5-2010

Flow-based Adaptive Split Signal Control

Airton G. Kohls
University of Tennessee - Knoxville, akohls@utk.edu

Recommended Citation

Kohls, Airton G., "Flow-based Adaptive Split Signal Control. " PhD diss., University of Tennessee, 2010.
https://trace.tennessee.edu/utk_graddiss/716

To the Graduate Council:

I am submitting herewith a dissertation written by Airton G. Kohls entitled "Flow-based Adaptive Split Signal Control." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Thomas Urbanik, Major Professor

We have read this dissertation and recommend its acceptance:

Lee Han, Christopher Cherry, Itamar Arel

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Airton Gustavo Kohls entitled "Flow-based Adaptive Split Signal Control." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Thomas Urbanik II, Major Professor

We have read this dissertation
and recommend its acceptance:

Lee D. Han

Christopher Cherry

Itamar Arel

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Flow-based Adaptive Split Signal Control

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Airton Gustavo Kohls
May 2010

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge my academic advisor, Dr. Thomas Urbanik II, for all his support towards my education. His paramount knowledge and recognized experience in traffic signal operations provided me unique guidance. I would like to particularly thank Dr. Lee Han, Dr. Christopher Cherry and Dr. Itamar Arel for serving on my doctoral committee. Two gentlemen were equally important in my academic life; thank you Dr. Frederick Wegmann and Dr. Arun Chatterjee.

Mr. Jerry Parker and Mrs. Bernice Parker are really special people. Long ago, they opened their home and welcomed me as their true son…and they continue to do so. I will always be grateful for their Volunteer spirit.

No words can describe the everlasting support I received from my parents, Adelmo and Cenilda. They were determined to provide better education for me and for my sister. They repeatedly told us that our education would be their legacy. I will always honor their decision. I will always cherish their ability to bond our family with love and integrity.

I am sincerely thankful for the unconditional help my sister and her family provided me during this journey. Thank you Marcia, Cenk, Aidan and Lara for your kindness and for the great times in Texas.

My amazing wife Caroline,…thank you for everything you do for me. My accomplishments are yours as well. Your devotion to our family is the foundation of our future. My son Nicholas,… he showed me everyday how powerful, replenishing and sacred a smile can be.

Finally, I would like to acknowledge the Tennessee Section of the Institute of Transportation Engineers for conferring me their scholarship for the past three years. Lastly, I need to acknowledge the game of golf where I developed patience and discipline to translate today into a better tomorrow.

ABSTRACT

Over the last 35 years many adaptive traffic signal control systems have been developed presenting alternative strategies to improve traffic signal operations. However, less than 1% of all traffic signals in the United States are controlled by adaptive systems today. The extensive infrastructure necessary including reliable communication and complex calibration leads to a time consuming and costly process. In addition, the most recent National Traffic Signal Report Card indicated an overall grade of D for the nation's traffic signal control and operations. Recent economic adversity adds to the already difficult task of proactively managing aged signal timing plans.

Therefore, in an attempt to escape the *status quo*, a flow based adaptive split signal control model is presented, having the principal objective of updating the split table based solely on real-time traffic conditions and without disrupting coordination. Considering the available typical traffic signal control infrastructure in cities today, a non centralized system is proposed, directed to the improvement of National Electrical Manufacturers Association (NEMA) based systems that are compliant with the National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) standards. The approach encompasses the User Datagram Protocol (UDP) for system communication allowing an external agent to gather flow information directly from a traffic signal controller detector status and use it to better allocation of phase splits.

The flow based adaptive split signal control was not able to consistently yield significant lower average vehicle delay than a full actuated signal controller when evaluated on an intersection operating a coordinated timing plan. However, the research proposes the ability of an external agent to seamless control a traffic signal controller using real-time data, suggesting the encouraging results of this research can be improved upon.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The effects of coordination timing plans at an individual intersection operating in unexpected traffic conditions can potentially produce unwanted and unnecessary delays. While vehicle actuated phases in a coordinated-actuated signal control system can partially address fluctuations in flow, an adaptive signal control system uses real-time detection data to recognize fluctuations in demand and update timing parameters that will potentially benefit the performance of traffic. The problem lies in the extensive infrastructure necessary for the implementation of adaptive signal control systems along with the reliable communication and complex calibration needs. In addition, the added capability of adaptive signal control does not always ensure responsiveness, due to predictive modeling, calibration maintenance, frequency of updates and hardware limitations.

Independent of what control system is being used, determining adequate split times (the time assigned to a phase during a cycle) can be challenging. If a split time is too long, other approaches may experience increased delays, while if a split time is too short, the demand may not be served. Time of day (TOD) scheduling attempts to address the recurrent variability of traffic but no account is taken of the fact that the cycle by cycle stochastic behavior is significant and unnecessary delay may be produced. Moreover, in coordinated systems, the constraints imposed by the traffic signal controller logic regulate unequally the ability of phases to reallocate unused time during a cycle, potentially producing unnecessary delay as well.

Most metropolitan areas do not have the resources to re-time their signals regularly. The Intelligent Transportation System (ITS) deployment tracking database shows that few areas re-time their signals each year. In fact, the Institute of Transportation Engineers (ITE) estimates that nearly 75 percent of all signals in the United States need to be re-timed. It has also been estimated that traffic experiences an additional 3% to 5% delay per year as a consequence of not retiming signals as conditions evolve over time [1].

Therefore, the motivation for this research is to provide a less complex system with adaptive split capability as a means to adjust to both changing patterns over time and more importantly, the stochastic nature of traffic. The traffic signal controller constraints on reallocation of time among phases will be relaxed and split updates would occur every cycle if necessary, based on real-time flow data provided by stop bar presence detection without disruption of coordination caused by changing timing plans. The model free algorithm follows NTCIP (National Transportation Communications for Intelligent Transportation System Protocol) standard and is NEMA (National Electrical Manufacturers Association) standard compatible. The main idea of the research is to determine whether the use of the proposed flow based adaptive split system yields better performance than a full actuated coordinated system operating in an intersection with a coordinated timing plan. The hypothesis states that *control logic* lost time can be reallocated between phases in a manner to reduce average delay in the intersection. The research is structured to focus on cycle by cycle fluctuations that do not exceed the overall capacity of the intersection.

The dissertation is developed in 6 Chapters. Following the introduction, Chapter 2 presents the literature review and is followed by Chapter 3 documenting the experimental system. Chapter 4 develops the proposed flow-based adaptive split algorithm and Chapter 5 presents and discusses the results. Finally Chapter 6 concludes the research with recommendations on future work.

CHAPTER 2

LITERATURE REVIEW

According to the NTCIP (National Transportation Communications for Intelligent Transportation System Protocol) standard, split is the time in the cycle allocated to each phase. More specifically it is the summation of the green and yellow time plus the red clearance time (commonly known as all-red) or the summation of the green time plus the pedestrian walk and clearance times, whichever is greater. For this research, pedestrian times will not be included in green time calculations to allow the greatest flexibility in split adjustment.

Coordination control strategies determine how splits are allocated and how demand is serviced. In the 1970's the Federal Highway Administration put forward a program called Urban Traffic Control Systems (UTCS) as part of a research project that aimed in developing and testing a variety of advanced control concepts and strategies [3]. The control strategies in the UTCS project are categorized into three generations. The First Generation Control (1-GC) uses offline calculation of signal timings in contrast to the online calculations presented on the Second and Third Control Generations. Adaptive control coordination represents the latter two control generations differentiating themselves mainly by the method of predicting traffic and the period after which timing plans are revised. The Second Generation Control (2-GC) predicts traffic from historic information and revises plans every 10 minutes while the Third Generation Control (3-GC) predicts traffic with smoothed current data and revises plans every 3 to 5 minutes.

The following is a description of the functionality of splits for different control strategies in traditional (non UTCS) control systems:

In fixed-time coordination, total split time is given to each movement every cycle regardless of changes in traffic conditions. Different coordination plans, including cycle, split and offset are calculated from historical data and implemented according to a time-of-day (TOD) schedule that identifies the time periods when the plans will be in operation. Fixed-time coordination is appropriate for areas where traffic demand is very predictable. Detection is not necessary.

In semi-actuated coordination, detection is provided to the non-coordinated phases (minor streets) and if necessary to pedestrian phases. The coordinated phase(s) will run its allotted split time every cycle, regardless of demand. The splits for the non-coordinated phases will only be served upon request during the permissive periods. Permissive period is a time period during which the controller unit is allowed to leave the coordinated phase(s) to go to other phases. Once served, the non-coordinated phases will time a pre-determined minimum green. After that, the green time can be extended according to demand up to a limit (dependant on force-off mode). Unused time during the cycle will ultimately be given to the "front" of the coordinated phase(s) causing what is known as "early return to green". Different pre-determined coordination plans are implemented according to a time-of-day (TOD) schedule. Semi-actuated operation is best suited for locations with low volume minor street traffic.

In full-actuated coordination, detection is provided to all phases and pedestrian phases. Non-coordinated phases will run their splits just like in semi-actuated coordination, giving any unused time during the cycle to the "front" of the coordinated phase(s). Coordinated phase(s), on the other hand, will benefit from an "actuated permissive period" at the "back" of the split green, when detection is monitored and green time is extended if demand exists or, the coordinated phase(s) terminates and unused green time is available to serviceable non-coordinated phases. Different pre-determined coordination plans are implemented according to a time-of-day (TOD) schedule. Full-actuated coordination is appropriate for intersections with less predictable volume on all approaches.

In addition to the capabilities presented in the above control strategies, traffic-responsive coordination monitors data from traffic detectors and, instead of time-of-day (TOD) scheduling, different pre-determined coordination plans are automatically selected to best suit current conditions. Usually volume and/or occupancy data is processed to calculate parameters that are compared to thresholds. Expertise is necessary to determine a set of fine tuned plans and thresholds to accommodate everyday traffic as well as benefit from unusual traffic occurrences such as incidents, extreme weather, sporting events, construction, etc. Traffic responsive plan selection

(TRPS) only selects a timing plan to operate and does not make changes to the splits or any parameters, for that matter, specified in the timing plan.

Adaptive control coordination consists of a higher level of control than traffic-responsive coordination because the real-time data collected through detection is not used to match current conditions to an existing plan, but rather an optimal timing plan is computed to accommodate fluctuations in demand. Areas with high rate of growth would potentially benefit from adaptive control coordination because timing plans would not need to be updated frequently.

2.1 Adaptive Traffic Control Systems

The main objective of adaptive traffic control systems is to readily recognize fluctuations in demand and implement the strategy that would best achieve the desired performance objective. The infrastructure necessary usually includes an extensive detection system to monitor traffic in real-time and gather accurate and comprehensive data. A reliable communications system is usually needed to collect and feed the data between intersections, regional and central computers. Due to calibration and the need of operational expertise, it is known that making adaptive control function properly is more of an art than science [4]. Most adaptive systems use traffic models to predict vehicular movements, estimate platoons arrivals, estimate queue size and evaluate alternative traffic control strategies. Adjustment of split, cycle and offset is determined by the evaluation of different performance metrics at individual intersections or system-wide. Typical performance metrics may be to minimize delay, to minimize stops, to minimize queue, to increase throughput and to maximize green band among others.

In 1963, Miller [5] described an algorithm for adjusting signal timings in small time intervals of 1 to 2 seconds. It was the beginning of the adaptive signal control concept. A decision to be made was whether to extend the current green duration or terminate it immediately. The algorithm calculated the difference in vehicle-seconds of delay between the gain made during an extension and the loss in the cross street resulting from that extension.

Rosdolsky [6] introduced a mathematical method for adaptive on-line signal program computation. The objective was to minimize number of stops. An algorithm was developed to advance green sufficiently to release the queue before the arrival of a platoon. The green time necessary for the platoon to pass the intersection was also calculated.

In the 1970´s, two of the most popular adaptive traffic control systems were developed: SCATS (Sydney Coordinated Adaptive Traffic System) [3, 7, 8, 9] and SCOOT (Split Cycle Offset Optimization Technique) [3, 10, 11, 12]. Both systems have evolved considerably. They have different operational philosophies with strengths and weaknesses. SCATS and SCOOT have to be back fitted to the NEMA (National Electrical Manufacturers Association) ring and barrier structure from the foreign stage based structure.

This research focuses on improving existing NEMA based systems and on adjustment of splits. Therefore, the review of the literature on adaptive systems is primarily related to how splits were calculated.

2.1.1 SCATS

Developed in Australia, SCATS is a reactive adaptive traffic control system. To adjust phase splits, SCATS uses a split plan library that is calculated by an off line computer program when the system is set up. Each plan determines the percentage of time that can be added to or reduced from each phase (up to 4% in each cycle). For every cycle, the degree of saturation is calculated for each plan using data from the last cycle. The plan that results in the best arrangement to maintain equal degree of saturation on critical approaches is selected. To calculate degree of saturation (the ratio of effectively used green time to the total available green time), SCATS uses stop bar detection to measure the space time between vehicles as they pass through each intersection. Unused green time is space time greater than or less than the daily calibrated standard space time at maximum flow. Minimum splits are user definable and maximum splits are limited by cycle length and minimum requirements of other phases. Split plan voting is carried out at the critical intersection. Minor intersections splits are controlled by the critical intersection. Phase splits can be biased to favor principle traffic movements when demand approaches saturation.

2.1.2 SCOOT

Developed in United Kingdom SCOOT is a proactive adaptive traffic control system. To adjust splits, SCOOT uses advance arrival information from upstream detectors to predict whether it is better to terminate the stage a few seconds earlier, a few seconds later, or as planned. The prediction takes place 5 seconds prior to each stage change. The split optimizer implements the decision that will minimize the maximum degree of saturation on all approaches to that intersection. Degree of saturation is “the ratio of the average flow to the maximum flow which can be passed through the intersection from the particular approach” [10]. For this calculation, account is taken of minimum safety timings, current estimates of queue lengths and of any congestion on the approaches to the junction. The SCOOT traffic model calculates the current degree of saturation at each signal stop bar. The “Temporary” changes of up to plus or minus 4 seconds are made to the green durations to take account of the cycle-by-cycle random variations in traffic flow. For such a temporary change, a smaller ‘permanent’ change of plus or minus 1 second is made to the stored values of green durations in the following cycle. SCOOT controls the exact green time of every phase on a traffic controller by sending “hold” and “force-off” commands to the controller. Split weighting can be used to favor principle traffic movements.

Garbacz [4] pointed out some weaknesses of the above adaptive traffic control systems. Since SCATS uses stop line detection it only knows what the demand was in the last cycle. Consequently, when a sudden, but short lived increase in traffic occurs in one approach, for example, SCATS is not able to increase the green time fast enough. By the time the green was increased for the approach the demand has dropped. With SCOOT, the system is trying to reduce queues, stops and delays in an entire signalized network. When one approach to an intersection is heavily saturated and the other approaches have light demand, the degree of saturation of the entire intersection is low causing the cycle length also to be low. When this happens, the cycle length may not be long enough to provide enough green time on the saturated approach to keep up with demand. By the same token, adaptive control is not always going to provide the progression everyone expects along an arterial. The author concluded that “adaptive control works best in demanding situations were minimal constraints are placed on the system’s ability to adjust signal timings”. He also noted that long pedestrian phases limit the ability of adaptive control to optimize signal operation.

Dey et al [13] suggested implementation results for SCATS being in the order of 6.6% to 32% reduction in travel time (average 7.8%); up to 28% reduction in delay; and up to 42% reduction in stops. The author also acknowledged results for SCOOT in the order of 8% reduction in travel time; 22% reduction in delay; and 17% reduction in stops.

More recently, the United States Department of Transportation (USDOT) and the Federal Highway Administration (FHWA) has encouraged the development and deployment of US adaptive signal systems including OPAC (Optimized Policies for Adaptive Control) [3, 14, 15, 16] RHODES (Real-Time Hierarchical Optimized Distributed and Effective System) [3, 17, 18, 19] and ACS-LITE (Adaptive Control Software – Lite) [3, 20, 21, 22].

2.1.3 OPAC

Developed at the University of Massachusetts, Lowell, OPAC is a proactive adaptive traffic control system. To adjust splits, OPAC calculates a flow profile for each approach over a user specified period of time (the rolling horizon concept). The beginning or head portion of the flow profile is obtained from upstream detectors counts. The tail part of the flow profile is predicted for near future using smoothed historical volume counts. A performance index of total intersection delay and stops for every possible signal switching pattern is then calculated. Dynamic programming techniques are used to minimize the performance measures. The signal switching combination with the best performance index (less delay and stops) is then considered to be the optimal solution. A decision is then made whether to terminate the current phase or extend it by one interval (1 or 2 seconds). The dynamic optimization process is carried out continuously to ensure that the signal control is always up-to-date. The duration of a phase is never pre-specified. It depends solely on the prevailing traffic flow conditions. For coordination, OPAC utilizes the Virtual Fixed Cycle (VFC) concept allowing the cycle time to start or terminate within a flexible range at each intersection. OPAC controls the exact green time of every phase on a traffic controller by sending “hold” and “force-off” commands to the controller.

2.1.4 RHODES

Developed at the University of Arizona, RHODES is a proactive adaptive traffic control system. At the highest level of its hierarchy (Dynamic Network Loading), RHODES captures characteristics of traffic and estimates the load on each particular link in vehicles per hour. At the Network Flow Control level, RHODES allocates approximate green time to each phase based on the load estimates. It is at the Intersection Control level that RHODES optimizes phase sequence and the actual phase start and end times using a dynamic programming algorithm based on the predicted vehicles arrivals from upstream detectors. A rolling horizon approach is used. A decision of allocating time to a phase has an associated value based on a performance measure such as total number of stops, maximum queue length and total delay. The value of the performance measure is determined by using the predicted vehicle arrivals, the current and prior decisions, and an imbedded traffic flow model that accounts for estimated queues, startup lost time, queue discharge and arrivals, as well as other traffic dynamics that relate the decision to the performance measure. A target phase evaluation order is provided to COP (Controlled Optimization of Phases).The dynamic program evaluates the value of the performance measure for each phase for the pre-determined rolling horizon time. A decision is then taken to determine the sequence of phases (if variable phase sequence is allowed) and phase durations that will result in the lowest value of the performance measure over the optimization horizon. RHODES continually re-solves its planned phase timings, every 5 seconds, to adapt to the most recent information. Stop-bar presence detection is necessary to control and adjust queue predictions. RHODES set the exact duration of each phase by sending “hold” and “force-off” commands to the controller.

2.1.5 ACS-LITE

ACS-LITE is a reactive adaptive traffic control system. To adjust phase splits, ACS-LITE uses stop-bar detection to collect occupancy data. The occupancy data is correlated to phase intervals and ACS-LITE develops phase utilization data for each phase determining how much of the available green time is being used. The phase utilization data is averaged for several cycles (usually 3 to 5 cycles). ACS-LITE then modifies the split parameters to adapt the performance of the system for oversaturated phases. The objective is to balance the degree of saturation on all approaches. Degree of saturation is the ratio of the averaged used green time to the averaged available green time. Coordinated phases can guarantee extra green time by addition of a bias to the algorithm. The split adjustments on ACS-LITE are executed on each controller independently. Each split optimization step occurs not earlier than the period necessary to time 3 cycles plus a minimum of 5 minutes. The system changes the splits and offsets only a small amount (2-5 seconds) [20]. It has also been deemed undesirable for ACS-LITE to remotely apply Hold, Omit, or Force-Off controls to each controller, because this option is not robust during intervals of unreliable communications [22]. The gap-out and force-off logic of the controller works normally with the updated parameters.

Other systems known to be operational but not widely used are ALLONS-D (Adaptive Limited-Look Ahead Optimization of Network Signals - Decentralized), ATSAC (Automated Traffic Surveillance and Control), IN-SYNC Traffic Adaptive System, ITACA (Intelligent Traffic Adaptive Control Area), MOTION (Method for the Optimization of Traffic Signals in On-line controlled Networks), MOVA (Microprocessor Optimized Vehicle Actuation), PRODYN, RTACL (Real-Time Traffic Adaptive Control Logic), SPOT / UTOPIA (System for Priority and Optimization of Traffic / Urban Traffic Optimization by Integrated Automation) and TACOS (Traffic Adaptive Control for Oversaturated Isolated Intersections).

Fehon [23] summarized the application of adaptive control in US up to 2004. The author introduced the successful installation of adaptive signals (e.g. SCAT and SCOOT) in other countries, and the FHWA sponsored research. The author pointed out three main obstacles that hold back traffic engineers in the United States from using adaptive signals. First, the traffic engineers either paid little attention to the issue or did not believe the claimed benefits of adaptive signals. The second and the third were the practical institutional and financial issues. The author appealed that US traffic engineers need a shift in attitude from the current signal control patterns and should be open minded to accept adaptive control system.

2.2 Adaptive Traffic Control Features in Signal Controllers Software

Some of today's traffic controllers make available the "Adaptive Split" feature that automatically seeks the most advantageous split possible for all non-coordinated phases. Coordinated phases are not adjusted. Coordination Adaptive Split (CAS) and Critical Intersection Control (CIC) (found respectively on the SIEMENS Eagle EPAC 300 Actuated Signal Control Software [24] and on the Naztec TS2 2070 signal controller software [25]) is achieved by monitoring the termination of each non-coordinated phase and determining whether the phase gapped out or was forced off. If for two consecutive cycles a phase gapped out with over one second left in its allotted green time, it is a candidate for a decrease in its split. If for two consecutive cycles a phase was forced off it is a candidate for an increase in its split. At the end of each cycle, the supply/demand situation is examined and, if possible, the splits of impacted phases are changed. A phase can lose one second after it gaps out twice in a row and an additional one second for each successive gap out, down to its specified minimum green. On the other hand, a phase can receive one second after two consecutive force offs and an additional second for each successive force off. There is no theory behind the choice of one second increments or number of gap outs and force-offs. The approach is not considered robust as occupancy is not directly related to flow rate [27].

Sunkari et al [28] pointed out that the adaptive split feature is useful for reclaiming some of the “unused” time in the cycle that normally goes back to the coordinated phase. Engelbreht et al [29] warned that coordination modes and force-off modes will impact the functionality of the adaptive split feature and a “maximum recall” on any phase will disable it.

The Advanced System Controller Series 3 (ASC/3) manufactured by Econolite Control Products Inc [26] present a similar “Adaptive Split” feature called “Direct Split” allowing the operator to select which following phase or phases receive any unused split time from a phase. The operator can select up to two phases to direct this unused time. The first preference phase is qualified to determine if the time will be directed to this phase. If the first preference phase does not qualify, or if it does not need the directed split time, a second preference phase is qualified. If neither of the two preference phases qualify or does not use the unused time, it will be added to the coordinated phase. The coordinated options must be programmed to floating force-offs. No further information on qualification or necessity rules is provided in the literature.

2.3 Additional Experiments with Adaptive Control Systems

There are several computational intelligence based techniques that have been applied for the designing of real-time traffic signal controllers, such as fuzzy logic system (FLS), neural networks (NN) and genetic algorithms.

Chiu and Chand [30] applied the fuzzy logic controller to adjust the signal timing parameters at a given intersection considering the local traffic conditions and the signal timing parameters at adjacent intersections. The author used fuzzy decision rules to adjust the splits based only on local information of the degree of saturation. The amount of change in the timing parameters during each cycle was limited to a small fraction of the current parameters in order to ensure a smooth transition.

Priyono et al [31] proposed the application of two-stage neural network in real-time adaptive traffic signal control capable of analyzing the traffic scene detected by video surveillance, process the data and by means of a fuzzy-genetic model estimate the objective values in the optimization process with iterative adjustment of signal parameters.

Park and Chang [32] used a simple event based simulation program of vehicle arrivals and departure times to explore adaptive signal control under perfect knowledge in vehicle arrivals. A Genetic Algorithm was utilized for the development of the signal timing plans. It was shown that the marginal benefit of adaptive signal control increases up to a certain volume level and then decreases.

Owen and Stallard [33] developed a control strategy denominated GASCAP (Generalized Adaptive Signal Control Algorithm Project) that uses queue estimates and a rule-based algorithm for effective distributed adaptive signal control of traffic networks. The signal control logic consists of a set of rules for uncongested control and an algorithm that creates a fixed time plan for congested control.

Diakaki et al [34] presented the TUC (Traffic Urban Control) strategy. Based on a store-and-forward modeling of the urban network traffic and using the linear-quadratic regulatory theory, the design of TUC leads to a multivariate regulator for traffic-responsive coordinated network-wide signal control that is particularly suitable also for saturated traffic conditions. Real-time decisions in TUC cannot be taken more frequently than at the maximum employed signal cycle. The strategy will need to be redesign in case of modifications and expansions of the controlled network.

Wunderlich et al [35] proposed an algorithm for scheduling signal phases at an isolated intersection so as to maximize traffic throughput while minimizing the average latency experienced by the traversing vehicles. A maximal weight matching algorithm is used to determine phase sequence and allocate green time considering queue sizes at each approach.

2.4 Traffic Monitoring and Data Collection

The infrastructure necessary to accommodate adaptive traffic signal control usually includes a vast detection system that monitors traffic in real-time and gather accurate and comprehensive data. The most widely used detection technologies include inductive loops, video and RTMS (Remote Traffic Microwave Sensor). Vehicle counts with stop bar presence detectors are possible with either inductive loops or video detection [36] that analyzes the inductive waveform of vehicles passing over a large presence detection zone, providing a short contact closure every time a vehicle is counted. Smaglik [37] assessed the accuracy of the inductive loop count detectors. Vehicle counting through presence detection has the added benefit of maintaining safety to the intersection. With presence detection, a call is placed for the duration of time that the vehicle occupies the detector, as opposed to a short 100ms blip as in pulse detection. The ability to obtain real-time flow rates offers the potential for controllers to monitor the operating efficiency of vehicle phases and identify inefficient use of green time. RTMS does not operate well as stop bar detection. [27].

Smaglik et all [38] proposed using vehicle counts with stop bar detection to develop real-time flow rate information to estimate real-time volume-to-capacity (v/c) ratios as well as to evaluate the performance of an intersection over a period of time. Video data was recorded concurrently with count detector status, binned in one minute intervals and compared; yielding satisfactory results.

Smaglik et all [39] developed an integrated general purpose data collection module that time stamps detector and phase state changes within a NEMA actuated traffic signal controller and uses the data to assess phase capacity utilization and served volume on a cycle-by-cycle basis.

Luyanda et al [22] mentioned that phase timing data available from signal systems can vary substantially. At best, second-by-second returns of phase status (red, yellow and green) are available for all phases using AB3418 (California Assembly Bill 3418) [40]. At worst, the phase timing data is available for all phases once per minute. Other systems can report phase termination reasons in cyclic measures of effectiveness (MOE) reports every 5 to 15 min (based on user configuration).

Despite research and development of a wide range of traffic signal control strategies, deployment of advanced systems has lingered due to financial limitations of public authorities and by the concern over changing to the non-established and non-field proven but potentially better system. Therefore the idea of a simple, non-intrusive and split oriented self-adjustable traffic signal control offers potential and will be described next.

CHAPTER 3

THE EXPERIMENTAL SYSTEM

Without the ability to test the proposed algorithm on live traffic flows due to obvious safety concerns, it becomes necessary to use computers to simulate traffic flows in order to facilitate testing and the evaluation process. Additionally, actuated traffic signal controllers typically lack the ability to bin flow data in a cycle by cycle manner. Nonetheless, available Ethernet capability allows real-time status data to be extracted from the NEMA TS2 signal controller being used. Therefore, cycle by cycle flow data is available through external data processing software allowing for real-time processing of information back and forward to the signal controller. The following is a detailed presentation of the system architecture, the data management and communication and a description of the modeled intersection characteristics.

3.1 System Architecture

A hardware-in-the-loop (HIL) simulation is used to implement and evaluate the proposed algorithm. The system architecture is shown in Figure 1. A computer runs a traffic simulator with signal phases being controlled by an actual traffic signal controller. Data collected from the traffic signal controller feed the control algorithm that, when necessary, update signal timing splits back to the signal controller.

3.1.1 Traffic Simulator

The traffic simulation environment is running VISSIM (Verkehr In Staedten SIMulation) which is a time-step microscopic multi-modal traffic simulator with user friendly controls over all aspects of the network, such as geometrics, vehicle type, driver behavior, intersection control, vehicular volume inputs and statistical data collection. VISSIM is running with default parameters and the Wiedemann 74 Car Following Model which is mainly suitable for urban traffic. The software was developed by PTV Traffic Mobility Logistics.

Figure 1 - System architecture for a hardware-in-the-loop simulation (1-Computer, 2-Traffic Simulator, 3-Controller Interface Device, 4-Traffic Signal Controller, 5-Communication Protocol, 6-Algorithm Software, 7-Ethernet Connection Provider)

3.1.2 Traffic Signal Controller

The Advanced System Controller Series 3 (ASC/3-2100) manufactured by Econolite Control Products Inc. is used to control the intersection signal phases presented in the simulation layout. The ASC/3 is a NEMA TS2 standard controller as well as NTCIP standard compliant.

Common coordination parameters were input in the appropriate plans and tables provided in the signal controller software. These values work as initial settings. Figure 2 illustrates the common initial signal timing to be used and Figure 3 demonstrates its application on the ASC/3 controller along with the selection of phases 2 and 6 as the coordinated phases. The offset value is set to zero. Yellow is set to 3 seconds and red clearance is set to 2 seconds. The controller is set to MAX INHIBIT, allowing the coordinator phase split to control the time a phase is allowed to be green in any coordination pattern. The transition method is set to SMOOTH according to NTCIP 1202 2.5.2 integer 3. The offset reference point is LEAD, referencing the start of the local dial to the start of the first-coordinated phase green. There is no actuated permissive period on the coordinated phase to avoid disruption of coordination when the new split table is updated during the yellow period of the coordinated phase. The Force-off method is set to Fixed. The extension time also known as passage time is set to zero.

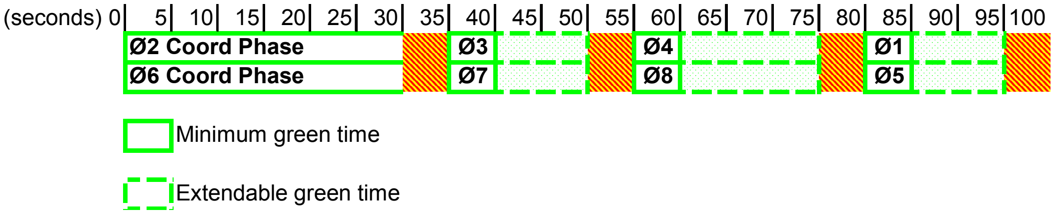


Figure 2 - Common initial signal timing

```
SPLIT PATTERN [  1]              MORE  v
                                 1 1 1 1 1 1 1
  PHASES 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
COORD PH . X . . . X . . . . . . . . . .

  PHASE        1       2       3       4
SPLIT....     20      35      20      25
MODE.....   NONE    NONE    NONE    NONE

  PHASE        5       6       7       8
SPLIT....     20      35      20      25
MODE.....   NONE    NONE    NONE    NONE

  PHASE        9      10      11      12
     PRESS 0..9 OR TOGGLE TO CHANGE
```

Figure 3 - ASC/3 controller application of the common initial signal timing

3.1.3 Algorithm Software

An external logic control method implements the algorithm using MATLAB®. MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation.

3.1.4 Data Management and Communication

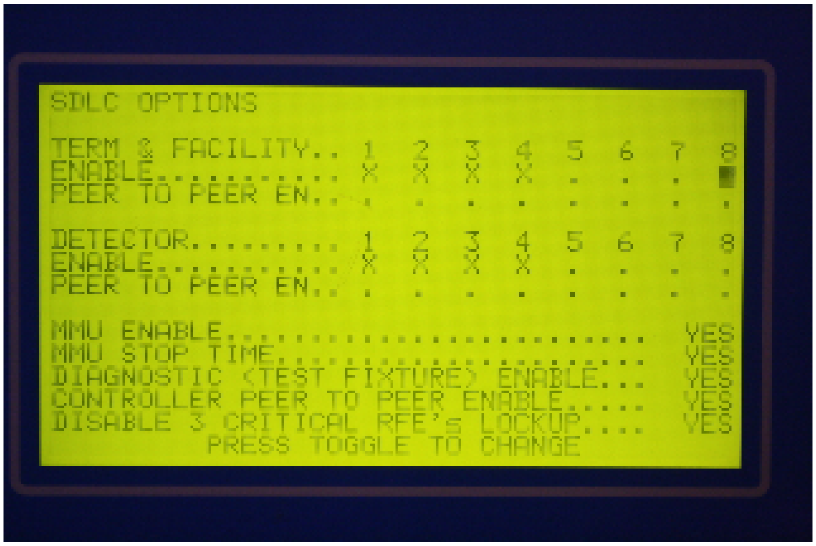
An important step on the hardware-in-the-loop (HIL) simulation is managing data and guaranteeing appropriate communication between entities. First, in order for the traffic simulator to interact with the traffic signal controller in a synchronous manner a controller interface device is necessary. Then, management of the data and communication between the algorithm and the signal controller will follow several protocol standards (NTCIP, TMP, and UDP) as described below.

3.1.5 Controller Interface Device

The Advanced Traffic Analysis Center's Controller Interface Device (ATACid) [41] was developed to interface a NEMA TS 2 (2003) compliant traffic controller with a personal computer running a traffic simulation model (VISSIM in this case) to perform hardware-in-the-loop simulation (HILS). The ATACid keeps track of phase data, and can update the controller via SDLC (Synchronous Data Link Control) cable with detector information received from VISSIM over its Ethernet connection. To allow for accurate time synchronization, provisions have been developed for holding its responses until the traffic controller has passed one real-time second. Therefore, this device is well suited for controller testing and real-time hardware-in-the-loop.

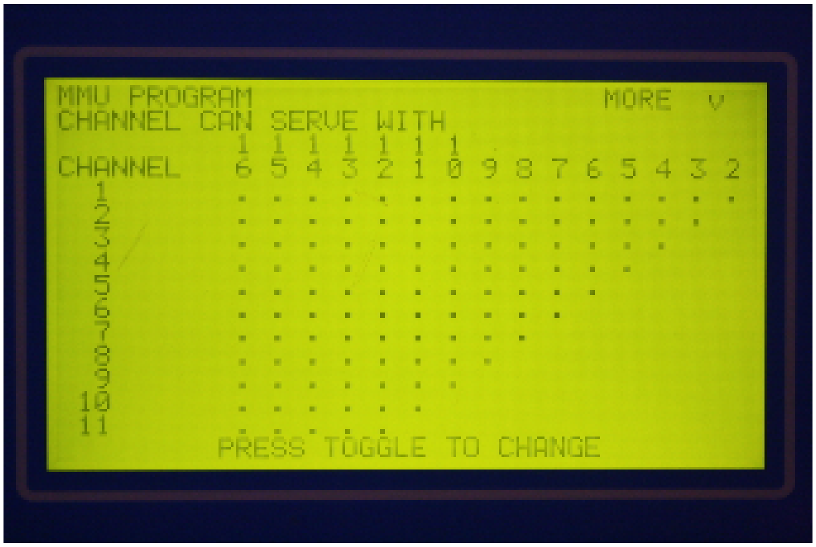
Before setting up the ATACid, the ASC/3 controller needs to be properly configured. Under SDLC option (MM-1-4-1), as shown in Figure 4, appropriate Terminal & Facility (T&F) and Detector Bus Interface Units (BIUs) should be turned on (typically 1-4 for both devices), so that, ASC/3 controller can read and set virtual detectors in VISSIM. The last step needed to set up the ASC/3 controller is to go under SDLC option (MM-1-4-2), as shown in Figure 5, and make sure that all of the channels are disabled. Therefore, the controller will not compare its programming with the MMU Program.

The next step in is to connect the ATACid and a computer with a RS232 “Null modem” serial cable, needed for setup purposes only. The SerUpdt.exe program as shown in Figure 6 is used for configuration. It is necessary to select the proper COM port to establish a communication between the computer and the ATACid. Since the ATACid is connected to the computer directly via crossover cable, manual mode is chosen instead of DHCP because of the direct connection. After choosing manual mode, pressing the retrieve button will show the detailed information on the ATACid, such as IP address, Subnet Mask, Gateway and port number, as shown in Figure 8.



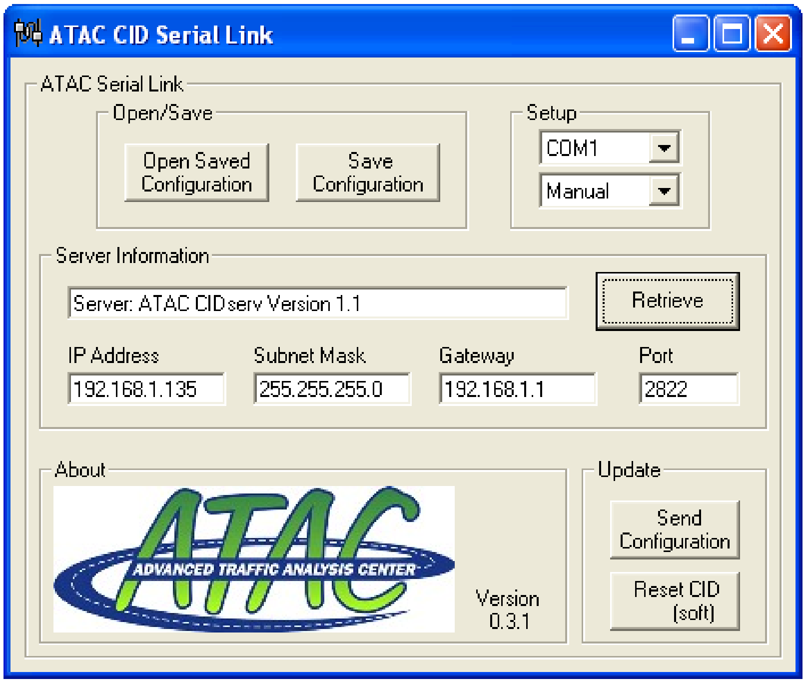
```
SDLC OPTIONS

TERM & FACILITY.. 1  2  3  4  5  6  7  8
ENABLE...........  X  X  X  X  .  .  .  █
PEER TO PEER EN..  .  .  .  .  .  .  .  .

DETECTOR......... 1  2  3  4  5  6  7  8
ENABLE...........  X  X  X  X  .  .  .  .
PEER TO PEER EN..  .  .  .  .  .  .  .  .

MMU ENABLE........................   YES
MMU STOP TIME.....................   YES
DIAGNOSTIC (TEST FIXTURE) ENABLE...   YES
CONTROLLER PEER TO PEER ENABLE.....   YES
DISABLE 3 CRITICAL RFE's LOCKUP....   YES
          PRESS TOGGLE TO CHANGE
```

Figure 4 - T&F and BIU



```
MMU PROGRAM                                  MORE  v
CHANNEL CAN SERVE WITH
                1 1 1 1 1 1 1
CHANNEL         6 5 4 3 2 1 0 9 8 7 6 5 4 3 2
   1            . . . . . . . . . . . . . . .
   2            . . . . . . . . . . . . . .
   3            . . . . . . . . . . . . .
   4            . . . . . . . . . . . .
   5            . . . . . . . . . . .
   6            . . . . . . . . . .
   7            . . . . . . . . .
   8            . . . . . . . .
   9            . . . . . . .
  10            . . . . . .
  11            . . . . .
           PRESS TOGGLE TO CHANGE
```

Figure 5 - MMU program



COM1

Manual

Server: ATAC CIDserv Version 1.1

IP Address: 192.168.1.135

Subnet Mask: 255.255.255.0

Gateway: 192.168.1.1

Port: 2822

Figure 6 - ATACID Serial Link – SerUpdt program

Those items should be set the same as the computer's, except for the IP address. However, both of the IP addresses have to be within the same section, which means only the last 3 digits can be different. Otherwise, they cannot be visible to each other. Lastly, the Port number is set to be 2822, the default one.

The next procedure is to test the connection between the ATACid and the computer using the CIDLink Interface, as shown in Figure 7. Under the "connection" menu of the CIDLink, IP address and Port have to be exactly the same as in the previous step. The connection will be set up after pressing "run" under the "connection" menu. Ideally, if the communications are successful, the number of packets transmitted (Packets Tx) should equal the number of packets received (Packets Rx). Detectors can be checked by pressing the virtual detectors under the signal heads in the CIDLink software. If the ASC/3 controller exactly reflects what the CIDLink indicates while testing the detectors, the computer and the ASC/3 controller are communicating successfully.

The last procedure is to copy several interface files to the VISSIM directory and to the working directory so that VISSIM can communicate to the ASC/3 controller via the ATACid. It is necessary to copy four DLL files, MSVCP71D.dll, MSVCR71D.dll,

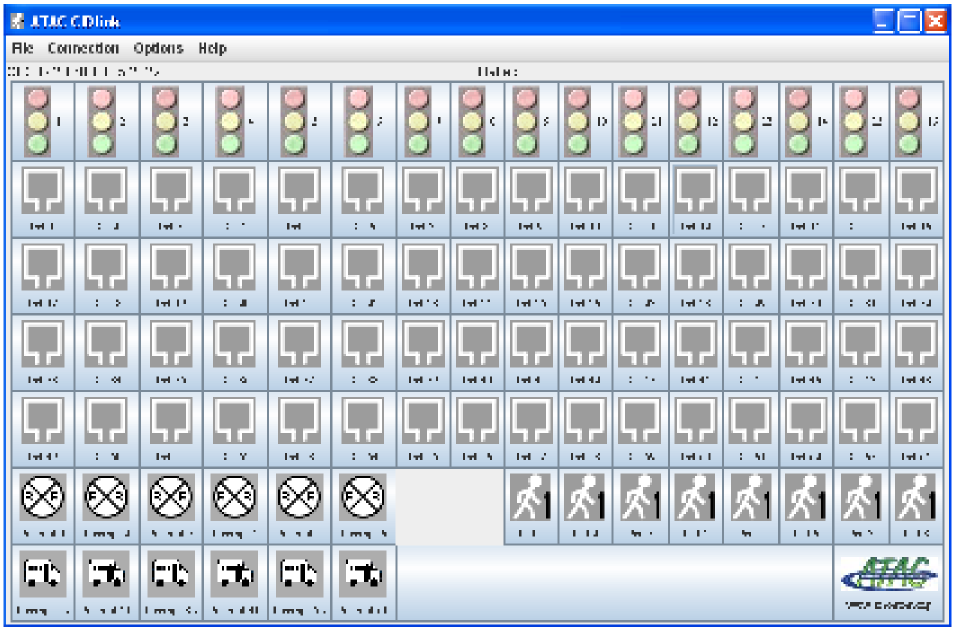
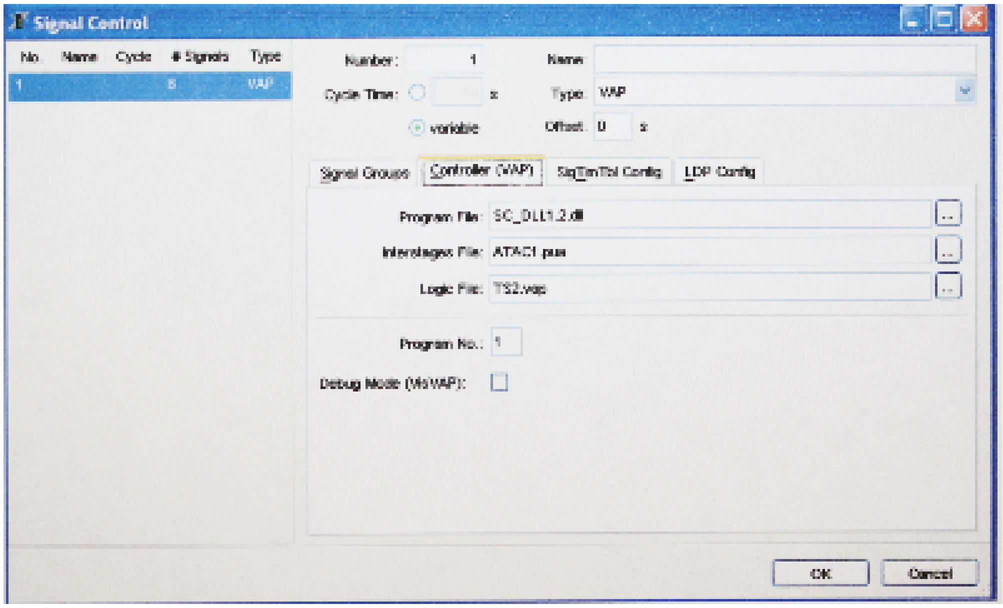


Figure 7 - CIDLink interface

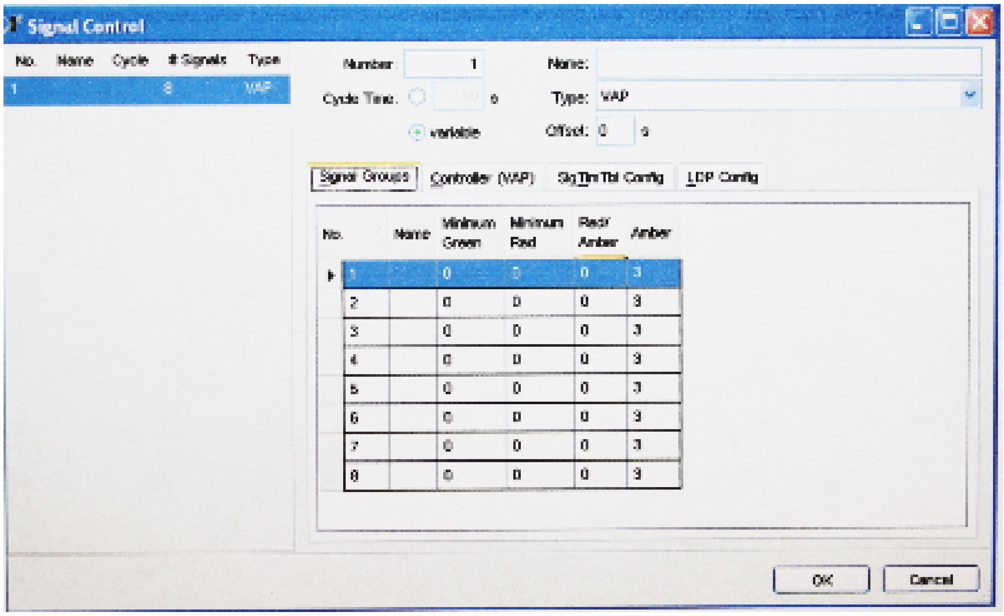
SC_DLL1.3.dll, and SC_DLL1.3.wtt files into the VISSIM\exe directory. The files ATAC1.pua and TS2.vap need to be copied to the working directory, where the project is located. These files allow VISSIM to treat the ATACid as a Vehicle Actuated Signal Control (VAP). The IP address and the port number in the file ATAC1.pua needs to match the ATACid IP address and Port number. In VISSIM, the signal control type under the "Signal control" menu needs to be changed to "VAP". These procedures are shown in Figures 8 and 9. The simulations should run at maximum speed and a resolution of 10 time steps per simulation second should be used. It is very important to notice that a CIDLink should not communicate simultaneously to an ATACid that is being used in a simulation.



No.	Name	Cycle	# Signals	Type
1			8	VAP

Number: 1
Name:
Cycle Time: ○ s
● variable
Type: VAP
Offset: 0 s
Program File: SC_DLL1.2.dll
Interstages File: ATAC1.pua
Logic File: TS2.vap
Program No.: 1
Debug Mode (VisVAP): ☐

Figure 8 - VISSIM signal control setting – Controller (VAP) tab



No.	Name	Cycle	# Signals	Type
1			8	VAP

Number: 1

Name:

Cycle Time: ○ s

● variable

Type: VAP

Offset: 0 s

No.	Name	Minimum Green	Minimum Red	Red/ Amber	Amber
1		0	0	0	3
2		0	0	0	3
3		0	0	0	3
4		0	0	0	3
5		0	0	0	3
6		0	0	0	3
7		0	0	0	3
8		0	0	0	3

Figure 9 - VISSIM signal control setting

3.2 Protocol Standards

3.2.1 National Transportation Communications for Intelligent Transportation System Protocol (NTCIP)

DeVoe and Wall [42] explained that in the past, each manufacturer of microprocessor based traffic controllers either developed or adopted a different, proprietary protocol for data communications. Extensive integration projects were necessary to incorporate different systems and to communicate between systems operated by adjacent agencies. NTCIP provides common standards for protocols that can be used by all manufacturers and system developers. A communication protocol defines a set of rules for messaging and how to encode the data contained in those messages for transmission between electronic devices. The NTCIP establishes the rules that allow bytes, characters, and strings to be organized into messages that are understandable by other NTCIP compliant devices. Therefore, NTCIP is a communication standard for transmitting data and messages between microcomputer controlled devices used in Intelligent Transportation Systems (ITS).

3.2.2 Transportation Management Protocol (TMP)

The Transportation Management Protocol (TMP) [43] is a composition of three distinct protocols all providing nearly identical services, but designed to meet different data exchanges and processing requirements. The three component protocols are as follows:

- Simple Network Management Protocol (SNMP);
- Simple Fixed Message Protocol (SFMP);
- Simple Transportation Management Protocol (STMP);

The information exchanged by all three protocols is in accordance with NTCIP. The TMP was carefully designed to provide 100% interoperability with the Internet-standard SNMP, but extends this protocol structure to provide for additional requirements of the transportation environment. STMP will be used in this project, but it is necessary to understand the functionality of SNMP first.

3.2.2.1 Simple Network Management Protocol (SNMP)

DeVoe and Wall [42] explained that SNMP is typically applied to managing network devices. Network management systems contain two primary elements: a manager and agents. The manager represents the traffic controller in NTCIP. Agents can be management centers, for example. Contained within the traffic controller are managed objects, or variables, that contain parameters that directly relate to the current operation of the intersection. These objects are arranged in a virtual information database called a management information base, or MIB. SNMP allows managers to communicate their MIB to agents for the purpose of accessing these objects. SNMP provides the means for retrieval and modification of information by using a get-set paradigm to exchange individual pieces of data (object). The exchange of data between the manager (traffic controller) and the agent (MATLAB®) will be provided by the ASC/3 SNMP Client management station by sending each object identifier along with a get or set request. Each object has a name, syntax and encoding. The name, an object identifier (OID), uniquely identifies the object.

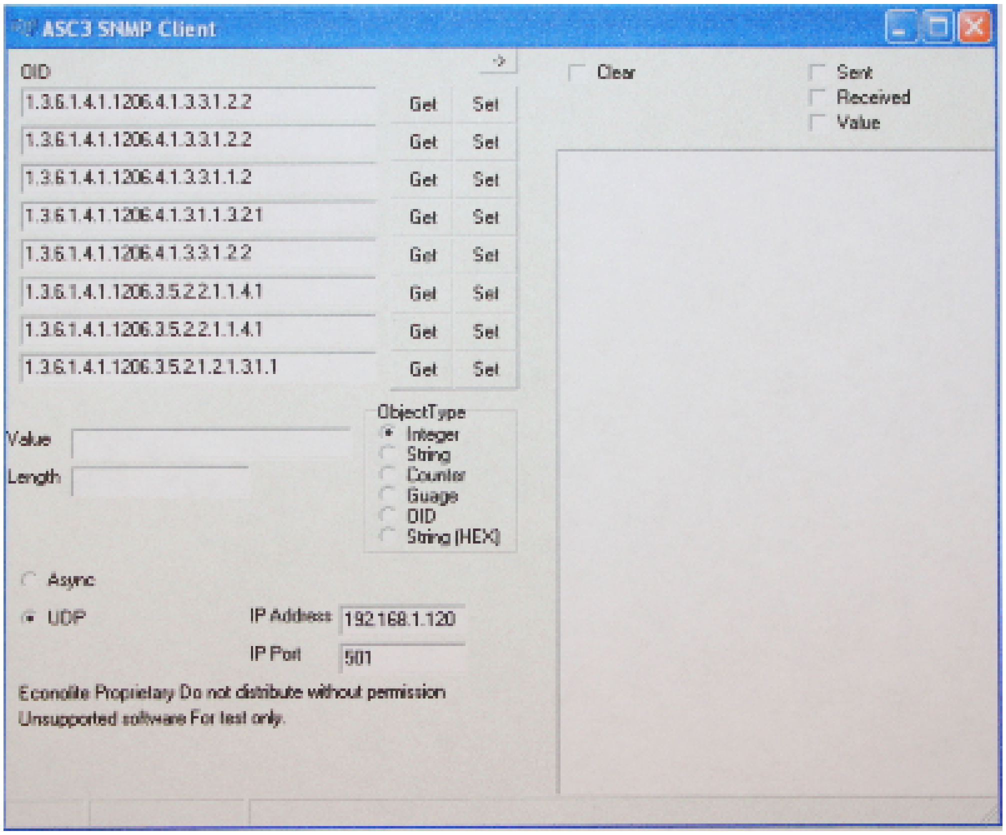
3.2.2.2 Simple Transportation Management Protocol (STMP) and Dynamic Objects

STMP is a simplified more compact version of SNMP. It has been designed to work with dynamic objects or block objects defined by the agent. This has the benefit of providing the management station with the flexibility required to define its messages. NTCIP dictates that up to 13 dynamic objects can be defined within the traffic controller and a sequence of 255 object identifier (OID) can be included in each dynamic object. Data packets can be largely reduced because there is no need to include object identifiers overheads since the transportation objects are under the same NEMA node (1.3.6.1.4.1.1206). The advantage of this approach is that it improves the polling frequency and reduces the communication bandwidth.

Accessing ASC/3 MIB by STMP

Liu [46] describe the problem concerning the deficit of low polling frequency of SNMP communication, and proposes a dynamic object configuration in Simple Transportation Management Protocol (STMP) which is able to speed up the polling frequency to 0.1 seconds.

As in SNMP, the exchange of data between the manager (traffic controller) and the agent (MATLAB®) will be provided by the ASC/3 Client management station by sending each object identifier along with a get or set request (as shown in Figure 10). Eleven dynamic objects containing object identifier (OID) information for two detector groups, system cycle time and individual phase split times have been configured for this project beforehand. In other words, initially the user defines what information will be needed from and what information will be sent to the ASC/3 controller. This is done through the ASC/3 Client management station that sends commands to the controller to declare how the dynamic objects will be build. The structure of the dynamic object is then stored in the controller. Ultimately, this communication mechanism allows the MATLAB® algorithm to continuously get detector information as well as system cycle time and update phase splits in an appropriate and timely manner.



OID
1.3.6.1.4.1.1206.4.1.3.3.1.2.2
1.3.6.1.4.1.1206.4.1.3.3.1.2.2
1.3.6.1.4.1.1206.4.1.3.3.1.1.2
1.3.6.1.4.1.1206.4.1.3.1.1.3.2.1
1.3.6.1.4.1.1206.4.1.3.3.1.2.2
1.3.6.1.4.1.1206.3.5.2.2.1.1.4.1
1.3.6.1.4.1.1206.3.5.2.2.1.1.4.1
1.3.6.1.4.1.1206.3.5.2.1.2.1.3.1.1

Value
Length

ObjectType
● Integer
○ String
○ Counter
○ Guage
○ OID
○ String (HEX)

○ Async
● UDP
IP Address 192.168.1.120
IP Port 501

Econolite Proprietary Do not distribute without permission
Unsupported software For test only.

Figure 10 - ASC/3 SNMP Client accessing data

The following are the configuration steps to define dynamic object 1. Due to problems encountered during the update process of the split table it was chosen to define one dynamic object for each individual split instead of one dynamic object with eight different object identifiers. The configuration steps to define dynamic objects 2-11 can be found in Appendix A.

Dynamic Object 1 - Detectors for phases 1-8

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**1** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**1** – dynObjConfigStatus
Value: 2
Object Type: Integer

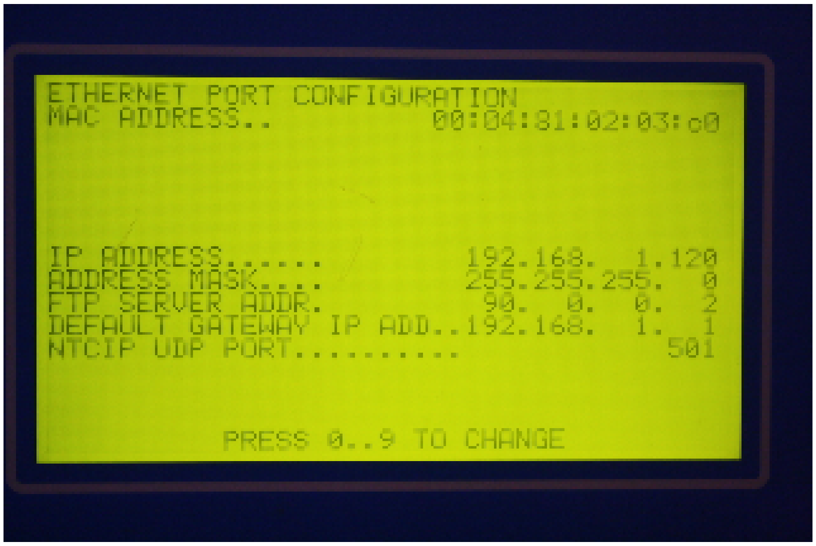
Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**1** – dynObjConfigOwner
Value: Detector1
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.1.**1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.2.4.1.2.1
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.**1.1** – dynObjConfigStatus
Value: 1
Object Type: Integer

3.2.3 User Datagram Protocol (UDP)

The UDP/IP Internet Transport Profile is used in this project for system communications between the algorithm and traffic signal controller, as defined in NTCIP 2022. It incorporates placing the data stream into an UDP datagram and then placing the UDP datagram into an IP packet. An IP defines the location of a device on a network. Because message arbitration could clutter network communication lines given only one communication channel, the Internet protocol standard provides up to 65535 channels, known as ports, for devices to communicate [42]. SNMP typically uses port 161 and for STMP communications, the NTCIP standard specifies that all communications be directed on port 501. Figure 11 shows the Ethernet port configuration for the ASC/3 controller.



```
ETHERNET PORT CONFIGURATION
MAC ADDRESS..              00:04:81:02:03:c0



IP ADDRESS......             192.168.  1.120
ADDRESS MASK....             255.255.255.  0
FTP SERVER ADDR.              90.  0.  0.  2
DEFAULT GATEWAY IP ADD..192.168.  1.  1
NTCIP UDP PORT..........                 501


             PRESS 0..9 TO CHANGE
```

Figure 11 - ASC/3 Ethernet port configuration

3.3 Geometric Design

Figure 12 illustrates the intersection modeled in the VISSIM traffic simulator. This is a four approach intersection with two through movement lanes, with one of them shared with the right turning movement. The right turning movement volume is set to be 20% of the total through movement volume. The left turning movement volume is variable and is described in Chapter 5. Platoon arrivals for the coordinated phases were not considered in this initial setup. There is a 100m left turning lane for each approach to avoid immediate blocking of the through movement. All approaches are 1000 meters long to ensure that arriving traffic is distributed properly, and that vehicles do not build up at the inputs of the network. Each phase in this intersection is labeled according to the NEMA (National Electrical Manufacturers Association) convention. Each lane received a 54 ft detection zone with four 6 ft x 6 ft loop detectors spaced 10 ft apart and set to standard mode. The first detector is positioned in front of

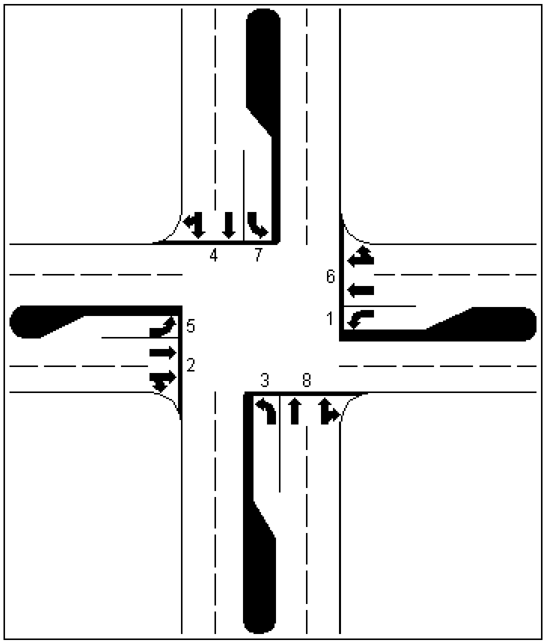


Figure 12 - Typical intersection

the stop bar and is used for counting purposes. It is important to notice that the flow-based adaptive split signal control system can be deployed with a different detector configuration as long as it includes the stop bar detection for counting purposes. The posted speed limit is 45MPH for all approaches. There are no pedestrians on the system and traffic is composed only by cars.

CHAPTER 4

THE FLOW-BASED ADAPTIVE SPLIT ALGORITHM

In coordinated systems, the constraints imposed by the traffic signal controller logic regulate unequally the ability of phases to reallocate unused time during a cycle, potentially producing unnecessary delay. Therefore, it is convenient to manipulate and update split tables in real-time giving any phase the opportunity to receive additional help to service demand. First, it is necessary to explore the concept of lost time during a cycle to understand the potential sources of unused time. Next, it is necessary to develop a strategy to capture any “slack” time and efficiently reallocate it in the cycle to potentially improve overall intersection conditions.

4.1 The Concept of Lost Time

Lost time is generally defined as the portion of time at the beginning of each green period (start-up lost time) and a portion of each yellow change plus red clearance period (clearance lost time) that is not usable by vehicles when demand is present. In this research, in more general terms, lost time is any available unused time during the cycle of a coordinated signal control that is not efficiently allocated to a phase to serve demand. A new interpretation of lost time is presented next, suggesting that there are two main sources that can potentially generate unused time in coordinated signal control systems.

Control logic lost time is any potential unused time during the cycle caused by the controller functionality. It can be caused by different mechanisms necessary to ensure coordination (hold and force-offs), for example, or to ensure minimum green time or even to allow for a phase to gap-out (passage time). First, related to coordination issues, non-coordinated phases cannot benefit from any available unused time from the coordinated phase, except on a very specific case, when the coordinated phase is actuated after the yield point (actuated permissive period). Also, the control logic is very restrictive in permitting unused time to be exchanged between non-coordinated phases (as demonstrated in the example provided in the next section). Then, it is not guaranteed that the minimum green time assigned to each phase during each cycle will be efficiently used, therefore generating potential unused time, mainly because it is based on an allowance for uncertainty (excluding the case of pedestrian clearance times). Finally, the time necessary to allow a phase to gap-out (passage time) may not be efficiently used either, especially if the detection zone is not located upstream. Improved split control can potentially reduce control logic lost time.

Driving behavior lost time is any potential unused time caused by the reaction of drivers to signal phase changes. It is the traditional start-up lost time or the time a driver takes to react to the initiation of the green phase and to accelerate. Clearance lost time is also a potential component of driving behavior lost time corresponding to a portion of each yellow change plus the red clearance period and is explained by drivers making different decisions on the onset of yellow at the ending stage of a phase. Due to its nature, driving behavior lost time exists but is not precisely quantified. Therefore it will have assumed values in this research based on common practice.

4.1.1 Allocating *Control Logic* Lost Time

Manipulating and updating split tables can be advantageous in reducing delay generated by *control logic* lost time. One specific example of addressing *control logic* lost time due to coordination issues is being able to provide extra time (when available) to any saturated phase in contrast to being restricted by the force-off logic of the controller. In other words, in a full-actuated coordinated system working with fixed force-off logic, a saturated phase will only receive time depending on its position in the ring and may not receive all the “slack” time available in the cycle due to constraints imposed by the logic. Figure 13 shows an example of a 80 seconds cycle with ring 1 original splits and demonstrates how time is allocated in fixed force-off logic to the saturated phase 4, when phase 3 gaps out early, phase 1 has no demand and phase 2 is the coordinated phase. Figure 14 demonstrates the same example with the manipulation of the split table, where phase 4 receives a hypothetical potential “slack” time of 5 additional seconds from phase 1, after calculation of the average traffic conditions for the last 3 cycles.

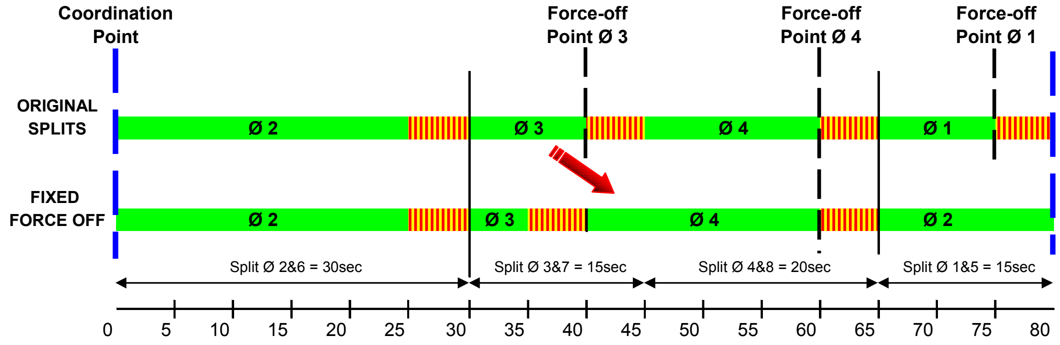


Figure 13 - Allocation of potential "slack" time using Fixed Force-off

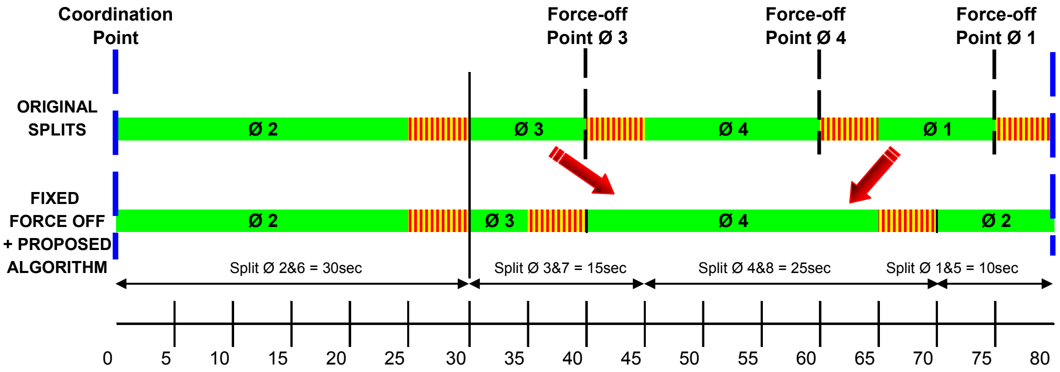


Figure 14 - Allocation of potential "slack" time using Fixed Force-off plus the proposed flow-based adaptive split algorithm

Therefore the split for phase 4 was increased from 20 seconds to 25 seconds, while the split for phase 1 was decreased by 5 seconds.

Analysis of current control logic produces Table 1 where the process of receiving potential “slack” time from another phase on a typical dual ring configuration is presented. The advantage of manipulating the split table is the potential to allocate “slack” time to any phase. In contrast with current technology only one phase (the coordinated phase) has the chance to inherit all the potential “slack” time available. Likewise, phases immediately following the coordinated phase generally can not get any slack-time unless the coordinated phase is actuated, which is not common in practice. Wise redistribution of potential “slack” time with different phases receiving a portion of time is not possible in current typical controller logic.

Following the same concept just presented is the ability to better control the “early return to green” phenomena. Early return to green is defined as the servicing of a coordinated phase in advance of its programmed begin time as a result of unused time from non-coordinated phases [2]. The drawback of the phenomena is the potential increase in system stops because of inefficient flow at downstream intersections.

Table 1 - Potential allocation of time between phases in fixed force-off logic

Potential "slack" time available from phase	Can be utilized by phase
1	2
2	3 or 4 or 1 *
3	4 or 1 or 2
4	1 or 2
5	6
6	7or 8 or 5 *
7	8 or 5 or 6
8	5 or 6

*Available only if the Actuated Permissive Period feature is enabled

4.2 Control Algorithm Development

The development of the flow-based adaptive split algorithm is based on HCM Quick Estimation Method (QEM) for critical movement analysis. Critical movement analysis is a simplified technique that has broad application for estimating phasing needs and is based on the basic fundamental principle that identifies the set of movements that cannot time concurrently and require the most time to serve demand.

Initially, in order for the algorithm to capture any potential "slack" time it is necessary to know the flow rate for every approach lane during the cycle with data from stop bar detection. Comparing the individual lane's actual flow rate volume to an assumed capacity expected for each approach lane, it is estimated if the green time for that phase is efficiently being used or if there is any unused time. The HCM considers that an intersection is operating under capacity when the volume to capacity ratio is below 0.85. Lacking more conclusive data, a threshold value of 0.85 for the volume being used over the capacity available for each phase was initially chosen to determine if a phase will receive any additional "slack" time, potentially preventing it to become oversaturated. The analysis will be done every cycle following the detailed step by step procedure laid out next. Data collection, data analysis and parameter updates are addressed in the development of the control algorithm.

4.2.1 Data collection

Volume count data from stop bar presence detection for every approach lane on the study intersection is collected for each cycle. If a phase is skipped the algorithm recognizes the no volume scenario. A three cycle moving average of vehicular discharge information is tabulated and recorded. At this point, it is important to note that the split table can be updated every cycle, characterizing the three cycle moving average as a smoothing mechanism but at the same time promoting the idea of responsiveness. The 3 cycle moving average is also important to prevent the system from "chasing" extreme changes in flow that might result from faulty data. The signal controller will feed the algorithm with detection information with a polling frequency of 0.1 seconds using the data packets of dynamic objects in STMP (described in the next chapter).

4.2.2 Data analysis

4.2.2.1 Effective green time

The effective green time is the duration of time between the end of the start-up delay on a green interval and the lost time during yellow extension. For each phase, the effective green time will be calculated according to:

$$g = G + Y + R - (l_1 + l_2)$$

where:

g = effective green time;

G = actual green interval;

Y = actual yellow change interval, considered to be 3 seconds in the algorithm;

R = actual red clearance interval, considered to be 2 seconds in the algorithm;

l_1 = start-up lost time, considered to be 2 seconds in the algorithm;

l_2 = clearance lost time, considered to be 2 second in the algorithm.

4.2.2.2 Capacity

Capacity for a movement at signalized intersection is the rate at which vehicles can pass through the intersection at saturation flow rate during the effective green time. For each phase, capacity will be calculated according to:

$$c = s*(g/C)$$

where:

c = capacity;

s = saturation flow rate

g = effective green time;

C = cycle length.

Saturation flow rate for a movement at a signalized intersection is the equivalent hourly rate at which vehicles can traverse the intersection assuming a constant green indication at all time and no loss time. The value of 1800 vehicles per hour per lane is the assumed value for the algorithm. The saturation flow rate value can be modified by actual data in the future, if deemed necessary.

4.2.2.3 Equivalent hourly volume

Equivalent hourly volume for a movement is the real-time cycle by cycle volume data collected from detectors transformed to an hourly rate. For each phase, equivalent hourly volume will be calculated according to:

$$EHV = (3600/C)*v$$

where:

EHV = equivalent hourly volume;

C = cycle length;

v = real-time cycle by cycle volume.

4.2.2.4 Volume to capacity ratio (v/c)

For any movement, the volume to capacity ratio is simply the ratio of the equivalent hourly volume to the capacity. For each phase, volume to capacity ratio (v/c) will be calculated according to:

$$v/c = EHV/c$$

where:

v/c = volume to capacity ratio;

EHV = equivalent hourly volume;

c = capacity;

4.2.2.5 Potential "slack" time calculations

First, the control algorithm provides the flexibility for the user to define how much each phase can be reduced, constrained by the value of the minimum green. An initial value of 50% reduction was chosen, meaning that each phase can have up to half of its time available for redistribution in the cycle. This feature is extremely important to determine how much time can be available from coordinated phases to any phase that needs additional green time. Care shall be taken regarding safety issues as driver expectancy of minimum green time and dilemma zone safety, when deciding how much flexibility will be given to the algorithm.

Now, for each phase, the vehicular discharge (real-time cycle by cycle volume data) for the last 3 cycles is averaged and the results are rounded up. The assumed saturation flow rate of 1800 vehicles per hour per lane yields headways of 2 seconds. Interpreting this time as the average time necessary to clear one vehicle through the intersection, one can estimate the total time necessary to clear the 3 cycle average of vehicular discharge by simple multiplication of the two variables. The potential "slack" time for each phase is then calculated by subtracting estimated total time necessary to clear the 3 cycle average of vehicular discharge from the last cycle effective green for the phase, if the 3 cycle average of vehicular discharge is larger than the minimum green. Otherwise, potential "slack" time is calculated by subtracting the phase minimum green from the last cycle effective green. Variation above the average is accounted for in the target v/c ratio of 0.85. One needs not to account for lost time in the calculation of total time necessary to clear the 3 cycle average of vehicular discharge, because lost time is already accounted for in the calculation of the effective green. Now, it is necessary to check for the added flexibility given by the user definable parameter of how much a phase can be reduced. Therefore, if the total time necessary to clear the 3 cycle average of vehicular discharge is smaller than the user definable reduced green time for the phase, then potential "slack" time is simply the difference between the last cycle effective green time for the phase and the user definable reduced green time. Otherwise, potential "slack" time is calculated by subtracting the 3 cycle average of vehicular discharge from the last cycle effective green time. For each phase, potential "slack" time will be calculated according to:

$$PST = g - (3600/s) * (\sum v/3)$$

where:

PST = potential "slack" time;

g = effective green time for the phase in the previous cycle;

s = saturation flow rate;

$\sum v$ = summation of last 3 cycles' real-time cycle by cycle volume;

For uniformity and ease of understanding, when mentioned, a "phase in need" is considered a phase that has reached the volume to capacity ratio (v/c) threshold value of 0.85, (as explained below). In contrast, a "helping phase" is any phase that is below the volume to capacity ratio (v/c) threshold value and is able to redistribute time.

4.2.2.6 Calculations of potential green time needed by a phase

The same concept to calculate potential "slack" time is applied here. Therefore, potential green time needed by a phase is the time in excess of the last cycle effective green time needed to accommodate the total time necessary to clear the 3 cycle average of vehicular discharge.

4.2.2.7 Monitoring volume to capacity ratio (v/c)

Volume to capacity has been calculated for each phase. The algorithm will test each phase's 3 cycle average volume to capacity ratio (v/c) against a target volume to capacity ratio (v/c) of 0.85. The 0.85 value was chosen in a proactive manner, thus when a phase is above the threshold value, the control algorithm will trigger potential modifications in the split table while traffic conditions are considered undersaturated and under stable operation.

Operating close to capacity can easily cause the demand during the cycle to exceed the green time on a given phase(s). Queues are likely to accumulate and affect intersection performance. The proposed algorithm is not intended to accommodate oversaturated conditions and significant different approaches are then necessary.

At this stage, volume to capacity ratio (v/c) for phase pairs and for the entire intersection are calculated, helping to evaluate future strategies to better accommodate any available "slack" time.

4.2.2.8 Time available from potential helping phases

Considering the standard 8 phase NEMA ring and barrier structure (Figure 15), a table is constructed demonstrating potential phases that can help a phase that reached the threshold value of 0.85 set for the volume to capacity ratio (v/c) (Table 2). It is also determined how much time is available from each phase for help, based on a decision if potential "slack" time is larger or smaller than potential time needed by the problematic phase.

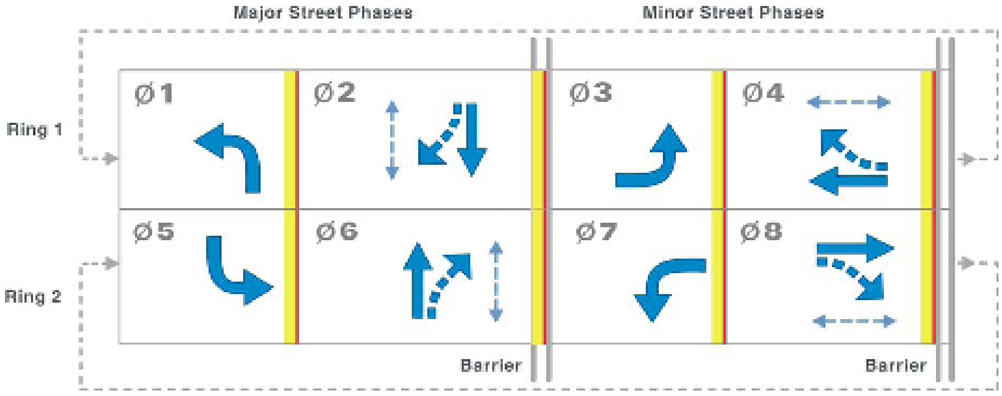


Figure 15 - Standard ring-and-barrier diagram [2]

Table 2 - Potential helping phases

HELPING PHASE	1	5	2	6	3	7	4	8
PHASE IN NEED								
1			X		X		X	
5				X		X		X
2	X				X		X	
6		X				X		X
3	X		X				X	
7		X		X				X
4	X		X		X			
8		X		X		X		

4.2.2.9 Barrier analysis

The easiest way to accommodate potential time needed by any phase is to allocate potential “slack” time from a helping phase inside the same barrier group. So, for example, if phase 1 is in need, phase 2 is the primary phase to provide help. In the same example, time from phase 2 may not be sufficient to accommodate the time needed by phase 1, therefore additional time is seek beyond the barrier group. During this step, for each phase, the algorithm first verifies how much time is necessary beyond the barrier to accommodate the phase needs. This is done by subtracting available helping time from a phase on the same barrier group from total time needed by a phase. Secondly, for each phase, time available for help beyond the barrier is calculated, aggregating information provided from the time available from potential helping phases’ table. Lastly, the algorithm will check if time available for help beyond the barrier is larger than time needed by a phase and proportionally reduces it to its needs.

4.2.2.10 Critical path analysis

Critical path analysis is a simple check performed by the algorithm to determine which phase pair is critical on each ring in each barrier group. A phase pair with the largest volume to capacity ratio (v/c) is considered to be critical and will serve as a constraining mechanism on the decision to allocate time to the other side of the barrier.

4.2.2.11 Time to be taken from each phase

Taking into consideration how much time is necessary for a phase to improve its volume to capacity ratio (v/c) and the time that is available from each phase to be allocated elsewhere in the split, as well as considering the barrier analysis and the critical path analysis, the algorithm decides how much time will be taken from each phase to help the problematic phase.

4.2.2.12 New green split calculation

First, the algorithm will seek the first phase to meet the following criteria: be the highest volume to capacity ratio (v/c) above the threshold value of 0.85 among all phases and have potential “slack” time to receive. The table developed for the “time available from potential helping phases” will dictate decisions at this stage. Phases in the same ring and in the same barrier group can simply exchange time between them. When additional time is needed, there will be the need to consider a barrier movement and time available will be added to the phase in need that shares the same ring. The non critical ring will redistribute time inherited proportionally, according to the volume to capacity ratios (v/c) of its phases.

4.2.2.13 Ring and barrier check

The cycle length will not be modified. Therefore, before deployment of parameter updates, ring and barrier structure is checked for consistent alignment of barriers and no modification of cycle length according to:

a) $\sum Sg\varnothing_{1} \,\&\, Sg\varnothing_{2} = \sum Sg\varnothing_{5} \,\&\, Sg\varnothing_{6}$

b) $\sum Sg\varnothing_{3} \,\&\, Sg\varnothing_{4} = \sum Sg\varnothing_{7} \,\&\, Sg\varnothing_{8}$

where:

Sg = split green;

$\varnothing_{i}$ = phase i.

4.2.2.14 Parameter updates

After every cycle, when necessary, the algorithm will have developed a new split table determining how much time of the cycle needs to be allocated to each phase to potentially improve the current traffic condition in the intersection. The final step of the flow-based adaptive split signal control algorithm is to update parameters without disruption of coordination. To accomplish that, the algorithm needs to avoid transition.

Transition is the process of either entering into a coordinated timing plan or changing between two coordinated plans. It may also be caused due to preemption or loss of coordination during pedestrian crossings. To better understand transition it is necessary to acknowledge that the concept of coordination relies on the ability to synchronize multiple intersections in time. To provide this synchronization, each local controller clock is referenced to a master clock (unchangeable background timing mechanism). When the local controller clock reaches a point where it is necessary to change the coordination plan (e.g.: peak or off-peak traffic), the cycle, split and offset may be changed. When changing the cycle length or the offset, the controller shifts the local offset reference point by means of a transition algorithm that may either shorten or lengthen the cycle. The offset reference point is a defined point in the cycle that creates the association needed between signalized intersections and the master clock. The transition period may vary from one to five cycles and may be very disruptive to traffic.

With the main objective of the algorithm being to update the split table every cycle if necessary, the determination of the point in the cycle where to implement the new split in real-time is of main importance to avoid transition. Understanding of the offset reference point is necessary for strategically deployment of the split table without transition. A TS2 controller is being used and the offset reference point references the start of the local dial (beginning of the cycle) to the start of the first coordinated phase green (LEAD) [26]. In practice what this means is that at least one coordinated phase is assured to be timing at the beginning of the cycle. Figure 16 illustrates the offset reference point when both coordinated phases have the same split. Figure 17 illustrates the offset reference point when coordinated phases have different splits.

For the flow-based adaptive split signal control algorithm updating the split table should be a trivial task without the necessity of transition because the offset reference point and the cycle length are not changed. The flexibility given in the algorithm to manipulate the coordinated phase yielded the need to validate the process. Preliminary tests with the system architecture in place proved to be not trivial and turn out to be a major problem.

First, the beginning of the cycle (after 0 seconds) was chosen as the point to implement the new split in real-time. This period of time was chosen because it was believed to be the only part in the cycle (up to the minimum reduced green time of the coordinated phases chosen by the user) that would be consistent in every cycle. It did work for light traffic that permitted an early return to green phenomena under fixed force-off. When phases 1 and 5 did not gap out the minimum green for the coordinated phases timed at the beginning of the cycle causing disruption of coordination during the split update. It led to the conclusion that updating the split parameters should occur when both coordinated phases are in “Green Rest” or in other words, have already timed their minimum green.

Even with that information available, the update process continued to disrupt coordination in some instances. It was later found that when updating the current splits with a new set of splits not every combination was accepted by the controller internal algorithm. Therefore, a transition process would start, causing major problems to traffic and compromising the premise that the algorithm should update parameters without disruption of coordination. After more than 100 hours of simulation, a library of occurrences was analyzed providing a common trend. In order to avoid transition the following condition need to be met:

$$Max\ (new) - Min\ (new) > Max\ (current) - Min\ (current)$$

where:

Max (new) = Maximum value between the two coordinated phases of the split to be updated;

Min (new) = Minimum value between the two coordinated phases of the split to be updated;

Max (current) = Maximum value between the two coordinated phases of the current split;

Max (current) = Minimum value between the two coordinated phases of the current split;

If the above condition is not true, the new set of splits was treated accordingly and was updated at two distinct periods, during the Green Rest period of the coordinated phases or during the yellow period of the coordinated phases.

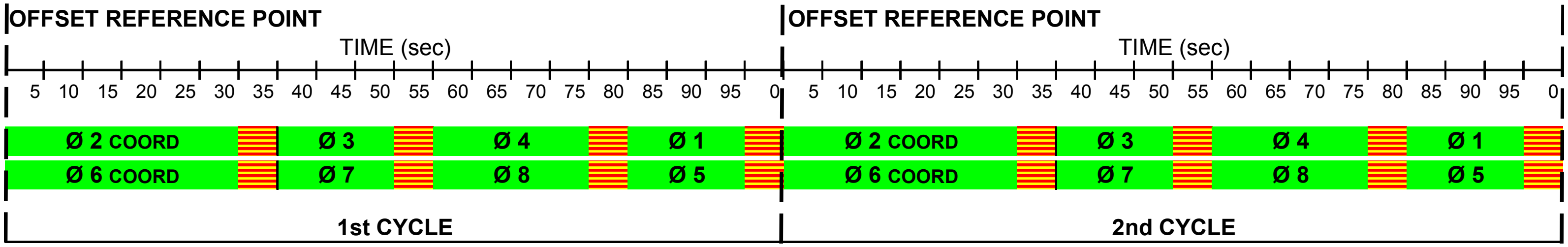


Figure 16 - Offset reference point with coordinated phases having similar splits

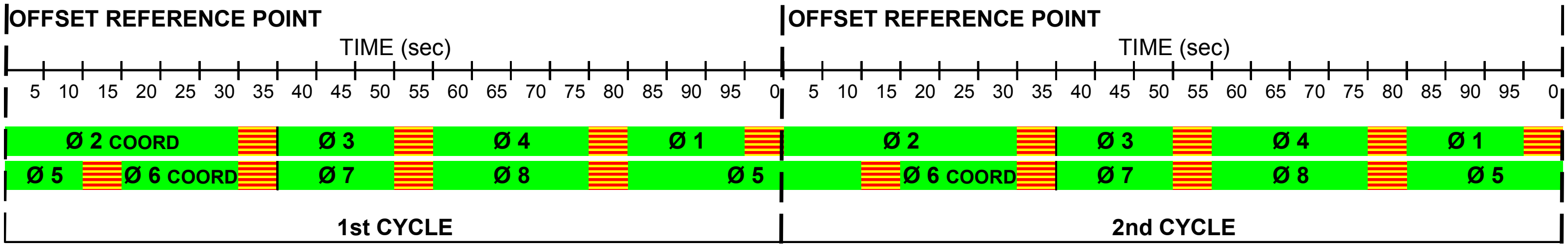


Figure 17 - Offset reference point with coordinated phases having different split

CHAPTER 5

ANALYSIS OF RESULTS

The proposed flow based adaptive split algorithm is compared to a full-actuated coordinated system and the hypothesis that *control logic* lost time can be reallocated between phases to reduce average delay in the intersection is evaluated. Average delay per vehicle is collected and analyzed as the appropriate measurement of effectiveness (MOE). Both systems run identical simulation environments. The control routine single steps through the traffic simulation, while controlling the signal group with the custom design control logic.

All simulation runs were performed for one hour. The initial 15 minutes were treated and discarded as "warm-up" period. Only the last 45 minutes of simulation were used to evaluate the performance of the system. A minimum of 30 runs was conducted for each scenario utilizing different random seeds.

One fundamental measurement of effectiveness (MOE) for evaluating the performance of a signal control strategy includes the average delay per vehicle. Analyzing the average delay experienced by a vehicle that has traversed the network is an indication of how long in average the vehicle has had to wait at the intersection prior to crossing it. It is important to notice that the loss time caused by acceleration or deceleration following other vehicles is part of the average delay. Performance data is collected from VISSIM traffic simulation utilizing "Node Evaluation".

The algorithm achieved the expected performance related to its ability of monitoring traffic flow, the capability of changing the split table in real-time and the ability to avoid disruption to coordination. A summary of related results is developed next.

5.1 Experimental System Performance

5.1.1 Monitoring Traffic Flow

During the simulation, the external agent MATLAB® continuously collected detector status data directly from the traffic signal controller translating the binary data into traffic flow information for each individual lane. Data were used to build a 3 cycle running average of vehicular flow. Each simulation run accounted for an average total number of 3.005 vehicles traversing the intersection during the analyzed 45 minutes period. An overall average error of -0.77% was detected when comparing the algorithm vehicular reading to VISSIM traffic simulator values. The majority of the missing readings occurred on the coordinated phases producing no impact on the results.

5.1.2 Adaptiveness

The algorithm was able to interpret the flow information extracted from the traffic signal controller and allocate any available “slack” time to the phase that presented the highest volume to capacity ratio (v/c) above the threshold value of 0.85. As an example, Figure 18 demonstrates the average vehicular flow per cycle for phase 4 of Scenario 1 (described in 6.2.1). Figure 19 shows the average, minimum and maximum split values per cycle for the same situation among the 30 simulation runs. It is possible to verify how the split for phase 4, with an original split of 25 seconds, received additional time responding to the increase in vehicular flow.

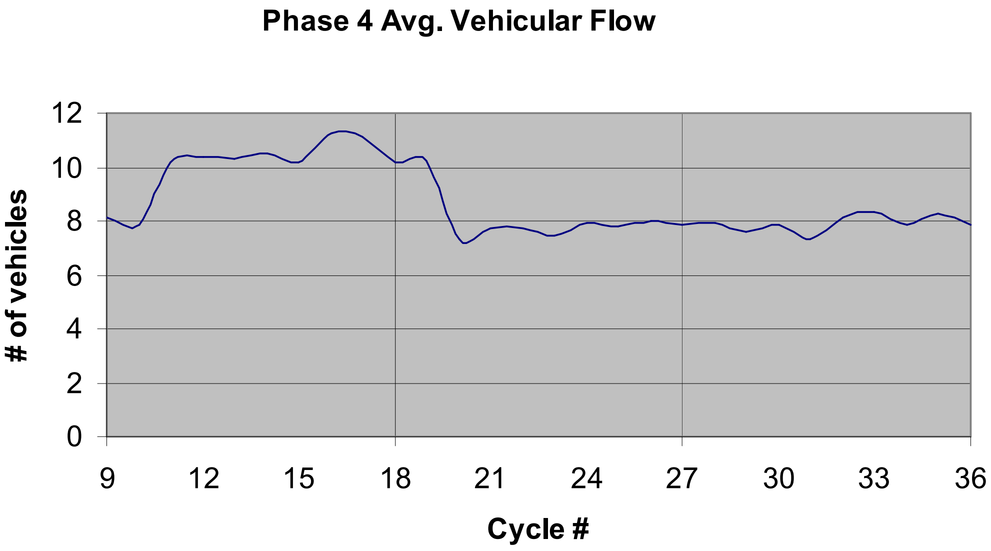


Figure 18 – Average vehicular flow per cycle – Phase 4 – Scenario 1

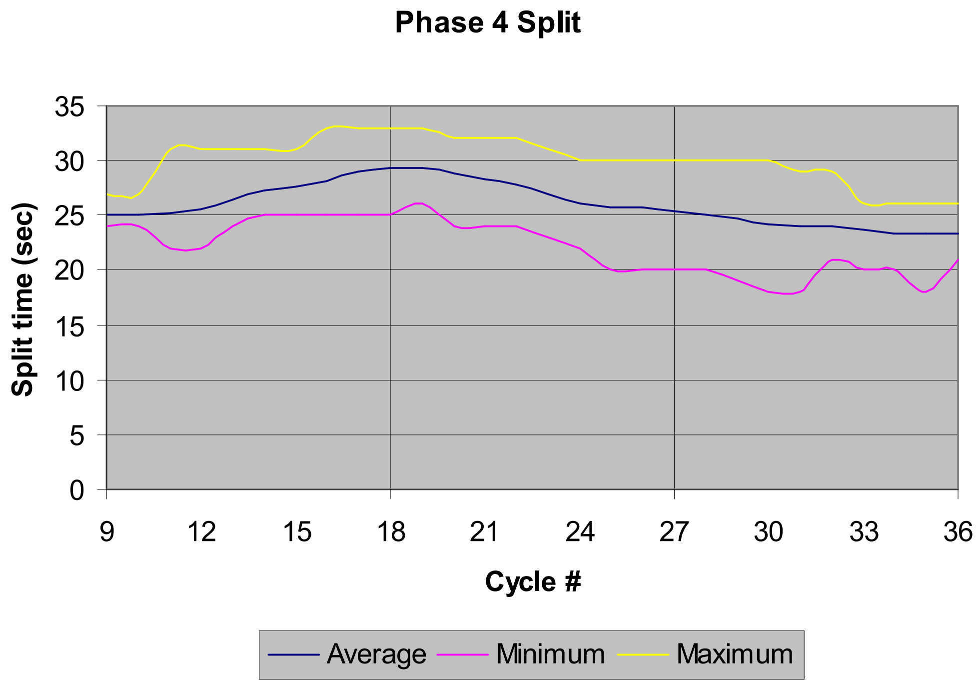


Figure 19 – Average, minimum and maximum split values – Phase 4 – Scenario 1

Additional data for Scenarios 1 and 2 can be found in Appendix B and Appendix C. A sample of split and volume data for simulation run of Scenario 2 Seed 735 is presented in Appendix D.

5.1.3 Robustness

To evaluate the robustness of the control algorithm it was important to check if the signal controller applied transition methods during any split table updates. The algorithm output data regarding individual phase vehicular counts would be affected by the event of disruption of coordination. While transitioning the traffic signal controller would maintain the coordinated phases active for at least one entire cycle promoting no vehicular flow data for all the remaining phases, making the problem readily detectable. Every simulation run was analyzed and no disruption of coordination was found during the total 60 hours of simulation running under the algorithm control.

The algorithm proved to be robust due to the implementation of corrections when a conflicting new split table was encountered. The output data were analyzed in search of corrections performed by the algorithm to avoid the already mentioned problem of transitioning or coordination disruption. Data for Scenario 1 presented a total of 86 corrections out of 810 cycles during the 30 hour simulation, representing 10.62%. Data for Scenario 2 presented a total of 118 corrections out of 810 cycles, representing 14.57%. Individual seeds presented up to 20% of the cycles being corrected. A table presenting the distribution of corrections during each individual run is presented in Appendix E.

5.2 Scenario Analysis

The main objective of the flow-based adaptive split signal control system is the ability of manipulating splits in favor of phases that need time to serve additional demand. Therefore, two different scenarios with 6 distinct 15 minutes intervals of traffic flow variation were constructed to evaluate the ability of the control system to address variations in flow.

5.2.1 Scenario 1

The objective for Scenario 1 was to determine the Measurement of Effectiveness (MOE) for a situation where only one approach would have traffic flow above the threshold v/c value of 0.85. For example, during the 900-1800 seconds period, the approach for phases 4 and 7 presents a v/c ratio of 0.95. For the next 15 minutes of simulation the phase 2 and 5 approach is affected with higher demand, and finally during the last part of the simulation, the phase 3 and 8 approach has higher traffic demand. Table 3 presents the distribution of traffic volume throughout the entire simulation.

Table 3 – Traffic volume distribution for Scenario 1

				0-900 (Warm-up)					**900-1800**				
					Volume					Volume			
Approach	**Phases**	**Capacity**	**Total**	v/c	**Total**	Left	Through	Right	v/c	**Total**	Left	Through	Right
1	1	288	1404	0.70	**983**	202	625	156	0.75	**1053**	216	670	167
	6	1116											
2	5	288	1404	0.70	**983**	202	625	156	0.75	**1053**	216	670	167
	2	1116											
3	3	288	1044	0.70	**731**	202	423	106	0.70	**731**	202	423	106
	8	756											
4	7	288	1044	0.70	**731**	202	423	106	0.95	**992**	274	575	144
	4	756											

				1800-2700					**2700-3600**				
					Volume					Volume			
Approach	**Phases**	**Capacity**	**Total**	v/c	**Total**	Left	Through	Right	v/c	**Total**	Left	Through	Right
1	1	288	1404	0.70	**983**	202	625	156	0.75	**1053**	216	670	167
	6	1116											
2	5	288	1404	0.95	**1334**	274	848	212	0.75	**1053**	216	670	167
	2	1116											
3	3	288	1044	0.70	**731**	202	423	106	0.90	**940**	259	544	136
	8	756											
4	7	288	1044	0.70	**731**	202	423	106	0.70	**731**	202	423	106
	4	756											

5.2.1.1 Average Delay

While analyzing each distinct 15 minutes interval for Scenario 1, it is important to have an understanding of how each phase is impacted by variations in traffic demand and by modifications of the split table throughout the entire cycle. Figure 20 and Figure 21 present the average delay in seconds per vehicle for each of the eight individual phases. The layout of the graphs follows the NEMA ring and barrier structure.

For the period of 900 - 1800 seconds, phases 4 and 7 are affected with higher demand. While average delay per vehicle for both systems tested went up during the period, further observation of the average delay graphs for both phases indicate that the flow-based adaptive split algorithm yielded lower values. The affirmative is confirmed in Table 4, with 13.92% lower average delay experienced in phase 7 and 4.29% lower average delay experienced in phase 4. Phases representing approaches 1 and 2 suffered an insignificant increase in delay. Phase 3 was affected by the algorithm in a negative way when compared to the full actuated coordinated system and its average delay increased by 7.27%. For now, the poor performance of phase 3 is directly related to the better performance of phase 4. Looking at the NEMA ring and barrier structure, phase 3 is the first phase to be able to help (with any “slack” time) the increase in demand experienced by phase 4. Section 5.3 will better address the reasons behind the performance of phase 3.

Table 4 – Average delay for Scenario 1 – 1st period

900-1800 (sec)				
Avg Delay				
Approach	**Phases**	**FACS ***	**FBASA ****	**% DIFF**
1	1	49.14	49.91	1.57%
	6	24.28	24.47	0.77%
2	5	50.29	50.63	0.69%
	2	24.31	24.48	0.68%
3	3	47.26	50.69	7.27%
	8	39.74	40.11	0.94%
4	7	64.01	55.10	-13.92%
	4	44.61	42.69	-4.29%
Overall		37.58	37.08	-1.34%

* FACS Full-actuated coordinated system
** FBASA Flow-based adaptive split algorithm

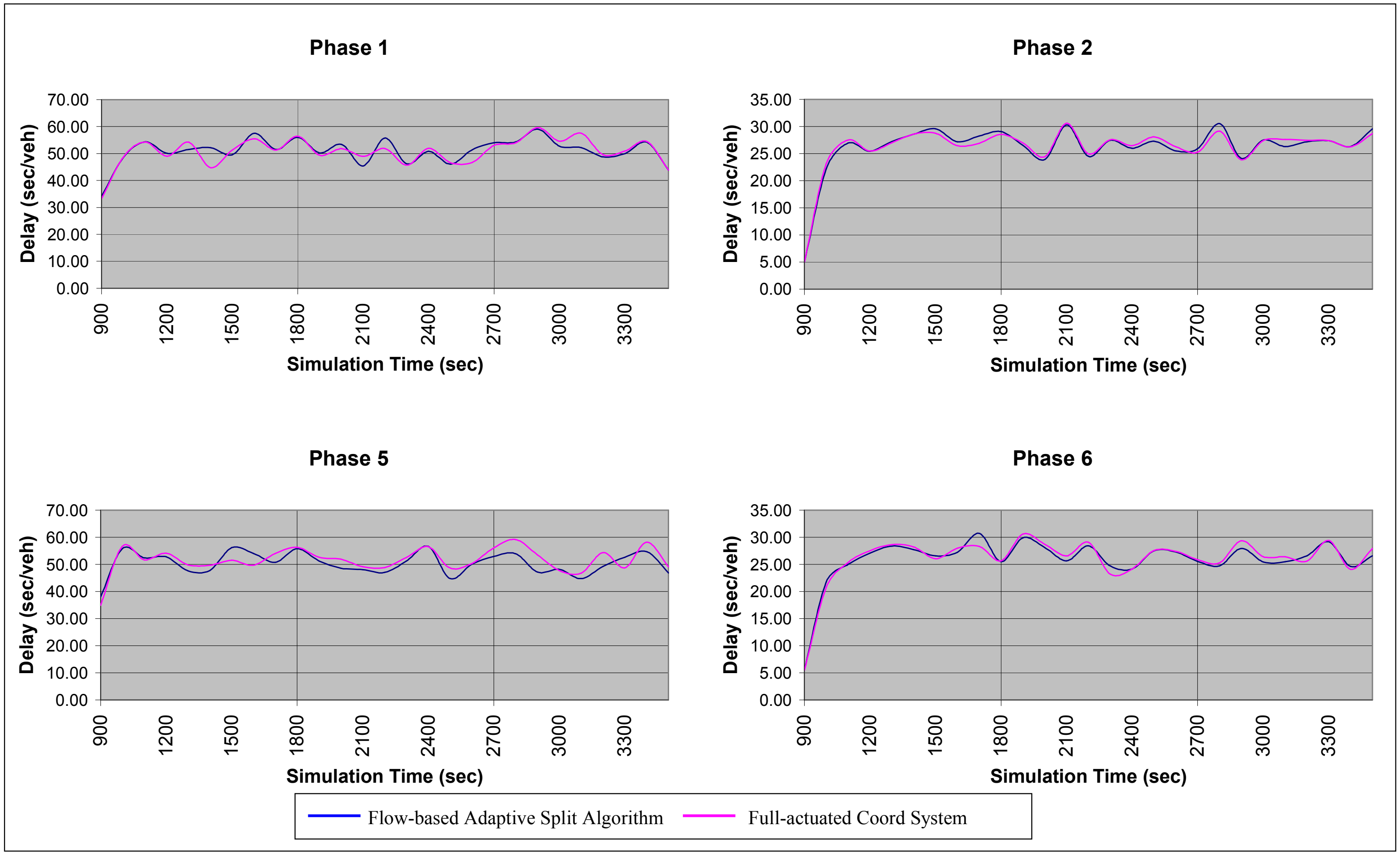


Figure 20 – Average delay for Scenario 1 – Phases 1, 2, 5 and 6

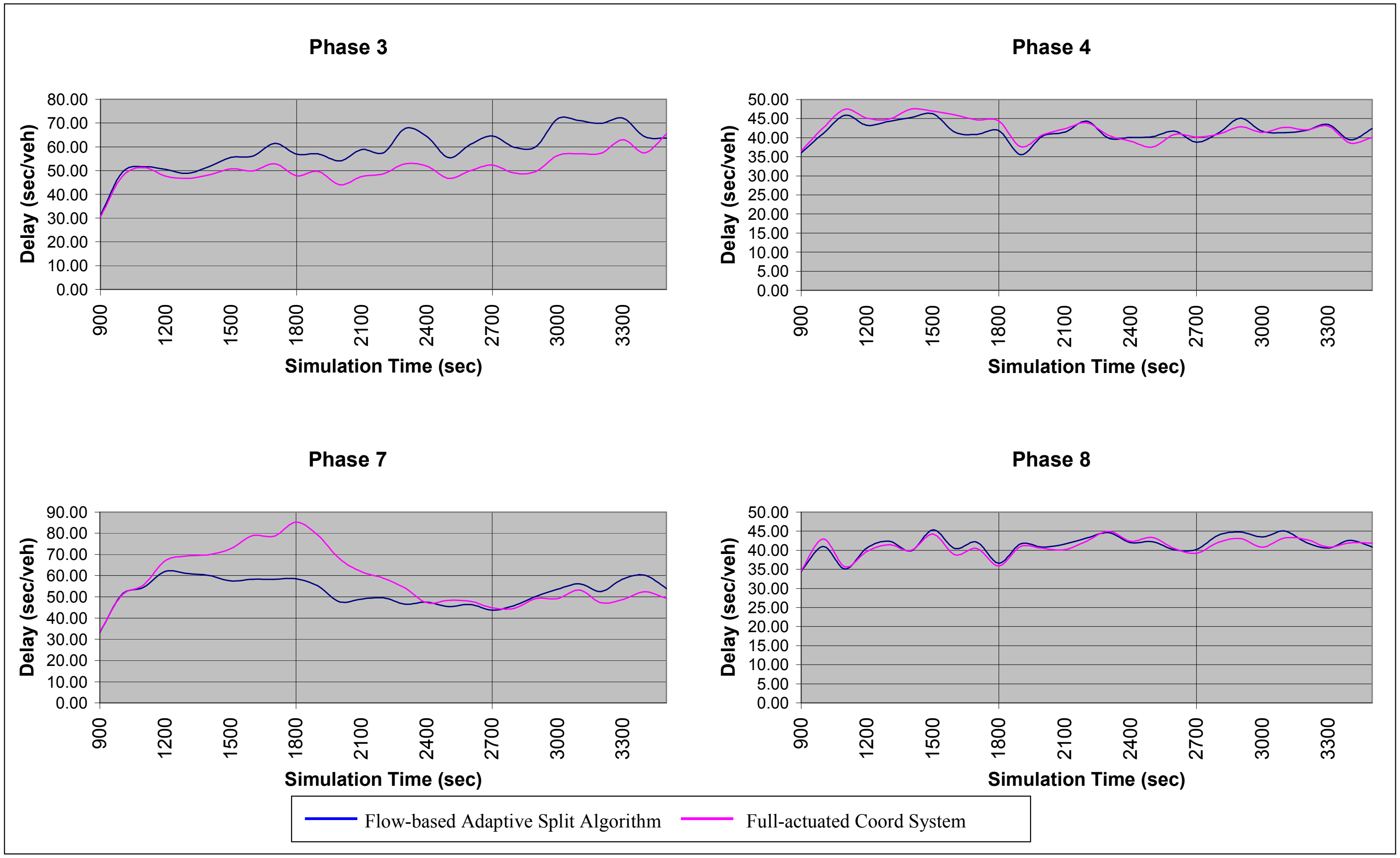


Figure 21 – Average delay for Scenario 1 – Phases 3, 4, 7 and 8

For the period of 1800 - 2700 seconds, phases 2 and 5 are affected with higher demand. Observation of the average delay graphs for both phases in conjunction with Table 5 indicate that the flow-based adaptive split algorithm yielded slightly lower values. Phase 3 experienced even higher average delays (21.37% increase) when the proposed algorithm was used and once again, the reasons leading to the poor performance of phase 3 will be discussed in Section 5.3. Nevertheless, phase 7 continue to benefit (19.07% decrease in average delay) from the extra-time acquired in the beginning of the cycle. No significant increase in the remainder of the phases was noticed.

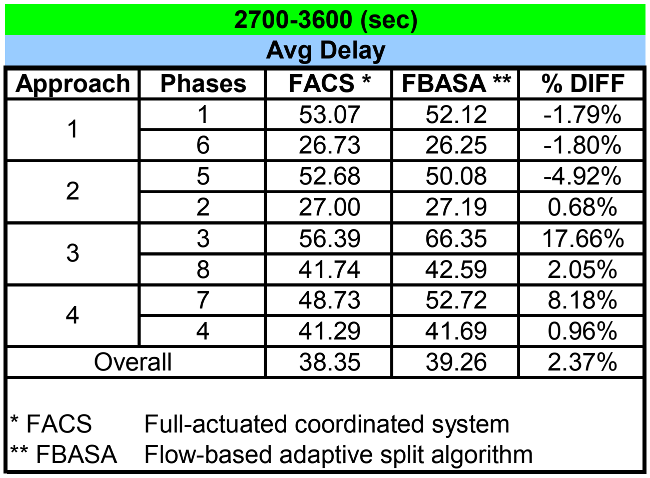
For the period of 2700 - 3600 seconds, phases 3 and 8 are affected with higher demand. Despite the higher demand the average delay for phase 3 was about 3.71% lower than the average delay experienced by phase 3 in the previous period of the simulation. The algorithm was able to improve the splits for phase 3 as can be noticed in the individual phase graphs for average, minimum and maximum split values found in Appendix B. Phase 7 and phase 4 ended up with higher average delays, being the primary phases to help phases 3 and 8, respectively. Once again, Section 5.3 will present a discussion on the problems evidenced during the simulation for Scenario 1.

Table 5 – Average delay for Scenario 1 – 2nd period

1800-2700 (sec)				
Avg Delay				
Approach	**Phases**	**FACS ***	**FBASA ****	**% DIFF**
1	1	49.93	50.58	1.32%
	6	27.01	26.83	-0.67%
2	5	51.91	50.41	-2.89%
	2	27.09	26.71	-1.38%
3	3	48.79	59.22	21.37%
	8	41.20	41.41	0.52%
4	7	61.18	49.52	-19.07%
	4	40.78	40.61	-0.42%
Overall		38.13	37.63	-1.30%

* FACS Full-actuated coordinated system
** FBASA Flow-based adaptive split algorithm

Table 6 – Average delay for Scenario 1 – 3rd period



2700-3600 (sec)

Avg Delay

Approach	Phases	FACS *	FBASA **	% DIFF
1	1	53.07	52.12	-1.79%
	6	26.73	26.25	-1.80%
2	5	52.68	50.08	-4.92%
	2	27.00	27.19	0.68%
3	3	56.39	66.35	17.66%
	8	41.74	42.59	2.05%
4	7	48.73	52.72	8.18%
	4	41.29	41.69	0.96%
Overall		38.35	39.26	2.37%

* FACS Full-actuated coordinated system
** FBASA Flow-based adaptive split algorithm

5.2.2 Scenario 2

The main objective for Scenario 2 was to determine the behavior of the algorithm for a situation where traffic increased above the threshold v/c value of 0.85 for more than one approach. Due to unexpected results on Scenario 1, it was decided to test the setup from the last 15 minutes of Scenario 1 at the beginning of Scenario 2 (900-1800 seconds) to verify if the reaction of phases 3 and 7 would be repeated. Table 7 presents the distribution of traffic volume throughout the entire simulation.

Table 7 – Traffic volume distribution for Scenario 2

				0-900 (Warm-up)					900-1800				
					Volume					Volume			
Approach	**Phases**	**Capacity**	**Total**	v/c	**Total**	Left	Through	Right	v/c	**Total**	Left	Through	Right
1	1	288	1404	0.70	**983**	202	625	156	0.75	**1053**	216	670	167
	6	1116											
2	5	288	1404	0.70	**983**	202	625	156	0.75	**1053**	216	670	167
	2	1116											
3	3	288	1044	0.70	**731**	202	423	106	0.90	**940**	259	544	136
	8	756											
4	7	288	1044	0.70	**731**	202	423	106	0.70	**731**	202	423	106
	4	756											

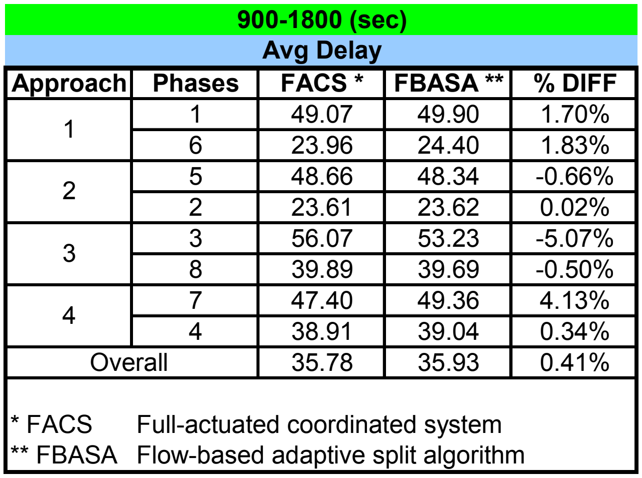
				1800-2700					2700-3600				
					Volume					Volume			
Approach	**Phases**	**Capacity**	**Total**	v/c	**Total**	Left	Through	Right	v/c	**Total**	Left	Through	Right
1	1	288	1404	0.70	**983**	202	625	156	0.75	**1053**	216	670	167
	6	1116											
2	5	288	1404	0.95	**1334**	274	848	212	0.95	**1334**	274	848	212
	2	1116											
3	3	288	1044	0.95	**992**	274	575	144	0.95	**992**	274	575	144
	8	756											
4	7	288	1044	0.70	**731**	202	423	106	0.95	**992**	274	575	144
	4	756											

5.2.2.1 Average Delay

As already mentioned, it is important to have an understanding of how each phase is impacted by variations in traffic demand and by modifications of the split table throughout the entire cycle while analyzing each distinct 15 minutes interval for Scenario 2. Figure 22 and Figure 23 present the average delay in seconds per vehicle for each of the eight individual phases. The layout of the graphs follows the NEMA ring and barrier structure.

For the period of 900 - 1800 seconds, phases 3 and 8 are affected with higher demand. Observation of the average delay graph for phase 3 indicate that the flow-based adaptive split algorithm yielded lower values as confirmed in Table 8 while Phase 8 did not present significant improvement. Phases representing approaches 1 and 2 suffered an insignificant variation in delay. Phase 7 was affected by the algorithm in a negative way when compared to the full actuated coordinated system and its average delay increased by 4.13%. The expectation that an anomaly with the behavior of phases 3 and 7 existed is confirmed. For now, the poor performance of phase 7 is directly related to the stable performance of phase 8, which split received extra time from phase 7 to accommodate the extra demand (see the individual phase graphs for average, minimum and maximum split values for phases 7 and 8 found in Appendix C). Section 5.3 will better address the reasons behind the irregular performance of phases 3 and 7.

Table 8 – Average delay for Scenario 2 – 1st period



900-1800 (sec)

Avg Delay

Approach	Phases	FACS *	FBASA **	% DIFF
1	1	49.07	49.90	1.70%
	6	23.96	24.40	1.83%
2	5	48.66	48.34	-0.66%
	2	23.61	23.62	0.02%
3	3	56.07	53.23	-5.07%
	8	39.89	39.69	-0.50%
4	7	47.40	49.36	4.13%
	4	38.91	39.04	0.34%
Overall		35.78	35.93	0.41%

* FACS Full-actuated coordinated system
** FBASA Flow-based adaptive split algorithm

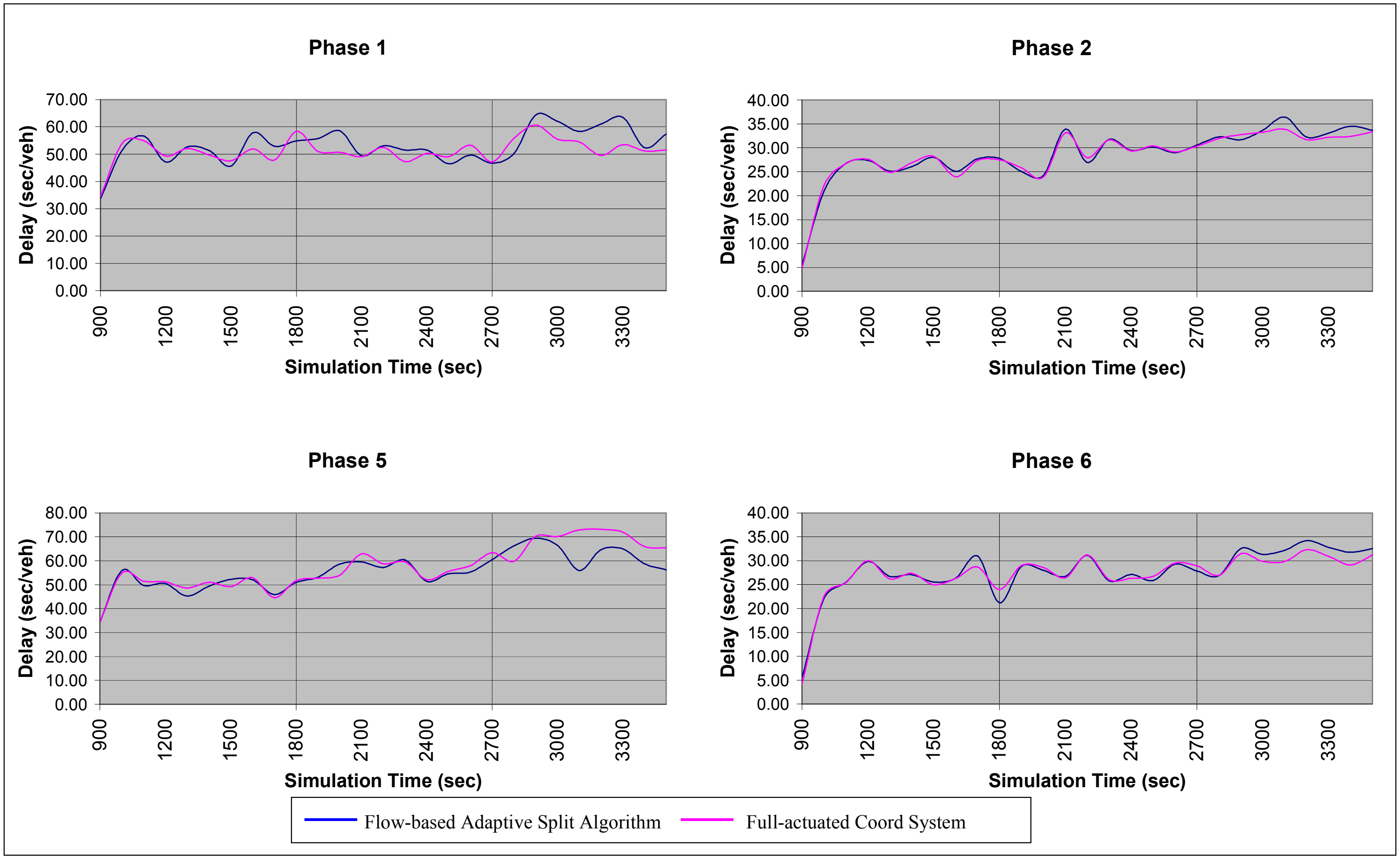


Figure 22 – Average delay for Scenario 2 – Phases 1, 2, 5 and 6

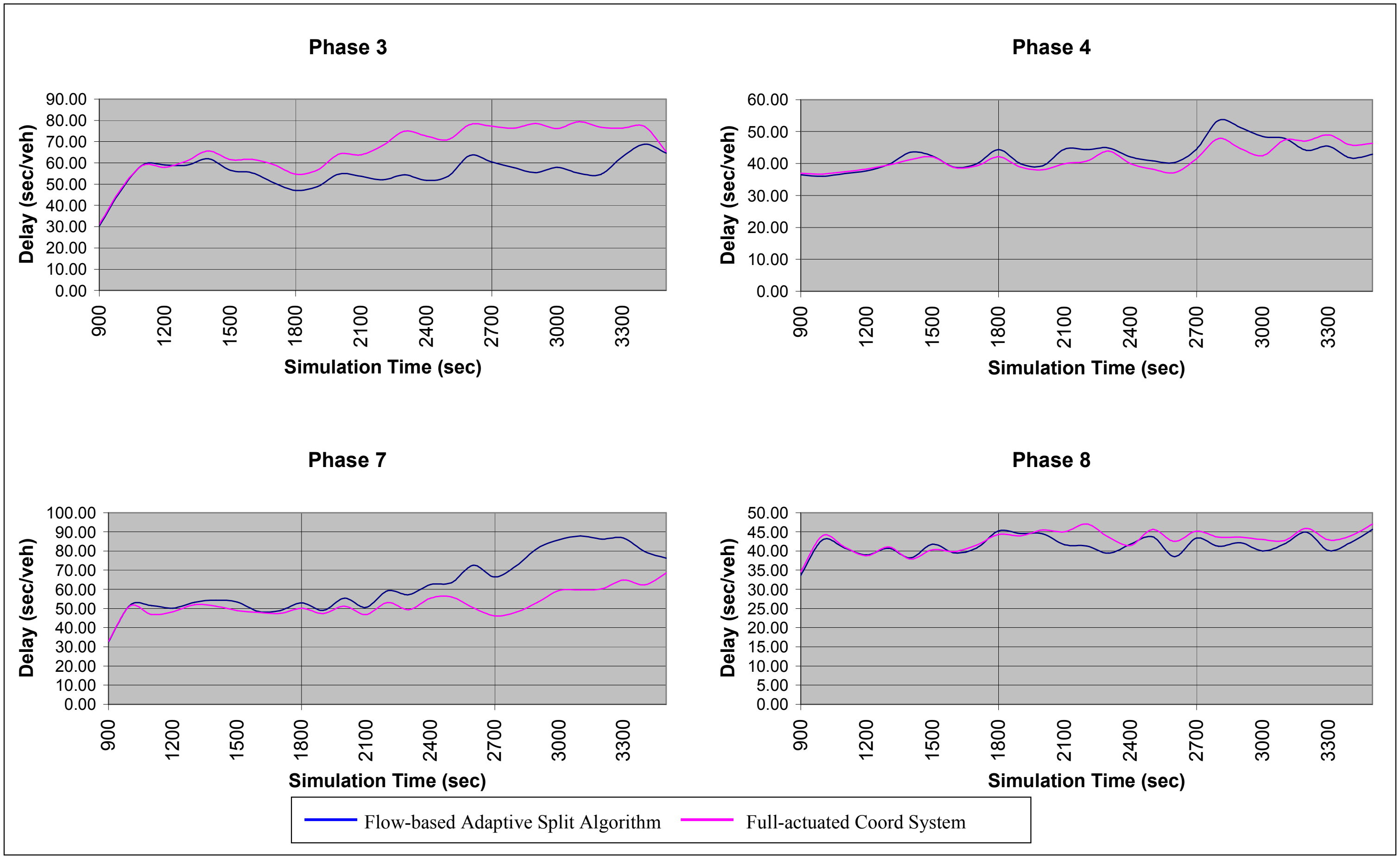


Figure 23 – Average delay for Scenario 2 – Phases 3, 4, 7 and 8

For the period of 1800 - 2700 seconds, phases 2 and 5, and phases 3 and 8 are affected with higher demand. Observation of the average delay graphs for both phases in conjunction with Table 9 indicate that phases 2 and 5 did not present significant improvement while the trend between phases 3 improving performance and phase 7 decreasing performance continued.

For the period of 2700 - 3600 seconds, only phases 1 and 6 are affected with higher demand. As can be noticed in the individual phase graphs for average, minimum and maximum split values found in Appendix C for respective phases, only phase 5 in the left side of the barrier (that includes phase 1, 2 and 6) received additional time in its split leading to a 8.11% lower average delay when the flow-based adaptive split algorithm was used, as shown in Table 10. Phases 1, 2 and 6 suffered significant increases in average delay for the period. Phases 3, 4, 7 and 8 exchanged extra time between themselves and received time from the left side of the barrier but were unable to stabilize the anomaly established since the 900 – 1800 seconds period of the simulation.

Table 9 – Average delay for Scenario 2 – 2nd period

1800-2700 (sec)				
Avg Delay				
Approach	**Phases**	**FACS ***	**FBASA ****	**% DIFF**
1	1	51.26	52.27	1.97%
	6	27.52	27.09	-1.56%
2	5	56.09	55.63	-0.81%
	2	28.75	28.70	-0.19%
3	3	67.12	53.32	-20.56%
	8	44.32	42.30	-4.57%
4	7	51.06	58.07	13.73%
	4	39.88	42.29	6.05%
Overall		40.04	39.10	-2.34%

* FACS Full-actuated coordinated system
** FBASA Flow-based adaptive split algorithm

Table 10 – Average delay for Scenario 2 – 3rd period

2700-3600 (sec)				
Avg Delay				
Approach	**Phases**	**FACS ***	**FBASA ****	**% DIFF**
1	1	53.24	57.30	7.63%
	6	30.05	31.33	4.26%
2	5	68.05	62.54	-8.11%
	2	32.43	33.13	2.16%
3	3	75.92	59.76	-21.28%
	8	44.20	42.37	-4.15%
4	7	58.04	80.29	38.35%
	4	45.78	46.66	1.92%
Overall		44.34	44.47	0.29%

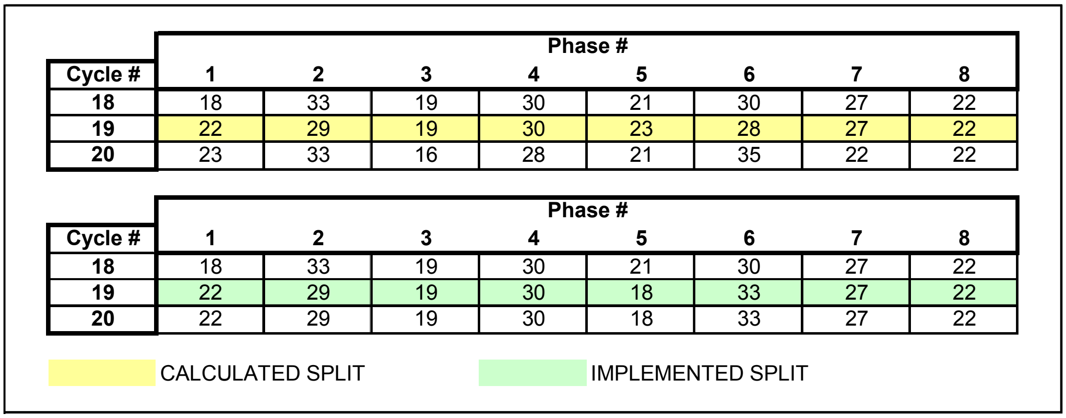
* FACS Full-actuated coordinated system
** FBASA Flow-based adaptive split algorithm

Scenarios 1 and 2 presented results for the average delay of individual phases demonstrating that the flow-based adaptive split was not able to consistently promote benefits to individual phases. After further analysis of the results and review of major functionalities of the algorithm, a discussion is presented next on the reasons that potentially inhibit the performance of the flow-based adaptive split algorithm.

5.3 Addressing the Problems

The ability of an individual phase to adapt to variation in traffic was achieved by the proposed algorithm, as results have shown. Unfortunately, results did not support the hypothesis that reallocation of *control logic* lost time would reduce intersection average delay due to problems that inhibit the overall performance. In order to better understand the origin of the inconsistent results it is necessary to revisit the functionality of the traffic signal controller as well as the functionality of the proposed algorithm.

Chapter 4 described the necessity of the flow-based adaptive split signal control algorithm to update new split parameters without disruption of coordination. The procedure became non-trivial when it was detected that the traffic signal controller would not accept the implementation of certain combinations of current and new splits without transitioning. Therefore the algorithm acknowledged the fact and promoted corrections to non-conforming splits. In Scenario 1 a total of 10.62% of the splits suffered corrections. In Scenario 2 a total of 14.57% of the splits suffered corrections. Table 11 presents an example in Scenario 1 seed 1, demonstrating that the split for cycle 19 was corrected by the algorithm. Notice that the implemented phase 5 and phase 6 sustained a 5 second differential from the calculated values. Even though a specific parameter was not establish to identify how much the corrections performed by the algorithm influenced the final results, it is believed that corrections do have the potential to inhibit the performance of the algorithm.



	Phase #							
Cycle #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
18	18	33	19	30	21	30	27	22
19	22	29	19	30	23	28	27	22
20	23	33	16	28	21	35	22	22

	Phase #							
Cycle #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
18	18	33	19	30	21	30	27	22
19	22	29	19	30	18	33	27	22
20	22	29	19	30	18	33	27	22

🟨 CALCULATED SPLIT

🟩 IMPLEMENTED SPLIT

Table 11 – Split update correction sample

During the simulation period close attention was designated to the behavior of the proposed Force-off method. The flow-based adaptive split signal control algorithm relies on stop bar detection to determine the number of vehicles utilizing each designated phase split. As expected, cycle by cycle variation of traffic promoted gap outs and force-offs. Since there is no actuated permissive period on the coordinated phase, only non-coordinated phases have the ability to gap out. Whenever a gap out occurred the following phase with demand received an additional green time. The algorithm utilizes the average of the previous 3 cycles' effective green time to calculate the volume to capacity ratio (v/c). Since the effective green time is calculated from splits and did not account for the extra green time received by phases that gap out, higher volume to capacity ratios (v/c) were calculated. At first, this was understood as a problem but later the conclusion was that with stop bar detection the utilization of the Fixed Force-off method helped the phase recognize the necessity of additional demand. In contrast, the Floating Force-off was also tested during simulation and since all the slack time in a cycle is directed to the coordinated phases; the algorithm struggled to recognize additional demand. Any change in split for the Floating Force-off methodology came only because the algorithm is triggered when a v/c is larger than 0.85, what does not necessarily mean additional demand.

That introduces the discussion on the anomaly presented in the analysis of results concerning phase 3 and phase 7. The coordinated phases would never gap out because the permissive period was not enabled. The coordinated phases are located before phases 3 and 7 in the NEMA ring and barrier structure. Phases 3 and 7 would not benefit from additional time provided by the controller functionality like phases positioned later in the ring (phases 4, 8, 1 and 5) would. Therefore when the flow-based adaptive split algorithm transferred time from phases 3 and 7 to help another phase, the ability to regain the time was impaired due to the lack of producing higher volume to capacity ratios (v/c) increasing the 3 cycle running average, as explained before.

In addition, the principle reason for the anomaly of phase 3 and phase 7 as well as the lack of consistent better performance for individual phases lies on the algorithm functionality itself. The algorithm allocated any unused time during the cycle to the phase that presented volume to capacity ratio (v/c) above the 0.85 threshold value. The problem with that is that for any given cycle more than one phase could be above the threshold value of 0.85. Therefore, a phase could be in need of additional time to serve its demand but would not be granted the benefit because the distribution of time was directed only to the phase with the highest volume to capacity ratio (v/c). The end result of the "unfair" distribution of "slack" time was the aggravation of the average delay. Going back to Scenario 1, phases 4 and 7 were the first ones to experienced higher demand. The algorithm performed its task of allocating additional unused time to both phases 4 and 7 and the "struggling" phase 3 was not able to compete with higher volume to capacity ratios (v/c) generated by phases 2 and 5 in the following 15 minutes of simulation. At the last 15 minutes of simulation, due to higher demand than other phases, phase 3 was able to produce higher volume to capacity ratios (v/c) in order to catch the attention of the algorithm and relatively reduce its average delay. The same rationale extends to Scenario 2, where phase 7 was the problematic phase.

CHAPTER 6

CONCLUSION AND REMARKS

The main objective of the flow-based adaptive split signal control was to adjust to both changing patterns over time and more importantly, to the stochastic nature of traffic. Unexpected traffic conditions can potentially produce unwanted and unnecessary delays when traffic signal systems operate under coordinated control. Therefore, the proposed flow-based adaptive split signal control algorithm was developed to address potential *control logic* lost time. Any unused time during the cycle should be reallocated to any phase that presented volume to capacity ratio (v/c) above the 0.85 threshold value. With that mechanism, the restrictive control logic imposed by current traffic signal controllers on non-coordinated phases would be relaxed. More than that, unused time from coordinated phases would be available to any phase.

Implementation of the proposed algorithm proved to be a challenge due to necessity of not disrupting coordination. The ability to implement any calculated split table was restricted by the traffic signal controller logic and potentially affected results. At the same time the strategy of focusing any potential "slack" time in the cycle solely to the phase with the highest volume to capacity ratio (v/c) above the threshold value of 0.85 led to inconsistent results. Nonetheless, future research can potentially improve the flow-based adaptive split signal control algorithm.

6.1 Recommendations for Future Research

The innovative methodology presented in this research brings opportunities for future research. The ability to manipulate the controller via its udp connection produced encouraging results that need to be further explored. The following are some recommendations that can potentially enhance the presented algorithm:

- distribution of available "slack" time among more than one phase for each cycle analyzed;
- exploration of different volume to capacity ratio (v/c) thresholds to trigger the modification of the split table;

- real time calculation of saturation flow rate values;
- utilization of actual green interval for the calculation of volume to capacity ratio (v/c) instead of previous cycle effective green time;
- modification of cycle time in real-time without disruption of coordination to address oversaturated conditions;
- utilization of one cycle data instead of a 3 cycle running average for the calculation of parameters;

The application of the proposed algorithm should be expanded to a network of intersections for evaluation of system wide performance. It is also necessary to acknowledge that only one timing plan with a 100 seconds cycle was examined and that different initial split distribution and scenarios need to be explored. Research findings on avoiding disruption of coordination can also be further explored to address minimization of current transition problems. Finally, capability of collecting flow data directly from the traffic signal controller can be utilized to test new or existing algorithms.

6.2 Conclusion

The flow-based adaptive split signal control was tested against a state-of-the-practice full actuated traffic signal control operating under a coordinated timing plan. The algorithm was not able to consistently produce lower average delay for phases approaching capacity saturation. Nevertheless, the research demonstrated the ability of an external agent to seamlessly control the traffic signal controller utilizing udp communication. It also introduced novel concepts of data gathering and manipulation, demonstrating that real-time flow data can be retrieved from the signal controller detector status itself with excellent results. At last, a better understanding of how to avoid transition in coordination was achieved.

Enhancement of the proposed algorithm is encouraged and potentially beneficial to minimize the everyday burden experienced by authorities in maintaining acceptable levels of traffic operation.

LIST OF REFERENCES

[1] M. C. Bell, and R. D. Bretherton, *Ageing of Fixed-Time Traffic Signal Plans*, Proc., IEE 2nd International Conference on Road Traffic Control, London, 1986

[2] Peter Koonce, Lee Rodegerdts, Kevin Lee, Shaun Quayle, Scott Beaird, Cade Braud, Jim Bonneson, Phil Tarnoff and Tom Urbanik, *Traffic Signal Timing Manual*, Federal Highway Administration, June 2008

[3] Rebecca Pearson, *ITS Decision – A Gateway to Understanding and Applying Intelligent Transportation Systems,* Institute of Transportation Studies at the University of California at Berkeley and Caltrans, 2001 http://www.calccit.org/itsdecision/serv_and_tech/Traffic_signal_control/trafficsig_report.html - visited Aug 2009

[4] Robert M. Garbacz, *Adaptive Signal Control: What to Expect*, ITS Cooperative Deployment Network, 2003

[5] A. J. Miller _1963_. *A computer control system for traffic networks*, Proc. 2nd International Symposium on the Theory of Traffic Flows, Paris, 1963, pp. 200-220, 1963

[6] H. G. Rosdolsky, *A Method for Adaptive Traffic Control*, Transportation Res. Vol. 7, pp. 1-16. Pergamon Press 1973. Printed in Great Britain.

[7] *SCATS 6 – Functional Description Manual* – Roads and Traffic Authority – Australia

[8] Neil R. Gross, *SCATS Adaptive Traffic System*, Transportation Research Board Adaptive Traffic Control Workshop, January 2000

[9] A. G. Sims and K. W. Dobinson, *The Sydney Coordinated Adaptive Traffic (SCAT) System Philosophy and Benefits*, IEEE Transactions on Vehicular Technology, Vol. 29, No. 2, May 1980.

[10] M. C. Royle and M. E. J. Harrison, *SCOOT Traffic Handbook – Executive Summary Description of Scoot*, Section 0414, February 2000

[11] R. Jayakrishnan, Stephen P. Mattingly and Michael G. McNally, *Performance Study of SCOOT Traffic Control System with Non-Ideal Detectorization: Field Operational Test in the City of Anaheim*, 80th Annual Meeting of the Transportation Research Board, Washington, DC, January 2001

[12] Peter T. Martin, *SCOOT, an Overview*, University of Utah Traffic Laboratory

[13] D. W. Dey, S. Fitzsimons, A. Morris and D. Ng, *Adaptive Traffic Signal Interconnect in Menlo Park and Sunnyvale, CA,* 2002

[14] Didier M.Valdes-Diaz and Alexander Paz, *Integration of Adaptive Traffic Control and Travel Information*, 83th Annual Meeting of the Transportation Research Board, Washington, DC, 2004

[15] Nathan H. Gartner, Farhad J. Pooran, and Christina M. Andrews, *Implementation and Field Testing of the OPAC Adaptive Control Strategy in RT-TRACS*, Transportation Research Board, Washington, DC, January 2002

[16] Lawrence C. Liao, *A Review of the Optimized Policies for Adaptive Control (OPAC)*, Institute of Transportation Studies, University of California, Berkeley, 1998

[17] Pitu Mirchandani and K. Larry Head, *A Real-Time Traffic Signal Control System: Architecture, Algorithms, and Analysis*, Transportation Research Part C: Emerging Technologies, Vol 9, No. 6, December 2001, pp. 415-432.

[18] K. Larry Head, Pitu Mirchandani and Steve Shelby, *The RHODES Prototype: A Description and Some Results,* Transportation Research Board, January 1998, pp. 1-12.

[19] Suvrajeet Sen and K. Larry Head, *Controlled Optimization of Phases at an Intersection*, Transportation Science, Vol. 31, No. 1, pp5-17, February 1997.

[20] *ACS-LITE - Adaptive Control Software,* Siemens Energy & Automation, Inc. Business - Unit Intelligent Transportation, http://www.itssiemens.com

[21] *Adaptive Control Software-Lite (ACS-LITE) - Implementation Template,* FHWA Resource Center, Operations Technical Support Team, September 2006

[22] Felipe Luyanda, Douglas Gettman, Larry Head, Steven Shelby, Darcy Bullock, and Pitu Mirchandani, *ACS-Lite Algorithmic Architecture Applying Adaptive Control System Technology to Closed-Loop Traffic Signal Control Systems*, Transportation Research Records 1856, 2003

[23] Kevin Fehon, *Adaptive Traffic Signals – Are We Missing the Boat?* ITE District 6 Annual meeting, 2004

[24] *Eagle EPAC300 Manual* – Siemens Energy and Automation Inc. – Austin, TX

[25] *NTCIP basedTS2/2070Controllers Manual* – Naztec Inc. – Sugar Land, TX

[26] *Advanced System Controllers ASC/3Programming Manual* – Econolite Control Products Inc., 2007

[27] Tom Urbanik, Scott Beaird, Doug Gettman, Larry Head, Darcy Bullock, Ed Smaglik, Rick Campbell and Matt Ablett, *Traffic Signal State Transition Logic Using Enhanced Sensor*, prepared for National Cooperative Highway Research Program, Transportation Research Board, National Research Council, 2003

[28] S. R. Sunkari, R. J. Engelbrecht, and K. N. Balke, *Advanced Coordination Features in Traffic Signal Controller,*Report 0-4657-1, Texas Transportation Institute, College Station, 2004

[29] R. J. Engelbrecht, S. Venglar, and Z. Tian, *Improving Diamond Interchange Operations Using Advanced Controller Features,*Research Report 4158-1, Texas Transportation Institute, October 2001

[30] Stephen Chiu and Sujeet Chand, *Adaptive Traffic Signal Control Using Fuzzy Logic*, Fuzzy Systems, Proc. 2nd IEEE International Conference on Fuzzy systems, Vol.2, pp. 1371-1376, San Francisco, CA, US, 1993.

[31] Agus Priyono, Muhammad Ridwan, Ahmad Jais Alias, Riza Atiq, O. K. Rahmat Azmi Hassan and Alauddin Mohd, *Application of LVQ Neural Network in Real-Time Adaptive Traffic Signal Control*, Jurnal Teknologi, Universiti Teknologi Malaysia, 42(B), pp. 29-44, Jun 2005

[32] Byungkyu Park and Myungsoon Chang, *Realizing Benefits of Adaptive Signal Control at an Isolated Intersection*, Transportation Research Records 1811, 2002

[33] Larry E. Owen and Charlie M. Stallard, *Rule-Based Approach to Real-Time Distributed Adaptive Signal Control*, Transportation Research Record 1683, 1999, pp. 95-101.

[34] Diakaki, C., Papageorgiou, M., Aboudolas, K., 2002. *A multivariable regulator approach to traffic-responsive network-wide signal control*. Control Engineering Practice 10, 183-195.

[35] R. Wunderlich, I. Elhanany, C. Liu, and Thomas Urbanik, *A Novel Signal Scheduling Algorithm With Quality of Service Provisioning for an Isolated Intersection*, IEEE Transactions on Intelligent Transportation Systems, Vol 9, No.3, 2008

[36] Edward J. Smaglik, Darcy M. Bullock, Thomas Urbanik, and Dan Bryant, *Evaluation of Flow-Based Traffic Signal Control Using Advanced Detection Concepts,* Transportation Research Record, Journal of the Transportation Research Board, No. 1978, pp.25-33, 2006

[37]E. J. Smaglik, *Enhanced Tactical and StrategicControl Methods for Traffic Signal Operation,* unpublished doctoral thesis, Purdue University, West Lafayette, Ind., 2005

[38] Edward J. Smaglik, Darcy M. Bullock, and Thomas Urbanik II, *Adaptive Split Control using Enhanced Detector Data*

[39] Edward J. Smaglik, Anuj Sharma, Darcy M. Bullock, Jim Sturdevant and, Gary Duncan, *Event Based Data Collection For Generating Actuated Controller Performance Measures,* Transportation Research Board Paper 07-1094

[40] *Standard Communications Protocol for Traffic Signals in California – Specification and Implementation Requirements* – Assembly Bill No. 3418 - Caltrans – November 1995

[41] *ATACID user manual*, Advanced Traffic Analysis Center, Upper Great Plains Transportation Institute, North Dakota State University

[42] Dustin DeVoe and Richard W. Wall, *Dynamic Objects for Smarter Pedestrian Control,* submitted for presentation at the 87th Annual Meeting of the Transportation Research Board

[43] *NTCIP 1103: Transportation Management Protocols (TMP) version 2.10b*, Washington, DC, AASHTO / ITE / NEMA, 2006

[44] Peter G. Furth, Burak Cesme and Theo H. J. Muller, *Lost Time and Cycle Length for an Actuated Traffic Signal,* submitted for presentation and publication to the Transportation Research Board, 2008

[45] Jun Xu, *The Development and Evaluation of a Detection Concept to Extend the Red Clearance by Predicting a Red Light Running Event*, Master Thesis – university of Tennessee, May 2009.

[46] Cuibi Liu, Itmar Arel, Thomas Urbanik, Airton Kohls, *Improved Truck Safety at Traffic Signals*, University of Tennessee, June 2009.

[47] Traffic Detector Handbook – Third Edition- Voume 1 – US Department of Transportation – Federal Highway Administration – October 2006

APPENDICES

Appendix A

Dynamic Objects Configuration (2-11)

Dynamic Object 2 - Detectors for adjacent lane of phases 2, 4, 6 and 8

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**2** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**2** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**2** – dynObjConfigOwner
Value: Detector2
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**2.1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.2.4.1.2.1
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**2** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 3 – System Cycle Time

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**3** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**3** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**3** – dynObjConfigOwner
Value: SystemCycleTime
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**3.1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.13.0

Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**3** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 4 – Split for phase 1

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**4** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**4** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**4** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**4**.**1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.1
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**4** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 5 – Split for phase 2

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**5** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**5** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object

OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**5** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**5**.**1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.2
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**5** – dynObjConfigStatus
Value: 1
Object Type: Integer

<u>Dynamic Object 6 – Split for phase 3</u>

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**6** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**6** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**6** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**6**.**1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.3
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**6** – dynObjConfigStatus
Value: 1
Object Type: Integer

<u>Dynamic Object 7 – Split for phase 4</u>

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**7** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation

OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**7** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**7** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**7.1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.4
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**7** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 8 – Split for phase 5

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**8** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**8** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**8** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**8.1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.5
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**8** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 9 – Split for phase 6

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**9** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**9** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**9** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**9**.**1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.6
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**9** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 10 – Split for phase 7

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**10** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**10** – dynObjConfigStatus
Value: 2
Object Type: Integer

Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**10** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**10**.**1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.7
Object Type: OID

Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**10** – dynObjConfigStatus
Value: 1
Object Type: Integer

Dynamic Object 11 – Split for phase 8

Action: Clearing any existing definition
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**11** – dynObjConfigStatus
Value: 3
Object Type: Integer

Action: Under creation
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**11** – dynObjConfigStatus
Value: 2
Object Type: Integer

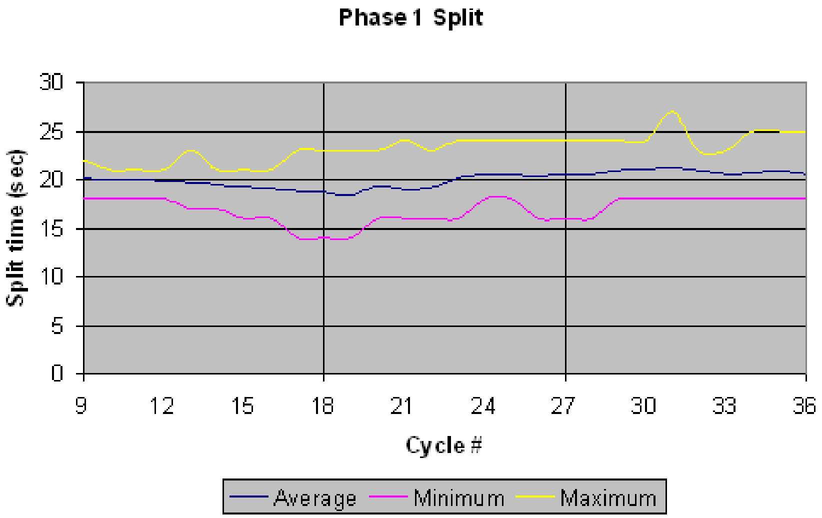
Action: Naming the dynamic object
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.1.**11** – dynObjConfigOwner
Value: Splits
Object Type: String

Action: Selecting the object identifier
OID: 1.3.6.1.4.1.1206.4.1.3.1.1.3.**11.1** – dynObjVariable
Value: 1.3.6.1.4.1.1206.4.2.1.4.9.1.3.1.8
Object Type: OID

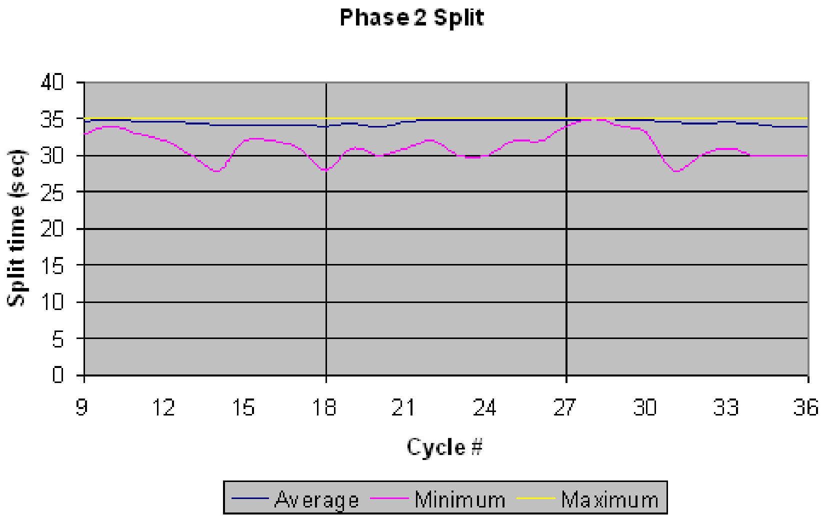
Action: Validating
OID: 1.3.6.1.4.1.1206.4.1.3.3.1.2.**11** – dynObjConfigStatus
Value: 1
Object Type: Integer

Appendix B

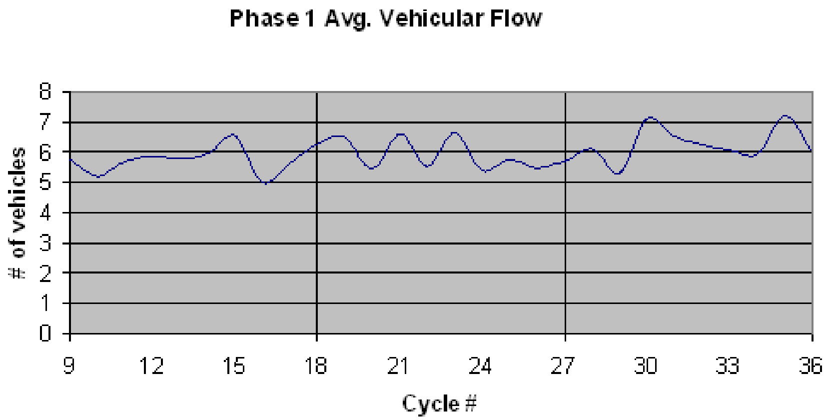
Scenario 1
Average, minimum and maximum split values
Average vehicular flow



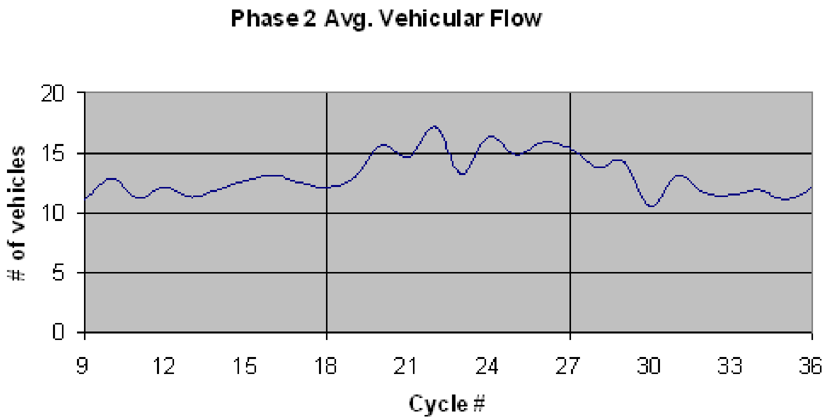
Phase 1 Split



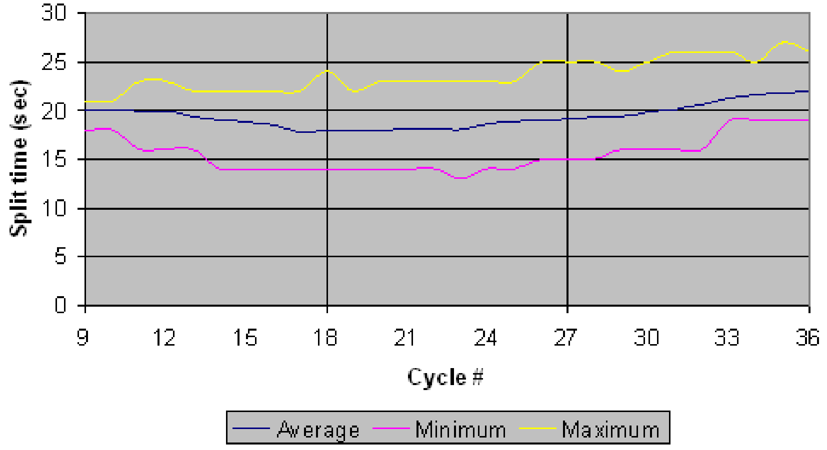
Phase 2 Split

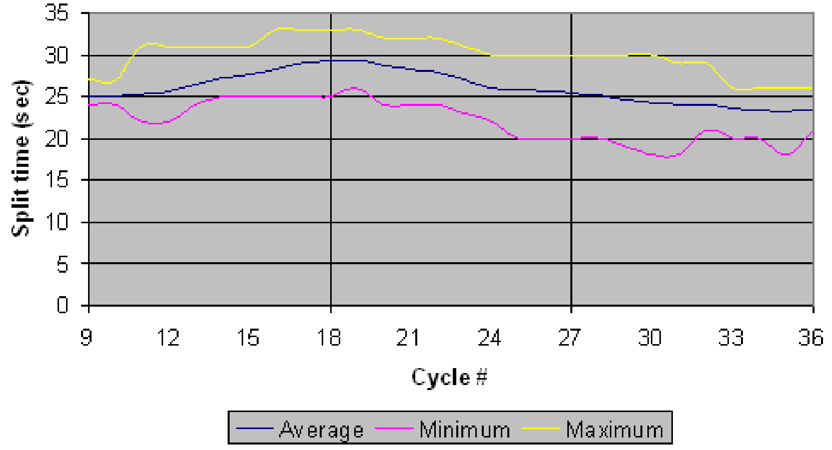


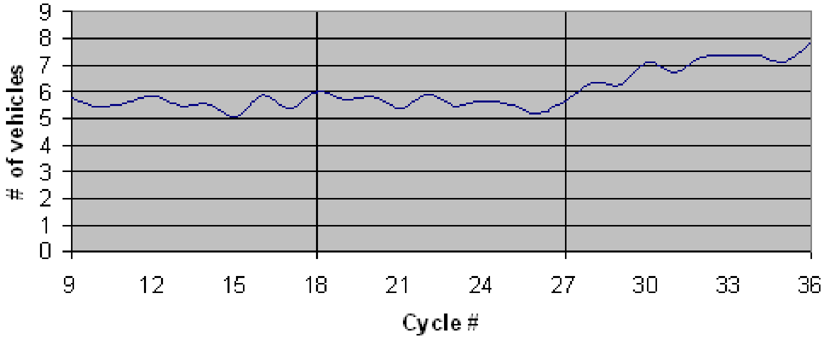
Phase 1 Avg. Vehicular Flow

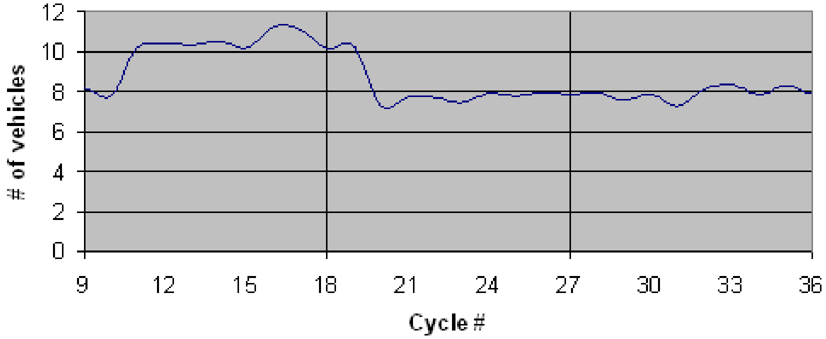


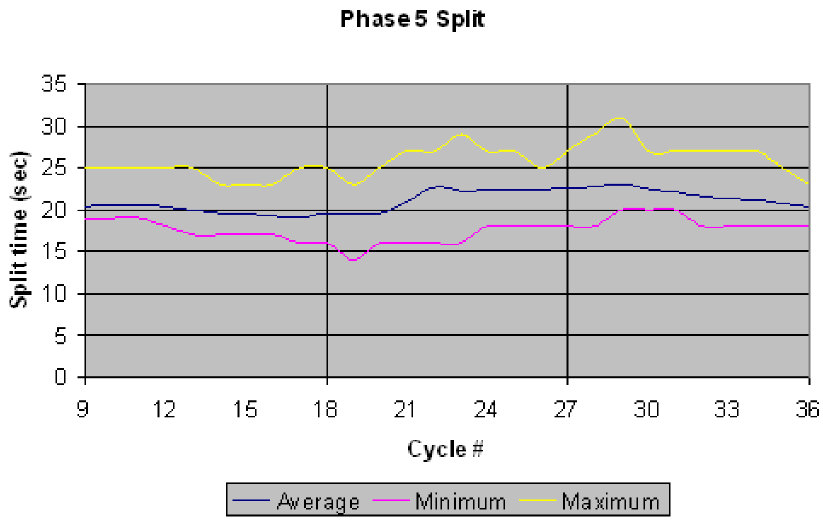
Phase 2 Avg. Vehicular Flow

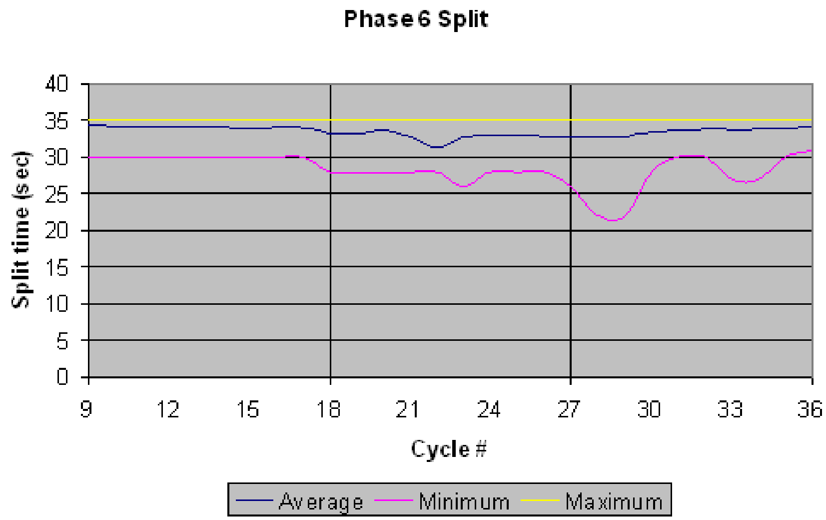

Phase 3 Split

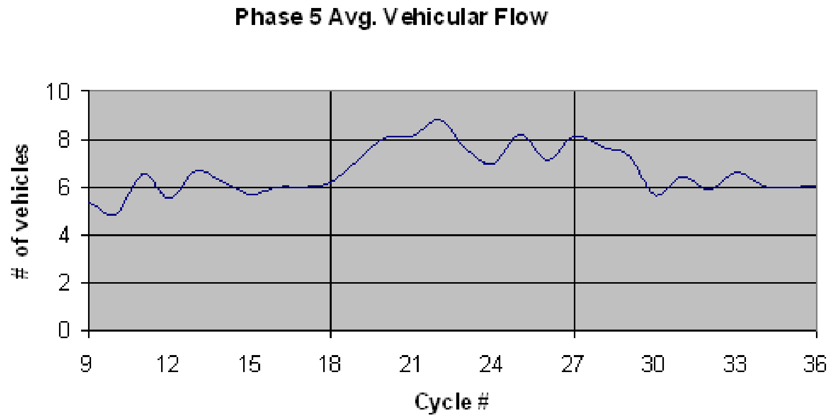

Phase 4 Split

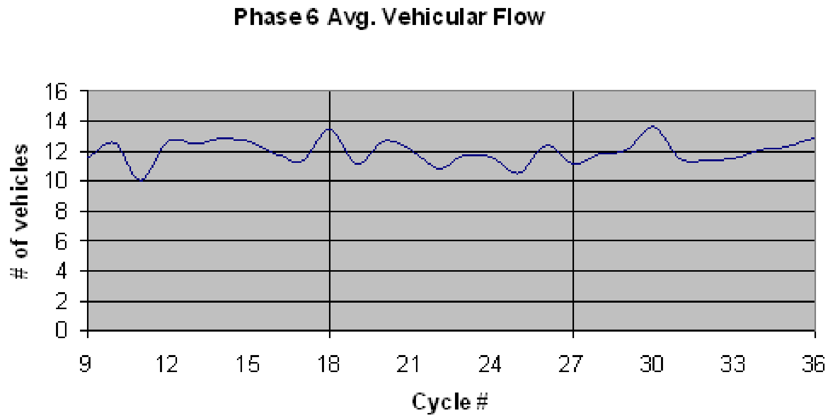

Phase 3 Avg. Vehicular Flow


Phase 4 Avg. Vehicular Flow

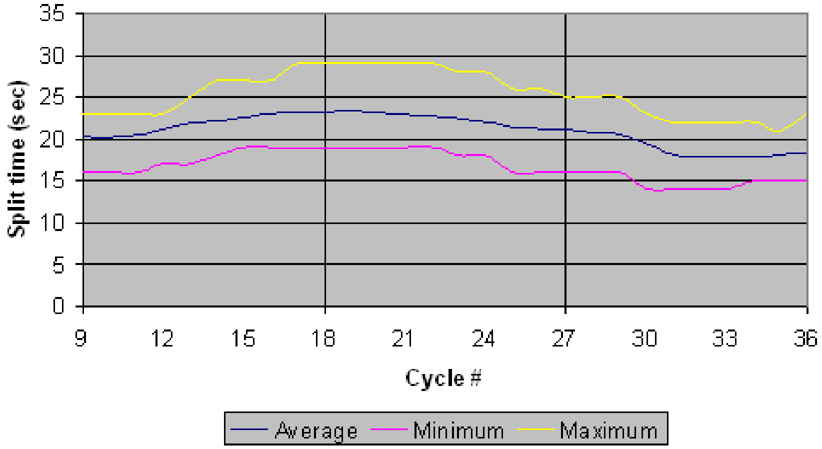

Phase 5 Split


Phase 6 Split

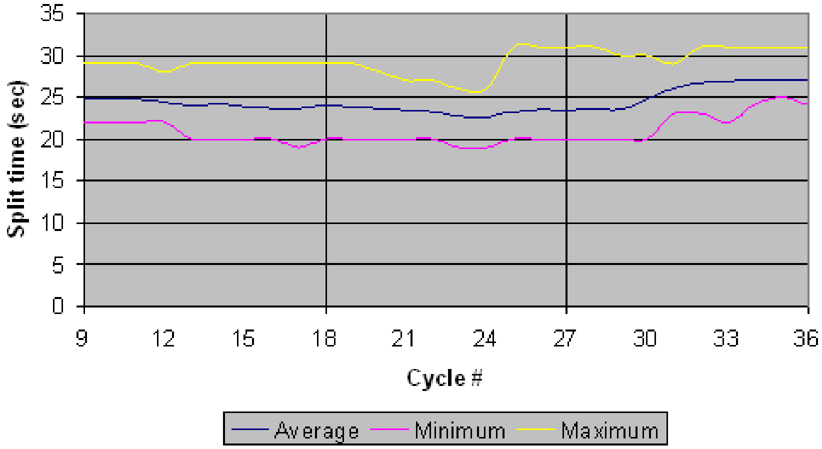

Phase 5 Avg. Vehicular Flow


Phase 6 Avg. Vehicular Flow

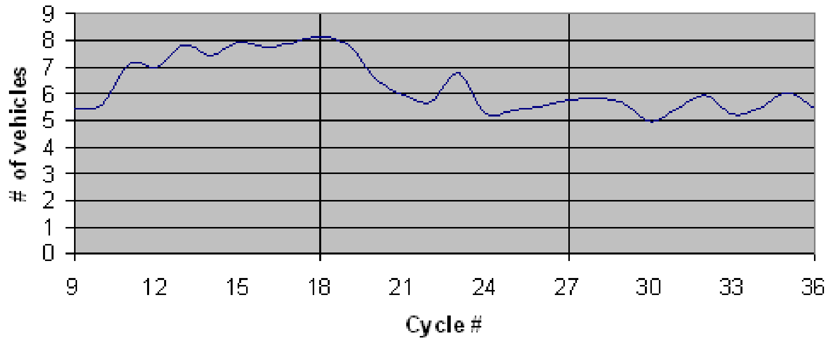
Phase 7 Split



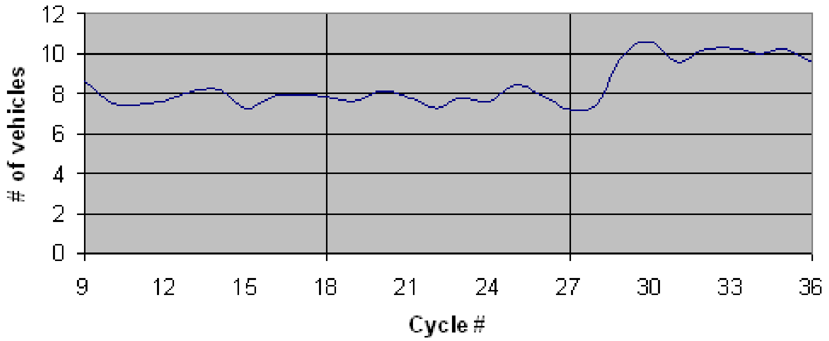
Phase 8 Split



Phase 7 Avg. Vehicular Flow

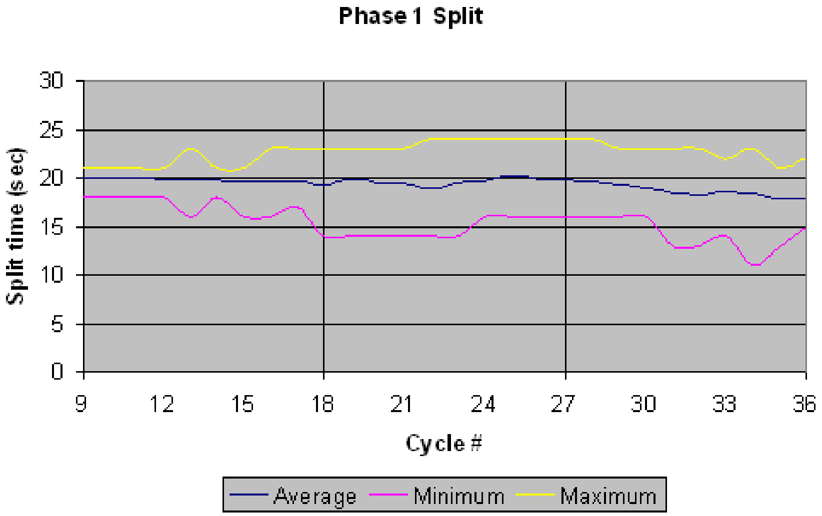


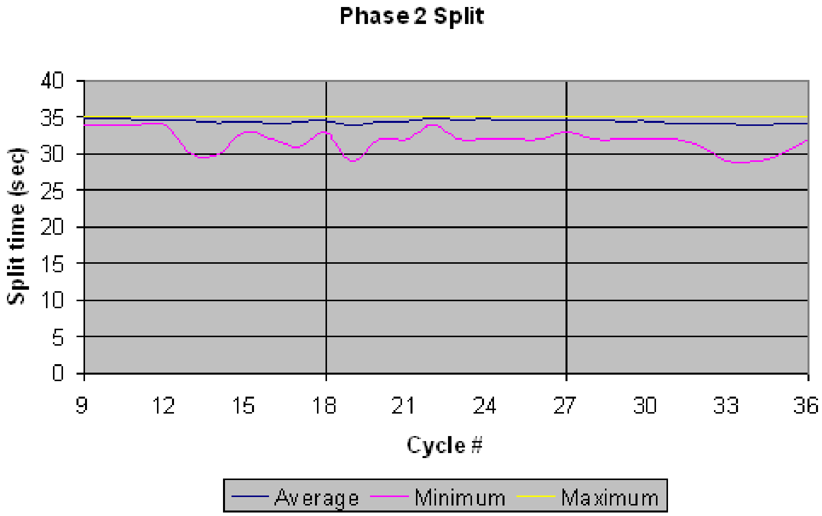
Phase 8 Avg. Vehicular Flow

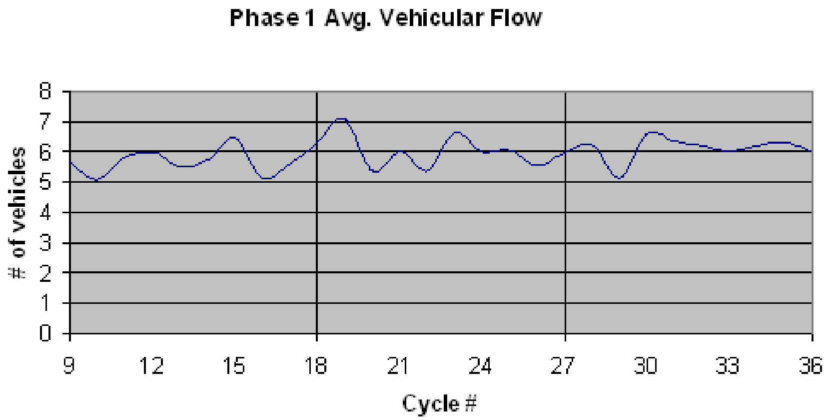


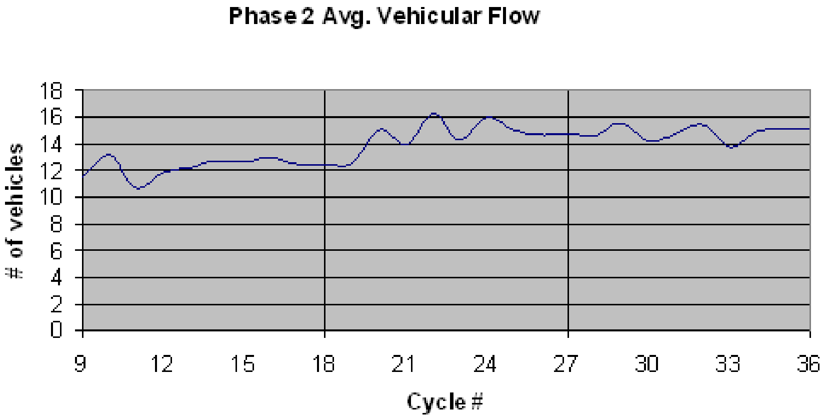
Appendix C

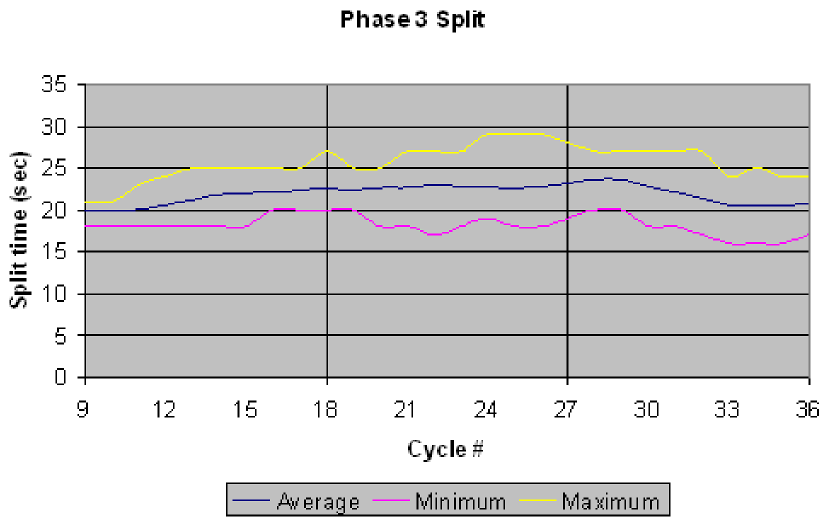
Scenario 2
Average, minimum and maximum split values
Average vehicular flow

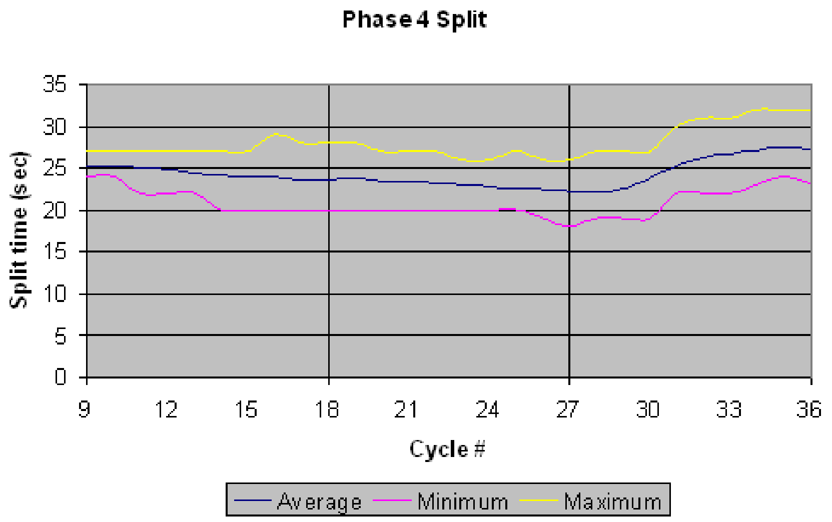

Phase 1 Split
Split time (sec)
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum

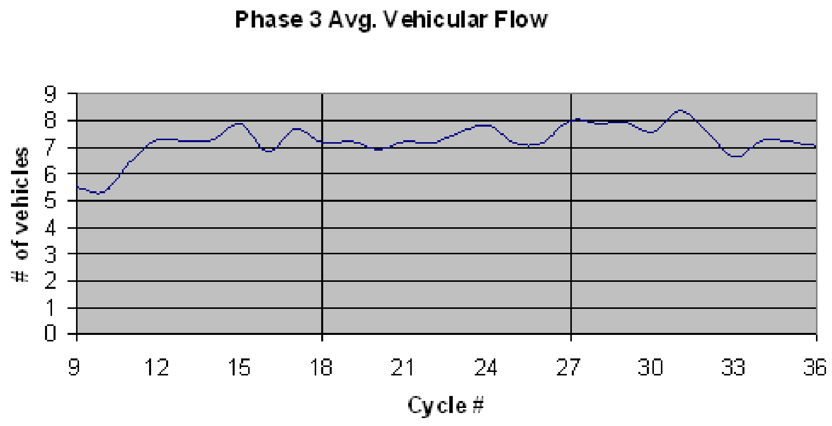

Phase 2 Split
Split time (sec)
40
35
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum

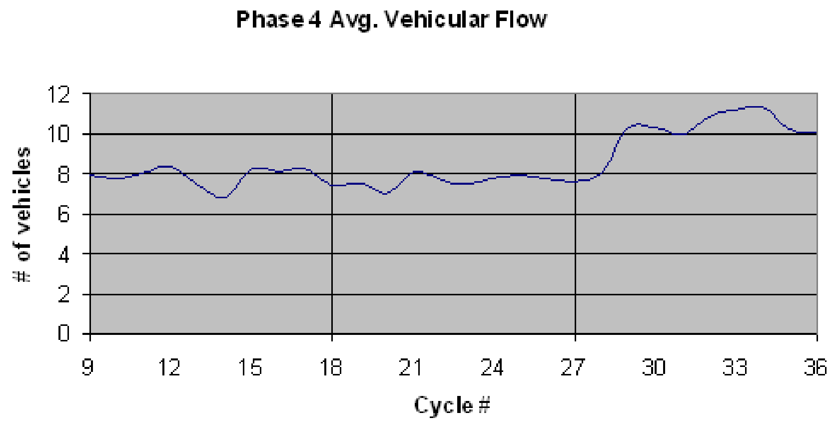

Phase 1 Avg. Vehicular Flow
# of vehicles
8
7
6
5
4
3
2
1
0
9
12
15
18
21
24
27
30
33
36
Cycle #

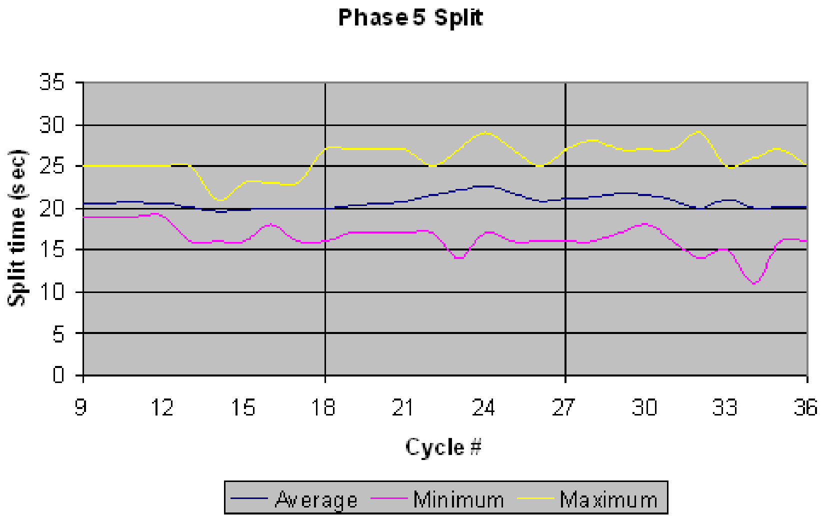

Phase 2 Avg. Vehicular Flow
# of vehicles
18
16
14
12
10
8
6
4
2
0
9
12
15
18
21
24
27
30
33
36
Cycle #

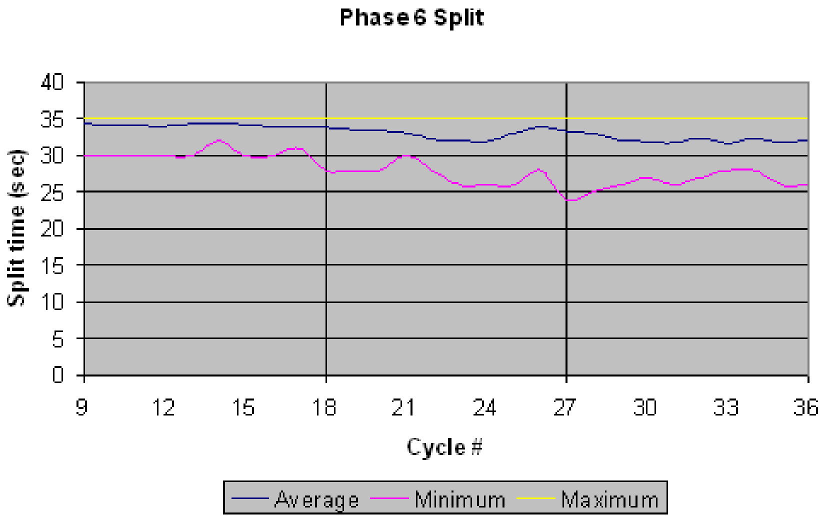

Phase 3 Split
Split time (sec)
35
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum

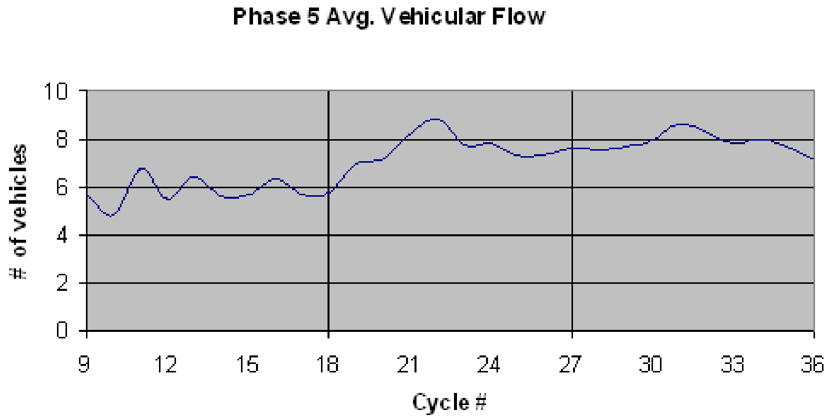

Phase 4 Split
Split time (sec)
35
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum

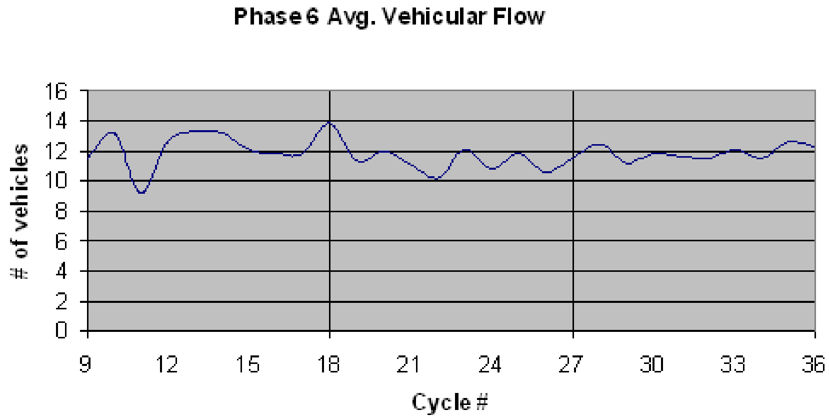

Phase 3 Avg. Vehicular Flow
# of vehicles
9
8
7
6
5
4
3
2
1
0
9
12
15
18
21
24
27
30
33
36
Cycle #

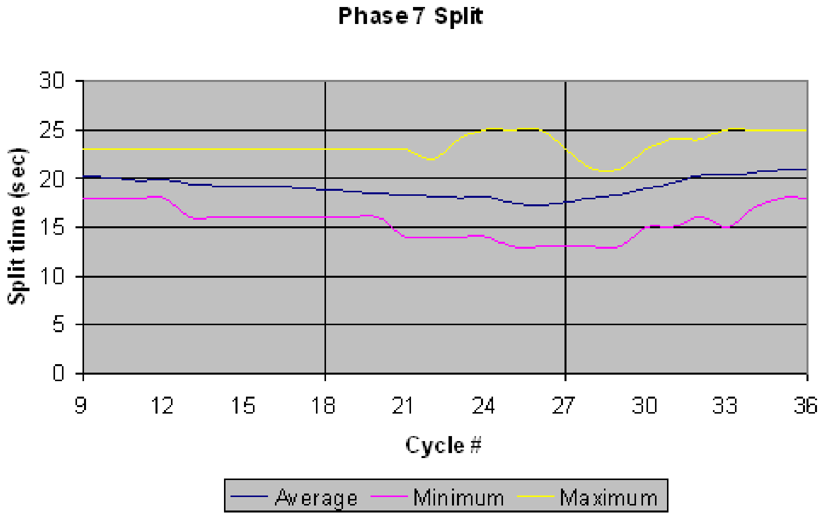

Phase 4 Avg. Vehicular Flow
# of vehicles
12
10
8
6
4
2
0
9
12
15
18
21
24
27
30
33
36
Cycle #

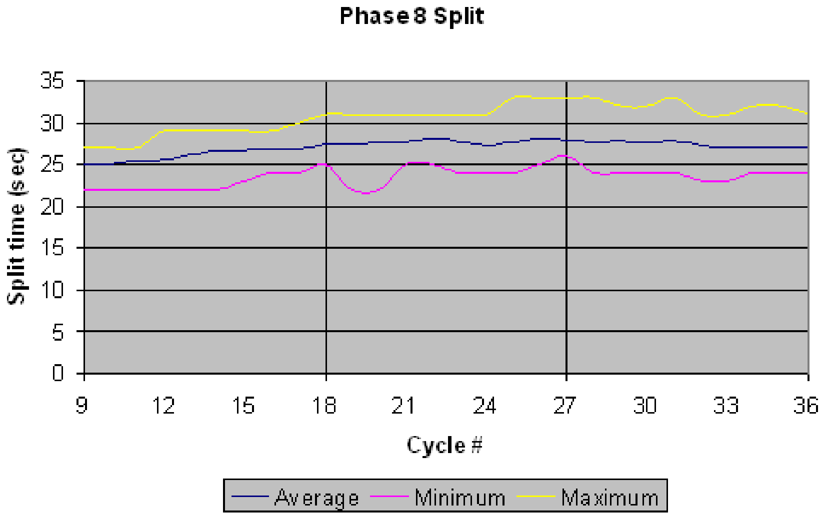

Phase 5 Split
Split time (sec)
35
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum

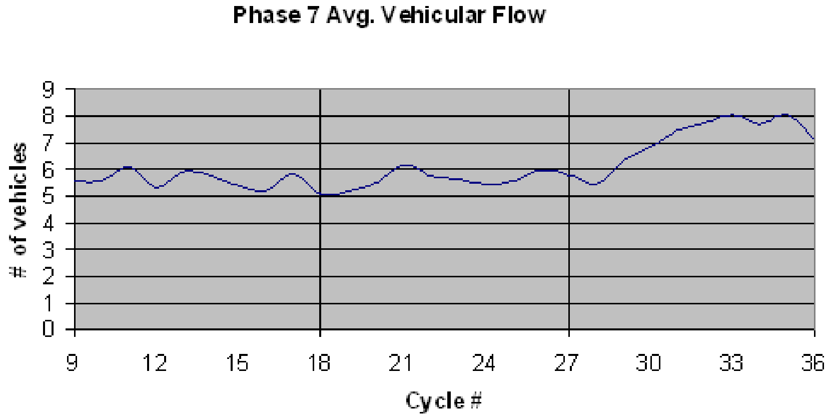

Phase 6 Split
Split time (sec)
40
35
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum

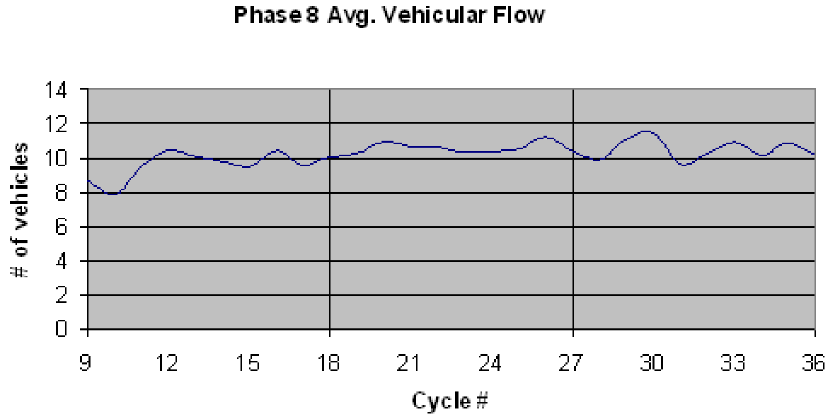

Phase 5 Avg. Vehicular Flow
# of vehicles
10
8
6
4
2
0
9
12
15
18
21
24
27
30
33
36
Cycle #


Phase 6 Avg. Vehicular Flow
# of vehicles
16
14
12
10
8
6
4
2
0
9
12
15
18
21
24
27
30
33
36
Cycle #


Phase 7 Split
Split time (sec)
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum


Phase 8 Split
Split time (sec)
35
30
25
20
15
10
5
0
9
12
15
18
21
24
27
30
33
36
Cycle #
Average
Minimum
Maximum


Phase 7 Avg. Vehicular Flow
# of vehicles
9
8
7
6
5
4
3
2
1
0
9
12
15
18
21
24
27
30
33
36
Cycle #


Phase 8 Avg. Vehicular Flow
# of vehicles
14
12
10
8
6
4
2
0
9
12
15
18
21
24
27
30
33
36
Cycle #

Appendix D

Split and volume data - Scenario 2 Seed 735

		Phase Split								Phase w/	Traffic Volume per Phase							
Time	**Cycle #**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Problem**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
900	**9**	20	35	20	25	20	35	23	22	7	3	14	4	10	6	12	8	7
1000	**10**	20	35	20	25	20	35	23	22	0	6	13	3	9	2	18	9	6
1100	**11**	20	35	20	25	20	35	23	22	0	6	9	9	10	7	7	2	9
1200	**12**	20	35	20	25	20	35	23	22	0	8	16	9	10	9	13	4	7
1300	**13**	20	35	23	22	20	35	23	22	3	0	20	9	6	4	23	5	7
1400	**14**	20	35	25	20	20	35	23	22	3	4	7	10	7	4	12	6	13
1500	**15**	20	35	25	20	20	35	22	23	8	7	13	8	6	4	16	4	7
1600	**16**	20	35	25	20	20	35	20	25	8	2	13	8	7	10	7	9	8
1700	**17**	20	35	25	20	20	35	20	25	0	5	11	10	8	1	16	7	9
1800	**18**	20	35	24	21	20	35	20	25	4	6	15	12	9	5	7	3	10
1900	**19**	20	35	25	20	20	35	20	25	3	6	18	6	6	3	15	3	10
2000	**20**	20	35	25	20	20	35	18	27	8	5	11	8	5	6	4	8	11
2100	**21**	20	35	25	20	20	35	18	27	0	6	16	4	6	3	20	6	7
2200	**22**	20	35	25	20	20	35	18	27	0	6	14	10	9	8	9	3	10
2300	**23**	20	35	24	21	20	35	18	27	4	6	12	10	8	9	9	8	10
2400	**24**	20	35	24	21	25	30	18	27	5	5	18	5	10	11	12	4	10
2500	**25**	20	35	24	21	25	30	16	29	8	6	8	7	6	10	13	6	15
2600	**26**	16	35	26	23	16	35	16	33	8	7	16	10	9	2	11	6	16
2700	**27**	16	35	26	23	16	35	17	32	7	7	19	10	5	10	4	6	9
2800	**28**	16	35	26	23	16	35	19	30	7	6	21	11	6	10	11	7	9
2900	**29**	16	35	26	23	25	26	19	30	5	8	12	8	14	8	18	8	10
3000	**30**	20	35	22	23	20	35	17	28	1	9	11	7	10	10	15	8	12
3100	**31**	23	32	22	23	20	35	17	28	1	9	17	9	6	7	15	7	7
3200	**32**	23	32	22	23	20	35	21	24	7	6	19	7	9	10	8	7	10
3300	**33**	20	35	22	23	20	35	21	24	2	9	17	9	10	9	9	9	11
3400	**34**	20	35	20	25	20	35	21	24	4	7	16	7	13	6	10	9	10
3500	**35**	20	34	20	26	19	35	22	24	7	7	11	6	9	5	10	9	11
3600	**36**	20	33	20	27	18	35	23	24	7	9	10	8	9	8	11	9	9

🟩 MODIFIED PHASES

Appendix E

Distribution of corrections during split table update

of Occurrences
Scenario 1

	Simulation Period		
Seed #	**900-1800**	**1800-2700**	**2700-3600**
1		1	
35		1	1
70		1	2
105			1
140	1	1	
175		2	
210			3
245		1	2
280	1	2	1
315			3
350		1	1
385		1	
420	1	3	2
455		1	1
490	1	3	2
525		1	1
560	1	2	2
595		3	1
630		2	1
665	1	4	2
700	2	1	
735			2
770		1	1
805		1	
840	2		
875		1	
910		2	
945	1	1	2
980	1	3	
1015		1	2
TOTAL	**12**	**41**	**33**
		86	

of Occurrences
Scenario 2

	Simulation Period		
Seed #	**900-1800**	**1800-2700**	**2700-3600**
1		2	3
35		1	
70	1	1	1
105		1	
140		2	1
175		1	2
210	1	2	4
245	1		3
280	2	1	2
315		1	5
350		2	1
385	2		1
420		2	1
455		2	2
490	1	3	2
525	1	1	2
560	1	1	5
595	1	4	1
630	1	2	2
665	1		3
700	3		4
735			2
770		2	2
805			5
840	2		1
875		1	
910		2	1
945	1	1	2
980	1	1	1
1015		1	2
TOTAL	**20**	**37**	**61**
		118	

VITA

Airton Gustavo Kohls was born in May 12th, 1973 in Santa Cruz do Sul, Rio Grande do Sul, Brazil. He graduated from the University of Tennessee in 1995 with a B. S. in Civil Engineering. He went back to Brazil where he worked as a Civil Engineer for his hometown, participating in the development of the city's infrastructure. Among many important projects, he was proud to be actively involved with the construction of the Santa Cruz do Sul International Raceway. He came back to the United States with his family in August 2006 and got his Masters degree as well as his PhD at the University of Tennessee with concentration in Transportation. He is interested in transportation operations and transportation infrastructure.