for left turn) at each intersection. The distance between intersections ranges from 0.15 to 0.35, for a total length of 2.1 miles. The posted speed limit is 30 mph. All signals operate as fixed-time coordinated with a common cycle length of 80 seconds.

In the Montrose Rd Network, there are 3 through lanes in each direction and left turn lanes (additional 1 to 3 lanes for turn) at each intersection. The distance between intersections ranges from 0.2 to 0.6, for a total length of 1.4 miles (note that the length of the top portion and the bottom portion of the network is very similar, therefore, the distance from the left-most intersection to any of the right-most intersection is around 1.4 miles.) The posted speed limit is 40 mph. All signals operate as fixed-time coordinated with a common cycle length of 120 seconds.

In the Live Oak Ave Network, there are 2 through lanes in each direction and left turn lanes (1 lane for turn) at each intersection. The distance between intersections ranges from 0.1 to 0.4, for a total length of 1.5 miles. The posted speed limit is 35 mph. All signals operate as fixed-time coordinated with a common cycle length of 120 seconds.

(a) Case study 1: San Pablo Ave Network in Berkeley, California

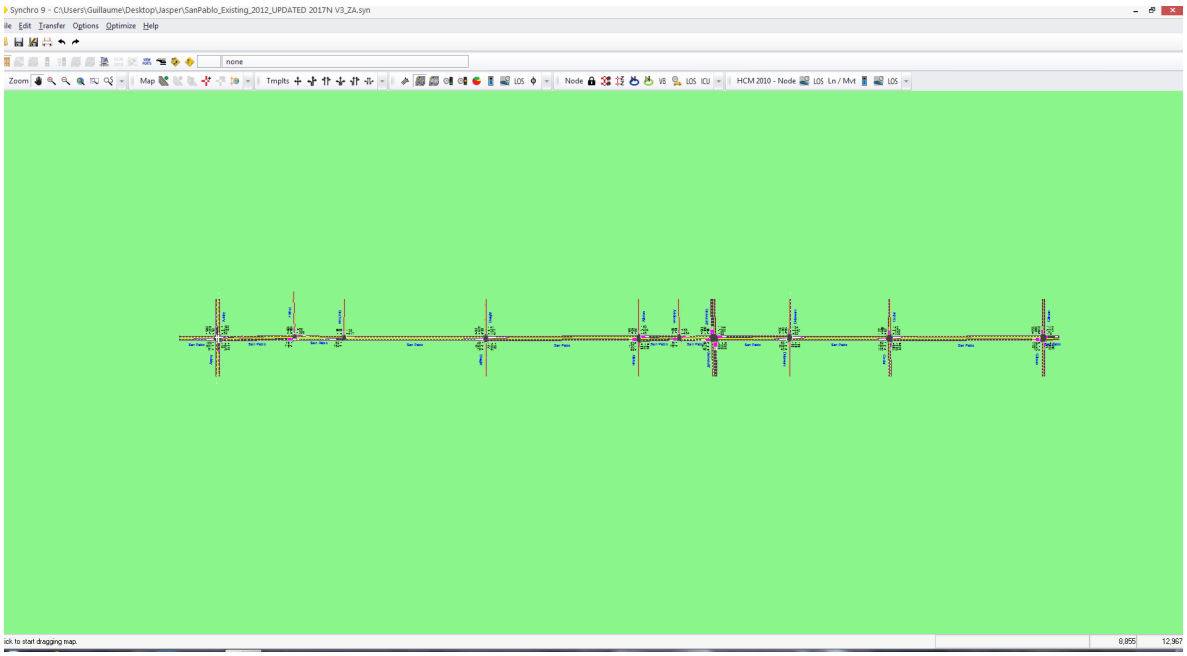


(b) Case study 2: Montrose Rd. Network in Montgomery County, Maryland.

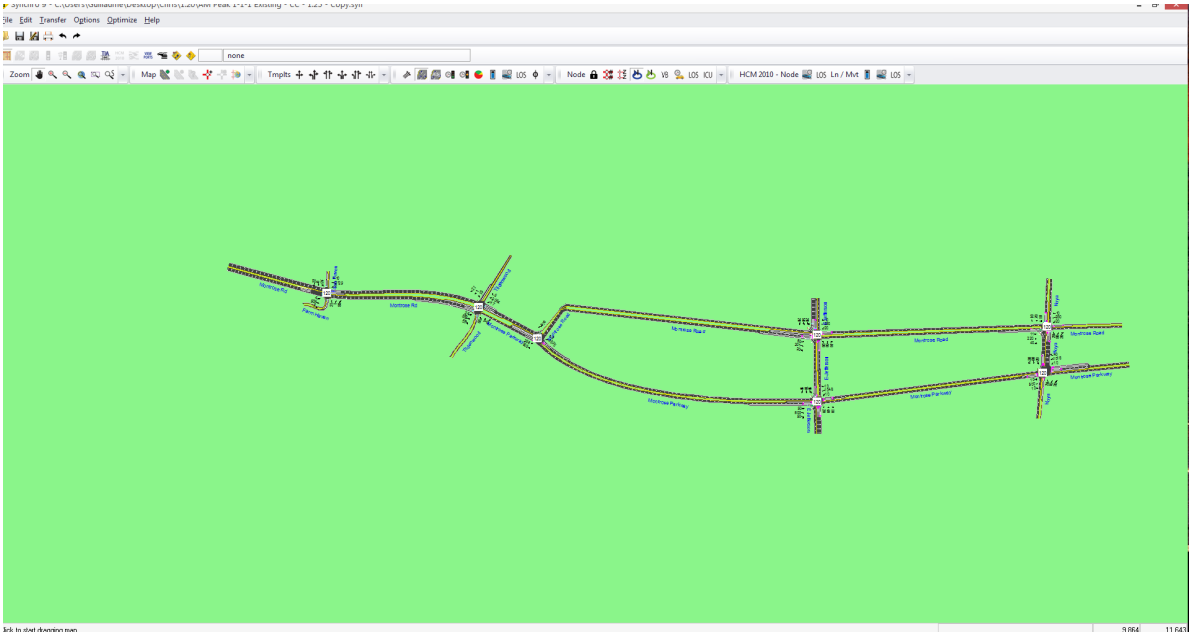
(c) Case study 3: Live Oak Ave in Arcadia, California.

Figure 4. 1 Case studies site schemes

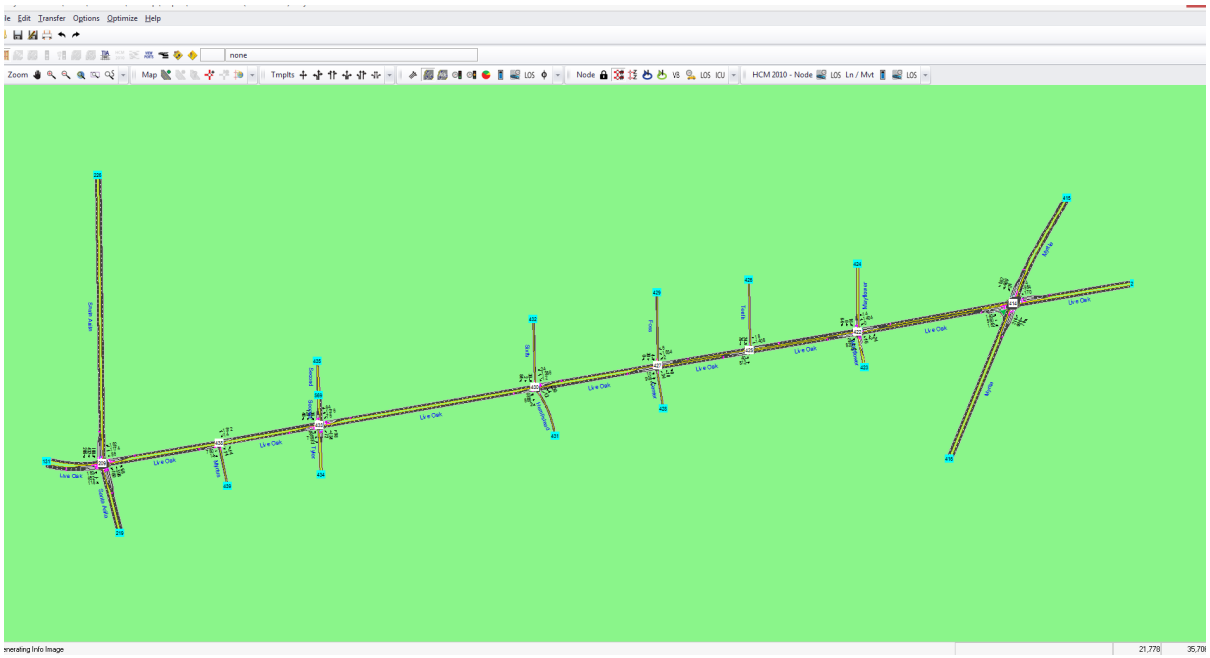
I used Synchro 9 to build a test-bed of the networks, as shown in Figure 4.2, and then used Synchro's offset optimization tool to optimize the intersection offsets, while other control parameters such as cycle time, green time, split ratio, and etc. are constant. As explained in the Synchro user manual, for each offset combination, Synchro reevaluates the departure patterns at the intersection and surrounding intersections and uses Highway Capacity Manual (HCM) delay equation to recalculate delay values [22]. It then chooses the offset values with the lowest delay as the optimum. I repeated this process for all the traffic profiles and recorded the offset values.



(a) San Pablo Ave Network Synchro model



(b) Montrose Rd. Network Synchro model

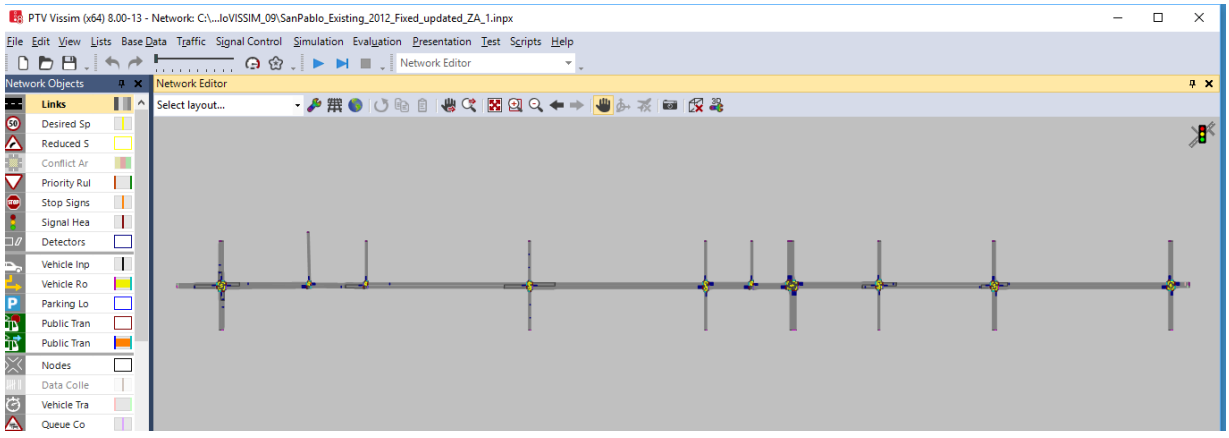


(c) Live Oak Ave Network Synchro model

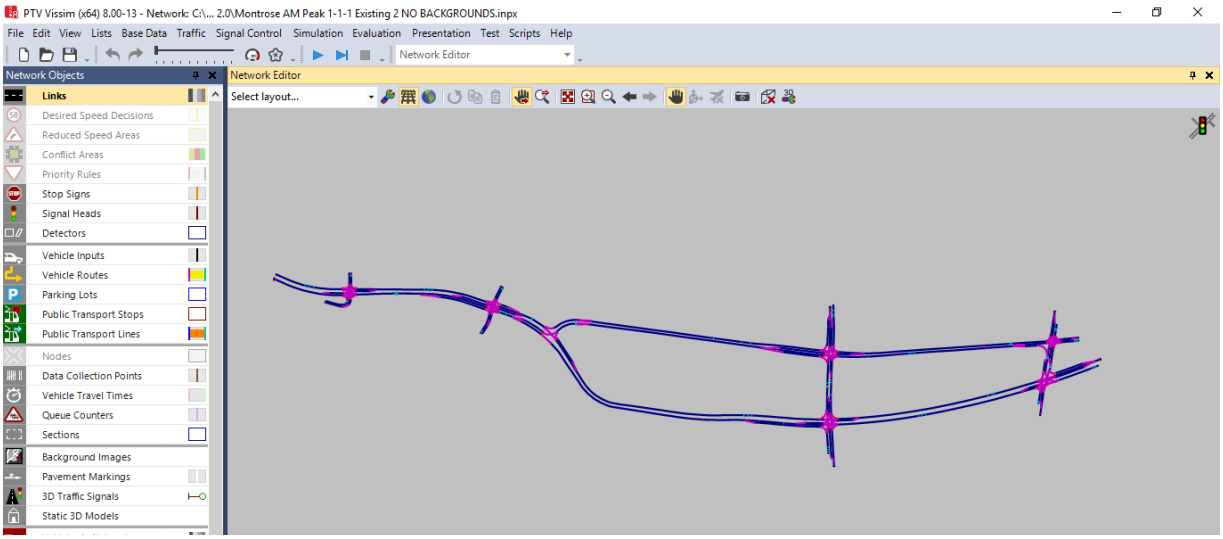
Figure 4. 2 Synchro models for the networks

In addition, a simulation test-bed was built in VISSIM 8 [42] as shown in Figure 4.3 for evaluating the performance of different offsets. I used the current signal settings and network information for modeling, and for each traffic profile I tested 2 scenarios:

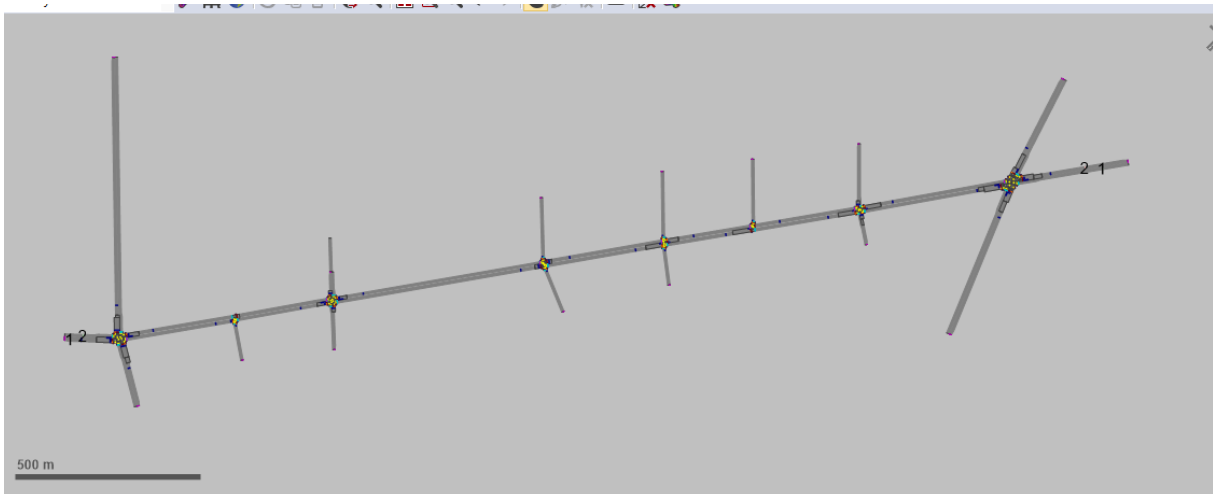
- Offsets determined by Synchro's optimization method.
- Offsets determined by the proposed offset optimization algorithm.



(a) San Pablo Ave Network VISSIM model



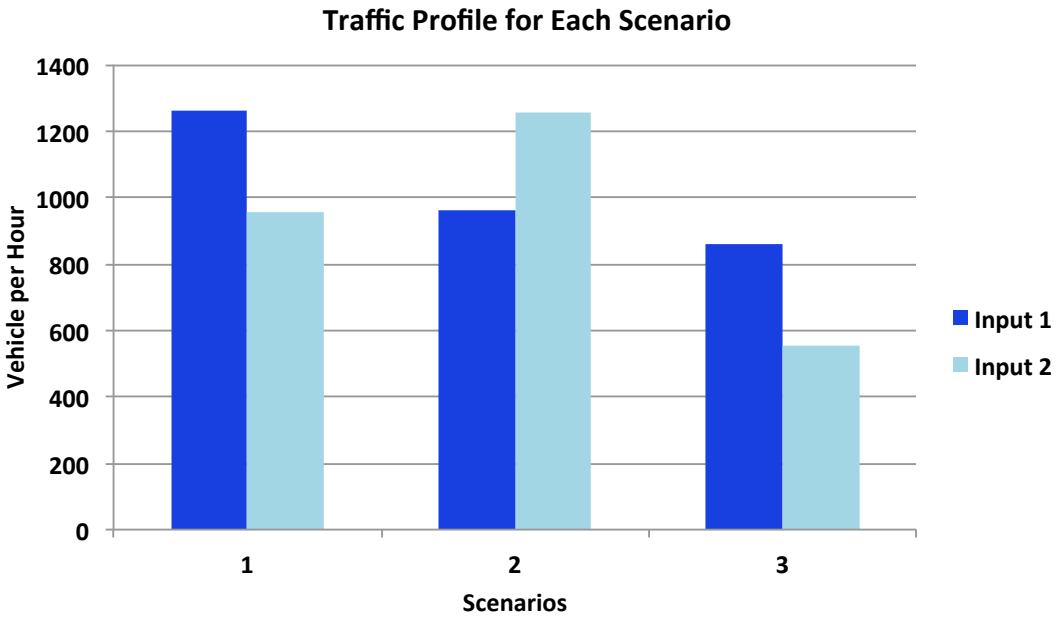
(b) Montrose Rd. Network VISSIM model



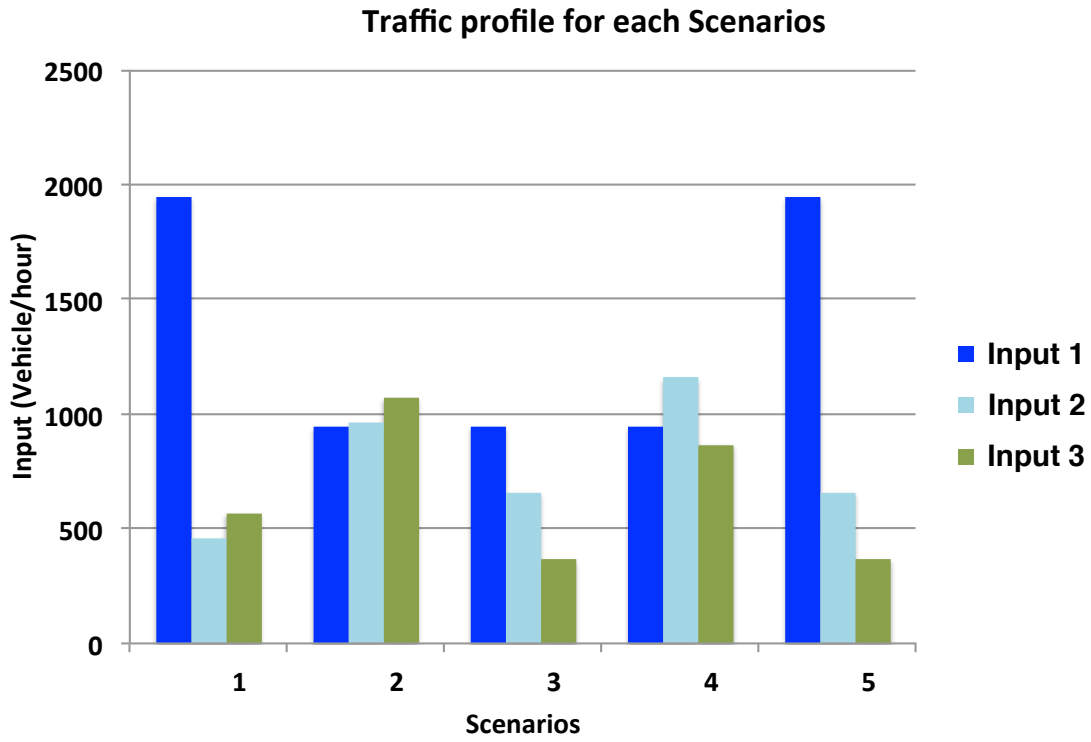
(c) Live Oak Ave Network VISSIM model

Figure 4. 3 VISSIM models for the networks

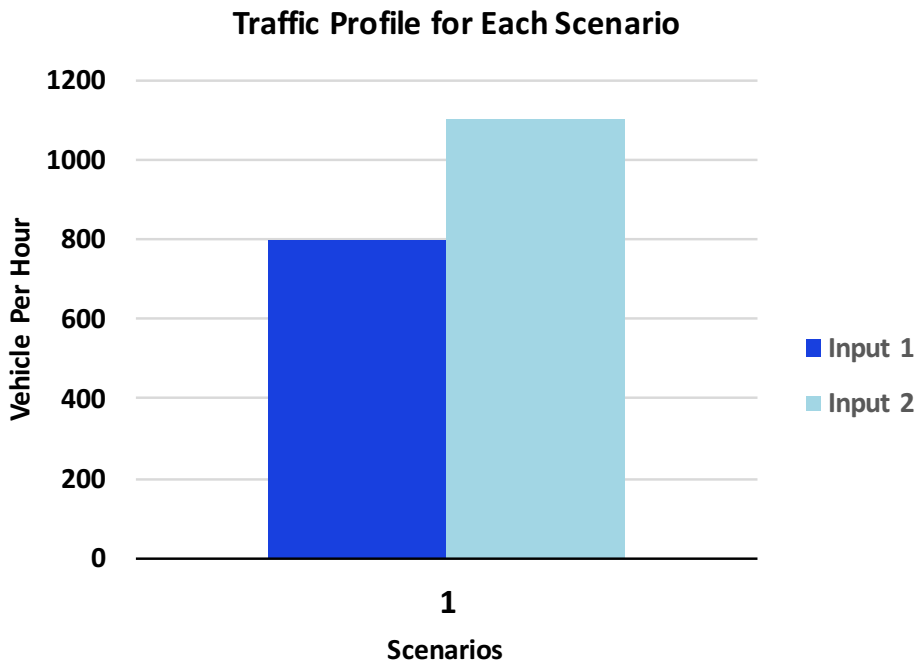
Based on available data, I designed several different traffic profiles for each network, which is showne in Figure 4.4. In Figure 4.4(a), San Pablo Ave Network traffic profile, scenario 1 and scenario 2 correspond to the traffic during the peak hour in different directions, and scenario 3 corresponds to the traffic during off-peak hours. Also, in Figure 4.4(b), Montrose Rd. Network traffic profiles, scenarios 1 and 5 represent the AM-peak condition while scenarios 2 and 4 are the PM-peak condition, and scenario 3 represents the mid-day traffic profile. Finally, Figure 4.4(c), Live Oak Ave Network traffic profiles, presents the AM-peak traffic condition in that network.



(a) San Pablo Ave Network traffic profiles



(b) Montrose Rd. Network traffic profiles



(c) Live Oak Ave Network traffic profiles

Figure 4. 4 Traffic profiles for all the networks

For the case study networks and for each traffic scenario, I tested the performance of the offsets suggested by the offset optimization algorithm and offsets suggested by Synchro. To do so, I ran the VISSIM simulation for one hour for each scenario. In order to evaluate the performance of the network under each offset configuration, I collected the following measures for vehicles moving along the major routes:

- Average number of stops that each vehicle experiences.
- Average vehicle delay that each vehicle experiences.

Major routes carry the most traffic volume. In the San Pablo Ave Network and the Live Oak Ave Network, highest traffic volumes are in the south-to-north and north-to-south approaches through all the intersections. In the Montrose Rd. Network, there are four major routes: from west to the upper east leg and vice-versa, and from west to lower east leg and vice-versa.

Before presenting the vehicle delay and number of stops results, let's take a look at an example of the offset values for Live Oak Ave Network, Scenario 1. As it is shown in Figure 4.5, offset values from Synchro and algorithm are different and there is no relationship between them. In all other networks and scenarios, there is no pattern between the offset values, because the methods that Synchro and algorithm use to optimize offset are different. Hence, the offset values will be different.

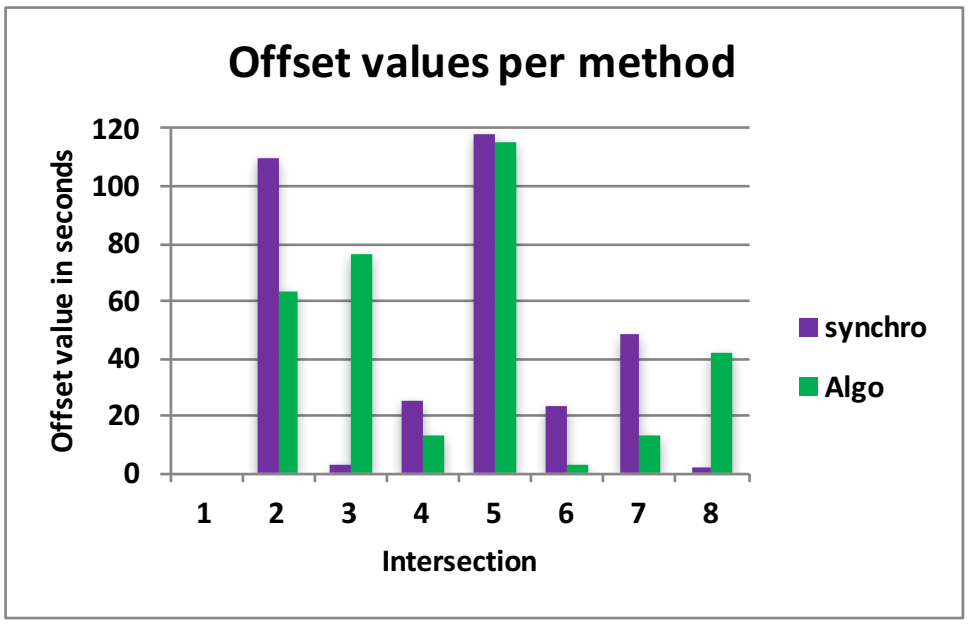
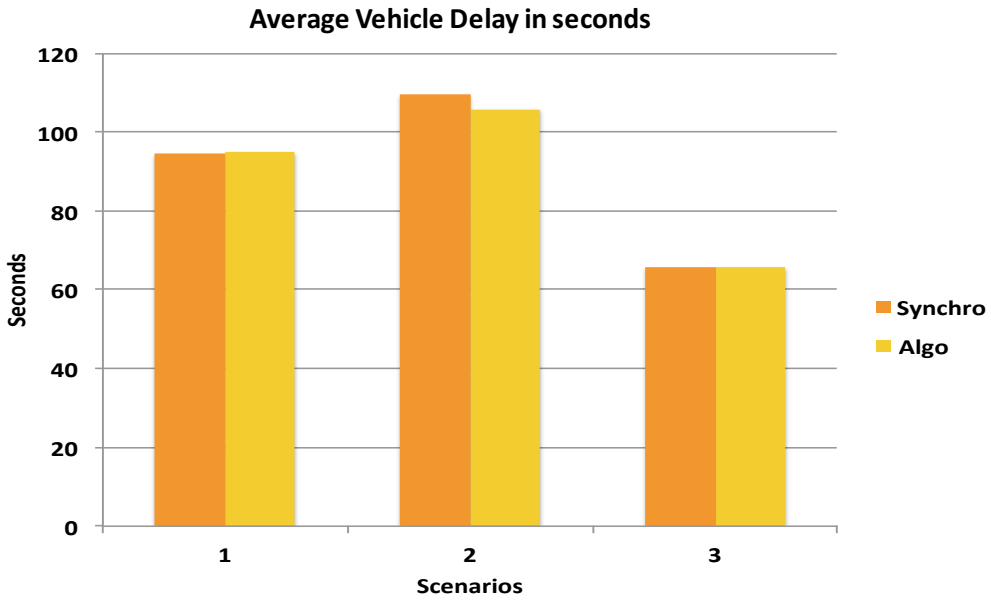
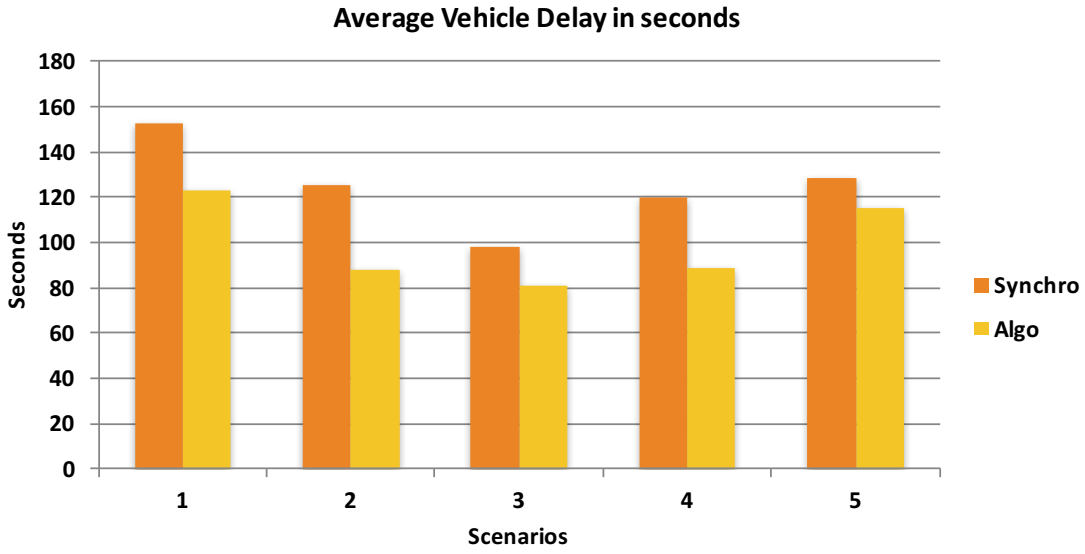


Figure 4. 5 Example of the offset values for Live Oak Network, Scenario 1

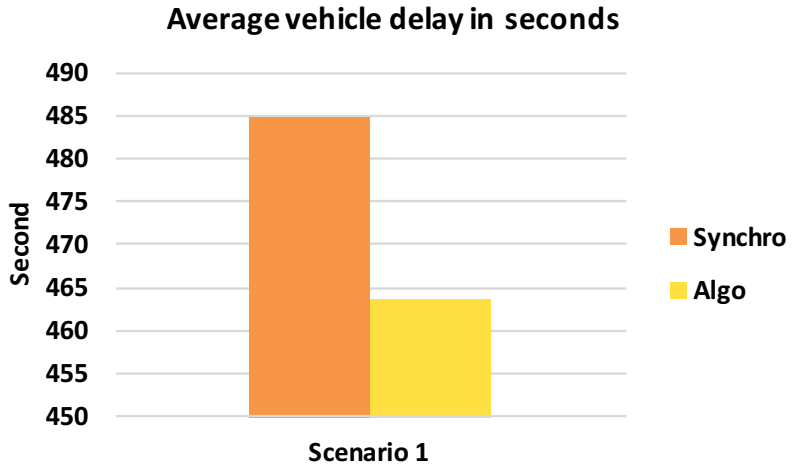
The following results present the average traffic measures over all the major routes in each network.



(a) Average vehicle delay results for San Pablo Ave Network



(b) Average vehicle delay results for Montrose Rd. Network



(c) Average vehicle delay results for Live Oak Ave Network

Figure 4. 6 Case studies average vehicle delay results for every scenario

VISSIM estimates the vehicle delay as the difference between the travel time in free flow condition and actual travel time of each vehicle. This delay includes the time that a vehicle is stopped at red lights and accounts for the acceleration and deceleration time as well. In Figure 4.6, the orange (respectively, yellow) columns show the average delay that vehicles experience under offset values suggested by Synchro (respectively, the proposed algorithm).

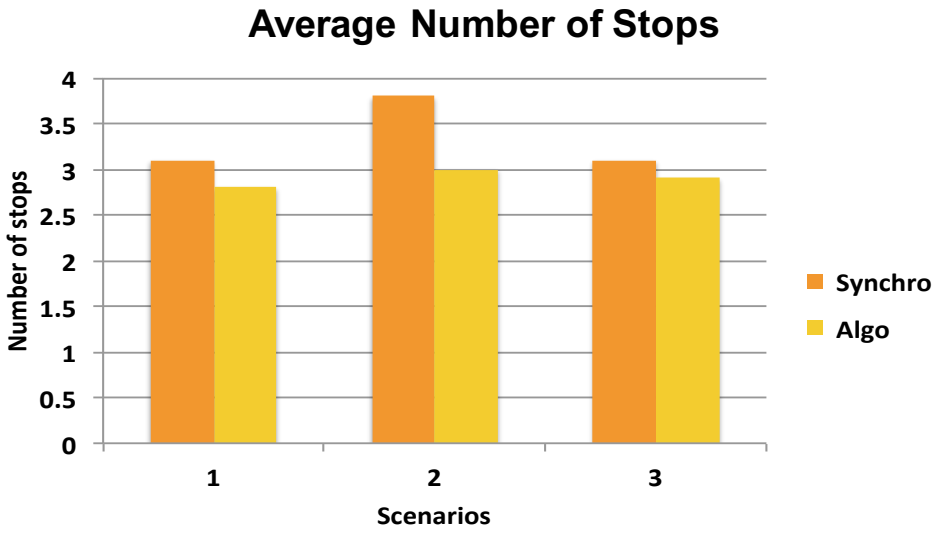
In the Montrose Rd. network, as seen in Figure 4.6(b), the average delay is lower for all traffic scenarios under offset values suggested by the proposed offset optimization algorithm compared with Synchro, and scenario 2 experiences the most improvement, with 30% reduction in delay.

Clearly, the algorithm offsets reduce vehicle delay in Live Oak Ave Network, Figure 4.6(c) as well. However, in the San Pablo Ave Network, Figure 4.6(a), the vehicle delay is almost same for both Synchro and algorithm's offset values under all the traffic profiles.

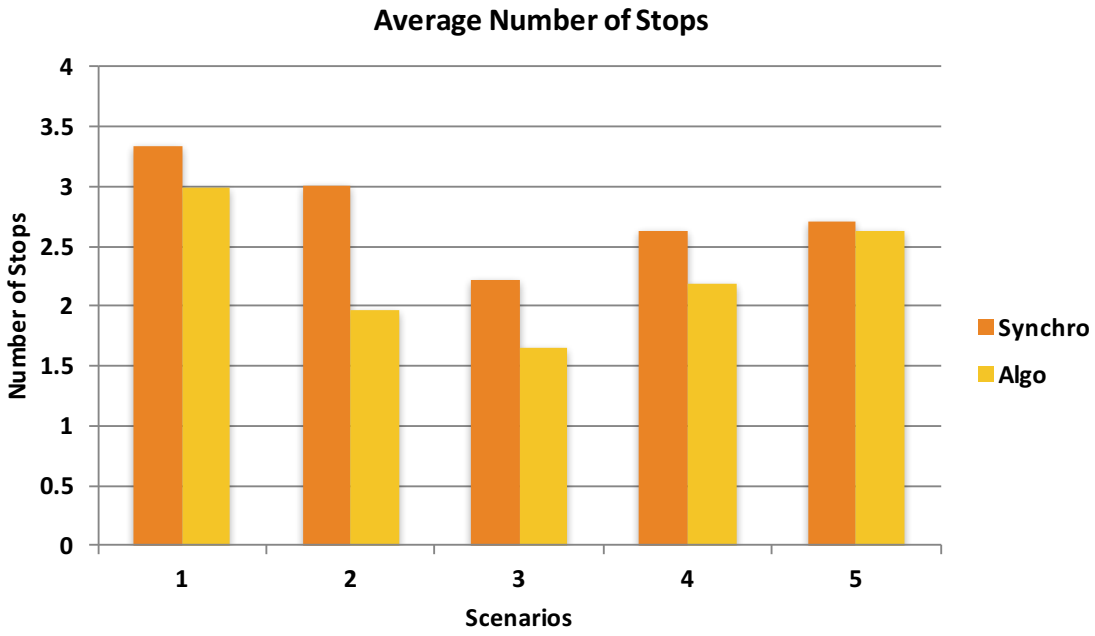
In Figure 4.7, the orange (respectively, yellow) columns show the average number of stops that vehicles experience under offset values suggested by Synchro (respectively, the proposed algorithm). Clearly, the algorithm offsets reduce the number of stops in all three traffic profiles for the San Pablo Ave Network, Figure 4.7(a) and up to 20% improvement in scenario 2.

In the Montrose Rd. Network, I tested 5 traffic profiles. Figure 4.7(b) shows the average number of stops for these profiles. In all scenarios, the proposed algorithm outperforms Synchro. Scenario 2 shows the most improvement of 30% reduction in the number of stops. Also, according to Figure 4.7(c) the vehicles during the AM-peak in the Live Oak Ave Network experience 6% fewer stops under the offset values from the algorithm compare to the Synchro's offset values.

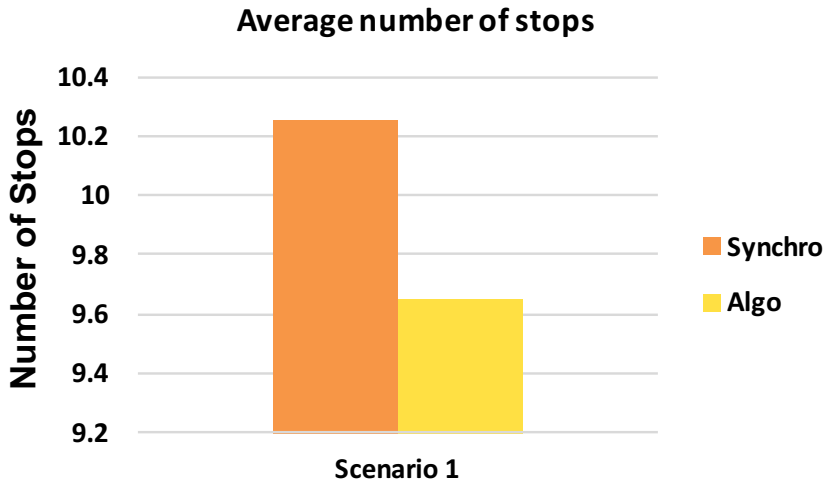
Note that the number of stops vehicles experience in the Live-Oak Ave Network is very large, it is because of the cycle failure in intersection 1 in the westbound direction. This intersection has about 40 second of green time and it only has 1 through movement lane, as a result, very large queue forms at this intersection and very quickly it grows and reaches upstream intersections. Hence, vehicles experience very large delay and number of stops in this direction because of the cycle failure.



(a) Average number of stops results for San Pablo Ave Network



(b) Average number of stops results for Montrose Rd. Network



(c) Average number of stops results for Live Oak Ave Network

Figure 4. 7 Case studies average number of stops results for every scenario

4.5 Algorithm Improvement

The result from case studies show the algorithm's suggested offset values improve the traffic condition in the networks. However, this algorithm implicitly assumes infinite storage capacity on links. While this assumption may be sometimes reasonable, as in the case studies above, some networks with short links or high traffic volumes may be susceptible to the spill-back phenomenon where a link exceeds its capacity and the queue blocks upstream traffic flow. In this

section I show how to modify the algorithm to work for networks with limited storage capacity links.

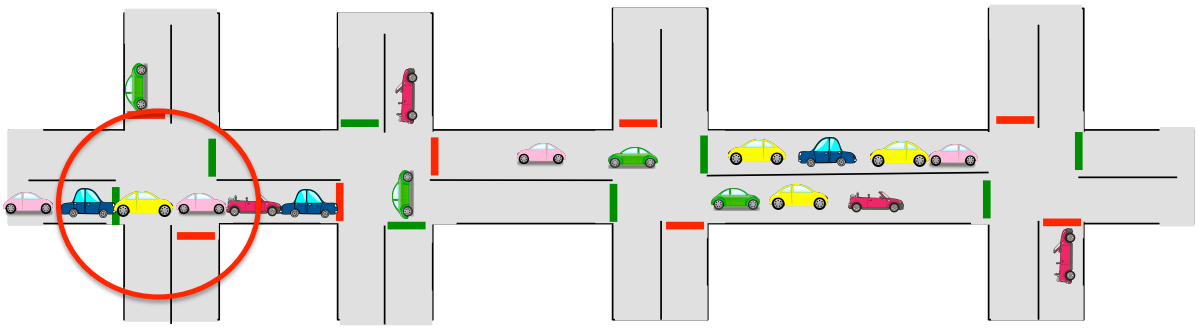


Figure 4. 8 Queue spill-back at the first intersection (red circle) blocks the entrance.

At this point I have a model of the average vehicle accumulation, equation 4.9, in each link as a function of the offsets θ_s for $s \in$ S. However, in reality, the number of the vehicles occupied each link cannot exceed the storage capacity of that link, and if it reaches the capacity, vehicles will not be able to enter the link and will block the upstream intersection; this situation is called *spill-back*. Figure 4.8 shows an example of the spill-back in a network. The intersection marked by the red circle is experiencing spill-back, and vehicles cannot enter the intersection even though the light is green.

In order to prevent spill-back, I constrain the maximum vehicle accumulation that can exist in each link, $2B_l$, to be equal or less than the maximum number of the vehicles that can be stored in the link under jam density, k_l. Section 4.2 explains that $\frac{Q_l}{\omega} \leq B_l$, so for each link $l \in L$, the storage capacity constraint is as following

$$2Q_l \leq \omega k_l \;\Leftrightarrow\; Q_l^2 - \left(\frac{\omega k_l}{2}\right)^2 \leq 0. \tag{4.19}$$

This leads to the final optimization problem by adding the constraint (4.19) to (4.11)

$$\max_{\{\theta_s\}\, s\in S} \sum_{l\in L} A_l \alpha_l \cos\left(\theta_{\tau(l)} - \theta_{\sigma(l)} + \varphi_l - \gamma_l\right)$$
$$s.t\;\; Q_l^2 - \left(\frac{\omega k_l}{2}\right)^2 \leq 0 \;\; \forall l \in L. \tag{4.20}$$

Now, to incorporate the new constraints in the QCQP formulation for each link $z^T C_l z \leq K_l$ where $K_l = -A_l^2 - \alpha_l^2 + \left(\frac{\omega k_l}{2}\right)^2$ and C_l is given by

$$C_1[s,u] = \begin{cases} -2A_l\alpha_l \cos(\varphi_l - \gamma_l) & if\;\; s = \tau(l)\;\; and\;\; u = \sigma(l) \\ 0 & otherwise \end{cases} \tag{4.21}$$

$$C_2[s,u] = \begin{cases} -2A_l\alpha_l \sin(\varphi_l - \gamma_l) & if\;\; s = \tau(l)\;\; and\;\; u = \sigma(l) \\ 0 & otherwise \end{cases} \tag{4.22}$$

with

$$\underline{C_l} = \begin{bmatrix} C_1 & C_2 \\ -C_2 & C_1 \end{bmatrix} , C_l = \frac{1}{2}\left(\underline{C_l} + \underline{C_l}^T\right). \quad (4.23)$$

The original optimization problem from 4.17 together with the new storage capacity constraints leads to

$$\begin{aligned} &\max_{z \in \mathbb{R}^{2|S|+2}} z^T W z \\ &S.t \quad z^T M_s z = 1 \quad \forall s \in S \\ &z^T C_l z \leq K_l \quad \forall l \in L \end{aligned} \quad (4.24)$$

Then, I relax the exact, non-convex QCQP into a convex Semi-Definite Programming (SDP) by removing the rank constraint

$$\begin{aligned} &\max_{z \in \mathbb{R}^{2|S|+2}} \boldsymbol{Tr}(WZ) \\ &S.t \quad \boldsymbol{Tr}(M_s Z) = 1 \quad \forall s \in S \\ &\boldsymbol{Tr}(C_l Z) \leq K_l \quad \forall l \in L \\ &Z \geq 0. \end{aligned} \quad (4.25)$$

When Z has rank equal to 1, the answer is the optimum solution but, if *Z* has rank higher than 1the result of the optimization problem is the upper bound solution. Therefore, one option is to use vector decomposition to estimate a feasible solution close to the optimum point.

Now, let's test the algorithm by assuming a network such as the one in Figure 4.8, with 4 intersections and fixed common cycle time for all intersections. Starting from the most left-hand side intersection, there is intersection 1, 2, 3, and 4. Also, there is link 1, connecting intersection 1 and 2, link 2, connecting intersection 2 and 3, and finally we have link 3, connecting intersection 3 and 4.

Input demands from minor streets to the network are minimal and there are traffic volumes equal to 800 and 500 vehicles per hour respectively on the eastbound and westbound. Moreover, signals have fixed predesigned timing plans but the offset vales were estimated by Offset optimization algorithm. I estimated the offset value under the following scenarios

- **Scenario 1**: Infinite storage capacities for links.
- **Scenario 2**: Storage capacity of link 1, 2, and 3 are respectively 3, 4, and 5 vehicles.

As a result, Table 4.1 shows the offset values for scenario 1 and 2. Moreover, Table 4.2 presents the algorithm's estimated values of the maximum number of the vehicles in each link under each scenario. According to this table, the maximum vehicle accumulation in link 1 WB, link 2 WB, and link 3 WB are smaller in scenario 2 and they are equal to the storage capacity values of the links. However, the maximum vehicle accumulation in link 1 EB, 2 EB, and 3 EB, increased due to the new offset values.

Intersection	1	2	3	4
Offset values in seconds for scenario 1	0	4	10	19
Offset values in seconds for scenario 2	0	3	7	12

Table 4. 1 Offset value of each intersection under the two scenarios

Link	Number of vehicles in Scenario 1	Number of vehicles in Scenario 2
Link 1 EB	1.8	2.2
Link 2 EB	2.3	3.0
Link 3 EB	2.5	4.6
Link 1 WB	3.2	3.0
Link 2 WB	4.4	4.0
Link 3 WB	6.0	5.0

Table 4. 2 Maximum number of vehicles in each link under the two scenarios

Clearly estimated vehicle accumulation values are not equal to the queue length, also the model does not consider car following theory. So, I will not be able to test the new algorithm in a simulator and on a real-world network. But the result from the algorithm and vehicle accumulation values from the model shows that algorithm is preventing spill-back by having a limit on the maximum number of the vehicles in the link.

4.6 Discussion and Future Research

In this chapter, I evaluated the performance of an offset optimization algorithm on two real world networks under several traffic profiles. In almost all cases, the offset values obtained by the proposed algorithm reduce the average number of stops and total delay that vehicles experience as compared to Synchro's extracted optimum offset values. I used Synchro's suggested optimum offset value as my baseline because it is a popular method among transportation engineers that is commonly used in practice.

The improvement varies depending on the geometry of the network and traffic profile. However, in all scenarios, the proposed offset optimization algorithm has better performance compared to Synchro. Networks that are larger and more complicated, in other words having multiple major traffic inputs, such as Montrose Rd. network, are likely to benefit more from the proposed offset optimization method because by considering the demand in all the existing links and traffic approaches, this algorithm can also improve network congestion in a time-efficient manner. For example, in the Montrose Rd. network, the offset optimization algorithm reduces the delay and number of stops by about 30% in some traffic profiles.

In section 4.5, I extended the original offset optimization algorithm by eliminating the assumption of infinite storage capacities for links. I added storage capacity constraints to the optimization problem, so the maximum vehicle accumulation in the links cannot pass the storage capacity of the links. The ultimate goal is to prevent spill-back in the network by controlling the maximum vehicle accumulation in critical links. In order to achieve this, I need to improve the current model in a way that it will estimate a queue length value. One approach for improving

the current algorithm in future, is to use multiple sinusoidal waves to model arrival and departure instead of using a single wave.

Chapter 5. Estimating Activation Intervals for Signal Plans

5.1 Introduction

Having the most efficient control parameter is only valuable if we apply it at the right time and place. Therefore, the fifth step of the flow-chart from Figure 1.1 is to estimate the suitable activation interval for control parameters. Existing responsive systems address the problem of responding to actual traffic condition by applying appropriate switching time between control plans [43].

There are different methods for estimating the switching time, such as using time of the day or the condition of the traffic. I specifically use the traffic profile of the intersection to estimate the most appropriate switching time. In order to do that I tested different machine learning methods and eventually picked K-means clustering to group the data points with the most similar traffic conditions together.

There are other parameters such as saturation flow rate and volume capacity ratio that could be used to estimate the switching times but traffic volumes are the most accessible information and it is available at all time.

Section 5.2 describes the switching time algorithms and next I test the algorithm on two real-world intersections and show the result in Section 5.3. Lastly, Section 5.4 presents a short conclusion.

5.2 Algorithm Description

K-means clustering method partitions observation points into k clusters in which each observation belongs to the cluster with nearest mean. The objective is to minimize the Euclidean distance of every item in each cluster from the center of the cluster, as it is shown in the following formula

$$\underset{S}{argmin} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||. \tag{5.1}$$

Where k is the number of clusters and μ_i is the center of the i^{th} cluster. Also, rows of S correspond to points and columns correspond to variables. Hence, every S_i is a subset of the set S, $S_i \in S$, but the intersection of the all the subsets is empty, $S_1 \cap S_2 \cap S_3 \ldots \cap S_k$, where $S_1 \cup S_2 \cup S_3 \ldots \cup S_k = S$. Point x presents a row from matrix S and includes the following information

$$S = \begin{bmatrix} \beta * t_1 & V_{1,1} & \cdots & V_{1,n} \\ \vdots & & \ddots & \vdots \\ \beta * t_\tau & V_{\tau,n} & \cdots & V_{\tau,n} \end{bmatrix} \tag{5.2}$$

There are τ rows in the S matrix and each row represents a time slot, which can vary from a single second to one hour or even several hours. Time slots are equal and their length depend on

two factors, first the resolution of the data, second the desired accuracy for the switching time. For example, some systems may only accept the whole hour values as the switching time, therefore the length of each slot is 1 hour and τ is equal to 24. Since data points are sorted by time, clusters include points that are in the same time range. However, to capture the increasing order of the time for all cases, I added an extra coefficient value, β, next to the time, which let me to increase the impact of the time by having higher coefficient. The default value for the coefficient β is 1 but it can go as high as necessary, until all the clusters include points in same time interval.

The first column of the S matrix is time, and the rest are the traffic volumes, V. Number of the columns that correspond to the traffic volume depends on the dataset as well. For example, if the data is available for every approach in each leg then there is one column for each approach, but if we only have the traffic volume per leg, then the number of the column is equal to the number of the legs. Moreover, for every day there is one set of columns, so for the cases that data is available for several days or months, there will be additional columns for every day. However, to estimate switching time for a specific day in the future, it is reasonable to use days with similar characteristic. For instead, if I want to estimate the switching time for a Monday, my S matrix only includes the previous weekdays (or just previous Mondays). In conclusion, each row includes the information about the traffic volume for a specific time of the day in all the approaches and for all the available dates.

After estimating the S matrix and assigning a value to k, I can solve the optimization problem in equation 5.1. As a result, I have k clusters and the index number of the members inside each cluster. Next, to find the switching time, I search for the times that cluster numbers change, for example if up to 7AM data points are in cluster number 1 but after 7AM data points are in cluster number 2, then 7AM is the switching time. The switching time is when the condition of the traffic changes and the difference between the traffic volume before and after switching time is significant.

For the case studies, I use the K-means clustering tool in MATLAB to group the data points and find the index of the member of each cluster.

5.3 Algorithm Evaluation

In this section I show the result of using the switching time algorithm on two real world intersections. The first intersection is Tildenwood and Montrose Rd. located in Montgomery County in Maryland and it is shown in Figure 5.1. The same figure shows the leg number as well. For this intersection, traffic volume per leg is available for July 17-30, 2016 and according to the control system information from the city, there are 5 control plans per day and switching time values should be in hour only.

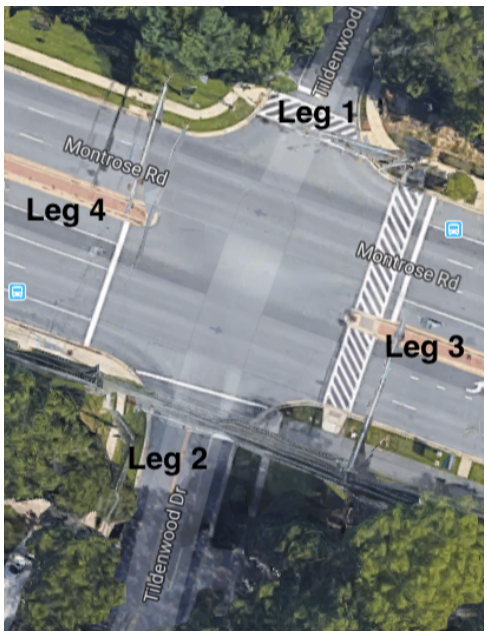


Figure 5. 1 Picture of Tildenwood and Montrose Rd. intersection in Maryland

As a result, I formulate the *S* matrix as follows

$$S = \begin{bmatrix} t_1 & \cdots & V_{l=m,d=n,t=1} \\ \vdots & \ddots & \vdots \\ t_{24} & \cdots & V_{l=m,d=n,t=24} \end{bmatrix}, m = 1,2,3,4, \quad n = 1,2,3,4,5. \tag{5.2}$$

where I use the first week of the data, July 17-23, 2016, to construct the *S* matrix and estimate the switching time for Tuesday July 26th. In equation 5.2, *m* represents the leg number and there are 4 legs. Also, *n* is the number of the days, which is five here because I use all the weekdays from the first week, July 18-22. As a result, I have the *S* matrix and number of cluster is equal to 5, $k = 5$, because the control system accept maximum of 5 control plans per day. Now I estimate the switching time by using the algorithm developed in section 5.2.

There are four curves in Figure 5.2 and each curve represents the traffic volume of a leg in the Tildenwood and Montrose Rd. Intersection during July 26th. In addition, on the same figure, there are black vertical lines that divide the day into 5 portions. These black lines represent the estimated switching times from the algorithm. As it is shown on the figure leg 3 and 4 have the highest traffic volume and the most contribution in the traffic demand of the intersection, accordingly, the black line are located at time when there are the most changes in the traffic behavior.

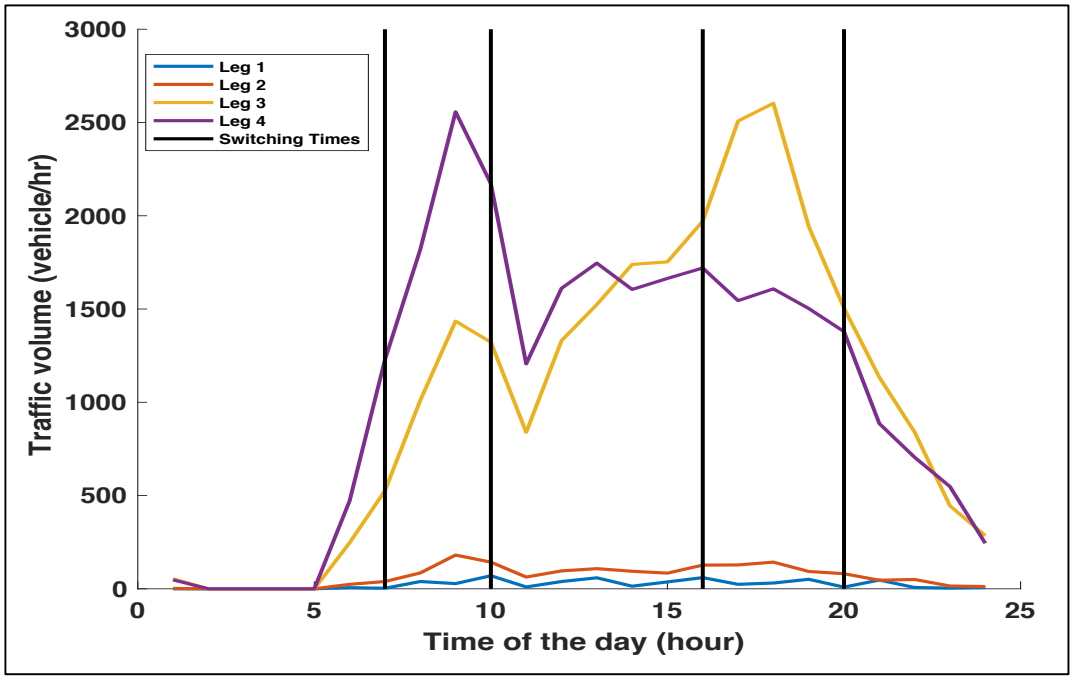


Figure 5. 2 Tildenwood and Montrose Rd. intersection traffic volume per leg from July 26th and estimated switching time values

The second case study is Ribaurt Rd. and Lady's Island Drive Intersection in Beaufort, SC from Figure 3.1 in Chapter 3, which the traffic volume for every 5 minutes per leg is available for Wednesday February 18, 2015 from 6:00AM to 7:00PM. In this intersection from 6:00AM to 7:00PM there are 3 different control plans (AM peak, Mid-day, PM-peak). However, the total traffic volume in Figure 5.3 shows that there is a time the in mid-day interval that has a total volume as high as AM-peak or PM-peak but the current control plan does not consider the high demand at the middle of the day.

According to my studies the high demand from 12PM to 1PM is due to the lunch break for the hospital and school close to that intersection and it happens every day. As a result, I construct the *S* matrix by using the same day traffic volume for every leg and assigned 5-minute interval for time slots, also have *k* equal to 4 (instead of 3, which is the default value) to run the algorithm. Note that, since the time slot length is 5 minutes, the switching times have higher accuracy and resolution compared to the first case study.

The y-axes on Figure 5.3 shows the total traffic volume of all the legs in the intersection for every 5 minutes and the x-axes it the time from 6AM to 7PM. On the same figure the black lines show the estimated switching time between control plans. Similar to the first case study, the switching times clearly presents the change in the traffic condition.

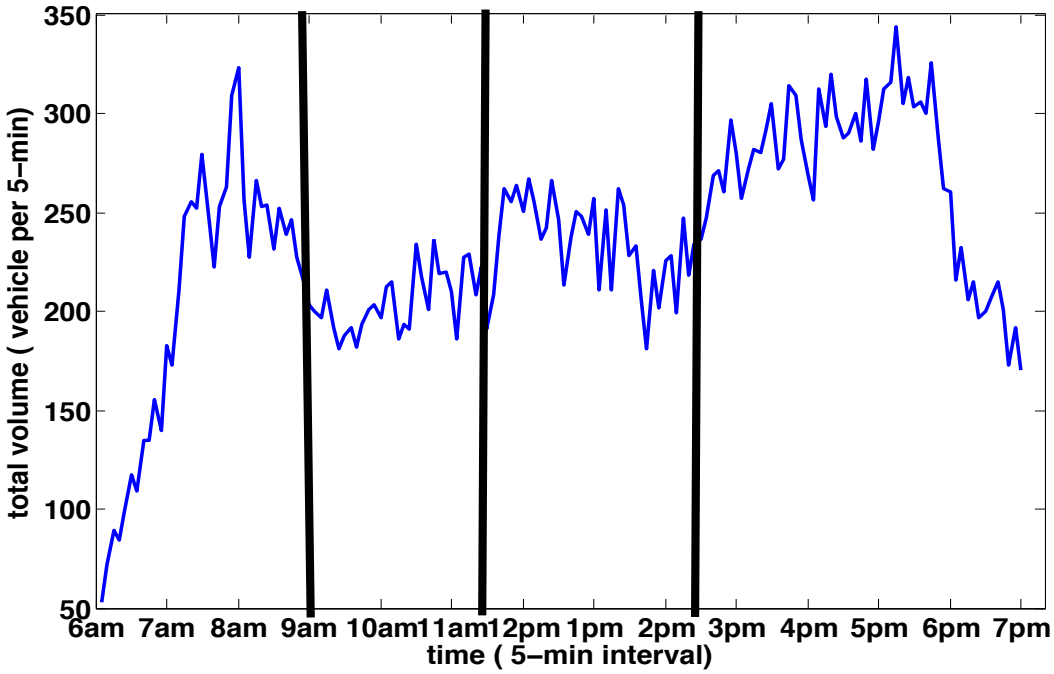


Figure 5. 3 Total traffic volume in Ribaurt Rd. and Lady's Island Drive Intersection with estimated switching time for control plans

Chapter 6. Conclusions

6.1 Summary of Conclusions

Studying the traffic condition and existing signal control strategies in urban transportation networks indicate that gaps exist in the areas of performance measurement, analysis tools, and control strategies. Adaptive control system was introduced to reduce these gaps; however, the high cost and complexity make it hard to implement such systems. On the other hand, robust pre-timed signal plan is an alternative to the adaptive control system, which is cheaper, and easier to implement. Also, recent advances in traffic data collection technology at intersections make it possible to design efficient robust timing plans that satisfy demand fluctuation.

In this dissertation, I studied a six-step process for designing and implementing robust pre-timed signal plans. This six-step process included, collect data, filter data, evaluate the control system, update the control parameters, design activation interval for new parameters, and implementation. Moreover, I presented my contribution on each one of the steps separately. For the first step, I used data on vehicle arrivals and departures plus the signal status at real-world intersections. The data included information on turning movement counts at various time resolutions and saturation flows, also it provided performance measures such as lost times, wasted green times, proportion of vehicles arriving on green, and delays. Next, I developed a data-filtering algorithm to reduce the systematic error in detector data up to 25%. Furthermore, analysis of the data provided an extensive set of metrics that used for the development and assessment of signal settings at signalized intersections. In addition, to evaluate the performance of the control system in the test intersection, an algorithm was developed to estimate the vehicle accumulation in each approach at any given time.

After conducting a complete evaluation on the performance of the control system, it was time to improve the weaknesses and update control parameters. At this step, I expanded my view from single intersection to network of intersections and focused on estimating offset values, because offsets constitute the main parameter for coordinated traffic movement among multiple traffic signals. I introduced an extended version of an offset optimization algorithm that optimizes offset by minimizing the vehicle accumulation in all the links. Also, I compared the performance of this algorithm with Synchro's offset optimization tool and used VISSIM to simulate changes in traffic condition in real-world networks. Offset optimization algorithm reduced the delay and number of stops up to 30% in some scenarios and in overall showed a better performance than Synchro's offset optimization tool. Lastly, a clustering algorithm was introduced to determine the appropriate switching time between signal plans during the day.

The main advantage of data-driven approaches over existing method is their flexibility and fast reaction time. Existing control systems and signal plans get revised every several years but with data-driven approaches, control systems can automatically update their control parameters rapidly and satisfy the demand growth and traffic volume fluctuations. In conclusion, data-driven approaches will have the main role in improving the traffic condition in future and they will fill out the gaps in performance measurements and control strategies.

6.2 Future Work

Data-driven approaches that I introduced in this dissertation were tested in simulator or on real-world database and showed promising results. Even that simulators provide environments very similar to the real world but still a real-world field test is necessary to estimate the actual benefits of the algorithms and identify problems that are not clear in the simulation. Since the data collection technology already exist in the studied networks, the field test of the algorithms would be relatively easy and it will not require any change of the infrastructures.

Considering all the works that have been done so far on developing robust pre-timed signal plans, there are still a lot to be done in this field. The six-step process that was introduced in this dissertation is a general framework for designing robust pre-timed signal plans and each step could be extended in the future. More importantly, most of the studies that have been done so far are not easily applicable to the real world and might work only under unrealistic assumptions. The algorithms that I have developed showed improvement on real world networks in simulation and they are also applicable in real world, due to their simplicity, low cost, and easy implementation process.

Lastly, algorithms that design control parameters can improve the safety of the intersection as well. For example, preventing spill-back at intersections by designing appropriate control parameters (e.g. green interval, offset value, and cycle time) will reduce the scenarios that vehicles block the intersection, which is an unsafe situation. Data-driven algorithms are practical methods that can address safety and efficiently in urban transportation networks.

References

1. National Traffic Operations Coalition. 2012 National Traffic Report Card, (2012). *http://www. ite.org/reportcard/*

2. Dowling, R., & Ashiabor, S. (2012). *Traffic signal analysis with varying demands and capacities, draft final report*. Tech. Rep. NCHRP 03-97, Transportation Research Board.

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

4. Day, C. M., Bullock, D. M., Li, H., Remias, S. M., Hainen, A. M., Freije, R. S., & Brennan, T. M. (2014). Performance measures for traffic signal systems: An outcome-oriented approach. Technical Report, Purdue University, Lafayette, IN. http://dx.doi.org/10.5703/1288284315333.

5. FHWA. "Measures of Effectiveness and Validation Guidance for Adaptive Signal Control Technologies," *http://www.ops.fhwa.dot.gov/publications/fhwahop13031/index.htm*

6. Yin, Y. (2008). Robust optimal traffic signal timing. *Transportation Research Part B: Methodological*, *42*(10), 911-924.

7. Skabardonis A. (2013) "Field Testing the Effectiveness of Adaptive Signal Control for Arterial Signal Management," PATH Research Report UCB-ITS-2014-3.

8. Muralidharan, A., Flores, C., & Varaiya, P. (2014). High-resolution sensing of urban traffic. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on* (pp. 780-785).

9. Argote-Cabañero, J., Christofa, E., & Skabardonis, A. (2015). Connected vehicle penetration rate for estimation of arterial measures of effectiveness. *Transportation Research Part C: Emerging Technologies*, *60*, 298-312.

10. Su, D., Lu, X. Y., Horowitz, R., & Wang, Z. (2014). Integrating Freeway Ramp Metering and Intersection Signal Control. *paper presented at the 93rd TRB Annual Meeting, Washington DC.*

11. Haoui, A., Kavaler, R., & Varaiya, P. (2008). Wireless magnetic sensors for traffic surveillance. *Transportation Research Part C: Emerging Technologies*, *16*(3), 294-306.

12. Muralidharan, A., Flores, C., & Varaiya, P. (2014, October). High-resolution sensing of urban traffic. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on* (pp. 780-785).

13. Krogmeier, J., Sinha, K., Fitz, M., Peeta, S., & Nof, S. (1996). Borman expressway ATMs equipment evaluation. *Joint Transportation Research Program*, 32.

14. Hughes Transportation Management Systems. (1996). Borman Expressway Advanced Traffic Management System (ATMS) Phase I Final Report. Prepared for Indiana Department of Transportation, Indianapolis.

15. Coifman, B. (1999). Using dual loop speed traps to identify detector errors. *Transportation Research Record: Journal of the Transportation Research Board*, (1683), 47-58.

16. Jacobson, L. N., Nihan, N. L., & Bender, J. D. (1990). *Detecting erroneous loop detector data in a freeway traffic management system* (No. 1287).

17. Chen, C., Kwon, J., Rice, J., Skabardonis, A., & Varaiya, P. (2003). Detecting errors and imputing missing data for single-loop surveillance systems. *Transportation Research Record: Journal of the Transportation Research Board*, (1855), 160-167.

18. Peeta, S., & Anastassopoulos, I. (2002). Automatic real-time detection and correction of erroneous detector data with fourier transforms for online traffic control architectures. *Transportation Research Record: Journal of the Transportation Research Board*, (1811), 1-11.

19. Anastassopoulos, I. (2000). Fault Tolerance and Incident Detection Using Fourier Transforms. *Purdue University, Westlafayette*.

20. Peeta, S., & Das, D. (1998). Continuous learning framework for freeway incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, (1644), 124-131.

21. Smith, B. L. (1995). Forecasting Freeway Traffic Flow for Intelligent Transportation Systems Applications. *Ph.D. dissertation. Department of Civil Engineering, University of Virginia, Charlottesville*.

22. Transportation Research Board. (2010) Highway Capacity Manual," National Academy of Sciences. Washington DC.

23. Newell, Gordon F. (1993) A simplified theory of kinematic waves in highway traffic, part I: General theory."*Transportation Research Part B: Methodological* 27.4 (1993): 281-287.

24. Newell, Gordon F. (1993) A simplified theory of kinematic waves in highway traffic, part II: Queueing at freeway bottlenecks." *Transportation Research Part B: Methodological* 27.4 : 289-303.

25. Lawson, T., Lovell, D., & Daganzo, C. (1997). Using input-output diagram to determine spatial and temporal extents of a queue upstream of a bottleneck. *Transportation Research Record: Journal of the Transportation Research Board*, (1572), 140-147.

26. Su, D., Lu, X. Y., Horowitz, R., & Wang, Z. (2014). Coordinated ramp metering and intersection signal control. *International Journal of Transportation Science and Technology*, *3*(2), 179-192.

27. Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, *36*, 177-195.

28. Kalathil, D., & Varaiya, P. (2017). Estimating Residual Phase Duration for SPaT Messages. *arXiv preprint arXiv:1710.05394*.

29. Little, J. D., Kelson, M. D., & Gartner, N. H. (1981). MAXBAND: A versatile program for setting signals on arteries and triangular networks.

30. Gomes, G. (2015). Bandwidth maximization using vehicle arrival functions. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 1977-1988.

31. Husch, D., Albeck, J. (2001) Trafficware Synchro 5.0 User Guide for Windows.

32. Robertson, D. I. (1969). TRANSYT: a traffic network study tool.

33. Robertson, D. I., & Bretherton, R. D. (1991). Optimizing networks of traffic signals in real time-the SCOOT method. *IEEE Transactions on vehicular technology*, *40*(1), 11-15.

34. Hu, H., & Liu, H. X. (2013). Arterial offset optimization using archived high-resolution traffic signal data. *Transportation Research Part C: Emerging Technologies*, *37*, 131-144.

35. Christofa, E., Argote, J., & Skabardonis, A. (2013). Arterial queue spillback detection and signal control based on connected vehicle technology. *Transportation Research Record: Journal of the Transportation Research Board*, (2356), 61-70.

36. Gentile, G., Meschini, L., & Papola, N. (2007). Spillback congestion in dynamic traffic assignment: a macroscopic flow model with time-varying bottlenecks. *Transportation Research Part B: Methodological*, *41*(10), 1114-1138.

37. Abu-Lebdeh, G., & Benekohal, R. (2000). Genetic algorithms for traffic signal control and queue management of oversaturated two-way arterials. *Transportation Research Record: Journal of the Transportation Research Board*, (1727), 61-67.

38. Daganzo, C. F., Lehe, L. J., & Argote-Cabanero, J. (2017). Adaptive offsets for signalized streets. *Transportation Research Part B: Methodological*.

39. Coogan, S., Kim, E., Gomes, G., Arcak, M., & Varaiya, P. (2017). Offset optimization in signalized traffic networks via semidefinite relaxation. *Transportation Research Part B: Methodological*, *100*, 82-92.

40. Kim, E. S., Wu, C. J., Horowitz, R., & Arcak, M. (2017, May). Offset optimization of signalized intersections via the Burer-Monteiro method. In *American Control Conference (ACC), 2017* (pp. 3554-3559). IEEE.

41. Muralidharan, A., Pedarsani, R., & Varaiya, P. (2015). Analysis of fixed-time control. *Transportation Research Part B: Methodological*, *73*, 81-90.

42. VISSIM PTV (2011) Vissim 5.40 user manual," *Karlsruhe, Germany*.

43. Mirchandani, P., & Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, *9*(6), 415-432.

Appendix A: Algorithm Implementation

In the traffic engineering community, applications programming and the development of COM application are regularly used to enhance or extend VISSIM (and other traffic) model capabilities, providing features or simulation abilities that cannot be otherwise accomplished. Typical uses of VISSIM-COM apps are dynamically rerouting or diverting traffic during major freeway incidents; dynamically changing driver/vehicle desired speed parameters as a response to speed reductions posted on changeable message signs, and implementing complex coordinated and demand responsive signal control strategies.

The VISSIM software has a COM interface as one of its built-in features. During a VISSIM model run, VISSIM's COM interface provides the ability to communicate with other apps that are also running. In order for VISSIM to communicate with other software applications, these other software apps need to have a built-in COM interface as well. These COM communications between VISSIM and MATLAB can provide the means to pass data, model runtime status or model parameters between the two communicating apps. By means of COM programming, MATLBA can also control or trigger events within another app. For example, MATLAB can invoke VISSIM, tell the VISSIM app to load a network, set model parameters, start a simulation model run, pause the simulation model run, retrieve and change VISSIM model parameters, and generally control the VISSIM model run.

COM programming can be used to detect VISSIM events and/or to change many of the model's aspects, parameters and/or conditions during VISISM simulation model runs. A typical COM program sequence of events would be:

- "Pause" the VISSIM simulation model:
    - Once or multiple times during the model run
    - At predetermined intervals or preselected times
    - Upon detection of specific conditions or events
- "Retrieve" data and information on VISSIM model runtime status:
    - Current signal controller values
    - Traffic volumes
    - Vehicle detector occupancy rates
    - Transit boarding/alighting counts
- "Set" or change VISSIM model parameters and/or status:
    - Signal controller offsets and/or splits
    - Ramp metering rates
    - Vehicle type and driver behavior model parameters
    - Vehicle route choice factors
    - Traffic toll collection booth operational parameters
- "Initiate" or "Terminate" events:
    - Freeway collision
    - Lane closures
- "Continue" the VISSIM simulation model

For my work, a MATLAB-VISSIM COM was built to:

1. Initiate the VISSIM model run

2. Periodically pause the VISSIM simulation model run to get traffic volumes and other performance measures from VISSIM
3. Change the traffic profiles or/and control parameters
4. Continue the model run

As I mentioned, I used this interface in Chapter 4 for testing the offset optimization algorithm and Figure A.1 shows the flow of information between the MATLAB, VISSIM and the Offset Optimization Module. The COM program and the Offset Optimization Module were developed in MATLAB. This means that, at runtime, MATLAB controls the VISSIM model and controls or calls the Offset Optimization Module. MATLAB initiates the VISSIM model run; it initiates and controls the VISSIM→MATLAB data transmissions (the data retrievals), the MATLAB→VISSIM data transmissions (the data pushes); and it invokes or calls the Offset Optimization Module (and controls parameter passing between MATLAB and the Offset Optimization Module).

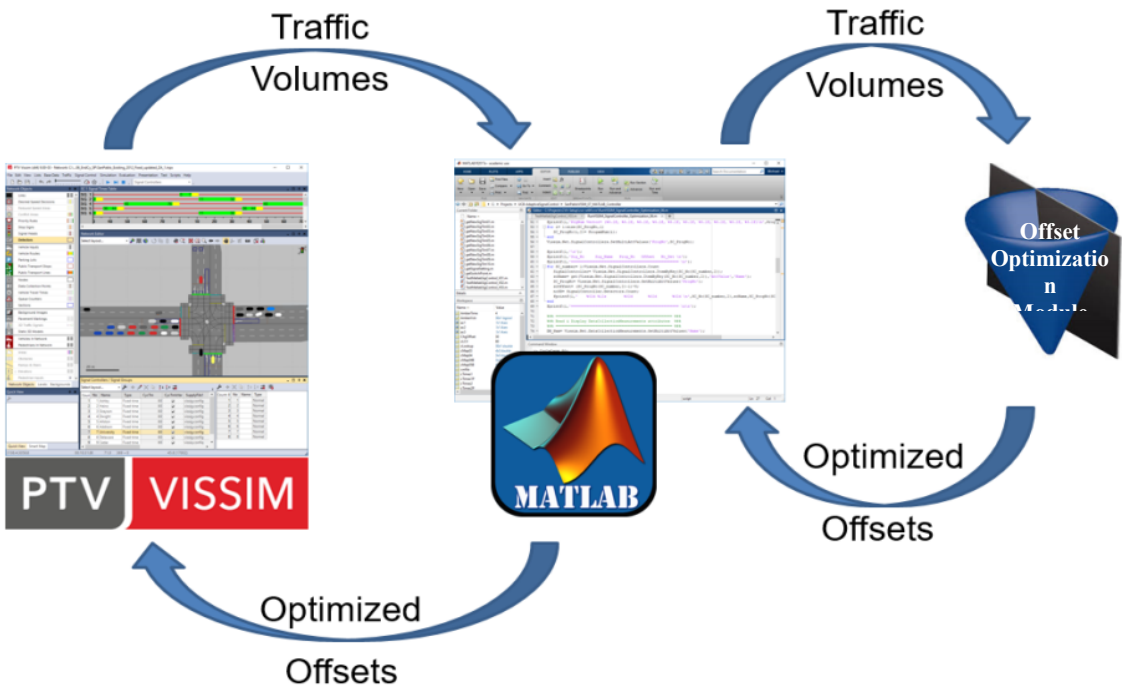


Figure A.1 Flow of information during MATLAB-VISSIM "COM" Controlled Model Run

Figure A.2 shows the iterative COM program's sequence of events. COM signal controllers like this one tend to be circular or iterative in nature, with their sequence of events being repeated with every signal cycle, or once every 5 cycles, or once every 10 (or so) cycles. It is up to the

VISSIM modeler to determine how often the signal timing parameters (e.g., offsets and splits) should be updated.

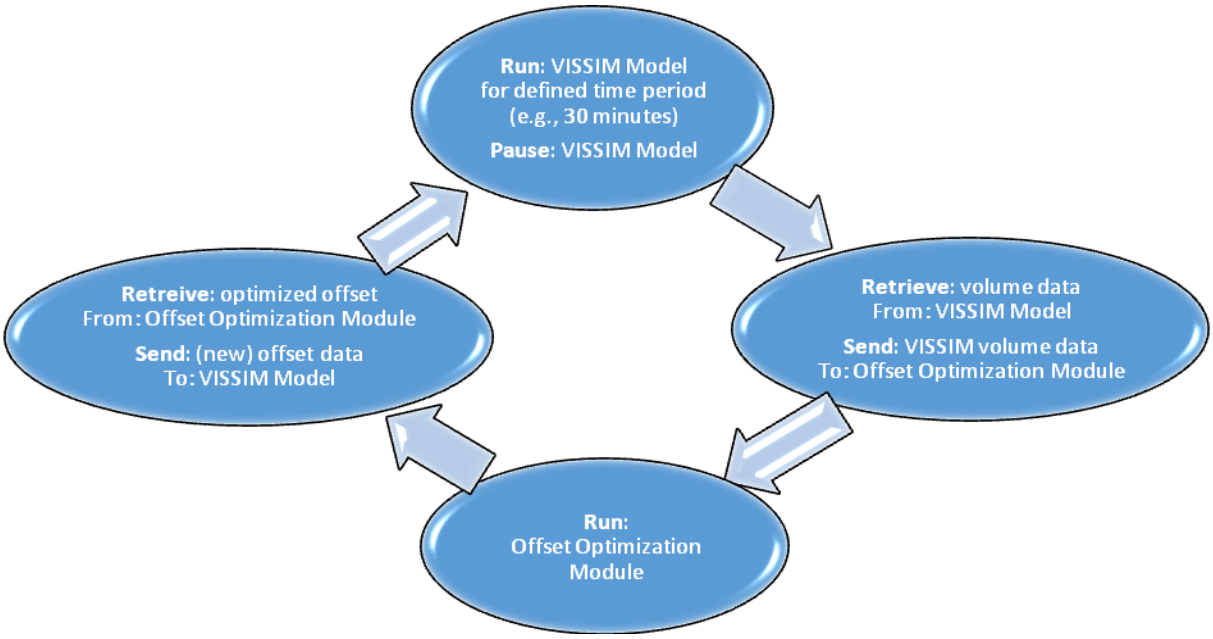


Figure A.2 Control of VISSIM Model and Offset Optimization Module

As a result, the performance of the algorithm was evaluated in the simulator. Such result helps the transportation engineer to have a more realistic understanding about the algorithm and performance of the algorithm in the real-world.