1995

Evaluation of Automatic Vehicle Specific Identification (AVSI) in a traffic signal control system

Alireza Kamyab
Iowa State University

Evaluation of Automatic Vehicle Specific Identification (AVSI)

in a traffic signal control system

by

Alireza Kamyab

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirement for the Degree of

DOCTOR OF PHILOSOPHY

Department: Civil and Construction Engineering
Major: Civil Engineering (Transportation Engineering)

Approved:

Signature was redacted for privacy.

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1995

UMI Number: 9531753

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

To Khojasteh, Mariam, Somayeh, and Ahmad

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 - INTRODUCTION

Intelligent Transportation Systems (ITS) involve the use of information, communications, and computer technology to update and enhance the capabilities of transportation systems (*1*, p. 3). Incorporating advanced technology into traditional transportation systems provides completely new functions to facilitate more productive and safe transportation of people and goods. Thus, ITS provides a new paradigm for planning, managing, operating, and maintaining transportation systems. The subject of this dissertation is to evaluate a new ITS concept for enhancing traffic signal control. The ultimate deployment of the concept is dependent on future incorporation of advanced technology into traffic control and vehicle systems.

The general emphasis of the United States ITS program, and programs throughout the world, is to promote safe and productive transportation systems that complement a multimodal environment through the application of technology (*2*, p. 1). Further, ITS is intended to promote the integration of transportation components into an integrated system. For example, vehicles, the roadway, and traffic control devices should be integrated to work as complementary components of a seamless system. This dissertation focuses on one aspect of the integration of vehicle and traffic control systems. It examines improved traffic signal control capabilities, which are supported by communication between the vehicle and traffic signal control, thus more highly integrating the operation of the vehicle and traffic signal control.

Technology allows the identification of specific vehicles and their location in the traffic stream, and information related to their performance can either be inferred from vehicle specific information or communicated directly from the vehicle to the roadside. For example, vehicles equipped with radio frequency transponders could communicate a variety of information to the roadside, including location, speed, classification of the vehicle, vehicle acceleration and deceleration performance, vehicle mechanical condition, and even whether the driver of the vehicle intends to turn right or left or travel through the next intersection. Further, other technologies are able to automatically gather the same or similar vehicle data. For example, current image processing technology can locate and classify vehicles. Current traffic signal systems and traffic control logic are, however, based only on the detection of the vehicle in the area of the signal and have previously not take into account in their control logic this higher order vehicle specific information provided by Automatic Vehicle Specific Identification (AVSI).

AVSI is a generic name for advanced vehicle detection systems. By automating the identification of vehicles by sensing the presence of vehicles with roadside detection sites or readers, AVSI is assumed to provide types and arrival times of the approaching vehicles to a local microprocessor-based traffic signal controller.

The fundamental question researched is whether there are benefits to making traffic signal control timing decisions (e.g., phase and cycle length, and green extension) based on vehicle specific information. If there are benefits to enhancing traffic control logic to take into account vehicle specific information, this research develops a measure of effectiveness to evaluate these enhanced capabilities. However, before the fundamental research question (what are the benefits of vehicle specific information) can be addressed, it is necessary to create traffic control logic to effectively apply AVSI to traffic signal control.

Because traffic signal control systems do not exist which take into account vehicle specific information and include Vehicle Specific Adaptive (VSA) control logic, a computer simulation is employed to obtain measures of effectiveness for such a system. Available traffic signal control and traffic computer simulations include existing control strategies and incorporate existing vehicle detection technology into underlying traffic control logic. Therefore, no known and available simulation model provides a viable testbed for the evaluation of AVSI in traffic signal control systems and thus one is constructed to achieve the research objectives.

1.1 Research Overview

The purpose of the simulation model is to provide a testbed to evaluate the merit of signal control supported by AVSI. It is intended to be a research tool and not a tool for conducting the design of traffic signal timing plans. An important and unique part of the simulation model developed is the VSA traffic signal control logic incorporated within the model. Implicit to the research is the assumption that the VSA control logic design is efficient. Future and more efficient control logic may result in superior performance.

The research tests the use of AVSI through the use of a relatively simple traffic control environment. The case study modeled in the simulation is a signalized isolated intersection with four phases. The simulation developed is a microscopic simulation (treating each vehicle as a unique entity) and assumes that headways between vehicles remain static throughout the simulation. The simulation periodically modifies the timing cycle splits between phases based on upstream traffic flow characteristics. At the end of each phase, the simulation determines whether to extend the green time based on a heuristic decision-making process. Although the heuristic is complex, in conceptual terms it compares the likely reduction in delay caused by extending the green and not stopping the approaching slow accelerating vehicle (e.g., trucks) versus the delay caused by reducing the green time available to other approaches. If delay is reduced by allowing a vehicle with slow acceleration performance to pass the intersection, green time is extended.

The VSA traffic signal control logic at the case study intersection significantly reduces average stopped delay when compared to traditional pretimed traffic control. Predominantly, the reduction of stopped delay is attributable to reducing the probability of stopping vehicles with poorer than average acceleration performance, although periodically adjusting the cycle splits to better conform with existing traffic patterns reduces delay for all traffic. To make this comparison, the model includes conservative assumptions regarding relative vehicle acceleration performance and assumes a modest difference in the acceleration performance between trucks and automobiles. The benefits of AVSI in traffic control systems increase when the difference in average acceleration performance between trucks and automobiles is increased. It is also believed that benefits would increase in traffic control systems with several coordinated traffic signals.

1.2 Contribution to the State of the Art

The application of AVSI in a traffic signal control system has not previously been reported in the literature. This evaluation of AVSI in traffic control systems has generated three significant contributions. First, the research explores the potential benefits of AVSI as a system for collecting traffic information as input to traffic control systems. If adoption of AVSI-dependent traffic control logic proves beneficial, then the evolution of traffic data collection systems with greater functionality is promoted. Second, a VSA signal control strategy is unavailable in the literature, and one is developed and described in this dissertation. Third, a new simulation model which incorporates VSA control logic is developed.

1.2.1 Traffic Information Collection Systems

One of the primary requirements of an effective adaptive control logic is accurate and advance traffic data (*3*). It is extremely difficult to gather adequate advance traffic data with common traffic detector technology. Therefore, there is an interdependence between the evolution of traffic information collection systems and the ability to perform more efficient adaptive control. That is, efficient adaptive control requires accurate advance traffic data (e.g., in the case of the issue being researched, it even requires vehicle specific information), and to justify the deployment of data collection systems requires evidence that the information used in traffic control will be beneficial. Because the research reported in this dissertation found significant benefits are possible through adaptive control logic based on vehicle specific information, it promotes the deployment of technology to allow more robust traffic data collection systems with greater functionality.

1.2.2. VSA Traffic Signal Control Strategy

Existing research of adaptive control systems considers the entire traffic stream when adjusting signal timings and does not consider differences in the performance of vehicles within the traffic stream (*3-6*). Traffic signal timing is adjusted regardless of the type of vehicle in the traffic stream.

The VSA traffic control logic reported in this dissertation makes cycle split adjustments and green time extension decisions based on individual vehicle performance. Since heavy vehicles have poorer acceleration performance, traffic signal time adjustments are made to provide priority to trucks and allow them to clear the intersection. This research illustrates that VSA control strategies reduced stopped delay at the intersection for both heavy vehicles and automobiles.

1.2.3 New Microscopic Simulation Model

A new microscopic simulation model is developed to provide a testbed for the evaluation of AVSI in traffic signal control systems. Although several other microscopic models exist, none has the capability to incorporate VSA traffic control logic. The model developed is a research tool for use only in evaluation and is applied to only one intersection (but one with common geometry). It does, however, provide a framework for the development of future vehicle specific traffic control systems and traffic signal timing plan computer analysis tools.

1.3 Document Organization

This dissertation consists of six chapters. The first chapter includes a brief description of AVSI, the objectives of the research, and an overview of the research. This chapter also identifies the three significant contributions of the research. Chapter two is a review of the related research and related traffic signal simulation models. The third chapter presents the problem statement. In chapter four the methodology employed to evaluate AVSI in traffic signal control systems is described. The fifth chapter presents the simulation results and includes a case study example and an analysis of the sensitivity of the model to parameter changes. Chapter six presents the conclusions of the research and recommendations for future research.

The dissertation includes four appendixes. Appendix I presents the derivation of the negative exponential distribution from the Poisson distribution. Appendixes II and III include listings of the computer simulation code. Appendix IV contains the delay study field data collection worksheets. Field data were collected to validate the result of the simulation.

CHAPTER 2 - LITERATURE REVIEW

The purpose of this research is to explore the benefits of AVSI traffic information in a traffic signal control system. Computer simulation is an appropriate tool for this purpose, however, the incorporated traffic signal control systems of the existing simulation models are incapable of utilizing AVSI information at intersections. Therefore, a new simulation model with an incorporated VSA traffic signal control strategy is developed to evaluate the applicability of AVSI traffic information at an isolated intersection. This chapter reviews the related studies to justify the employed methodology.

The related studies can be divided into three categories which are reviewed in the four sections of this chapter. AVSI is considered to be an advanced vehicle detector system in this research. Thus, in the first section vehicle detector systems are reviewed. The second section examines traffic signal simulation models. The third and fourth sections review adaptive control strategies. The chapter closes with some concluding remarks.

There are no studies in applications of AVSI to traffic signal control systems. This research has the potential to begin to fill this gap in the literature.

2.1 Vehicle Detectors

Traffic sensing is accomplished by a device commonly referred to as a detector. Detectors are an important part of actuated signal control systems. For example, the decision of local controllers to extend green time depends upon data provided by detectors. The performance level of a traffic control system depends primarily upon the accuracy of data which detectors provide at an intersection. This is particularly true in adaptive control systems, because their effectiveness depends largely on accurate and advance data (*3*).

Primarily, detectors operate on two concepts: mechanical-force or energy-pattern change principles (*7*, p. 5.3). Outdated pressure detectors and pedestrian pushbuttons are examples of detection by the principal of mechanical force. Inductive loops, magnetic, and microwave detectors are classified under the energy-pattern change concept.

The priority control system for emergency and transit vehicles sometimes use a special-purpose high-intensity light detector. This type of detector utilizes a high-intensity light relayed at a specific frequency from a transmitter mounted on the vehicle and a detector mounted on the traffic signal mast (*8*, p. 82; *9*).

The most commonly used detectors are inductive loops. Installation is performed by embedding wire loops in the pavement. Embedding the wire requires sawcuts of the pavement, which in turn weakens the roadway and increase the need for road surface maintenance. The maintenance associated with inductive loop detectors is not limited to road surfacing; the imbedded loops themselves should also be maintained regularly.

New vehicle detection systems are designed to avoid the maintenance costs associated with the old detectors. For installation of these detectors, there is no need to interrupt traffic operations. Some of the newest detectors available in the market are Autoscope (*10*), Remote Traffic Microwave Sensor (RTMS) (*11*), and a Dual Mode (*12*) which use image processing, microwave, and radar and utrasonic technologies, respectively.

One of the advantages of these new detectors is they replace the inductive loops. In Ontario, Canada, the RTMS has been considered as an option to replace loop detectors to enhance traffic monitoring conditions at signalized intersections (*13*). A single Autoscope system, a video detection system, is capable of replacing up to 32 inductive loops. The Dual Mode detector also contains a microprocessor and memory which can accumulate data for periodic polling.

Except for Autoscope, which has the capability of recognizing different types of vehicles within a traffic flow, none of the previously mentioned detectors has this potential. With this capability and tested reliability (*14*), Autoscope has the ability to improve traffic flow at an intersection in the adaptive control system.

The classification of vehicles within a traffic flow adds a new challenge to research in the area of adaptive control strategy which presently examines total traffic volume in its signal timing adjustment algorithm. An adaptive control strategy which considers vehicle classifications information in its signal timing is referred to here as a VSA traffic signal control logic. By adjusting cycle splits at an intersection based on performance characteristics of vehicles in a traffic stream, a VSA control strategy is shown to be capable of reducing stop delay. VSA control logic is a new concept which is developed in this research.

2.2 Simulation Models

Simulation is the process of modeling the operation of an actual system. Its purpose is to provide a better understanding of the behavior of actual systems and to evaluate the operation of potential system designs. Traffic simulation on digital computers has attracted considerable interest since the late 1950s (*15*). Since then, many traffic simulation models have been constructed to evaluate traffic operation for roadway facilities at a microscopic or macroscopic level of analysis. In an effort to combine the various categories of simulation models in an integrated, coherent manner, FHWA (Federal Highway Administration) developed the TRAF systems (*16*). This section reviews NETSIM and FREFLO, two of TRAF systems simulation models, and five other non-TRAF simulation models.

NETSIM, TEXAS and EVAPAS are examples of microscopic traffic control simulation models which are reviewed in the first subsection. NETSIM is a network simulation model, while TEXAS and EVAPAS models are capable of evaluating the performance of a single intersection. The second subsection reviews the macroscopic models which include TRANSYT-7F and FREFLO. TRANSYT-7F is the most frequent used arterial simulation and signal optimization model. FREFLO is a freeway simulation model. These simulation models are commercially available; however, there are many other simulation models which remained at a research level (*17-20*). Two of these non-commercial simulation prototypes designed for ITS applications are reviewed in the last subsection.

2.2.1 Microscopic Models

NETSIM is one of the most detailed network simulation models available for street networks. It is a microscopic, stochastic simulation model. Vehicles are represented individually and can be identified as passenger cars, buses, or trucks. Vehicles move each second according to car-following logic while responding to traffic control devices.

NETSIM does not perform design tasks and has no optimization capability for signal timing. The current version of this simulation model is referred to as the TRAF-NETSIM model (*21*).

The NETSIM model capable of simulating bus preemption is called NETSIM/BPS (*7*, p. 3.39). The implemented preemption logic extends green time or terminates red time for the phases which service an oncoming bus. The NETSIM/BPS can only simulate preemption systems with pretimed controllers.

There are other microscopic traffic simulation models such as Traffic Experimental and Analytical Simulation (TEXAS) and Enhance Value Iteration Process Actuated Signals (EVAPAS). These two models are designed for actuated as well as pretimed, controlled isolated intersections. EVAPAS is also an optimization model capable of determining optimum signal timing for various types of traffic control systems at isolated intersections (*22*, p. 31).

With the TEXAS model, drivers and vehicles are classified individually. The model allows up to 15 vehicle classes (i.e., small, large, bus, full trailer, sport car) and up to 5 driver classes (i.e., aggressive, average, slow) (*23*). The particular driver characteristics and vehicle generation are treated stochastically in this model.

In EVAPAS each vehicle type is assigned a particular acceleration rate which is analogous to the driver classifications in the TEXAS model. The average driver in the TEXAS model is, for instance, comparable to the uniform acceleration rate in the EVAPAS model. The vehicle processing in the EVAPAS and NETSIM models is constructed according to a linear car-following model (*24*), while in the TEXAS model each vehicle processes through an intersection in a fixed, discrete time increment (7, p. 3.21).

2.2.2 Macroscopic Models

FREFLO is a macroscopic simulation model for freeway traffic. The model is based on a conservation equation and an equilibrium speed-density relationship with a dynamic speed equation (*16*). FREFLO can model both the regular and HOV (High Occupancy Vehicles) lanes of a freeway. Incidents are represented in this model by lane reduction or in the form of capacity restraint on the freeway section.

TRANSYT-7F (Traffic Network Study Tool), is one of the most widely used models in the United States and Europe for signal network timing design (*25*, p. 285). It is designed to optimize signal timing on coordinated arterials and grid networks with either pretimed systems and fixed-phase sequence. Although this model is essentially a traffic signal timing optimization program, it also contains a traffic simulation model. Unlike the previously mentioned models, TRANSYT-7F is a macroscopic model which considers the entire intersection or arterial.

2.2.3 Dynamic Traffic Assignment Simulation Models

The two dynamic traffic assignment models which are discussed in this section are simulation models designed for ITS related applications. These models evaluate the effectiveness of in-vehicle real time information about the performance of a congested traffic network.

The approach adopted in the first model integrates traffic flow, driver behavior, and network information into a signal simulation model (*26*). In this simulation model, the traffic flow is modeled using a fluid conservation equation. It is assumed that drivers are being provided with the basic information regarding the travel times on the best (i.e., shortest route) and alternative routes. This model is capable of simulating corridors as well as general network systems.

The second model, Vehicular Traffic Analysis Capability (VTAC), is capable of evaluating Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS) (*27*). This model is intended to simulate the performance of a network where drivers are informed of the shortest paths to their destinations.

Unlike most traffic simulation models, which are coded in procedural languages such as FORTRAN, the VTAC is developed using object oriented programming techniques. The decision-making feature in the VTAC model has been developed by object oriented programming with knowledge-structuring techniques (*28*).

2.2.4 Concluding Remarks

Dynamic route selection models and adaptive signal control systems are among those topics which still require more research to allow development of simulation models suitable for use by practicing traffic engineers (*29*). Dynamic route selection and driver information are partially addressed in the dynamic traffic assignment models such as VTAC model. It should be noted, however, that no simulation model for evaluating vehicle specific adaptive control systems have been developed.

In this research, a simulation model with an incorporated VSA control strategy is developed. This control strategy could be considered an adaptive control system, allowing cycle splits to be adjusted based upon performance characteristics for each vehicle.

2.3 Adaptive Control Strategy

The growth of car ownership and truck shipments has resulted in increased congestion on the nation's highways and city streets. New road building has not kept pace with this dynamic vehicle growth rate. For years, conventional traffic control systems (i.e., pretimed and actuated) have provided service to cities around the world. To appreciate their roles in our communities, it is noted that gridlock would be the norm at busy intersections without appropriate traffic control systems.

To partially alleviate traffic congestion, the concept of real time traffic control systems was conceived by Miller in 1963 (*30*). Real time traffic control is an element of ITS technology, commonly referred to as an adaptive control system. The objective of this control system is to minimize delay at intersections.

Thirty years after this conception, adaptive control systems are still in research and development stages. Most research in this area is conducted for isolated intersections. In the following section of this dissertation, some of these studies are reviewed.

2.3.1 Miller Algorithm

The pioneer study of an adaptive control strategy was conducted by Miller more than 30 years ago (1963) (*30*). In this traffic control algorithm, called a timing control system, a computer scans the provided information from the detectors for each intersection at regular intervals of two seconds (h) and decides whether to change the signals or not. These decisions are based upon estimates of delay which will be incurred if the signals are changed immediately or in h seconds. The signals are left unchanged unless it appears that less delay will result from an immediate change. For arterial control, this control logic considers the delay which vehicles are likely to experience at the next intersection.

Based on Miller's algorithm, Traffic Optimization Logic (TOL) was developed (*4*). In this method the time interval for checking the detectors information, is changed to one second. The decision to extend green time is based on the costs and benefits imposed upon motorists due to green extension. For a simple two phase signal intersection, one control function calculates the benefit for the traffic that can pass during the green extension of the running phase. Simultaneously, another control function calculates the cost to vehicles that will be further delayed on the other phase because of the green extension of the running phase. If the benefits exceed costs, the controller extends the green time by one second, otherwise it terminates the green as scheduled.

The criteria for determining costs and benefits in the TOL method are based on two parameters in the control function. The first is the cost of delay per second. The second is the cost to bring a vehicle to a complete stop. The same cost of delay is considered for each vehicle type. It is noted that in reality, the cost of delay for a commercial truck is greater than that for a passenger vehicle. This methodology could be improved if the information provided by the detectors also indicated vehicle classifications, as in AVSI where vehicle types are known.

Another model which used the Miller algorithm for its optimization process, is the Modernized Optimized Vehicle Actuation (MOVA) (*31*). This strategy relies less on forecasted data and more on actual information. Two detectors for each approach lane are used, one at 40 meters and one at 100 meters upstream of a stop line. Initially, each lane receives sufficient green time to discharge vehicles waiting within the 40 meter zone. The time intervals between the 40 meter and 100 meter detectors are examined for any changes in saturation flow rate. If the flow rate is increased, suggesting that more vehicles are approaching the intersection, the current green will be extended. In situations where one or more approaches fail to completely clear the queued vehicles at the end of their green time, MOVA switches from a delay minimizing process to capacity maximizing routine to clear those congested approaches.

2.3.2 Demand-Responsive Traffic Control Strategy

The next model reviewed in this section is Optimization Policies for Adaptive Control (OPAC) (*5*). The development of this model began with global optimization of a multistage adaptive control system, using a dynamic programming technique. This approach required advance traffic data for an entire period. The approach was simplified to require less advance data, but it still needs future traffic arrival information for the entire stage (i.e. 50-100 seconds--cycle length of a pretimed signal).

In the final approach, the rolling horizon concept which could function well with available traffic flow data were introduced. The rolling horizon concept is used by operation research analysts in production-inventory control. In this approach the stage length was divided into k intervals. For a period of r intervals (the head period) actual arrival data would be available. For the next k-r intervals (the tail period) flow data are assumed or forecasted. The entire stage is optimized but the result applies only to the head period. The entire stage is then shifted (rolled) r units forward, new traffic data are obtained for the head and tail, and the process is continuously repeated.

Other studies found that the 50-100 second stage length recommended in OPAC is excessive, since only the first 10-15 seconds of the stage (i.e., the head) can be expected to have reliable vehicle arrival information (*3,32*). These studies recommended 25 seconds for signal timing optimization. Even for a 25 second stage, however, the data for the tail of the stage should be forecasted. With the shorter tail period, it is noted that more accurate optimization would result. It was also found that stage subintervals (i.e., k intervals) of 3-5 seconds are adequate for an optimum solution (*33*).

The OPAC system received considerable attention during the 1980s. In one study, the required interfaces among the OPAC strategy, the signal control equipment, and vehicle detectors for a field test were examined (*34*). In another study a modified version of OPAC was evaluated using only predicted data (*35*). Three different predictors were used to forecast required data. Although none of the errors for the predictors was sufficiently small, the predicted data produced by an exponential smoothing technique were selected for the control policy evaluation for simplicity.

The evaluation of the modified OPAC points out two issues of importance. First, traffic flow is difficult to predict. As the prediction period becomes larger, the data obtained become less reliable. Second, adaptive control systems perform better using real data rather than forecasted traffic information, as adaptive control systems need accurate and advance traffic data in real time. To provide such data, detectors can be moved further upstream. Nonetheless, as more advance traffic information is supplied, the accuracy of the data decreases because of the fluctuation of traffic flows. This problem can be solved by using AVSI. The variability of the information provided by the AVSI is minimal, because it could continuously update the controller with accurate and advance data.

2.3.3 Stepwise Signal Control System

The decision to extend or terminate green in the Stepwise Adjustment of Signal Timing (SAST) model is divided into four levels (*3*). The decisions at each level are based on specific conditions. At level one, the current green is extended if no vehicles are expected from the competing phases within the next few seconds (e.g., six seconds). At level two, the maximum expected queue length of the green phase is checked with a threshold value (e.g., four vehicles). If the expected value is larger than the threshold value, the current green is extended. At levels three and four, other criteria are examined for extending green time. If green time is extended in one level, the model does not continue to check subsequent steps. SAST does not impose any constraint on the placement of detectors.

Simulation analyses of the SAST model against a loop occupancy control strategy (i.e., uses long loop detectors to detect the presence of vehicles) have shown that SAST is capable of reducing an intersection delay by 8 to 15 percent.

2.3.4 Adaptive Control System for Networks

SCOOT (Splits, Cycle, Offset Optimization Technique) is an adaptive control strategy designed to coordinate networks of traffic signals. SCOOT, developed by the Transport and Road Research Laboratory (TRRL), was first tested in Glasgow, England in 1979 (*36*). Since then, 53 SCOOT systems have been installed in approximately 25 cities in the United Kingdom, as well as a number of other countries (*37*,*38*). The application of SCOOT to network systems in the United States will be addressed by a test implementation project of the system in Anaheim, California. This particular area generates large and unpredictable traffic flows. The purpose of this project is to assess the degree of expertise required to install a new system (*38*).

As an alternative to Urban Traffic Control (UTC) systems, which operate on fixed timing plans, SCOOT operates on a flexible time plan that can be expanded or contracted depending on the traffic information provided. This responsiveness is achieved by optimizing splits, offsets, and cycle length (*36*,*39*).

A few seconds before every phase change, the split optimizer calculates whether it is better to make the split change earlier or later than the scheduled time by up to four seconds or to leave it unchanged. The objective of split optimization is to minimize the maximum degree of saturation on approaches to a particular intersection. At the end of each cycle, the offset optimizer evaluates whether delay on streets around the intersection can be reduced by increasing or decreasing the offset by four seconds. The suggested split and offset changes are implemented immediately. Any alteration to the green duration for each split would be temporary, and for each adjustment, a permanent change would be made to the signal timing plan. Accordingly, the cycle time optimizer would change the cycle length of a group of intersections by a few seconds every few minutes (e.g., 2.5 minutes).

The SCOOT model has been relatively successful for optimizing traffic flow in uncongested situations (*40*). It has also been shown that SCOOT can improve traffic flow operations in networks as well as for isolated intersections (*41*), during afternoon peaks, under low flow conditions (*42*), and in cases of road closure because of an incident (*43*).

The key to the success of the SCOOT system is the inclusion of concepts (i.e., making many small signal timing adjustments) which most other adaptive control strategies do not include. These concepts were suggested by Robertson in 1972 (*36*). In summary, SCOOT makes many small, but frequent, changes to signal timings and offsets. Typically, in a network of 100 intersections over 10,000 small optimization decisions per hour will be made (*39*). In this way SCOOT controls the traffic on a signal timing plan that evolves through time.

2.3.5 Concluding Remarks

Adaptive control systems have the potential to improve traffic flow at intersections. Nonetheless, more work is required before adaptive control systems can effectively replace vehicle-actuated systems. There is one principal difference between vehicle-actuated systems and adaptive control systems. In adaptive systems, the objective is to minimize delay at intersection, while in actuated systems no optimization process is performed. With an actuated system, green is granted to the minor street regardless of the number of waiting vehicles detected. With an adaptive control system, however, since the objective is to minimize delays at intersections, green time granted to the minor street is based upon the number of vehicles waiting.

The adaptive control system most widely accepted by transportation agencies is the SCOOT model. The system works for two reasons. First, it does not depend upon predicted traffic data. Second, if split or offset timing needs to be adjusted, it occurs in small frequent changes.

Field test results indicate some adaptive traffic control models perform better when the approaching traffic flow is high. The Miller algorithm was more effective at traffic flows of 3,000 vph (*44*), whereas TOL strategy performed well at 4,000 vph (*4*).

Adaptive control systems are still in the research stage. Most of the adaptive control strategies are for isolated intersections and they remain as prototypes. The only practical adaptive control system is SCOOT, which was developed for network control.

2.4 Expert Systems and Traffic Signals

An expert system can be defined as: "a computer program that embodies the expertise of one or more experts in some domain and applies this knowledge to make useful inferences for the user of the system" (*45*). Several characteristics distinguish expert systems from conventional computer programming systems. One of the most important characteristics is the separation of knowledge and control algorithm in expert systems. Knowledge base and control algorithms are the two primary components of an expert system. In expert systems, the control algorithm is referred to as the *inference engine*. The inference engine interprets the knowledge and applies it to solve a particular problem.

Expert systems can be developed by using available computer packages, as well as with low-level expert system computer languages such as LISP, PROLOG, and object oriented programming techniques. Commercial expert system shells, however, are not as general and flexible as programming languages since they are specially designed for only certain groups of problems. Of the more than 30 expert system shells reviewed by Chang, et al. (*46*) for traffic signal design, only two packages were recommended for this purpose.

Expert systems hold significant potential for solving problems which do not have explicit numerical algorithms and that require human judgment and experience for the final answers. Transportation problems such as traffic control and pavement management can make good use of expert systems, since their final solution is based on judgment and human expertise. Most of the transportation expert system models are at the prototype stage and will require extensive testing before becoming commercially available (*47*).

There are few expert system models available for traffic control systems. SCII (*48*), SCII-2 (*6*), and INTEL (*49*) are three models which use expert systems for controlling signalized intersections. In this section expert system prototypes used for designing signal timing, as well as the SCII and SCII-2 models are reviewed.

2.4.1 Semi-Actuated Control Knowledge Base

A Knowledge base is a critical element of an expert system. A Knowledge base is constructed by using the knowledge and experience of human experts. An alternative method of building this critical element of an expert system is by using a simulation model.

Lin (*50*) used a simulation model to build a knowledge base for designing a semi-actuated control system at isolated intersections. Through this approach, the design parameters were tested for their best responses in each simulation run. In order to find the optimum green extension time, for example, the detectors' distances from the stop line were varied in different simulation runs. As a result, zero second green extension was recommended for detector distances of 65-80 ft.

The design parameters used in Lin's study were phasing plan, detector distance, and minimum and maximum green time. This study has concentrated only on developing a knowledge base. This knowledge base is an informational tool which can help traffic engineers to design semi-actuated traffic signals. Using this knowledge base, an expert system for designing semi-actuated signals could be developed.

2.4.2 Traffic Signal Timing Design Expert Systems

Traffic signal timing design is a process which lacks explicit numerical algorithms. Therefore, the knowledge and experience of traffic engineers are important factors in this process. A few expert system prototypes have been developed which use a computer routine to emulate the traffic signal timing design procedures. Three of these expert systems prototypes are reviewed in this subsection.

The first prototype, TRALI, is an interactive expert system which assists traffic engineers in the designing procedures to identify an optimum phasing plan and cycle length (*51*). After a user inputs traffic flows and intersection geometry, TRALI proposes a phase plan. The proposed phase plan is evaluated by calculating the delay using Webster's formula (*52*). By specifying a cycle time range, the user is able to ask the model to suggest an alternative phase plan.

Similar to the TRALI model, PHAST (*53*) uses heuristics to determine phases and green times for single intersections. PHAST, a knowledge based expert system (KBES) model, emulates the design procedures that an experienced traffic engineer will take into account while designing a traffic signal. It determines initial phase and timing plans, evaluates intersection performance, and modifies suggested plans if they are determined to be inadequate. The evaluation aspect of this model is done by the microcomputer version of Highway Capacity Manual (HCM) and the results are manually fed back into the model. TRALI, on the other hand, carries out the evaluation automatically in the model. The graphic input interface and output display in PHAST is a plus when compared with the limitations of TRALI.

The next expert system prototype proposed a framework for an intelligent traffic signal design (*54*). It was proposed that at the first level, an expert system determines a phase plan for the signalized intersection. A simulation program then evaluates the performance of the intersection for the proposed phase plan. At the final stage, through another expert system module, based on simulation outputs, the system is able to decide whether to accept the suggested phase plan. This prototype only discussed the first stage of the framework. For this stage an expert system called PHASES was developed to select the best phasing pattern.

The first two stages of this framework are similar to the structure that has been used in TRALI and PHAST. In TRALI, instead of using a simulation process, the corresponding delay and the optimum cycle length are calculated through Webster's formula. In PHAST, on the other hand, this stage is performed manually. This proposed framework is intended more for an inexperienced traffic engineer because at the end, the expert system decides whether or not to accept the design.

2.4.3 Adaptive Control Strategies Expert Systems

TRALI, PHAST, and PHASES could be considered to be off-line traffic signal models which interactively assist traffic engineers in signal timing decision-making. On the other hand, the next model, Signal Control at Isolated Intersection (SCII), could be regarded as an on-line expert system prototype for traffic signal control (*48*).

SCII has taken a different approach than the previously explained adaptive control logic to model adaptive control strategies. Models such as OPAC and TOL continuously monitor traffic information in interval of one or two seconds to decide to whether extend or terminate green time. In contrast, SCII evaluates the performance of the signalized intersection at the end of each cycle. It determines the levels of service for each approach and the intersection. Based on heuristics, the levels of service are converted to a value between 0-100. If this performance grade exceeds the user-set threshold value, it continues with the existing signal timing. Otherwise, based on the predicted traffic volume for the next cycle, it recalculates the signal timing and determines the appropriate cycle length. If the last traffic volume data reflects a new trend in the database, it updates the database by adding the new data and deleting the old. It then repeats the entire procedure at the end of each cycle.

Since SCII uses predicted data, it does not require advance traffic information. To predict the next cycle traffic volume it uses a forecasting model which is a function of current volume and the previous volume data stored over past cycles.

The enhanced version of SCII, named SCII-2, was developed to handle different types of intersection geometries and conventional traffic signal operations (*6*). In this model the performance grade for each cycle is determined based on the combination of delay and queue length. This prototype was evaluated by the TEXAS simulation model using 20-minute traffic volume data.

The SCII-2 also includes a logic for switching controller operation from actuated to pretimed during heavy traffic conditions and vice versa (*55*). Based on the traffic volume at the end of each cycle, it determines the transition point for switching the control strategy. The appropriate signal operation is determined by the methodology outlined in *NCHRP Report 233*. It is believed that this part of SCII-2 did not involve any expert systems methodology.

2.4.4 Concluding Remarks

There are a limited number of expert systems studies in the area of traffic signal design and control. This is particularly true for signal control systems where only a few models are available. Furthermore, not all of the discussed models apply expert systems in their approaches.

The review of the computer program code developed by Elahi (*56*) suggests that SCII and SCII-2 are not traditional expert system models with separate knowledge bases and inference engines. Radwan et al. (*57*) provided some of their experience during the development of SCII. Through one of their experiences they state: "...we are often asked whether the system is a KBS (knowledge base system)." They continue with: "since the definition for KBS to which many people subscribe involves the inclusion of an inference engine and a knowledge base, they expect descriptions in these terms." They also point out they did not anticipate the controversy that centers around whether or not SCII is a KBS.

Building a traditional expert systems model for signal timing in a traffic control system would be a difficult, perhaps impossible, task. This is because a true knowledge base requires explicit rules, which would be very difficult to establish in the case of signal timing.

2.5 Chapter Summary and Conclusion

The capabilities of existing vehicle detectors are limited to detecting the presence and passage of vehicles. They are incapable of providing advanced traffic data, and their accuracy declines over the time due to pavement deterioration and exposure to all extremes of weather (particularly moisture). As such, they require frequent maintenance. New vehicle detection systems, however, are designed in such a way as to avoid frequent maintenance. They can also provide more accurate and advanced traffic data.

The review of the simulation models indicates that they are not suitable for evaluating the effectiveness of AVSI in a traffic signal control system since their incorporated traffic signal control systems (i.e., pretimed and actuated) are incapable of using the additional vehicle specific traffic information that AVSI could provide. Therefore, in this research, a new microscopic simulation model with an incorporated VSA control logic is developed.

VSA control logic is an important element of the simulation model which could utilize AVSI traffic information at signalized intersections. The examination of adaptive control systems indicates that the consideration of performance of vehicles is not part of their decision-making processes. They only consider total traffic volume in adjusting signal timing at intersections.

One of the unique features of AVSI is its capability to provide the classification of vehicles within a traffic flow. This unique feature of AVSI provides an opportunity for a VSA traffic signal control strategy to adjust cycle splits according to the characteristics of each vehicle within a traffic stream.

This research is expected to fill some of the gaps in the existing literature by developing a simulation model and a VSA signal control system which facilitate the evaluation of AVSI performance as a part of a traffic signal control system.

CHAPTER - 3 PROBLEM STATEMENT

Existing vehicle detectors, whether magnetic, inductive, or microwave, are designed primarily to sense the presence and passage of vehicles. When coupled with a traffic signal controller, detectors are used to derive traffic volume, vehicle speed, lane occupancy, and queue length *(25,* p. *285).* Vehicle speed, for example, can be determined based on the spacing between detectors and the time which a vehicle takes to travel the distance between two detectors. They are, however, unable to determine the type and performance characteristics (e.g., size, weight) of the detected vehicles.

As an alternative, AVSI would be capable of detecting vehicles and providing vehicle specific information. In the application to traffic signal control systems, AVSI is assumed to be capable of providing vehicle specific traffic information to a local microprocessor-based traffic signal controller. Based on the AVSI traffic information, the controller then adjusts its signal timing to reduce intersection delay.

The purpose of the research is to explore the potential benefits of AVSI vehicle specific information in a traffic signal control system. The AVSI traffic information considered in this research includes vehicle types and arrival times. Also, to limit the scope of the research to a manageable size, the benefits of use of AVSI information are evaluated at an isolated intersection. Future research may investigate the benefits of AVSI to traffic signal control systems.

If delay reduction at an intersection is the measure of effectiveness of a traffic signal control system, a tool is needed to estimate benefit. Computer simulation is a well known and powerful technique for testing the impact of changes in variables or parameters for systems that cannot be solved analytically (*58*, p. 227). It is also an appropriate tool for traffic experiments which for one reason or another, cannot be applied in the field. The use of AVSI information in a traffic control system can be categorized as a system where it may be impossible to evaluate system performance analytically, and it would be extremely costly to examine its applicability through experiments at an actual intersection. Thus, in this research, a simulation model is used to evaluate the effectiveness of AVSI in a traffic signal control system.

3.1 Simulation Model

Existing traffic control simulation models, such as NETSIM and TEXAS, are not suitable for evaluation AVSI, since their incorporated signal control system strategies (i.e., pretimed and actuated systems) are not designed to utilize vehicle specific information. The strategies engaged in these signal control systems are based on the assumption that heavy vehicles make up a constant portion of the traffic stream. In an actuated control system, for example, the green phase may be extended, up to an assigned maximum green, for every actuation generated by a detector regardless of vehicle characteristics.

Customizing the existing simulation models to use AVSI information would be difficult, if not impossible. Therefore, a new simulation model is developed to evaluate the use of AVSI information for adjusting the timings of a traffic signal control system. This new simulation model is a microscopic, stochastic, single-server model.

3.2 VSA Traffic Signal Control strategy

An important part of the simulation model is the development of a VSA traffic signal control strategy. A VSA traffic control system adjusts the signal timing based on AVSI traffic information. Compared with an actuated control system which adjusts signal timing in response to fluctuating traffic volumes, the VSA traffic signal control strategy examines individual vehicle performance characteristics before extending a phase green time or implementing a new cycle split.

It is suggested that traffic throughput at an intersection would be improved if heavy trucks could clear an intersection without delay. The objective of the VSA control strategy used in this research is to facilitate truck traffic at intersections. Although in VSA signal control logic trucks are given green time extension priority, the control strategy is not a truck preemption control system (i.e., cycle splits in this control logic are not adjusted each time a truck is detected). The VSA control strategy examines the status of other phases, prior to granting a green time extension to an approaching truck. As an illustration, if adjusting cycle splits would cause increased intersection delay, cycle splits are not adjusted for the detected truck.

VSA traffic signal control strategy is a new signal control logic which has not previously been investigated. The methodology used in this research to model VSA traffic signal control logic consists of two parts. In the first part, Webster's equation is used to compute optimum cycle length (*52,* p. 13) and appropriate cycle splits (*59,* p. 9-67). In this case, the actual traffic flow and heavy vehicle percentages are considered changeable parameters in Webster's equation. Based on traffic volume and percent of heavy vehicles in a traffic stream, the VSA control strategy periodically (e.g., every 5 minutes) adjusts its cycle splits according to the present traffic demand.

The first part of the VSA control system lacks decision-making capabilities, therefore, excessive delay may often be incurred at an intersection. To address this problem, the second part of the VSA control logic readjusts cycle splits to account for the most recent traffic conditions. By readjusting the green times, a slow-moving truck may be prevented from stopping, thereby resulting in decreased intersection delay. The second part of the VSA control logic involves decision-making processes using an heuristic. The heuristic could be considered to be nontraditional expert systems technology, since it may still require input from an expert.

Using the simulation model developed for this research, the incorporated VSA control strategy is tested against a pretimed control system. The simulation results indicated that through the use of AVSI traffic information, the VSA control logic can improve intersection performance by reducing vehicles stopped delay at an intersection. Moreover, the validation of the simulation model has established a level confidence in the obtained simulation results.

CHAPTER 4 - METHODOLOGY

The purpose of the research is to explore the potential benefits of traffic information, provided by AVSI, in a traffic signal control system. AVSI is capable of detecting and providing vehicle specific traffic information. However, this research only examines the use of AVSI to determine vehicle types and arrival times at a traffic signal controlled intersection, as well as those vehicles approaching the intersection. It is assumed that AVSI provides arrival time and type of each approaching vehicle to a local traffic signal controller at an isolated intersection. Based on the AVSI information, the controller then adjusts its signal timing to reduce intersection delay.

In this research, a simulation model is developed to measure the effectiveness of AVSI traffic information in reducing delay at an intersection. The first four sections of this chapter describe the structure and the strategies used in the simulation model. In the fifth section the VSA signal control strategy, which is incorporated in the simulation model, is reviewed.

4.1 Simulation Model Structure

The steps involved in the formulation of the simulation model are adopted from the *Transportation Research Board special report 165* (*60,* p. 178). This report suggests a systematic approach for simulating traffic flows at an isolated intersection. This approach is applied to the case study described in this research.

4.1.1 Step 1 - Case Study

The formulation of a simulation model starts with the selection of the traffic situation. The selected case study in this research is a four-leg, four-phase isolated intersection, shown in Figure 4.1. The phase patterns and layout of the selected intersection resemble the intersection of Lincoln Way and Duff Avenue located on the east side of the City of Ames, Iowa.

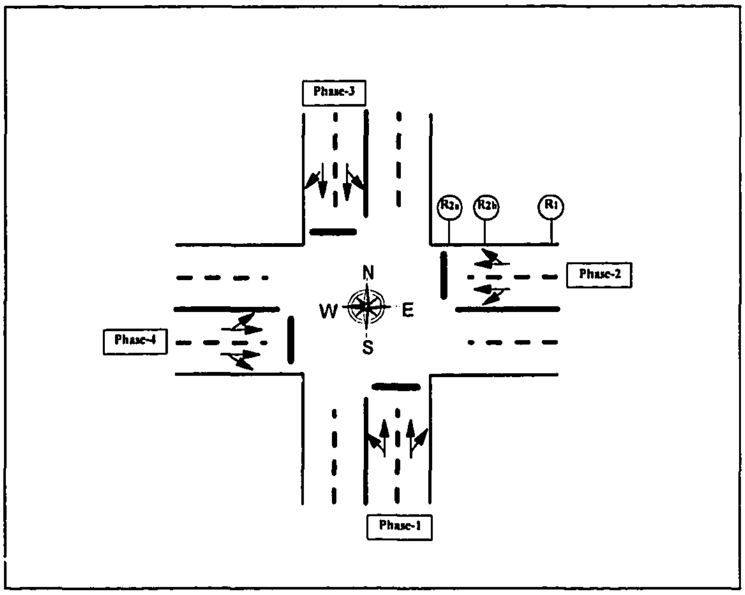

Figure 4.1 Four-Leg Four-Phase Isolated Intersection

4.1.2 Step 2 - Measure of Effectiveness

The second step in a simulation formulation is to define a figure of merit or measure of effectiveness (MOE). The MOE of the new simulation model is the average stopped delay of vehicles at an intersection. The average stopped delay of vehicles can further be classified into the delay of heavy vehicles or trucks and the delay of all other vehicles. The output of the simulation model, however, is not limited to only average stopped delay. It could also include maximum stopped delay and maximum and average queue length of each intersection approach.

4.1.3 Step 3 - Degree of Complexity

The third step of a simulation development requires defining the degree of the model's complexity. The simulation technique for modeling a single intersection can be classified into two categories (*58,* p. 235). Simulation models under the first category deal with the mutual interaction of vehicles on a roadway. One of the most useful techniques for simulating the vehicle interactions in a traffic flow is the use of car-following theory (*61*). CARSIM (*19*) and WEASIM (*20*) are two examples of car-following models which simulate stop-and-go conditions and weaving sections of freeways, respectively.

Under the second category, the distance component is neglected, and a model only simulates the sequence of vehicle arrivals and departures (*17,18*). The pattern of vehicle arrivals at an intersection is defined by the arrival time distribution of vehicles from connecting streets. Thus, each connecting street or link can be dealt with as a "black box" (*58,* p. 236). In this context, simulation models have no capability to capture vehicles' interaction. A vehicle arrives, waits in a queue, "seizes" an intersection, "holds" the intersection for a designated time, and leaves the intersection. There is no physical space assumed between vehicles; rather they are separated by a time component. Thus, vehicles in a queue can be visualized as a stack of cards where the bottom one would be the next to enter the intersection, once the intersection is available.

A car-following model is capable of simulating intersection traffic flow more accurately. However, with a lesser degree of complexity, the second simulation technique can also provide satisfactory results, when stopped delay at an isolated intersection is under investigation. Therefore, the second technique is appropriate for simulating traffic flow in this research. Accordingly, each connecting street of the intersection is treated as a "black box" in the new simulation model. This assumption is valid as traffic delay on connecting streets is usually caused by a traffic signal and not by mutual interactions of vehicles in the link (*58,* p. 236).

4.1.4 Step 4 - Vehicle Arrivals Pattern

In the fourth step of a simulation model formulation, the pattern of vehicle arrivals on each link needs to be determined. Arrival of vehicles at an intersection could be described as a random event. That is, the number of vehicles arriving in a given time interval is independent of the number of vehicles that arrived during any other time interval. Based on the assumption that arrival of vehicles is randomly distributed over time, the Poisson distribution can be used to predict vehicle arrivals and, in turn, the exponential distribution can be used to describe vehicle interarrival times (i.e., time between vehicle arrivals) or headways , (*62*, p. 354).

The Poisson distribution is related to the exponential distribution in that, if vehicle arrivals in a given period of time are Poisson-distributed with a mean of X, then the vehicle interarrival times are exponentially distributed with a mean of *1/X* (*63*, p. 47). Appendix I includes the mathematical derivation of the exponential distribution from the Poisson distribution.

The exponential distribution, often named negative exponential distribution, is only applicable when traffic flow is light (*60*, p. 23; *62*, p.352), that is, when there is no interaction between vehicles. The new simulation model also assumes no vehicle interactions, because each link is treated as a "black box." Moreover, since the objective of the simulation model is the computation of intersection delay, the negative exponential distribution is an appropriate choice for describing traffic flow in each link (*60*, p. 31).

One shortcoming of the negative exponential distribution is the generation of some short headways. Vehicles require front-to-front physical spacing equal to at least a car length (*62*, p. 356). One approach to correct this situation is to define a minimum allowable headway. This can be visualized by shifting the exponential curve to the right by an amount equal to the defined minimum allowable headway. This is called shifted exponential distribution.

In order to describe traffic flow with a shifted negative exponential distribution in the simulation model, vehicles are initially generated by a negative exponential distribution. If the generated headway is less than a required headway, the simulation model then determines the adequate headway, and uses the new value instead. The minimum allowable headway in the simulation model is a variable number. It depends on the types, lengths and speeds of the two vehicles that follow each other. For example, when a truck (i.e., assumed

60 ft) is followed by a passenger car (i.e., assumed 15 ft), the required headway should be at least 1.59 seconds. A 1.59 second headway is determined by assuming approaching speed of 30 mph (i.e., 44 ft/sec), and clearance distance of 10 feet between a truck and a car. That is:

$$Headway = \frac{10+60}{44} = 1.59 \text{ seconds.}$$

If the generated headway is less than 1.59 seconds, the simulation program discards the generated value and uses the calculated headway.

4.1.5 Step 5 - Vehicle Processing

The fifth step of the simulation formulation is to define a model for processing vehicles through the intersection. As early as 1947, Greenshields, et al. (*64*) studied the performance of drivers and vehicles at signalized intersections. They measured the time required for vehicles to enter an intersection after the traffic signal turned green. According to this study, the first car at the head of a queue enters the intersection 3.8 seconds after the signal turns green. The second car enters the intersection 3.1 seconds after the first car. After the sixth car, the headway remains as 2.1 seconds for the rest of the cars in queue. The study also indicated that because of lower acceleration ability, truck headways were about 1.5 times passenger car headways. The corresponding headway values reported by Greenshields, et al., are tabulated in Table 4.1. Figure 4.2 shows these headway values graphically. Each box shown in Figure 4.2 represents a vehicle.

The performance of vehicles has been changed dramatically since 1940s; however, some later studies confirmed the headway values estimated by Greenshields, et al.. In a 1961 study, the headways of the first few cars were reported to be less than these calculated values;

however, the minimum headway was still reported as about 2.1 seconds (*65*). A 1975 study, also suggested the same headway values calculated by Greenshields, et al. for simulating traffic flows at a signalized intersection (*60*, p. 179).

In order to determine whether the headway values estimated by Greenshields, et al. are still valid in 1994, a limited field study was conducted at several intersections in Ames,

Table 4.1 Greenshields Headway Values

Sequence of Cars in a queue	Seconds Before Each Car Enters the Intersection
First	3.8
Second	3.1
Third	2.7
Fourth	2.4
Fifth	2.2
Sixth	2.1

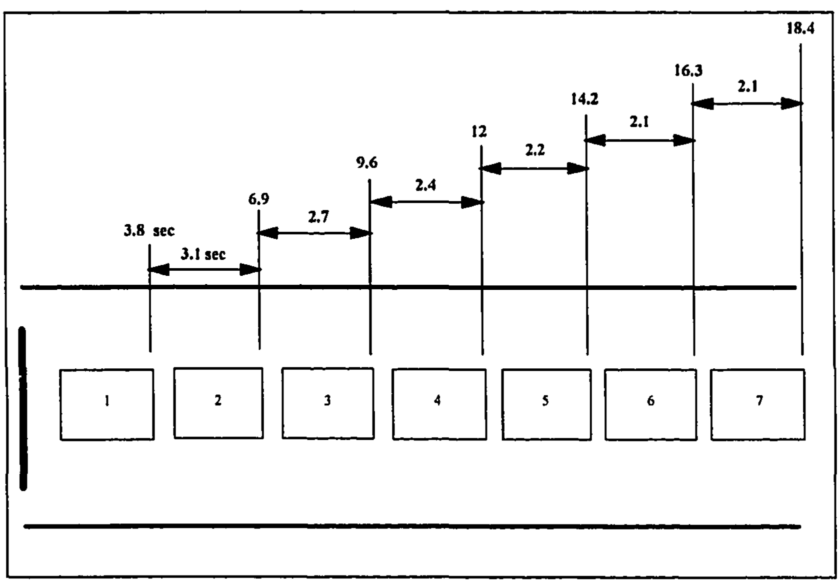

Figure 4.2 Greenshields Headway Values

Iowa. In this study, immediately after the traffic signal turned green, the time that each vehicle in the queue took to cross the stopline was recorded. Depending on the responsiveness of the drivers and the performance of the vehicles different headway values were recorded. However, on average, the headway for passenger cars was found to be about two seconds. Obviously, trucks need more time to clear an intersection. This extra time, of course, depends on the position of a truck in a queue. If a heavily loaded truck is the first in the line, it takes more time to leave an intersection than, if it is the seventh vehicle of a queue.

The simulation model employs the logic developed by Greenshields et. al. for discharging vehicles through the intersection, except here, it is assumed that each passenger car in a queue takes two seconds to cross a stopline. Since an average headway value is used for passenger cars, it seems appropriate to use an average headway for trucks. Therefore, a headway of three seconds, that is 1.5 times a passenger car headway, is used for trucks. The MOE of the simulation is defined as stopped delay, thus, a vehicle clears an intersection without delay only when a traffic signal is green and there is no queue. Otherwise (i.e., green signal with queue, or red signal), vehicles are processed through an intersection with the assigned headways.

4.1.6 Step 6 - Computer Language

The last step of the formulation of a simulation model is to select an appropriate computer language. In this research, the simulation is coded in SIMAN (SIMulation ANalysis) simulation language (*63*). SIMAN has been selected among other simulation languages both because of its advanced features and because of the author's prior knowledge of SIMAN.

4.1.7 Section Summary

In summary, a new simulation model is developed to determine average stopped delay at an isolated four-leg, four-phase intersection. The intersection is modeled as a single-server facility, that is, green time for a given signal phase is red time for other phases. In this regard, by serving the vehicles in one phase, more queue will be accumulated in other phases. Furthermore, the simulation model assumes no interaction between vehicles in the links. It only simulates the sequence of arrival and departure of vehicles at the intersection. Therefore, each link is considered as only a feeder. Vehicle arrivals in each link are described by the negative exponential distribution. Vehicles are processed through the intersection according to a modified version of logic developed by Greenshields, et al., using a constant average headways for passenger cars and trucks. The assumption of using constant average headways in a simulation model is found to be a good approximate solution to the problem of determining traffic delay at intersections (*66*, p. 52). Finally, this simulation model can be considered to be a stochastic model, since the interarrival time of vehicles is modeled as variables which have probability associated with each of the possible values (*67*, p. 271).

4.2 Simulation Modules

The simulation model consists of two modules: traffic flow and traffic signal control. The traffic flow module is responsible for generating vehicles on each link, registering arriving vehicles, moving vehicles through an intersection, and recording a vehicle's stopped delay. The traffic signal control module, on the other hand, emulates a local traffic signal controller at an intersection.

An important part of the signal control module is the inclusion of the VSA traffic signal control logic. The implemented VSA control strategy adjusts the signal timing based on AVSI traffic information which is complied in the traffic flow module. Figure 4.3 shows the flow of data in the two modules of the simulation model. The next two sections contain detailed description of these two modules.

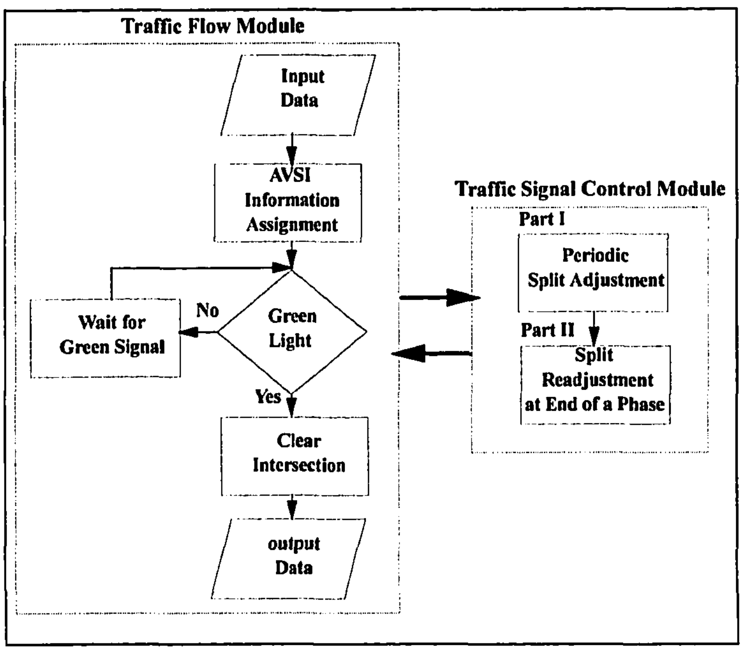

Figure 4.3 Structure of the Simulation Model

4.2.1 Traffic Flow Module

The traffic flow module consists of four independent submodules. Each submodule models a signal phase and its associated link at an intersection. Vehicles are generated in each submodule according to a negative exponential distribution. Once a vehicle is generated, based on an estimated discrete distribution (e.g., 30% trucks, 70% passenger cars), it is classified as a truck or a passenger car. Whether a vehicle is turning or moving through the intersection, is next attribute which is assigned according to a discrete distribution (i.e., 15% right turn, 30% left turn, and 55% through).

Next, based on the assigned direction, the model selects an appropriate lane for the vehicle to enter the intersection. As shown in Figure 4.1, each intersection approach has two through lanes. Unlike right or left turning vehicles, through traffic has the option of choosing either lane. In this multiple-queue system, it is assumed that through vehicles select the shortest queue lane to enter the intersection.

Once a vehicle is at the intersection, depending on traffic signal status, it either clears the intersection or comes to a complete stop before leaving the intersection. In the first case when a traffic light is green and no vehicles are waiting, the service time, or the time a vehicle occupies an intersection, is zero seconds. In this case, no delay is recorded for the vehicles. In the second case when a traffic light is red, or other vehicles are waiting to be discharged, a predetermined service time (i.e., passenger cars two second, trucks three seconds) is assigned to each vehicle. The vehicle delay, in this case, is the sum of its service time and the time that a vehicle is stopped before it is served.

The traffic module is responsible for clocking the time that each vehicle spends in the system. Thus, the average delay at any given time can be obtained directly from the simulation output. The average delay that vehicles experience at the end of each phase is an important parameter which is used in decision-making processes of the VSA traffic signal control logic.

4.2.2 Traffic Signal Control Module

The traffic signal control module represents a local traffic signal controller at an intersection. The sequence of this traffic signal is green, all-red, and red for each approach. The yellow light is considered as a green signal, since it is assumed that drivers treat the yellow light as the last portion of the green period. The all-red period compensates for the start-up and signal change interval lost time, which has not been included in the constant average service time of vehicles. Thus, the all-red period is assumed to represent the lost time of a cycle.

The initial cycle length and cycle splits are determined in the traffic signal control module. Based on the information sent from the traffic flow module, the incorporated VSA signal control logic periodically (e.g., five minutes) determines a new set of cycle splits according to current traffic volume and truck percentages. By engaging the second part of the VSA control logic, the signal control module further readjusts the cycle split at the end of each phase to meet the most recent traffic condition. The VSA signal control strategy is described entirely in section 4.2.

4.3 Simulation Input and Output Data

The vehicle interarrival times is the only data set which needs to be input into the simulation model through the traffic flow module. The interarrival times are defined by a negative exponential distribution. Other vehicle attributes, such as a vehicle type, direction, and service time are the predefined values which will be assigned to a vehicle after it enters the system. The output data including, the stopped delay of vehicles at each phase, queue length of each lane, and number of vehicles served are estimated by the simulation model.

The simulation model is run under two scenarios using peak traffic conditions. In the first scenario, the cycle length and splits remain unchanged during the whole simulation run. This case, where the VSA control strategy is disengaged, is essentially a pretimed signal control system. In the second case, the VSA signal control logic is engaged, thus the cycle splits are variable during a simulation run. The comparison of the average stopped delay resulting from the two cases determines if the VSA signal control strategy produces less stopped delay at an intersection.

In further analysis, some of the vehicle attributes are changed in different simulation runs to find the effects of that particular attribute on the average stopped delay. For example, the sensitivity of intersection average stopped delay to an increase in truck service time (e.g., changing from three seconds to four seconds) is determined by two simulation runs.

4.4 Validation of Simulation Results

Validation can be defined as the process of reaching an acceptable level of confidence that the inferences drawn from the model are correct and applicable to the real-world system being simulated (*63*, p. 133). A simulation model is a simplified version of an actual system. Thus, it is not expected that the model and the real system have identical behavior.

There are different types of systems that can be modeled by simulation. A system that exists and is accessible, a system that exists but is not available for direct experimentation, or a system which does not yet exist (*64*, p. 154). The simulation model in this research is of the third kind. That is, it simulates the performance of AVSI which has not yet been deployed in a traffic signal control system. Though one may argue whether it is even possible to validate such a system, the observed model behavior should give some indication of how AVSI would perform if implemented at an intersection.

Limited time and resources, preclude building a VSA traffic signal control facility to collect traffic data. Therefore, by validating the simulation model under a pretimed signal control system, a level of confidence can be established that indicates a similar level of confidence in the model outputs under the VSA signal control system. Confidence in the usefulness of the model can be established as new points of correspondence between the model and its actual state are found. For instance, it can be shown that cycle length and delay resulting from the simulation have the same relationship as that originally discovered by Webster (*52*, p. 13).

In order to validate the simulation model under a pretimed signal control system environment, a delay study at the intersection of Lincoln Way and Duff Avenue (Ames, Iowa) was performed. This study provided data on the level of delay for each approach. There are a variety of methods for performing field measurement of intersection stopped delay. The most common method of estimating intersection average stopped delay is based on direct observation of the number of stopped vehicles at frequent intervals (*25*, p. 71). The direct observation method of estimating average stopped delay at an intersection is presented in the 1985 *Highway Capacity Manual* (*59*, p. 9-71).

The method requires two observers for each approach. One observer with a stopwatch counts stopped vehicles, while the second person maintains a volume count by counting vehicles as they cross the stopline. At first, the furthest extent of a standing queue is identified. This point is used as the limit of stopped vehicle counts. At a regular interval time of between 10 and 20 seconds, the number of vehicles stopped at the intersection are counted and recorded on the field sheet in the appropriate time slot. A minimum sample of 60 measurements of stopped vehicles is recommended (*68*, p. 4.8). During the entire period of the study, a volume count is maintained by the second observer. At the end of the study period, the sum of the stopped vehicle counts and the total volume count of the approach are recorded in the appropriate boxes on the field form. The average stopped delay of each intersection approach can be calculated by using the following equation (*59*, p. 9-83):

$$D = \frac{\sum V_s \times I}{V} \tag{4.1}$$

where,

D = average stopped delay; sec/veh,
ΣV_s = sum of stopped vehicle counts,
I = interval between stopped vehicle counts; sec,
V = total volume observed during study period.

The Lincoln Way and Duff intersection is controlled by a fully-actuated signal control system. However, an actuated signal control system at a peak traffic condition can be treated as a pretimed system, since it is assumed that during a peak hour all the phases at an intersection reach maximum green. The delay study is conducted at the afternoon peak hour; thus, during the study the intersection is assumed to be functioning under a pretimed control system. Therefore, the stopped delay obtained through the field study is considered to be compatible with the simulation results obtained under the same traffic conditions. The actual intersection cycle length and cycle splits are used in obtaining the simulation stopped delay.

4.5 VSA Traffic Signal Control Strategy

VSA traffic signal control logic is a new strategy which has not yet been investigated. The VSA control strategy consists of two parts. In the first part of the strategy, based on advance traffic information, cycle split is updated periodically (i.e., every five minutes). The traffic information, assumed to be provided by AVSI, includes periodic traffic volume and percent of trucks. The cycle splits are determined by using an HCM equation (*59*, p. 9-67).

In the second part of the VSA control logic, cycle split could be readjusted to possibly prevent slow-moving heavy vehicles from stopping at a red signal. Based on the VSA control strategy, trucks are given the first priority to clear an intersection; however, splits are not adjusted every time a truck is detected. The VSA signal control strategy examines other phases of the intersection before granting a green extension to a phase. The objective of this control logic is to increase the efficiency of the entire intersection by reducing overall delay.

The two-part VSA signal control strategy is incorporated in the traffic signal control module of the simulation model. The next two subsections review this new traffic signal control system.

4.5.1 Part I - Periodic Cycle Split Adjustment

Based on fluctuation of the total number of vehicles and truck percentages in a traffic stream, the first part of the VSA signal control strategy adjusts the cycle splits periodically. The assumed time period should be a multiple of an intersection cycle time, since a new cycle split can not be implemented at the middle of a cycle. Periods of 75, 150, 225, or 300 seconds, for instance, are acceptable time periods for updating cycle splits of a 75 second cycle.

An AVSI roadside reader, R_1, is assumed to be located five minutes away from the stopline of each approach (see Figure 4.1). As a vehicle passes reader R_1 the reader records the type and arrival time of each vehicle. Table 4.2 presents a typical sample of AVSI traffic information which could be collected by a reader. The entries in this table are hypothetical, and are provided for better understanding of the data format.

The assumed time period and assigned vehicle speed determine the distance from the roadside reader R_1 to the stopline. It is assumed all vehicles approach the intersection with a

Table 4.2 Typical AVSI Traffic Information

Arrival Time	Type
7:20:35	Passenger Car
7:25:36	Passenger Car
7:26:40	Truck
7:29:42	Passenger Car
7:46:10	Truck

predefined, constant speed. An assumption of a five minute period and a vehicle speed of 30 miles per hour results in a distance of 2.5 miles from a reader to a stopline. Although positioning an AVSI reader 2.5 miles from an intersection seems to be a realistic distance in an ITS environment, a shorter distance might also be assumed.

By the end of each five minute period, the roadside readers inform a local controller of the expected total number of vehicles and trucks which will be arriving at an intersection in the next five minutes. Upon receiving the data, a microprocessor-based controller immediately determines a new set of cycle splits. This procedure is repeated every five minutes. The part-I portion of the traffic signal control module shown in Figure 4.3 is zoomed out in Figure 4.4 to show this procedure graphically.

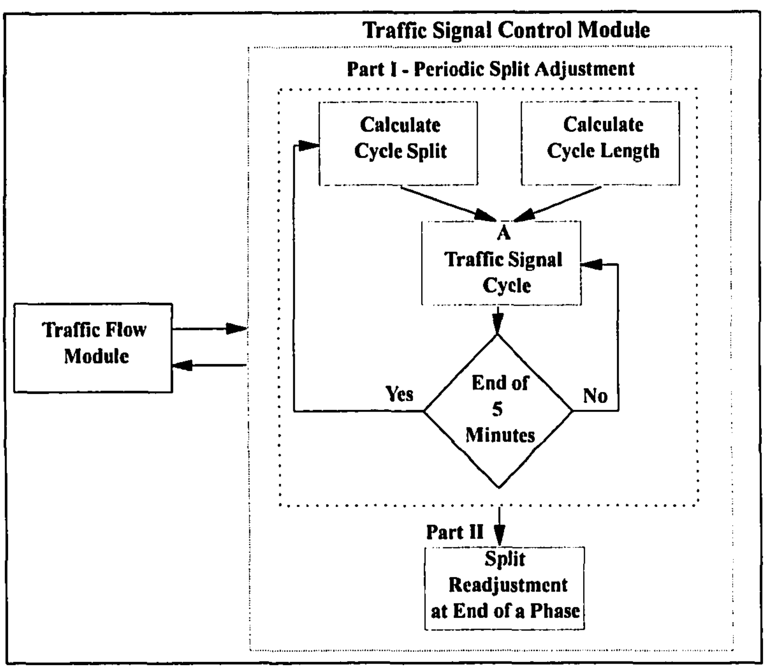

Figure 4.4 VSA Signal Control Logic - Part I Periodic Cycle Split Adjustment

As shown in Figure 4.4, initially, cycle length is determined by Webster's equation using estimated hourly average traffic volume and percent of trucks at an intersection. Once a new data set is provided, the cycle split is updated for the next five minutes. Cycle length remains unchanged all the time. The reason behind keeping cycle length constant will be explained later in this chapter. The following equations (*52*, p. 13; *59*, p. 9-11,9-67) are used to determine cycle length and cycle split.

$$C = \frac{1.5L+5}{\left[1-\sum_i (\frac{v}{s})_i\right]} \tag{4.2}$$

$$g_i = (\tfrac{v}{s})_i \left(\frac{C}{X_c}\right) \tag{4.3}$$

where,

$$s = s_0 N f_w f_{HV} f_g OAF \tag{4.4}$$

$$X_c = \frac{\sum_i (\frac{v}{s})_i C}{(C-L)} \tag{4.5}$$

C = cycle length,
L = lost time per cycle,
v = actual flow rate,
s = saturation flow rate,
$(\frac{v}{s})_i$ = flow ratio for lane group i,
g_i = effective green time for lane group i,
X_c = critical $\frac{v}{c}$ ratio for the intersection,
s_0 = ideal saturation flow rate per lane, usually 1800 pcphgpl,
N = number of lanes in the lane group,
f_w = adjustment factor for lane width,
f_{HV} = adjustment factor for heavy vehicles in the traffic stream,
OAF = other adjustment factors.

The flow ratio, $(v/s)_i$, is considered to be a variable parameter in the VSA traffic signal control strategy. The variability of flow ratio, and thus the cycle splits, is the only factor which differentiates the first part of the VSA control logic from a pretimed signal control system. Despite the similarity between the two signal control systems, the VSA control system provides an improvement in utilization of AVSI traffic information. The second part of the VSA control strategy could further enhance the signal control system through the readjustment of cycle splits during the five minutes time intervals.

4.5.2 Part II - End of a Phase Cycle Split Readjustment

The first part of the VSA signal control logic adjusts the cycle splits according to current traffic conditions. It is possible, however, that readjustment of the cycle splits at the end of a phase within the five minute intervals, may produce even less intersection delay. The objective of the second part of the VSA control strategy is to provide an optimum cycle split at an intersection, in order to minimize the stopping of slow-moving trucks at a red signal. It is believed that by facilitating the movement of heavy vehicles, the overall intersection delay is reduced.

The periodic adjustment of cycle splits in the first part of the traffic control system depends on AVSI traffic information which is collected by the distant roadside reader, R_1. The readjustment of cycle splits, however, is based on the data collected by two other AVSI readers located at each approach. As shown in Figure 4.1, reader R_{2a} is posted in the vicinity of a point where a queue of vehicles usually reaches during a red phase, the other one, R_{2b}, is located upstream at about 20 seconds from reader $R2_a$.

During the five minute periods for which cycle splits remain unchanged, AVSI traffic data collected by R_{2a} and R_{2b} readers are examined a few seconds (t_{chk}) before the end of each phase. The advance inspection of AVSI information provides ample time for a detected truck to stop safely if green time has to be terminated. The checking time (t_{chk}) is assumed to be

4.4 seconds which is determined based on vehicle's comfortable deceleration rate of 10 ft/sec^2 (*25*, p. 41) and approaching speed of 30 mph.

Upon examination of AVSI information, if a detected truck is not able to clear an intersection within the given green time, the second part of the VSA control strategy determines whether to readjust cycle split or to leave it as scheduled. Before the control logic reaches a decision, it goes through different levels of a decision-making process. The decision-making process involved in the second part of the control system is based on an heuristic approach, which is reviewed in the remainder of this section.

Before proceeding, however, a few terms need to be defined. A phase which is green is called the current phase. All remaining phases of an intersection are called competing phases. The AVSI traffic information collected by the readers of a current phase is referred to as the current database. Data collected in competing phases are referred to as competing databases.

At about 4.4 seconds (t_{chk}) before the green light switches from the current phase to the next competing phase, the current database of reader R_{2a} is examined. If no truck is detected by reader R_{2a}, the current database of reader R_{2b} is examined for possible detection of a truck which may arrive within the next 20 seconds. If again there is no truck detected by reader R_{2b}, the current phase green time is terminated as has been scheduled. However, if a truck is listed in either database, the VSA traffic control system starts its decision-making processes by checking the status of the competing databases. The flow chart presented in Figure 4.5 graphically shows the decision-making processes involved. This figure zooms in on the part-II portion of the VSA traffic control system. The following rules represent different levels of decision-making processes shown in Figure 4.5.

4.5.2.1 Rule 1 - Queue Length

- IF the queue length of one of the competing phases, q_{com}, is less than a threshold value, qv, THEN the next level will be executed. Otherwise, cycle split remains unchanged.

The purpose of the first rule is to make sure that there is no so-called "spillover condition" in any of the intersection approaches. The threshold value is determined based on the distance between two adjacent intersections. For example, a distance of 750 ft and average vehicle length of 25 feet, produces a threshold value of 30.

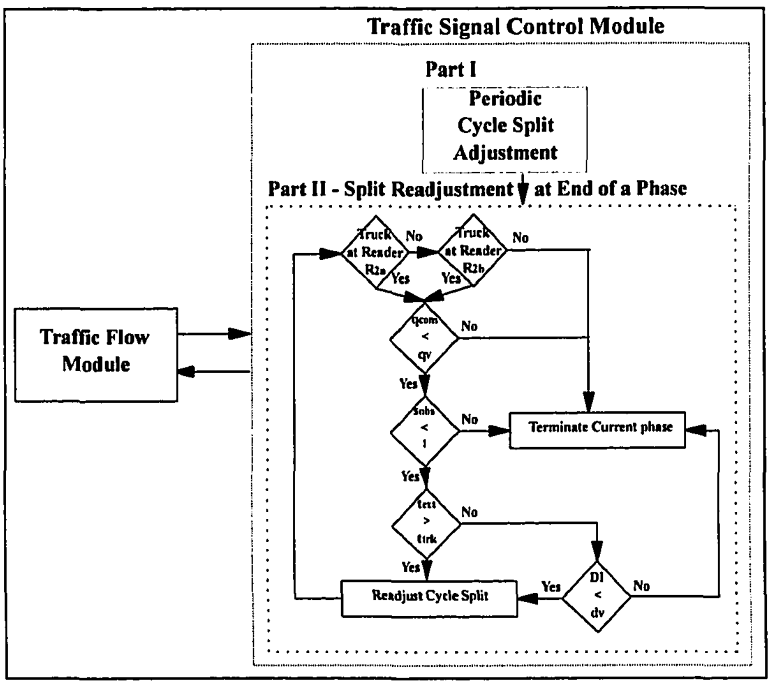

Figure 4.5 VSA Signal Control Logic - Part II End of a Phase Cycle Split Readjustment

4.5.2.2 Rule 2 - Degree of Saturation

- IF the recent degree of saturation of the next competing phase, s_{obs}, is less than one, THEN, the next level will be executed. Otherwise, cycle split remains unchanged.

The purpose of the second rule is to make sure that the possible green time extension would not cause a saturation condition in the next phase. The recent degree of saturation of the next competing phase can be defined as the degree of saturation created during the phase's last red time period. The degree of saturation is determined by the following equation (*58*, p. 9-4):

$$S_{obs} = \frac{v_i}{\left[s_i \times \left(\frac{g_i}{C}\right)\right]} \tag{4.6}$$

Where,

S_{obs} = degree of saturation,
v_i = actual flow rate for group lane or approach i; vph,
s_i = saturation flow rate for group lane or approach i; vphg,
g_i = green time for group lane or approach i; sec,
C = cycle length; sec.

The saturation flow rate (s_i) in Equation 4.6 is determined according to the "direct observation" method suggested in *1985 Highway Capacity Manual* (*59*, p. 9-73). The direct observation of saturation flow rate is designed to be conducted at an actual intersection; however, in this situation, the required data are directly obtained from the simulation model. According to this method, the time between the start of a phase green time and the time that the last vehicle in the queue is discharged, or the time that a phase ends, is measured. This time is then divided by the number of counted vehicles during that time period to obtain the saturation flow rate of the observed lane. The saturation flow rate is estimated by the following equation:

$$s_i = \frac{3600 \text{ (sec/hr)}}{\frac{\text{queue dischage time OR phase green time (sec)}}{\text{counted vehicles during the period(veh)}}} \quad (4.7)$$

Once the saturation flow rate is determined, given the actual flow rate, green time and cycle length, the recent degree of saturation of the next competing phase is obtained by calculating Equation 4.6.

4.5.2.3 Rule 3 - Extra Time

- IF the next competing phase has any extra green time, t_{ext}, to spare, AND this is greater than the time that a detected truck needs to be discharged, t_{trk}, THEN the current phase receives the requested green time extension. Otherwise, cycle split remains unchanged.

Depending on whether a green time extension request comes from a detected truck by R_{2a} or R_{2b} readers, different expressions are used to determine the requested green time extension, t_{trk}. If a green time extension is requested for a truck detected by the R_{2a} reader, the requested time is determined by the following equation:

$$t_{trk} = \{[(J-1) \times dt_{pc}] + [dt_{veh} - (TNOW - t_{in})] + dt_{trk}\} - t_{chk} \quad (4.8)$$

where,

t_{trk} = requested green time extension,
dt_{veh} = discharge time of a vehicle which is in service,
$TNOW$ = current simulation clock time,
t_{in} = the time that the last vehicle entered the intersection,
dt_{trk} = discharge time of a truck,
J = position of the first detected truck in the queue,
dt_{pc} = discharge time of a passenger car,
t_{chk} = 4.4 sec; checking time.

Equation 4.8 consists of four terms. The first term calculates the time a detected truck needs to reach the intersection. For calculating the value of this term, the location of the truck in the queue should first be determined. Once the position of the truck is determined by the value of J, the detected truck has to wait for J-1 number of passenger cars in front of it to be discharged before it enters the intersection.

The second term of Equation 4.8 determines the additional time that the detected truck, once it reaches the intersection, needs to stop for a vehicle in service to be discharged. The value of this term could be either the whole discharge time of a vehicle or a portion of its discharge time. In cases where a portion of a vehicle's discharge time is requested, a vehicle has already entered the intersection and is in the middle of its service. The discharge of the vehicle in service (dt_{veh}) is determined based on its type.

The third term of the equation simply is a truck's discharge time which the detected truck needs to completely clear the intersection, once it reaches the stopline. By deducting the last term from the first three terms the requested green time extension is determined. If the obtained value is negative, the detected truck then needs no green time extension and will clear the intersection within the remaining 4.4 seconds (t_{chk}) of the phase green time.

In order to clarify the terms used in Equation 4.8, consider the following example. Assume at the time that the status of the reader R_{2a} is examined (t_{chk}), a truck which is the third vehicle in the queue is detected. Therefore a value of 3 is recorded for J. Since the first two vehicles in the queue must have been passenger cars, the time that the detected truck needs to reach the intersection is the total discharge time of the two passenger cars. Given a passenger car's discharge time of 2 seconds (dt_{pc}) the detected truck therefore needs 4 seconds to reach the intersection (i.e., the equation's first term). Once the detected truck is at the stopline, it has to wait for the vehicle in service to clear or "release" the intersection. Assume the intersection is busy with a passenger car which has entered the intersection at 18 seconds (t_{in}). The simulation clock time (TNOW) indicates 18.5 seconds. Given the vehicle's discharge time (dt_{vch}) of 2 seconds, the detected truck needs to stop 1.5 additional seconds before it enters the intersection (i.e., the equation's second term). However, before the detected truck is being discharged, it needs 3 more seconds (dt_{trk}) to clear the intersection (i.e., the equation's third term). Deducting the 4.4 seconds (t_{chk}) from the numerical values calculated for each term, the detected truck needs 4.1 seconds (i.e., 4+1.5+3-4.4) of green time extension (t_{trk}) to clear the intersection.

For calculating a green time extension of a truck that is detected by the R_{2b} reader, which will arrive within the next 20 seconds, again the position of the truck on the road should first be determined. Depending on the status of the queue at the intersection at the time that the detected truck will arrive, the following equations are used to determine the requested green time extension:

$$t_{trk} = [dt_{tot} + ((J-1) \times dt_{pc})] + dt_{trk} - t_{chk} \tag{4.9}$$

$$t_{trk} = (av - TNOW) - t_{chk} \tag{4.10}$$

where,

t_{trk} = requested green time extension,
dt_{tot} = total discharge time of vehicles waiting in the next competing queue,
J = position of the first detected truck in the queue,
dt_{pc} = discharge time of a passenger car,
av = arrival time of the detected truck,
$TNOW$ = current simulation clock time,
dt_{trk} = discharge time of a truck,
t_{chk} = 4.4 seconds; checking time.

The requested green time extension, t_{trk}, is determined by Equation 4.9 if there are other vehicles waiting in the queue to be discharged when a detected truck will arrive at the intersection. However, in the cases where no vehicles are stopped at the intersection, the detected truck clears the intersection with no stop delay. Therefore in these cases, the requested green time extension is determined by Equation 4.10 which is simply the time before the detected truck arrives minus 4.4 seconds, the checking time.

In order to clarify the terms used in Equations 4.9 and 4.10, consider the following example. Assume the first truck detected by reader R_{2b} is the second vehicle (J=2) on the road, that is, one (J-1) passenger car is in front of it. If the stopped vehicles at the intersection need a total of 6 seconds (dt_{tot}) to clear the intersection, the value of the first term in Equation 4.9 is 8 seconds (i.e., 6+(2*1)). Given the truck's discharge time of 3 seconds (dt_{trk}) and checking time of 4.4 seconds (t_{chk}), the detected truck therefore needs 6.6 seconds (i.e., 8+3-4.4) to clear the intersection.

If, however, no vehicles are waiting at the intersection when the detected truck will arrive, Equation 4.10 is used to determine the requested green time extension. Assume the current simulation clock time (TNOW) indicates 45 seconds. A detected truck will arrive at 50 seconds. Therefore, the detected truck in this case only needs 0.6 seconds (50-45-4.4) green time extension to clear the intersection.

Once the requested green time is determined, it is compared to the extra green time of the next competing phase (t_{ext}). Extra green time is calculated by subtracting the total discharge time of vehicles waiting in the competing queue from the competing phase's assigned green time. This can be shown in the following equation:

$$t_{ext} = g_{com} - dt_{tot} \qquad (4.11)$$

where,

t_{ext} = extra green time of the next competing phase,
g_{com} = assigned green time of next competing phase,
dt_{tot} = total discharge time of vehicles waiting in the next competing queue.

As an illustration, assume dt_{tot} and g_{com} are 13 and 20 seconds, respectively. By inserting these values into Equation 4.11, the next competing phase is determined to have seven seconds of surplus green time (t_{ext}). If the condition defined in the third rule is satisfied, that is, t_{ext} is greater than t_{trk}, the requested green time extension can be awarded unconditionally to the current phase. However, if t_{ext} is less than the requested green time extension, the next rule is examined. This rule ensures that the vehicles waiting in the next competing phase do not experience excessive delay.

4.5.2.4 Rule 4 - Delay Index

- IF delay index, DI, is less than a predefined threshold value, dv, THEN the current phase receives the requested extension time. Otherwise, the cycle split remains unchanged.

The delay index is a weighted combination of passenger cars' and trucks' average stopped delay of the next competing phase, which has been built up during the phase's last red time. The delay index is defined by the following expression:

$$DI = (w_{pc} \times d_{pc}) + (w_{trk} \times d_{trk}) \tag{4.12}$$

where,

$$d_{pc} = \frac{(V_{pc} \times TNOW) - at_{pc}}{V_{pc}} \tag{4.13}$$

$$d_{trk} = \frac{(V_{trk} \times TNOW) - at_{trk}}{V_{trk}} \tag{4.14}$$

DI = delay index; sec/veh,
w_{pc} = assigned weight factor of passenger cars,
w_{trk} = assigned weight factor of trucks,
d_{pc} = p. cars' delay of the next competing phase during the phase's last red time; sec/veh,
d_{trk} = trucks' delay of the next competing phase during the phase's last red time; sec/veh,
$TNOW$ = current simulation clock time,
V_{pc} = observed volume of passenger cars during the next competing phase's red time; veh,
V_{trk} = observed volume of trucks during the next competing phase's red time; veh,
at_{pc} = total passenger cars' arrival time; sec,
at_{trk} = total trucks' arrival time; sec.

The weight factors used in Equation 4.12 are determined based on vehicles' travel time values. Trucks' travel time values are reported as about 2.7 times of those for passenger cars (*69*, p. 74; *70*, p. 3-19,3-20). Running the simulation model with different combinations of weight factors also indicates that the weight factors of 0.25 and 0.75 are the appropriate factors for passenger cars and trucks, respectively.

The other two parameters involved in delay index calculation are d_{pc} and d_{trk}, which are determined by using Equations 4.12 and 4.13. The numerators used in Equations 4.11 and 4.12 are the total stopped delay that passenger cars and trucks are experiencing in the next competing phase, since the phase's last green time has ended. Dividing the total stopped delay by its associated number of observed vehicles determines the average stopped delay of passenger cars and trucks in the simulation model environment.

In order to determine the delay index, DI, in Equation 4.10, Equations 4.11 and 4.12 need to be solved first. Consider the following example:

Assume,

V_{pc} = 10 passenger cars,
V_{trk} = 3 trucks,
at_{pc} = 43 seconds,
at_{trk} = 27 seconds, and
TNOW = 44 seconds.

By applying these values in Equations 4.11 and 4.12, the numerators of these equations (i.e., total stopped delay values) compute to 397 and 105 seconds; respectively. Dividing these total stopped delay values by the associated number of observed vehicles, the average stopped delay of passenger cars (d_{pc}) and trucks (d_{trk}) since the last red signal of the next competing phase are calculated to be 39.7 sec/veh and 35 sec/veh; respectively. Finally, by inserting the values of d_{pc} and d_{trk} and their assigned weight factors into Equation 4.10, the delay index is determined as 36.2 sec/veh.

Once the delay index is calculated, it is compared to a predefined threshold value, dv. If delay index is less than dv, the current phase receives the requested green time extension. Therefore, the higher the value of dv is, the more chances the current phase has to receive green time extension. For example, a value of zero for dv, provides no chance for the current phase to receive a green time extension at this stage of the decision-making process. An appropriate value of dv is selected by running the simulation model under different dv values. Based on the obtained simulation results, a value of 15 is assigned to dv.

If the condition stated in the fourth rule is satisfied, the current green time is extended for the amount of time which is requested. In the meantime, an equal amount of green time is subtracted from the next competing phase. Once the extension time is ended (i.e., 4.4 seconds before the end of the phase), the procedure continues, again checking the current database of readers R_{2a} and R_{2b} for another truck. If there is a truck and the stated conditions are satisfied, the current green time is extended for the second time. The procedure is repeated until it violates any one of the stated rules.

After the green light moves to the next competing phase from the current phase, the previous competing phase becomes the new current phase, and the old current phase now becomes one of the competing phases. If a truck is detected at the end of this new current phase, the green time extension now is requested from the next competing phase. In a four-phase signal control system, for example, phase-1 requests green time extension from phase-2, phase-2 requests from phase-3, phase-3 requests from phase-4, and phase-4 could request green time extension from phase-1. The only restriction to this procedure is that at the last cycle of the defined period in the first part (i.e., 5 minutes), phase-4 is not permitted to request green time extension from phase-1, since a green time extension at this time would change the defined intersection cycle length, which, as stated before, needs to remain constant throughout the simulation run.

4.5.3 Variable Splits versus Variable Cycle

Based on the VSA traffic signal control logic, cycle splits are adjusted based on the fluctuation of traffic flows, while cycle length remains unchanged. The variable splits algorithm is chosen over the variable cycle algorithm (i.e., varying both cycle length and splits at the same time), because the defined five minute period needs to be a multiple of the cycle length. The simulation results, however, indicate that the variable splits algorithm produces less intersection delay. Although it may seem intuitive that a variable cycle option would reduce intersection delay, a thorough investigation has shown that, for the selected five minute period, the variable splits approach is better for a VSA traffic control system. The rationale behind this conclusion is discussed extensively in the next chapter.

CHAPTER 5 - APPLICATION

A simulation model is developed to evaluate the applicability of AVSI traffic information at an isolated intersection. The computer listings of the simulation model are included in Appendix II. The AVSI traffic information utilized in this evaluation consists of vehicle types and arrival times. An important component of the simulation model is the VSA traffic signal control strategy, which is implemented in the traffic signal control module of the model. By utilizing AVSI traffic information, the VSA signal control system has been shown to be capable of improving signal timing at intersections by reducing stopped delay.

This chapter includes six sections. The first section defines the case study employed to obtain the simulation results. The second section presents the simulation results indicating the superiority of the VSA traffic signal control logic over the pretimed signal control system. The third section investigates the reasons behind employing a variable splits algorithm rather than a variable cycle algorithm in the simulation model. The fourth section describes the procedures involved in validating the simulation model. The fifth section examines sensitivity of the intersection average stopped delay to some of the attributes and variables used in the simulation model. For example, the effect of truck percentages changes on the average stopped delay at an intersection are studied in this section. The last section includes concluding remarks.

5.1 Case Study

The simulation model is developed based on an isolated intersection as shown in Figure 5.1. This is a four-leg, four-phase intersection with phase patterns resembling the intersection of Lincoln Way and Duff Avenue in Ames, Iowa. The north bound is assumed to be phase-1. Turning counterclockwise, west bound is phase-2, south bound is phase-3, and the east bound is phase-4. Each intersection approach has two lanes. The outer lane is used for left turning and through traffic. The inner lane is used for right turning and through traffic.

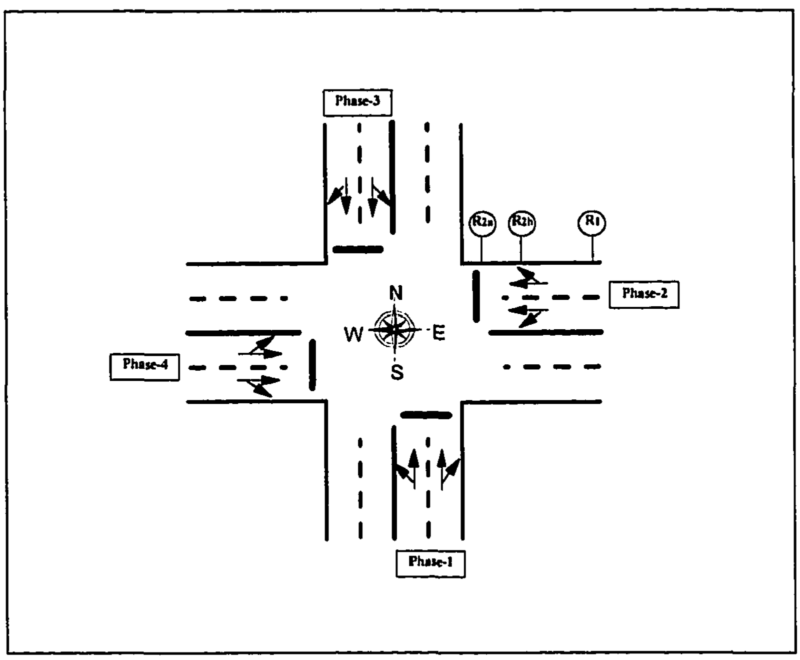

Figure 5.1 Four-Leg, Four-Phase Isolated Intersection

The average hourly volume of traffic at the afternoon peak hour is assumed to be 600 vehicles per hour (vph) on each approach. A rather high truck percentage of 30 percent is assumed in this case study, since the performance of a VSA control strategy can be better evaluated when a high number of trucks is present in the traffic flow. The assumed traffic flow rate and truck percentage are used to calculate the intersection cycle length and its initial cycle split in the simulation model.

It is assumed that AVSI roadside readers R_1, R_{2b}, and R_{2a} are posted upstream at about 2.5 miles, 1200 feet, and 300 feet, respectively, from the stopline of each intersection approach. Initially, it is assumed that there are no vehicles present at the intersection or on its approaches. At t_0 , the R_1 readers start detecting the approaching vehicles. At t_5 (i.e., five minutes after t_0), the first detected vehicle arrives at the intersection. At this time the R_1 readers transmit the current values of traffic flow rates and truck percentages which were detected during the past five minutes in each approach to the local traffic signal controller. Based on the received traffic information, the local controller determines a new optimum set of cycle splits, which will satisfy the upcoming traffic demand for the next five minutes. This procedure, which is defined in the first part of the VSA traffic control system, is repeated every five minutes. Thus, the local traffic signal controller is provided a chance to adapt its cycle split for the vehicles arriving in the next five minutes.

Through the information received from R_{2a} and R_{2b} readers, the traffic signal controller is provided other opportunities during the five minute period to readjust the cycle split at the end of each cycle phase. For example, if a truck is detected by any one of these readers, the local controller decides whether to readjust the cycle split to provide additional green time to the current phase, or leave it as it has been scheduled. This decision is based on the decision-making processes defined in the second part of the VSA traffic signal control logic.

5.2 AVSI Evaluation

The AVSI performance at an intersection is evaluated by conducting a "before and after" study. In the absence of AVSI, the intersection is assumed to be operating under a pretimed control system. With the AVSI traffic information, however, the intersection is assumed to be operating under this new VSA signal control system. By comparing the results obtained from the simulation model run under the two described traffic signal control systems, the AVSI performance at an intersection is evaluated.

The first part of the VSA traffic control strategy involves a mechanism that periodically informs a local controller regarding the upcoming traffic demand during the next five minutes. In order to capture this strategy in the simulation model, a pre-generator model (i.e., a limited version of the developed simulation model) is used to generate vehicles through each intersection approach for the entire simulation running period. The computer listings of the pre-generator model are included in Appendix III.

Vehicle interarrival times are generated according to a negative exponential distribution with a mean of six seconds per vehicle, resulting in a flow rate of 600 vph. Once a vehicle is created it is classified as a passenger car or a truck. The next attribute identifies the vehicle direction, which determines if a vehicle will turn right or left or go through the intersection. This data is recorded on the separate files for each approach (i.e., files t11.txt, t12.txt, t13.txt, and t14.txt). As an illustration, Table 5.1 includes the information assigned to the first five vehicles generated for the north bound approach in the "t11.txt" file.

At the end of each five minute period, the number of observed vehicles and the percent of the detected trucks are calculated. For the purpose of calculating cycle splits, these values are converted to hourly volumes and truck percentages, which are recorded in file "t.txt". For example, Table 5.2 presents the data recorded for the first few periods in file "t.txt".

Table 5.1 Data Assigned to Vehicles on North Bound Approach

Arrival Time (sec)	Type[1]	Direction[2]
0.01	1	3
13.03	2	1
29.59	2	3
30.16	2	1
32.81	1	1

[1] 1 = Truck, 2 = Passenger Car

[2] 1 = Straight, 2 = Right, 3 = Left

Table 5.2 Converted Traffic Flow Rates and Truck Percentages for Each Approach

Time (min)	North bound	West bound	South bound	East bound
5	564[1] (36%)[2]	576 (38%)	516 (26%)	588 (29%)
10	708 (29%)	636 (21%)	708 (32%)	888 (30%)
15	396 (21%)	516 (19%)	600 (32%)	588 (37%)
20	672 (36%)	588 (37%)	684 (32%)	720 (35%)
25	624 (38%)	552 (24%)	540 (33%)	672 (29%)

[1] Traffic flow rates

[2] Truck percentages

Once the pre-generator model generates the data files, they are read by the simulation model. The simulation model is run under the two traffic signal control systems for 2 hours. In the first case, the simulation model is run under the pretimed traffic signal control system. In this case, the cycle length and cycle split are determined based on the traffic information provided in the case study (i.e., traffic flow rate 600 vph, truck percentage 30%). The cycle length and cycle split remain unchanged during the simulation run. The average stopped delay is calculated at the end of each five minutes period, as well as at the end of the simulation run period (i.e., two hours). The recorded average stopped delay is further broken down to passenger cars, trucks, and intersection delay.

In the second case, the simulation is run using the VSA traffic signal control logic. Initially, the intersection cycle length is calculated using the same traffic information used in the first case (i.e., traffic flow rate 600 vph, truck percentages 30%). Thus, the cycle lengths for the both cases are the same, and they are assumed to remain unchanged throughout the signal control operation. However, unlike the pretimed case, the VSA control system adjusts cycle splits according to the fluctuation of traffic volumes.

Once the cycle length is determined, the cycle split is calculated using the traffic flow rates and truck percentages reported for the first five minute period. At the end of the period, a new cycle split is determined using the upcoming flow rate and truck percentage information. This procedure continues for every five minutes. During each five minute period, by engaging the second part of the VSA control strategy, the cycle split is readjusted if necessary to provide extra green time for discharging the trucks detected at the R_{2a} or R_{2b} readers.

Table 5.3 presents the average stopped delay resulting from the two traffic signal control systems at the end of the simulation run period. As these results indicate, the utilization of the AVSI traffic information in a traffic signal control system could provide reduced delay at intersections. Table 5.3 also includes the time that each vehicle could save in a VSA traffic control system during its travel through the intersection. Figure 5.2 shows that even during each five minute period the VSA control logic produces less intersection delay than the pretimed system.

Table 5.3 Average Stopped Delay

	Average Stopped Delay (sec/veh)		
	Truck	Passenger Car	Intersection
Pretimed Control System	47.63	45.45	46.11
VSA Control Strategy	33.07	36.47	35.45
Time Saved by AVSI	14.56	8.98	10.66

During the two hours run period, the simulation output shows that about 4,800 vehicles are discharged from the intersection. Thus, under the present case study, the VSA control logic could reduce approximately 5.8 hours of total stopped delay during the two-hour period at the intersection. The complete list of total time that passenger cars and trucks could gain in a VSA control system is included in Table 5.4.

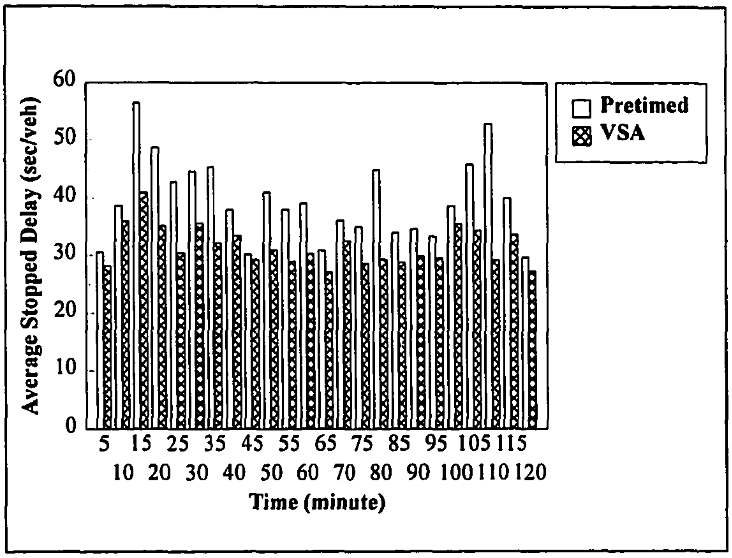

Figure 5.2 Average Stopped Delay at End of Each Five minutes Period

Table 5.4 Total Time Saved by VSA Traffic Control Strategy During a Two-hour Period

	Truck	Passenger Car	Intersection
Time Saved by AVSI (sec/veh)	14.56	8.98	10.66
Observed Vehicles (veh)	1,440.00	3,360.00	4,800.00
Total Time Saved by AVSI (hour)	5.82	8.38	14.21

A VSA traffic signal control strategy is also capable of reducing maximum stopped delay and queue length at an intersection. Table 5.5 includes a sample of these maximum values. As indicated in the table, the maximum stopped delay of trucks on the east bound approach, for instance, could be reduced by more than 93 seconds with a VSA traffic control system. Maximum queue length in the left lane of the same approach could also be shortened by eight vehicles.

Table 5.5 Maximum Stopped Delay and Queue Length

	Maximum Stopped Delay (sec)	
	Pretimed Control System	VSA Control Strategy
Trucks in North bound	125.14	103.51
Passenger Cars in West bound	143.41	93.44
Trucks in East bound	175.87	81.92
	Maximum Queue Length (veh)	
	Pretimed Control System	VSA Control Strategy
Right lane of West bound	14	8
Left lane of South bound	15	10
Left lane of East bound	18	10

The reason why such positive outcomes are obtained is due to the adaptiveness of the VSA traffic control strategy. The VSA traffic control system not only adapts the cycle splits according to the fluctuation of traffic volume, it also adjusts cycle splits based on the performance characteristics of vehicles in the traffic flow. Although, trucks are given the priority to clear an intersection in order to experience less delay, passenger cars are the main beneficiary in a VSA traffic control system. This can be noted in Table 5.4, as the total time that passenger cars could save during a two-hour period in a VSA control system is greater than that for trucks. Based on these results, it is fair to conclude that facilitating of the movement of heavy vehicles in a traffic signal control system causes a reduction in overall intersection delay.

5.3 Variable Splits versus Variable Cycle

There are different algorithms for signal timing adjustment which could be used in a traffic signal control system. According to a variable splits algorithm, only cycle splits are adjusted to accommodate the incoming traffic demand. However, when a traffic signal control system employs a variable cycle algorithm for adjusting its signal timing, both cycle length and cycle splits are adjusted.

The VSA traffic control strategy employed in the current case study adopts a variable splits algorithm for its signal timing adjustment. The reason for choosing a variable splits algorithm in this study is because the time interval selected for the cycle splits adjustment (i.e., five minutes) should be an exact multiple of the cycle length since a new cycle split can not be implemented at the middle of a cycle. Therefore, a periodic change of cycle length may violate this requirement. For example, a time interval of five minutes is an adequate period for a cycle length of 75 seconds. However, if the cycle length is changed to 85 seconds, the five minute period is no longer an exact multiple of the new cycle length. Moreover, the simulation results indicate that VSA control logic under a variable splits algorithm produces less intersection delay than under a variable cycle algorithm.

Although the VSA control strategy is designed to employ a variable splits algorithm for its signal timing adjustment, the question of why a variable splits algorithm produces less intersection delay still remains. This question is answered by conducting a thorough investigation. It is found that a five-minute interval is a short time for using a variable cycle algorithm, since the calculated cycle length should either be significantly increased or decreased to satisfy the previously defined requirement. Therefore, the new cycle length no longer holds its original optimum value to provide a best signal timing at an intersection.

To understand how this conclusion (i.e., a five-minute interval is a short time for using a variable cycle algorithm) is reached, the simulation model is run assuming a zero truck percentage and under the two signal timing adjustment algorithms for one hour. The first run used a variable splits algorithm where cycle splits are adjusted every five minutes, while the cycle length remained unchanged. In the second simulation run under the variable cycle algorithm, both cycle length and cycle splits are adjusted at the end of each five minutes. Table 5.6 presents the average stopped delay obtained from these two simulation runs. As indicated in this table, when the time interval for signal timing adjustment is five minutes, the variable splits algorithm provides less intersection delay than the variable cycle algorithm.

Table 5.6 Average Stopped Delay - The Five-minute Time Intervals Case

Signal Timing Adjustment Algorithm	Intersection Delay (sec/veh)
Variable Splits	25.95
Variable Cycle	28.44

In next round of simulation runs, the five minute period is changed to a one hour period, and the simulation model is run for 12 periods or 12 hours. For consistency, the same total traffic volumes generated for each five minute period are used to generate vehicle interarrival times for each one hour period of the second round of simulation runs. As in the first round, the simulation model is run using the two signal timing adjustment algorithms. In the first run, cycle splits are adjusted at the end of each one hour, while in the second run, cycle length and cycle split are adjusted at the end of each one hour period. Table 5.7 shows the simulation results obtained from the second round of simulation runs. Unlike the previous case, as expected, variable cycle algorithm produces less intersection delay than the variable splits case.

Table 5.7 Average Stopped Delay - The One-hour Time Intervals Case

Signal Timing Adjustment Algorithm	Intersection Delay (sec/veh)
Variable Splits	30.97
Variable Cycle	29.58

The reason for the discrepancy in the results obtained from the two rounds of simulation runs (i.e., the five minutes and one hour periods) can be explained by examining the calculated cycle length for each period. Tables 5.8 and 5.9 list the cycle lengths that are engaged in the first and second rounds of simulation runs under the variable cycle algorithm; respectively. These tables also include the actual values of cycle lengths. As indicated in these tables, the cycle lengths used differ from the values originally computed, since in some cases the cycle length might not be an interval that evenly divides into the designated periods. For example, in order to fit a cycle length of 55 seconds into a five minute period, the cycle length would have to be changed to either 50 or 60 seconds.

Table 5.8 Cycle Length - The Five-minute Time Intervals Case

	Cycle Length (sec)	
Period	Variable Cycle	Actual
1	50	45.13
2	100	92.73
3	43	40.81
4	60	65.38
5	50	50.51
6	60	56.67
7	50	47.22
8	50	51.01
9	50	54.26
10	60	61.45
11	43	43.59
12	50	51.52

Table 5.9 Cycle Length - The One-hour Time Intervals Case

	Cycle Length (sec)	
Period	Variable Cycle	Actual
1	46.75	46.65
2	92.31	90.13
3	40.91	40.48
4	64.29	63.16
5	48.65	48.46
6	52.94	52.41
7	46.75	46.39
8	50.71	50.58
9	55.38	54.69
10	66.67	66.59
11	41.86	41.75
12	52.17	51.91

The comparison between the cycle lengths listed in Tables 5.8 and 5.9 indicates that, in the case of five minute time interval, the cycle lengths used under a variable cycle algorithm deviate from their original optimum values by a larger margin than in the case of a one hour period. This is clearly shown in Figure 5.3, which compares the two cases by examining the absolute differences of the cycle lengths used in a variable cycle algorithm and their actual values. Therefore, the large deviations of the engaged cycle lengths from their actual values observed in the case of five minutes period explain the ability of a VSA traffic control system to generate a higher stopped delay under a variable cycle algorithm than under a variable splits algorithm at an intersection.

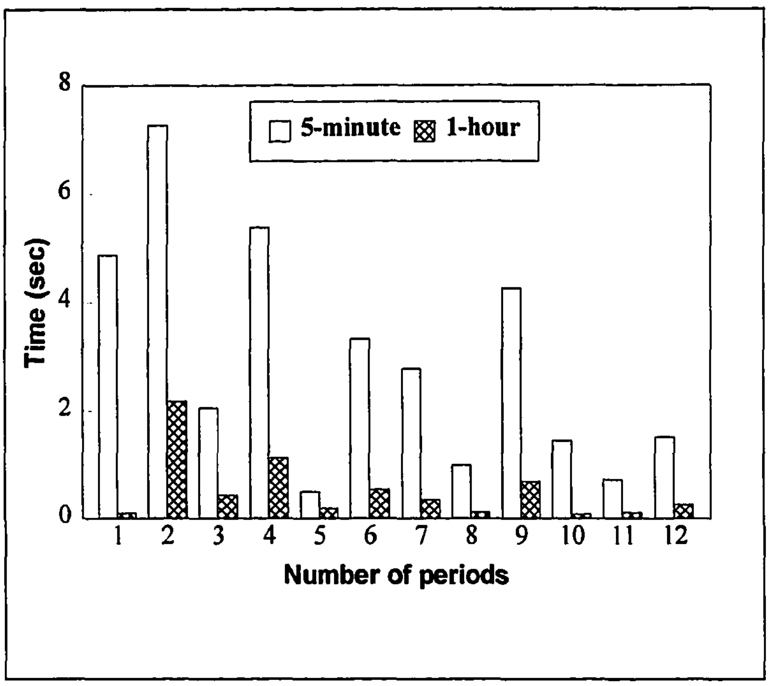

Figure 5.3 Absolute Cycle Length Differences

5.4 Validation of the Simulation Model

In this research a simulation model is developed that evaluates the applicability of AVSI in a traffic signal control system. According to the simulation results, a VSA traffic control strategy reduces intersection delay from the amount a pretimed system would produce. In order to establish an acceptable level of confidence that the obtained simulation results are correct, the simulation model is validated.

The validation of the simulation model is accomplished in two parts. First, an adequate level of confidence in the simulation model is established by comparing the results obtained from a field study and the ones obtained from the simulation model. Second, the established level of confidence is reinforced by showing a similarity between the model outputs and the Webster's equation.

5.4.1 Part 1 - Field Study

In validating a simulation model, the simulation results need to be compared to the actual system being simulated. The actual system in this case is a hypothetical VSA traffic signal control facility which does not yet exist. However, when the procedures defined in the simulation model for the VSA traffic control strategy are disengaged , the model simulates a pretimed control system. Because a VSA traffic control facility does not currently exist, the simulation model could only validated under a pretimed traffic control system. The purpose of this partial validation of the simulation model is to build a similar level confidence in the results obtained under the VSA signal control logic.

The first step in validating the simulation model is the field measurement of the average stopped delay produced under a pretimed control system at the intersection defined in the case study. While the intersection of Lincoln Way and Duff Avenue is operated under an actuated traffic control system, during the afternoon peak hour when the delay study was performed, the actuated system could be considered as a pretimed system since at a peak hour nearly all phases reach maximum green. The field measurement of stopped delay at the intersection was conducted by using the direct observation method (*59*, p. 9-71).

The direct observation method requires two observers to collect data at each approach. One observer with a stopwatch counts stopped vehicles within the approach, while the other person maintains a volume count by counting vehicles as they cross the stopline. A time interval of 20 seconds was selected for recording the number of stopped vehicles. The study started at 4:30 PM and it lasted in one hour. Thus, a total of 180 stopped vehicle counts were taken for each approach. These measurements and traffic volume counts were recorded on the provided field sheets. Samples of provided field sheets are included in Appendix IV.

The data collected are then entered into computer spreadsheets. A summary of these data are presented in Table 5.10. Given an interval time (I) of 20 seconds, and the information provided in first two rows of Table 5.10, the average stopped delay vehicles experienced during the one hour study at the intersection of Lincoln Way and Duff can be calculated by using Equation 4.1. The last row of the Table 5.10 shows the calculated average stopped delay of each approach.

Table 5.10 Summary of Data Collected During the Delay Study

	North bound	West bound	South Bound	East bound
Stopped Vehicles, V_s	1,887.00	1,332.00	2,001.00	1,510.00
Traffic Volume, V	665.00	633.00	665.00	630.00
Avg. Stopped Delay, (sec/veh)	56.75	42.09	60.18	47.94

The next step used to validate the simulation model is to run the model under a pretimed control system with the information given in Table 5.11, which was collected during the field study. The traffic flow rates, shown in Table 5.11, are adjusted to reflect unequal lane utilization. Using the lane utilization factor of 1.05, the actual flow rates are increased to reflect the flow rates in the lane with the highest utilization (*59*, p. 9-10).

Other information collected during the study include the intersection cycle length and its cycle splits. Since the intersection is controlled under a fully-actuated system, the observed cycle lengths are not exactly equal to each other. However, during the peak hour when the study was conducted, the intersection is assumed to be operating under a pretimed

Table 5.11 Observed Traffic information - Used in Simulation Model

	North bound	West bound	South Bound	East bound
Traffic Volume[1]	669.0	665.0	699.0	662.0
Truck Percentage	0.9	0.8	1.5	2.9
Right Percentage	11.3	25.9	12.9	15.9
Through Percentage	59.1	42.8	66.9	47.8
Left Percentage	29.6	31.3	20.2	36.3

[1] Adjusted volume - using a lane utilization factor of 1.05

control system. Therefore, an average value of 128 seconds is assumed for the cycle length. Moreover, the green time is assumed to be the same for each phase.

With the data collected during the field study, the simulation model is run under ten different seeds (i.e., sampling methods for the simulation random-number generators) for one hour. Each seed generates a different random number sample which in turns generates different vehicles' interarrival times. Table 5.12 includes the average stopped delay obtained in each simulation run which shows no significant changes in the results obtained in each run. The mean values of the average stopped delay, calculated in the last row of Table 5.12, and the average stopped delay obtained in the field study are included in Table 5.13 to make an easy comparison between the two methods.

Table 5.12 Average Stopped Delay (sec/veh) - Different Seeds

Seed	North b.	West b.	South b.	East b.	Intersection
4	53.05	47.92	49.97	50.65	50.46
15	55.55	46.53	48.55	50.14	50.31
52	55.41	49.57	54.11	51.85	52.87
342	51.42	47.56	53.95	46.79	50.07
845	52.36	47.72	52.12	46.52	49.82
5,896	52.74	47.01	55.89	54.69	52.67
13,571	55.04	46.93	51.21	48.57	50.54
18,777	52.93	50.09	52.47	47.35	50.76
55,555	55.92	47.68	54.91	48.35	51.86
697,321	50.56	50.59	48.54	50.39	50.01
Mean	53.49	48.16	52.17	49.53	50.94

Table 5.13 Average Stopped Delay (sec/veh) - Validation Results

	North b.	West b.	South b.	East b.	Intersection
Delay Study	56.75	42.09	60.18	47.94	51.74
Simulation Model	53.49	48.16	52.17	49.53	50.94

While the average stopped delay obtained from the two methods do not match exactly, they are close enough to establish a level of confidence that the model is capable of simulating an intersection operating under a pretimed traffic control system. The confidence in the simulation model yields a similar level of confidence in the model outputs obtained under a VSA traffic control strategy.

As stated earlier the simulation model and the actual system being simulated are not expected to have identical outputs. In addition to potential human error occurring during field data collection, other reasons for the discrepancy in the results obtained from the two methods include the followings:

- the assumption of pretimed control system for the intersection,
- the assumption of a lane utilization factor, and
- the assumption of a traffic arrival pattern.

The potential errors introduced by these three assumptions are discussed below.

5.4.1.1 Traffic Control System

Treating an actuated traffic control system during peak traffic conditions as a pretimed system may not accurately represent the intersection under study. While most of the phases reached maximum green time during the field observations, there were times that different cycle lengths and splits were observed. In order to show the effect of cycle length changes on the average stopped delay, the simulation model is run using various cycle lengths observed at the intersection. Figure 5.4 indicates that stopped delay is sensitive to the cycle length changes. Therefore, using an average cycle length and a constant green time in the simulation model to approximate a pretimed system in the simulation could be a potential reason for the differences between results obtained.

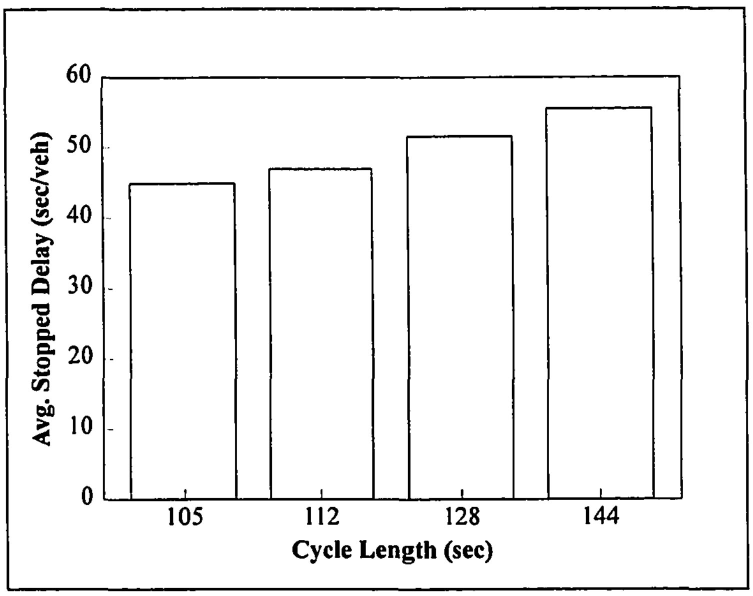

Figure 5.4 Intersection Average Stopped Delay - Different Cycle Length

5.4.1.2 Lane Utilization Factor

A lane utilization factor of 1.05, suggested by *1985 Highway Capacity Manual* (*58*, p. 9-10) for a 2-lane approach, was used for each approach. This is an estimated value which may not be applicable to all four approaches of the intersection. An estimation of lane utilization factor, based on the total number of stopped vehicles observed in each lane, indicates that the lane utilization in each approach could be different. Using the total number of stopped vehicles presented in Table 5.14, and the following expression, a different lane utilization factor can be estimated for each approach.

$$LUF = \frac{2X}{X+Y}\ ;X > Y \tag{5.1}$$

where,

LUF = estimated lane utilization factor,
X = total number of stopped vehicles in the lane with the higher utilization,
Y = total number of stopped vehicles in the lane with the lower utilization.

Table 5.14 Observed Total Stopped Vehicles at the Intersection

North bound		West bound		South bound		East bound	
In lane	Out lane	In lane	Out lane	In lane	Out lane	In lane	Out lane
880	1,007	652	680	743	1,258	681	829

Table 5.15 lists the lane utilization factors calculated from Equation 5.1. Using these calculated lane utilization factors, a different stopped delay value can be obtained for each approach. Figure 5.5 compares the average stopped delay obtained by a different set of lane utilization factors (i.e., calculated and suggested presented in Table 5.15) with the field average stopped delay. This figure indicates the effect of different lane utilization factors on the resulting average stopped delay at an intersection. Therefore, using the inappropriate lane utilization factors in the simulation model could be the second potential reason for discrepancies of the results.

Table 5.15 Different Sets of Lane Utilization Factors

	North b.	West b.	South b.	East b.
Suggested	1.05	1.05	1.05	1.05
Calculated	1.07	1.02	1.26	1.09

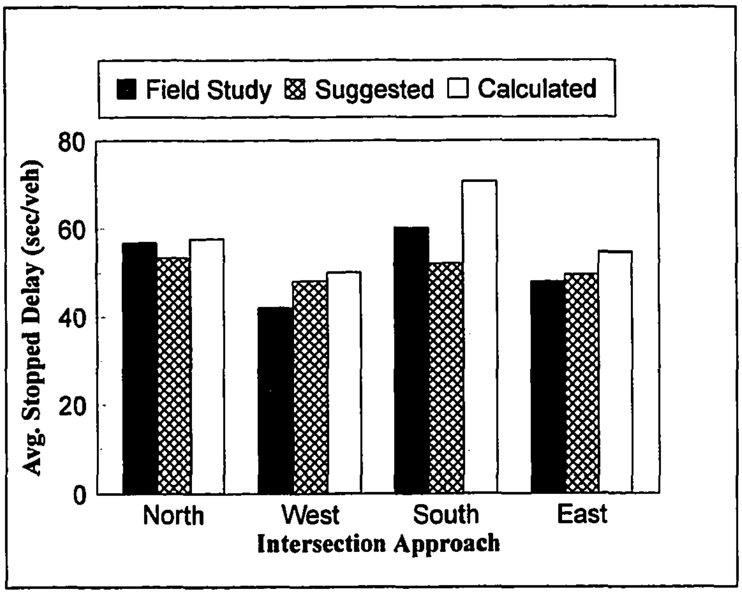

Figure 5.5 Intersection Approach Avg. Stopped Delay - Different Lane Utilization Factors

5.4.1.3 Traffic Arrival Pattern

The vehicle interarrival times generated by the negative exponential distribution in the simulation model may not exactly represent the actual traffic flow at the intersection. Using different vehicle interarrival times in the simulation model would generate different values for the resulting stopped delay. Thus, the dissimilarities between the observed vehicle interarrival times and the ones generated by the simulation model could be the third potential reason for obtaining different results.

5.4.2 Part 2 - Reinforcing the Simulation Validation

The established level of confidence in the simulation model can be reinforced by showing a similarity between the model's outcomes and the Webster's equation. The procedure used in the simulation model for obtaining stopped delay at an intersection is different from the Webster's equation, however, it can be shown that both models provide a similar relationship between stopped delay and cycle length. The stopped delay resulting from the simulation model is the measurement of the time that each single vehicle spends in the system, from its point of entry to the point that it leaves the intersection. In Webster's case, however, stopped delay can simply be calculated by solving the stopped delay equation, presented in *1985 Highway Capacity Manual* (*59*, p. 9-18).

Webster studied the effect of cycle length changes on average delay at an intersection. He developed a graph showing a relationship between the average delay and cycle length (*52*, p. 13). This graph indicates that the minimum average delay occurs at an optimum value of cycle length. However, at lower or higher values of cycle lengths, the intersection average delay increases.

In order to recreate the Webster's graph, the stopped delay equation is coded into the simulation model. In order to show the similarity between the average stopped delay resulted from the simulation model and the ones obtained from the stopped delay equation, the simulation model is run with different cycle lengths under the defined case study. With the assumption that there are no trucks in the traffic flow, the simulation model is run for one hour. At the end of each run, the average stopped delay resulting from the two procedures along with the engaged cycle length are recorded. Figure 5.6 graphs the result of 20 simulation runs showing the effect of cycle length changes on the average intersection stopped delay. The graphs obtained are not plotted on the same scale, that is, minimum stopped delay occurs at different cycle length in the two models. However, by considering the differences between the two procedures, they are similar enough to reemphasize the established level of confidence in the simulation model.

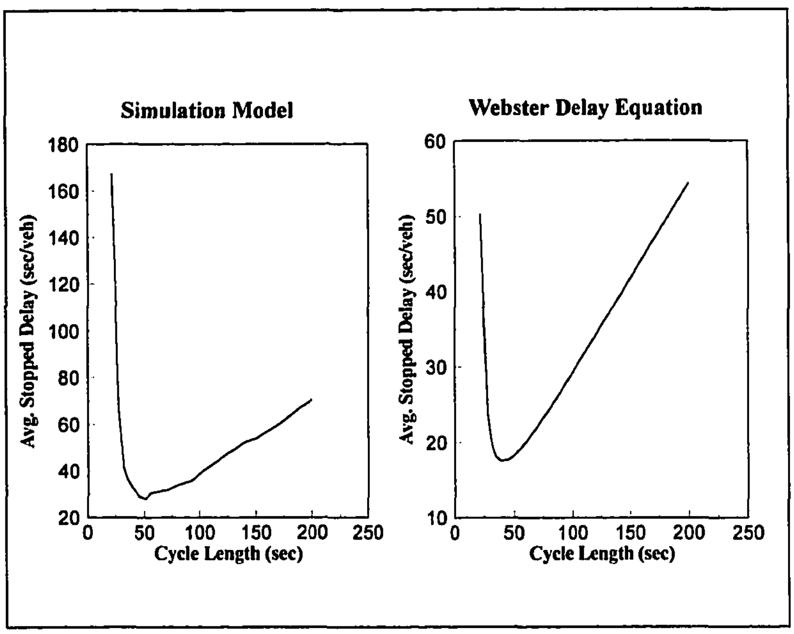

Figure 5.6 Average Stopped Delay versus Cycle Length

5.5 Average Stopped Delay

This section examines the sensitivity of the average stopped delay at an intersection controlled by a VSA traffic control system, with respect to the changes of following attributes and variables:

- truck percentage
- traffic flow rate
- truck discharge time

In this part of the analysis the simulation model is run under the defined case study while the questioned attribute or variable changes in each simulation run. The simulation running period is set for one hour for the first two attributes and two hours for the last one. The changes in average stopped delay are defined in terms of the time that vehicles could save at an intersection controlled by a VSA traffic control logic as compared to a pretimed system. The stopped delay reduction or the time saved by an engagement of a VSA control strategy at an intersection is determined by taking the average stopped delay resulting from a pretimed system from the one produced by a VSA control strategy. The remainder of this section examines the effect of each one of these parameters changes on the average stopped delay at an intersection.

5.5.1 Truck Percentage

As the number of trucks increase in a traffic flow, vehicles experience additional delay during their travel. However, the amount of this added delay may not be the same in all traffic signal control systems. For example, the lack of signal timing adjustment in a pretimed control system generates a higher delay than a VSA control strategy at an intersection. Figure 5.7 shows that a pretimed control system does not perform as well as a VSA control strategy at high truck percentages, thus vehicles save more time in a VSA traffic control system. Figure 5.7 also indicates that even at a truck percentage of zero, each passenger car could save as much as three seconds at an intersection operated under a VSA control system.

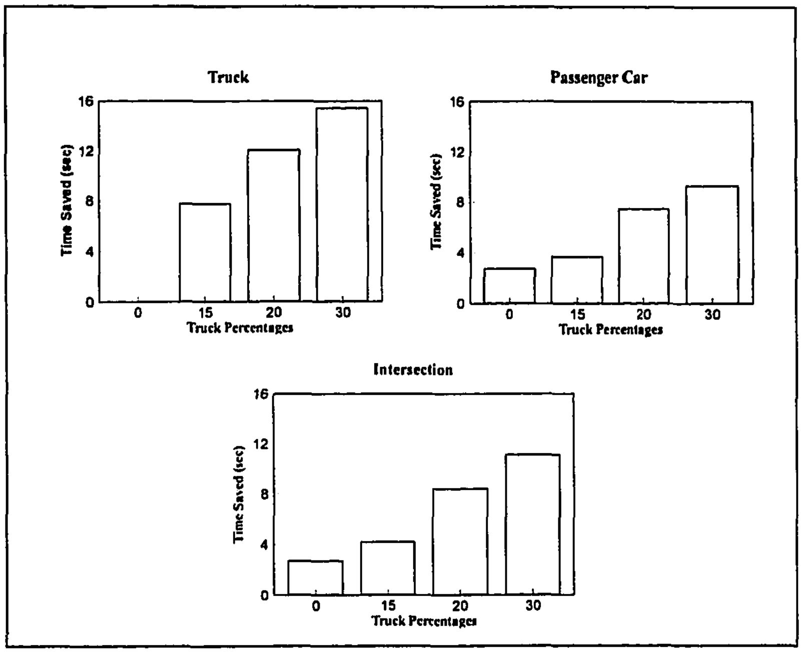

Figure 5.7 Travel Time Saved - Changing Truck Percentages

5.5.2 Traffic Flow Rate

Intersection stopped delay is also sensitive to the traffic flow rate. Figure 5.8 shows that, a VSA control system could save as much as 27 seconds of delay compared to a pretimed control system for each vehicle at an intersection, when traffic flow rate reaches to 700 vph in each approach. At flow rate of 700 vph, Figure 5.8 indicates that because of oversaturation condition a pretimed system experiences a lot more difficulty to handle the traffic flow at an intersection.

As shown in Figures 5.7 and 5.8, the increment of traffic flow rates and truck percentages at an intersection also increases the time that vehicles could save in a VSA controlled intersection. This hypothesis supports the conclusion reached in another study (3).

The advantage of adaptive control strategy over other types of traffic control systems is small or nonexistent under light traffic flow conditions.

5.5.3 Truck Discharge Time

This study used a conservative estimate of the time needed for a truck to clear an intersection. The time used is 1.5 times the discharge time of a passenger car. However, by

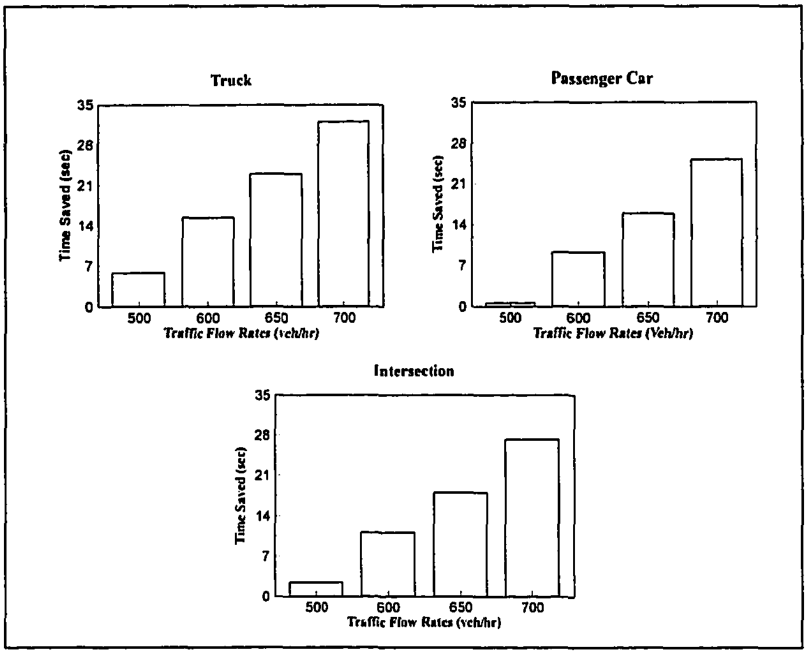

Figure 5.8 Travel Time Saved - Changing Traffic Flow Rates

changing the truck's discharge time from three seconds to four seconds (i.e., two times the discharge of a passenger car), Figure 5.9 shows that a VSA traffic control system could reduce intersection stopped delay substantially. In this case, the total number of trucks observed during the two hours simulation run (i.e., 1,440 trucks), could save more than 18 hours in a VSA traffic control system. Therefore, in an actual traffic control system where trucks may take more time to clear an intersection, a VSA control logic would be a valuable tool to reduce traffic delay at an intersection.

5.6 Concluding Remarks

It has been shown that VSA traffic control logic is capable of reducing stopped delay at an isolated intersection. As Figure 5.10 indicates the first part of the VSA control strategy has a larger contribution in reducing intersection stopped delay. As shown in this figure, passenger cars only receive a small portion of stopped delay reduction from part two of the control strategy. This may be because this part may only adjust signal timing for trucks. With trucks, however, part two could reduce an additional 5.62 seconds of stopped delay per truck. This contributes to more than 38 percent of the total trucks' stopped delay reduction.

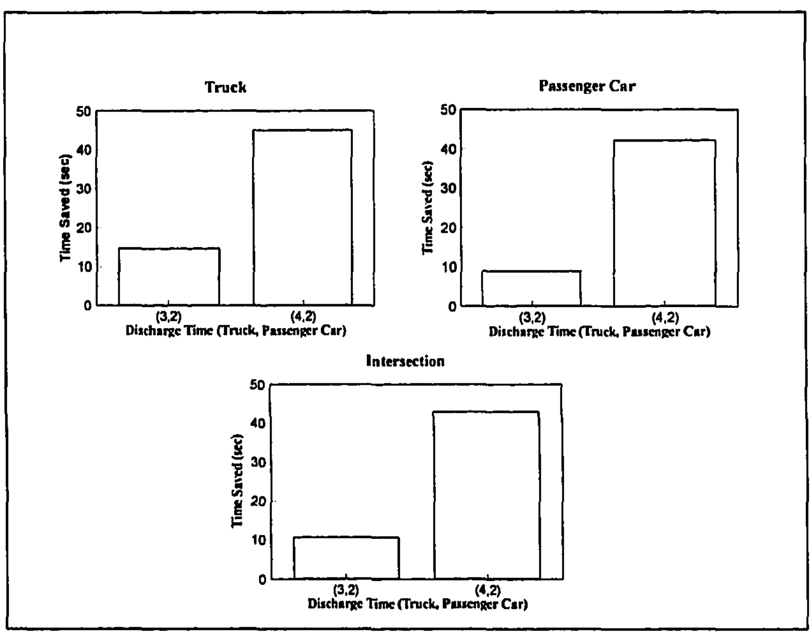

Figure 5.9 Travel Time Saved - Changing Trucks' Discharge Times

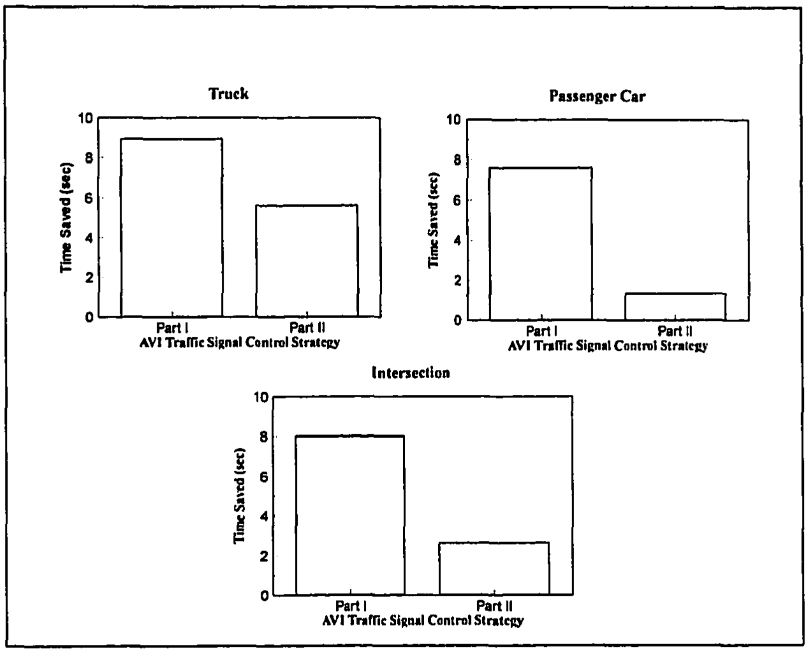

Figure 5.10 Travel Time Saved - The Two Parts of VSA Traffic Signal Control Strategy

The decision-making processes involved in the second part of the VSA control logic are responsible of providing a logical decision whether to extend a green time or leave it as it has been scheduled. Most of the requested green time extensions are granted when the next competing phase has extra green time to lend. Otherwise, the green time extension requests are transferred to the delay index level, where only a few requests are granted. This indicates that delay index is an additional safety measure in the VSA control logic which ensures that vehicles waiting in the next competing phase do not experience excessive delay.

Another feature of the VSA control strategy is the detection of trucks which may arrive within the next 20 seconds. If no truck is detected at R_{2a} readers (see Figure 5.1), the VSA control logic checks to see if any truck is detected at R_{2b} readers. The simulation results indicate that not many trucks receive green time extensions at this stage. In this case, the requested green time extensions are often greater than the next competing phase's extra green time, since the detected truck has not yet arrived. These long green time extension requests make it difficult for the next competing phase to lend the requested green time. The simulation results, however, indicate even the few observed cases where the next competing phase provides its surplus green time to a detected truck which will arrive, for example, in 10 seconds, have noticeably reduced truck's stopped delay at the intersection.

CHAPTER 6 - CONCLUSIONS

AVSI is a generic name for advanced vehicle detection systems. It is assumed that AVSI is capable of detecting vehicles and providing vehicle specific information. In the application of AVSI to traffic signal control systems, AVSI is assumed to provide types and arrival times of the approaching vehicles to a local microprocessor-based traffic signal controller. Based on the AVSI traffic information, the controller then adjusts its signal timing to reduce intersection delay.

The purpose of the research is to explore the potential benefits of AVSI vehicle specific information in a traffic signal control system. To limit the scope of the research, the benefits of use of AVSI traffic information are evaluated at a four-leg, four-phase isolated intersection, shown in Figure 5.1. Future research may look at the benefits of AVSI in traffic signal control systems.

A microscopic simulation model is developed to evaluate AVSI performance at an isolated intersection. An important element of the simulation model is the development of a VSA traffic signal control strategy. The VSA traffic control system adjusts the signal timing based on AVSI traffic information. The objective of VSA control logic is to provide optimum cycle splits at an intersection, in order to minimize the stopping of slow-moving heavy vehicles at a red signal.

The simulation results indicate that through the use of AVSI traffic information a VSA control strategy is capable of reducing average stopped delay of vehicles at the intersection. Compared with pretimed control systems, it is found that average stopped delay of trucks and passenger cars at the isolated intersection is reduced by 14.56 and 8.98 second per vehicle, respectively.

This research also found that VSA control logic is capable of reducing maximum stopped delay and queue length at the intersection. In one particular scenario, the obtained simulation results indicate that the maximum stopped delay of trucks on an approach is reduced by more than 93 seconds. On the left lane of the same intersection approach, the simulation results show that maximum queue could also be shortened by eight vehicles.

The benefits of AVSI are more apparent in a traffic control system with high traffic volumes and increasing truck percentages. It has been shown, however, that even at zero truck percentage in a traffic flow rate of 600 vph per intersection approach in the case study, each passenger car could save as much as 2.73 seconds when a VSA traffic control system is used.

The logic involved in a VSA control strategy not only could alleviate the drivers' frustration by reducing intersection stopped delay, it also reduces vehicles' travel times. Assuming a value of 16.25 dollars per hour for the travel time savings of a larger semi-truck (*70*, p. 3-20), the simulation results of the case study indicate that each truck could save about 6.6 cents in a VSA traffic controlled intersection. Therefore, 1,440 trucks observed during the two hour simulation run period could save a total of 94 dollars in a VSA control system.

The truck travel time savings in a VSA signal control system is calculated based on the assumption of a conservative average discharge time of three seconds for each truck. In an actual intersection, however, the discharge time of trucks could vary based on the position of a truck in a queue. A truck at the head of a queue will obviously take more time to clear an intersection after the signal turns green than a truck in the fifth position will. Therefore, by assuming a more realistic average discharge time for each truck (i.e., 4 seconds), it is found that VSA control logic could reduce intersection stopped delay substantially. With the assumption of 4 seconds discharge time for each truck, the simulation model results indicate a 18 hour travel time savings for the 1,440 trucks observed during the two hour run period in a VSA traffic controlled intersection. Assuming an hourly truck travel time savings of 16.25 dollars, each truck saves 20 cents which adds up to more than 292 dollars savings for all trucks during a two hour peak period.

The application of AVSI in a traffic signal control system has been presented in its simplest form in this research. The vehicle travel time savings in a VSA traffic signal controlled intersection can of course be multiplied, if AVSI is applied to intersections in an arterial or a network system. The present version of the simulation model is developed for the defined case study; however, the model's flexibility allows it to be enhanced to simulate a more complex traffic control system. For example, inclusion of additional intersections would make the model capable of evaluating AVSI benefits at an arterial.

The system evaluated in this research does not yet exist; however, with the emphasis of ITS technology on ATMS it may not be far from reality that in the next few years the concept of this research is actually applied. The major obstacle in implementing AVSI in traffic control systems at this time is not related to the technical adequacy of AVSI; rather, it is the inadequacy of the existing traffic control systems (i.e., pretimed and actuated systems) which are not designed to utilize additional information that AVSI is providing. The development of the VSA control strategy in this research can be considered a step in the actual implementation of AVSI in traffic control systems. ITS technology is expected to facilitate the further research required for actual implementation of AVSI in traffic control systems.

6.1 Future Studies

The quality of this research can be improved by enhancing the simulation model to better emulate the traffic signal control system under study. The following suggestions will help improve the simulation model:

6.1.1 Additional AVSI Vehicle Specific Information

The VSA control logic used in the current version of the simulation model only considers arrival times and types of vehicles (i.e., trucks and passenger cars) in its signal timing adjustment. The consideration of other AVSI vehicle specific information such as acceleration-deceleration rate, speed, weight, and size of vehicles may provide a better decision-making base for a VSA control strategy to adjust cycle splits more appropriately, because the signal timing adjustment in a VSA control strategy is based on vehicle performance. In addition, classifying vehicles into more classes may also provide an improved decision-making base for VSA control logic. Thus, the more detailed the information provided to a VSA control system, the better it could perform at intersections.

6.1.2 Car-following Technique

The simulation model could further be enhanced when a car-following technique is added to the traffic flow module of the model. Using the acceleration-deceleration rate of vehicles, a car-following model is capable of considering the interactions between vehicles in a traffic flow. If, for example, a leading vehicle decelerates, the following vehicle also decelerates. Although the technique used in this simulation model is an appropriate approach for simulating traffic flows at intersections, a car-following model seems to better capture the movement of vehicles on the intersection approaches.

6.1.3 AVSI in Traffic Signal Control systems

The application of AVSI to an arterial or a network system would require to examine a traffic control system with more than one intersection. This task could be begun with a simple case study evaluating AVSI at two adjacent intersections. In this case, a VSA control strategy should not only adjust the cycle splits, it should also be capable of adjusting the offset time of intersections. Once a model is built for two adjacent intersections, it could be further enhanced to include more intersections to simulate an arterial or a network system with a higher degree of complexity.

6.1.4 Variable Discharge Time

The present version of the simulation model considers constant discharge times for processing vehicles through the intersection. Assigning different discharge times based on the position of the vehicles in the queue, however, would provide a better estimation for the times that vehicles need to be discharged, once they arrive at the intersection. Because of trucks' low acceleration rates, a truck positioned at the head of a queue requires more discharge time than a passenger car does. In the case of variable discharge times, therefore, based on the location of vehicles in the queue appropriate discharge times are assigned to the vehicles. The assigned discharge times, however, should be changed if vehicles remain in the queue after the green time is terminated. If, for example, a vehicle which is assigned a seventh vehicle discharge time can not clear the intersection with the provided given green time, based on its new position in the queue a new discharge time will be assigned to the vehicle.

REFERENCES

1. R. Whelan. *Smart Highways, Smart Cars.* Artech House, Boston, MA, 1995.

2. G. E. Euler and R. H. Douglas. *National ITS Program Plan: Intelligent Transportation Systems.* ITS America, Washington, D.C., 1995.

3. F. Lin and S. Vijayakumar. *Adaptive Signal Control at Isolated Intersections.* Transp. Engineering Journal of ASCE, No. 5, 1988, pp. 555-573.

4. K. L. Bang. *Optimal Control of Isolated Traffic Signals.* Traffic Eng. Control. 17, 1976, pp. 288-292.

5. N. Gartner. *OPAC: A Demand-Responsive Strategy for Traffic Signal Control.* Transp. Res. Record, No. 906, 1983.

6. S. M. Elahi, A. E. Radwan, and K. M. Goul. *Knowledge-Based System for Adaptive Traffic Signal Control.* Transp. Res. Record, No. 1324, 1991.

7. Institute of Transportation Engineers. *Traffic Control Systems Handbook.* Washington, D.C., 1985.

8. J. H. Kell and I. J. Fullerton. *Manual of Traffic Signal Design.* 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

9. Tomar Electronics Inc.. *Traffic Control Preemption System*, Gilbert, AZ.

10. Econolite Control Products Inc. *AUTOSCOPE-2003LE*, Anaheim, CA.

11. Electronic Integration Systems Inc. *Remote Traffic Microwave Sensor*, Toronto, Canada.

12. Peek Traffic Inc.. *Dual Mode Traffic Detector-2001-DMTD by Westinghouse.* Sarasota, FL.

13. J. W. Wang, E. R. Case, and D. Manor. *The Road Traffic Microwave Sensor (RTMS).* Third Int. Vehicle Navigation & Information Systems Conference, Oslo, Norway, 1992, pp. 83-90.

14. P. G. Michalopoulos and R. Jacobson. *Field Implementation of the Minnesota Video Detection System*, Third Int. Vehicle Navigation & Information Systems Conference, Oslo, Norway, 1992, pp. 104-111.

15. S. Y. Hsu and P. K. Munjal. *Freeway Digital Simulation Models*. Transp. Res. Record, No. 509, 1974.

16. A. J. Santiago and H. Chen. *Traffic Modeling Software for IVHS applications*. Proceedings of the 1990 Winter Simulation Conference, 1990, pp. 759-762.

17. K. Yamada and T. N. Lam. *Simulation Analysis of Two Adjacent Traffic Signals*. Proceedings of the 1985 Winter Simulation Conference, 1985, pp. 454-463.

18. A. A. Saka, C. Anandalingam, and N. J. Garber. *Traffic Signal Timing at Isolated Intersection Using Simulation Optimization*. Proceedings of the 1986 Winter Simulation Conference, 1986, pp. 795-800.

19. R. F. Benekohal and J. Treiterer. *CARSIM: Car-Following Model for Simulation of Traffic in Normal and Stop and Go Conditions*. Transp. Res. Record, No. 1194, 1988.

20. M. Zarean and Z. A. Nemeth. *WEASIM: A Microscopic Simulation Model of Freeway Weaving Sections*. Transp. Res. Record, No. 1194, 1988.

21. A. K. Rathi and A. J. Santiago. *Urban Network Traffic Simulation: TRAF-NETSIM Program*. Transportation Engineering Journal of ASCE, No. 6, 1990, pp. 734-743.

22. McTrans Catalog. *Traffic Engineering-Signal Timing & Warrants*. Transportation research center, University of Florida, June 1993.

23. C. E. Lee and R. F. Inman. *TEXAS Model for Intersection Traffic-Guide to Data Entry*. Center for Transportation Research, University of Texas at Austin, 1985.

24. A. G. R. Bullen, N. Hummon, T. Bryer, and R. Nekmat. *EVAPAS: A Computer Method for the Optimal Design of a Vehicle-Actuated Traffic Signal*. Transp. Res. Record, No. 1114, 1987.

25. J. L. Pline. *Traffic Engineering Handbook*. 4th Ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.

26. R. Jayakrishnan and H. S. Mahmassani. *Dynamic Simulation-Assignment Methodology to Evaluate In-Vehicle Information Strategies in Urban Traffic Networks*. Proceedings of the 1990 Winter Simulation Conference, 1990, pp. 763-769.

27. M. F. McGurrin and T. R. Wang. *An Object-Oriented Traffic Simulation with IVHS Applications*, Vehicle Navigation & Information Systems Conference Proceedings Part 1, Dearborn, MI, 1991, pp. 551-561.

28. G. F. Luger and W. A. Stubblefield. *Artificial Intelligence-Structures and Strategies for Complex Problem Solving*. 2nd Edition, The Benjamin/Cummings Publishing Company Inc., Redwood City, CA, 1992.

29. A. J. Santiago and A. K. Rathi. *Simulation and Intelligent Vehicle Highway Systems*. Third Int. Vehicle Navigation & Information Systems Conference, Oslo, Norway, 1992, pp. 490-498.

30. A. J. Miller. *A Computer Control System for Traffic Network*. Proc. 2nd Int. Symp. On Theory of Road Traffic Flow, London, U.K., 1963, pp. 201-220.

31. R. A. Vincent and C. P. Young. *Self-Optimization Traffic Signal Control Using Microprocessors-the TRRL 'MOVA' Strategy for Isolated Intersections*. Traffic Eng. Control. 27, 1986, pp. 385-387.

32. F. Lin. *Efficiency and Information Availability in Adaptive Control of Individual Intersection*. Traffic Eng. Control. 26, 1985, pp. 434-438.

33. F. Lin and D. J. Cooke. *Potential Performance Characteristics of Adaptive Control at Individual Intersections*. Transp. Res. Record, No. 1057, 1986.

34. H. Chen, C. C. Liu, S. L. Cohen, P. J. Tarnoff, and R. L. Sumner. *An adaptive Control Strategy for Signalized Intersections*. Proc. Microcomputer Applications in Transportation-II International Conference, ASCE, New York, 1987, pp. 55-63.

35. F. Lin, D. C. Cooke, and S. Vijayakumar. *Use of Predicted Vehicle Arrival Information for Adaptive Signal Control-An Assessment*. Transp. Res. Record, No. 1112, 1987.

36. P. B. Hunt, D. I. Robertson, and R. D. Bretherton. *The SCOOT on-line Traffic Signal Optimization Technique*. Traffic Eng. Control. 23, 1982, pp. 190-192.

37. J. C. Greenough and B. Y. Quan. *The Metropolitan Toronto SCOOT Demonstration Project-The Next Frontier*. Prepared for: Urban Traffic Engineers Council, Institute of Transportation Engineers, 1991.

38. P. T. Martin and S. L. M. Hockaday. *SCOOT-An Update*. Institute of Transportation Engineers Journal, No. 1, January, 1995, pp. 44-48.

39. D. I. Robertson and R. D. Bretherton. *Optimizing Networks of Traffic Signals in Real Time-The SCOOT Method*, IEEE Transactions on Vehicular Technology, No. 1, 1991, pp 11-15.

40. E. Sauthier and B. Faltings. *Model-Based Traffic Control.* Artificial Intelligence in Engineering, No. 7, 1992, pp 139-151.

41. P. Carden and M. McDonald. *The Application of SCOOT Control to an Isolated Intersection.* Traffic Eng. Control. 26, 1985, pp. 304-310.

42. R. D. Bretherton and G. I. Rai. *The Use of SCOOT in Low Flow Conditions.* Traffic Eng. Control. 23, 1982.

43. P. B. Hunt and T. R. Holland. *The Effect of an Incident in a SCOOT System-In Simulation and On-Street.* Traffic Eng. Control. 26, 1985, pp. 55-58.

44. L. de la Breteque and R. Jezequel. *Adaptive Control at an Isolated Intersection-a Comparative study of Some Algorithms.* Traffic Eng. Control. 20, 1979, pp. 361-363.

45. F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. *Building Expert Systems.* Addison-Wesley Publishing Company, Reading, MA, 1983.

46. A. T. Chang and R. Y. Tang. *A Prototype Expert System for Traffic Signal Control in real time.* 1st Int. Conf. on Applications of Advanced Technologies in Transportation Engineering, San Diego, CA, 1989.

47. S. G. Ritchie and R. A. Harris. *Expert Systems in Transportation Engineering.* Expert Systems for Civil Engineers: Technology and Application, Edited by Mary Lou Maher, ASCE, 1987, pp. 129-143.

48. K. M. Goul, T. J. O'Leary, A. E. Radwan. *Expert Systems for Traffic Signal Control*, Proc. Microcomputer Applications in Transportation-II International Conference, ASCE, New York, 1987, pp. 629-638.

49. G. L. Chang, C. S. Chin and N. Kartam. *Selection of Expert System Shells for Traffic Signal Design.* Journal of Advanced Transportation, No. 3, 1992, pp. 241-273.

50. F. Lin. *Knowledge Base on Semi-Actuated Traffic-Signal Control.* Transp. Engineering Journal of ASCE, No. 4, 1991, pp. 398-417.

51. C. Zozaya-Gorostiza and C. Hendrickson. *Expert System for Traffic Signal Setting Assistance*, Transp. Engineering Journal of ASCE, No. 2, 1987, pp. 108-126.

52. F. V. Webster. *Traffic Signal Settings*. Road Research Technical Paper No. 39, Road Research Laboratory, 1958, pp. 1-44.

53. J. S. Linkenheld, R. F. Benekohal, and J. H. Garrett Jr.. *Knowledge-Based System for Design of Signalized Intersections*. Transp. Engineering Journal of ASCE, No. 2, 1992, pp. 241-257.

54. S. B. Pattnaik, S. Rajeev, and A. Mukundan. *Toward Intelligent Traffic Signal Design System*. Transp. Engineering Journal of ASCE, No. 5, 1991, pp. 524-539.

55. S. M. Elahi, A. E. Radwan, and K. M. Goul. *Traffic Signal Using Mixed Controller Operations*. Transportation Engineering Journal of ASCE, No. 6, 1992, pp. 866-880.

56. S. M. Elahi. *Signal Control Strategies Using Expert Systems*. Ph.D. Dissertation, College of Engineering and Applied Sciences, Arizona State University, Tempe, AZ, 1990.

57. A. E. Radwan, M Goul, T. J. O'Leary, and K. E. Moffitt. *A Verification Approach for Knowledge Based Systems*. Transportation Research, No. 4, 1989, pp. 287-300.

58. H. Inose and T. Hamada. *Road Traffic Control*. University of Tokyo Press, Tokyo, Japan, 1975.

59. *Highway Capacity Manual-Special Report 209*. Transportation Research Boarch, National Research Council, Washington, D.C., 1985.

60. D. L. Gerlough and M. J. Huber. *Traffic Flow Theory A Monograph*. Trans. Res. Board, Special Report 165, Washington, D.C., 1975.

61. S. O. Simonsson. *Car-Following as a Tool in Road Traffic Simulation*. Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference, Ottawa, Canada, 1993, pp. 150-153.

62. M. Wohl and B. V. Martin. *Traffic System Analysis for Engineers and Planners*. McGraw-Hill Book Company, New York, NY, 1967.

63. C. D. Pegden, R. E. Shannon, and R. P. Sadowski. *Introduction to Simulation Using SIMAN*. McGraw-Hill Company, New York, NY, 1990.

64. B. D. Greenshields, D. Shapiro, and E. L. Erickson. *Traffic Performance at Urban Street Intersections*. Tech. Report No. 1, Yale Bureau of Highway Traffic, New Haven, CT, 1947.

65. D. G. Capelle and C. Pinnel. *Capacity Study of Signalized Diamond Interchanges.* Highway Research Board Bulletin, No. 291, 1961, pp. 1-25.

66. W. R. Blunden. *The Land-Use Transportation System Analysis and Synthesis.* Pergamon Press Ltd., Headington Hill Hall, Oxford, England, 1971.

67. E. K. Morlok. Introduction to Transportation Engineering and Planning. McGraw-Hill Book Company, New York, NY, 1978.

68. W. R. Reilly, J. H. Kell, and I. J. Fullerton. *Design of Urban Streets Student Textbook.* U.S. Department of Transportation, September, 1977.

69. *A Manual on User Benefit Analysis of Highway and Bus-Transit Improvements.* American Association of State Highway and Transportation Officials, Washington, D.C., 1977.

70. B. Allen, C. P. Baumel, D. J. Forkenbrock, and D. Otto. *Guide to the Economic Evaluation of Highway Projects.* Prepared for Iowa Department of Transportation, Wilbur Smith Associates, 1993.

ACKNOWLEDGMENTS

Fulfillment of this research effort would not have been possible without the generosity and support of so many. I am first and foremost indebted to Almighty God whose blessing and guidance was always evident, particularly when I needed it most.

I wish to thank my committee members Dr. Kenneth Brewer, Dr. James Cable, and Dr. Robert Strahan for their constructive feedback throughout the development of this research. Four other members of my committee deserve special recognition. Dr. Reginald Souleyrette took the initiative in acting as a champion for this research and to assist me in numerous intangible ways. Dr. Kathleen Waggoner provided practical advice and expert editing that immeasurably facilitated the research process. Dr. Tom Barta, my co-major professor was always accessible and supportive. Finally, my deepest gratitude is extended to Dr. Tom Maze, my advisor and major professor, for providing me with the opportunity to pursue my goals, supplying financial backing, and challenging me to acquire the skills essential of a researcher and educator.

My sincere appreciation is extended to the Iowa Transportation Center and its staff for supplying the technological and administrative resources necessary to complete this manuscript. I am grateful to Mr. Bill McCall for demonstrating the attributes of a true professional and for understanding the commitment that this research required. I wish to thank my colleagues and office mates, especially my good friend James Hunt for making those weekends and long nights at the office bearable.

The contribution made by my family and friends in helping me achieve this goal can not be underestimated. The source of all my inspiration comes from the memories of my father, Rahmatollah Kamyab, and my brother, Parviz Kamyab. I like to express my gratitude to my mother Zahra Kamyab. I am forever grateful to my brother, Khosro Kamyab, for sacrificing so much in order to raise me and instill in me the values of perseverance and dedication. My sister, Shirin Kamyab, deserves a special thanks for her love and warmth. I also wish to thank my father-in-law, Hossain Ebnejalal, for teaching me virtue of integrity and for always providing me with his unconditional support. Words can not describe the respect I have for my wife's aunt, Kobra Ebnejalal, and how her prayers and kindness consistently uplift me.

Finally, I am forever grateful to my wife, Khojasteh, and my children Mariam, Somayeh, and Ahmad, who made this all possible with their consistent love and reassurance. Khojasteh demonstrated the true meaning of marital sacrifice through her strength, tenderness, and willingness to always boost my moral. Because I am able to share my life with her is yet another example of the greatness of God.

APPENDIX I - EXPONENTIAL DISTRIBUTION

For Poisson-distributed arrivals, *p(n)*, the probability of exactly *n* vehicles arriving in any *t* second interval, is given by (*61*, p. 353):

$$p(n) = \frac{\mu^n e^{-\mu}}{n!} \qquad \text{(A.1)}$$

where,

$\mu = \lambda t$ = average number of vehicle arrivals in time t,
$\lambda = \frac{V}{T}$ = mean rate of arrivals per unit time interval,
V = total volume of vehicles arriving during time T.

If no vehicles arrive in time t, then there must have been a gap or time headway of at least *t* second(s) between successive vehicles. Therefore, the probability of a headway *h* being greater than or equal to *t* is:

$$P(h \geq t) = e^{-\lambda t} \qquad \text{(A.2)}$$

Because of the mathematical form of this distribution, it is often termed the negative exponential distribution.

APPENDIX II - SIMULATION PROGRAM LISTINGS

Simulation Model - MODEL

Lincoln Way and Duff Avenue Intersection

```
;Traffic Signal Control Module

        BEGIN;

        CREATE;

        ASSIGN: StartTime=TNOW;

        ASSIGN: ChkTime=Spd/Decel;

;Saturation Flow Rates

        ASSIGN:
                SatFlow1=2*1800*TF(hv%,Truck%1):
                SatFlow2=2*1800*TF(hv%,Truck%2):
                SatFlow3=2*1800*TF(hv%,Truck%3):
                SatFlow4=2*1800*TF(hv%,Truck%4):
                VoverS1=ActFlow1/SatFlow1:
                VoverS2=ActFlow2/SatFlow2:
                VoverS3=ActFlow3/SatFlow3:
                VoverS4=ActFlow4/SatFlow4;

;Cycle Length Calculation

        ASSIGN:
                CycleLength=((1.5*LostTime)+5)/
                        (1-(VoverS1+VoverS2+VoverS3+VoverS4));

;Keeping Track of Number of Cycle in each period

        ASSIGN: Multi=1;
Add     ASSIGN:
                Multi=Multi+1:
                Adj=CycleLength*Multi;

        BRANCH,1:
```

```
                IF,Adj.GT.Period1,Subs:
                ELSE,Add;

Subs    ASSIGN:
                Multi=Multi-1:
                CycleLength=((Period1-(CycleLength*Multi))/Multi)+CycleLength;
;
;Selection of AVSI or Non AVSI System

        BRANCH,1:
                IF,Option==0,AVI:
                ELSE,NoAVI;

;In an AVSI System Traffic Volume and Truck Percentages Varies

AVI     READ, F,FREE: ActFlow1,ActFlow2,ActFlow3,ActFlow4,
                      Truck%1,Truck%2,Truck%3,Truck%4;

        ASSIGN:
                SatFlow1=2*1800*TF(hv%,Truck%1):
                SatFlow2=2*1800*TF(hv%,Truck%2):
                SatFlow3=2*1800*TF(hv%,Truck%3):
                SatFlow4=2*1800*TF(hv%,Truck%4):
                VoverS1=ActFlow1/SatFlow1:
                VoverS2=ActFlow2/SatFlow2:
                VoverS3=ActFlow3/SatFlow3:
                VoverS4=ActFlow4/SatFlow4;

;Cycle Splits Calculation

NoAVI ASSIGN:
                Xc=(VoverS1+VoverS2+VoverS3+VoverS4)*CycleLength/
                   (CycleLength-LostTime):
                Green1=VoverS1*(CycleLength/Xc):
                Green2=VoverS2*(CycleLength/Xc):
                Green3=VoverS3*(CycleLength/Xc):
                Green4=VoverS4*(CycleLength/Xc);

;Webster Delay Equation
;
;       ASSIGN:
;               Cap1=SatFlow1*(Green1/CycleLength):
```

```
;               Cap2=SatFlow2*(Green2/CycleLength):
;               Cap3=SatFlow3*(Green3/CycleLength):
;               Cap4=SatFlow4*(Green4/CycleLength):
;               X1=ActFlow1/Cap1:
;               X2=ActFlow2/Cap2:
;               X3=ActFlow3/Cap3:
;               X4=ActFlow4/Cap4:
;               Delay1=((0.38*CycleLength)*(((1-(Green1/CycleLength))**2)/
;                       (1-((Green1/CycleLength)*X1))))+
;                       (173*(X1**2)*((X1-1)+((((X1-1)**2)+(16*(X1/Cap1)))**0.5))):
;               Delay2=((0.38*CycleLength)*(((1-(Green2/CycleLength))**2)/
;                       (1-((Green2/CycleLength)*X2))))+
;                       (173*(X2**2)*((X2-1)+((((X2-1)**2)+(16*(X2/Cap2)))**0.5))):
;               Delay3=((0.38*CycleLength)*(((1-(Green3/CycleLength))**2)/
;                       (1-((Green3/CycleLength)*X3))))+
;                       (173*(X3**2)*((X3-1)+((((X3-1)**2)+(16*(X3/Cap3)))**0.5))):
;               Delay4=((0.38*CycleLength)*(((1-(Green4/CycleLength))**2)/
;                       (1-((Green4/CycleLength)*X4))))+
;                       (173*(X4**2)*((X4-1)+((((X4-1)**2)+(16*(X4/Cap4)))**0.5))):
;               WebDelay=((Delay1*ActFlow1)+(Delay2*ActFlow2)+
;                          (Delay3*ActFlow3)+(Delay4*ActFlow4))/
;                          (ActFlow1+ActFlow2+ActFlow3+ActFlow4);
;
;Traffic Signal Timing

        ASSIGN: CPh=1;

Cycle   QUEUE, Dum1Q;

        PREEMPT: PhaseTR2;              East, T_R Red
        PREEMPT: PhaseTL2;              East, T_L Red

        PREEMPT: PhaseTR3;              South, T_R Red
        PREEMPT: PhaseTL3;              South, T_L Red

        PREEMPT: PhaseTR4;              West, T_R Red
        PREEMPT: PhaseTL4;              West, T_L Red

        DELAY: Green1-ChkTime-TimeNeedT;

;PART II of the VSA Traffic Signal Control Strategy
```

```
        BRANCH,1:
                IF,PARTII==2.AND.Option==0,PII1:
                ELSE,Fine1;

PII1    ASSIGN:
                K=1:
                NewG1=Green1-TimeNeedT:
                TimeNeedT=0;

More1 BRANCH,1:
                IF,NQ(PhaseTR2Q).GT.440.OR.NQ(PhaseTL2Q).GT.440.OR.
                NQ(PhaseTR3Q).GT.440.OR.NQ(PhaseTL3Q).GT.440.OR.
                NQ(PhaseTR4Q).GT.440.OR.NQ(PhaseTL4Q).GT.440,Fine1:
                IF,NQ(PhaseTR1Q).GE.JR2a,QSrch12:
                IF,NQ(PhaseTL1Q).GE.JL2a,QSrch13:
                ELSE,Cond1;

QSrch12 SEARCH, PhaseTR1Q,JR2a: VehType==1;

        BRANCH,1:
                IF,J.GT.0,J2a12:
                IF,NQ(PhaseTL1Q).GE.JL2a,QSrch13:
                ELSE,Cond1;

J2a12   ASSIGN: JR2a=J+1: NEXT(QTrk12);

QSrch13 SEARCH, PhaseTL1Q,JL2a: VehType==1;

        BRANCH,1:
                IF,J.GT.0,J2a13:
                ELSE,Cond1;

J2a13   ASSIGN: JL2a=J+1: NEXT(QTrk13);

Cond1 BRANCH,1:
                IF,NR(ReaderTR1)==1.AND.TypArLog12(2)==1
                  .AND.TagR==0,Busy12:
                IF,NR(ReaderTL1)==1.AND.TypArLog13(2)==1
                  .AND.TagL==0,Busy13:
                IF,NQ(ReaderTR1Q).GE.JR2b,QSch12:
                IF,NQ(ReaderTL1Q).GE.JL2b,QSch13:
                ELSE,Fine1;
```

```
QSch12 SEARCH, ReaderTR1Q,JR2b: VehType==1
                         .AND.ArrTime.LE.TNOW-TravTime+R2b;
    BRANCH,1:
        IF,J.GT.0,J2b12:
        IF,NQ(ReaderTL1Q).GE.JL2b,QSch13:
        ELSE,Fine1;

J2b12  ASSIGN: JR2b=J+1: NEXT(QTk12);

QSch13 SEARCH, ReaderTL1Q,JL2b: VehType==1
                         .AND.ArrTime.LE.TNOW-TravTime+R2b;
    BRANCH,1:
        IF,J.GT.0,J2b13:
        ELSE,Fine1;

J2b13  ASSIGN: JL2b=J+1: NEXT(QTk13);

QTrk12 ASSIGN:
        TimeNeed=((Discharge(1)+((J-1)*Discharge(2)))+
                (Discharge(TypeLog12(2))-(TNOW-TimeLog12)))-
                ChkTime: NEXT(SubZ1);

QTrk13 ASSIGN: TimeNeed=((Discharge(1)+((J-1)*Discharge(2)))+
                (Discharge(TypeLog13(2))-(TNOW-TimeLog13)))-
                ChkTime: NEXT(SubZ1);

Busy12 BRANCH,1:
        IF,DurationT12.GT.((ArrLog12(2)+TravTime)-TNOW),Bsy12:
        ELSE,Bs12;

Bs12   ASSIGN:
        TagR=1:
        TimeNeed=((ArrLog12(2)+TravTime)-TNOW)-
                ChkTime: NEXT(SubZ1);

Bsy12  ASSIGN:
        TagR=1:
        TimeNeed=DurationT12+Discharge(1)-ChkTime: NEXT(SubZ1);

Busy13 BRANCH,1:
        IF,DurationT13.GT.((ArrLog13(2)+TravTime)-TNOW),Bsy13:
```

```
            ELSE,Bs13;

Bs13  ASSIGN:
            TagL=1:
            TimeNeed=((ArrLog13(2)+TravTime)-TNOW)-
                  ChkTime: NEXT(SubZ1);

Bsy13 ASSIGN:
            TagL=1:
            TimeNeed=DurationT13+Discharge(1)-ChkTime: NEXT(SubZ1);

QTk12 BRANCH,1:
            IF,(DurationT12+((J-1)*Discharge(2))).GT.
              ((ArrLog12(J+2)+TravTime)-TNOW),Qk12:
            ELSE,Qkk12;

Qk12  ASSIGN:
            TimeNeed=(DurationT12+((J-1)*Discharge(2)))+
                  Discharge(1)-ChkTime: NEXT(SubZ1);

Qkk12 ASSIGN:
            TimeNeed=((ArrLog12(J+2)+TravTime)-TNOW)-
                  ChkTime: NEXT(SubZ1);

QTk13 BRANCH,1:
            IF,(DurationT13+((J-1)*Discharge(2))).GT.
              ((ArrLog13(J+2)+TravTime)-TNOW),Qk13:
            ELSE,Qkk13;

Qk13  ASSIGN:
            TimeNeed=(DurationT13+((J-1)*Discharge(2)))+
                  Discharge(1)-ChkTime: NEXT(SubZ1);

Qkk13 ASSIGN:
            TimeNeed=((ArrLog13(J+2)+TravTime)-TNOW)-
                  ChkTime: NEXT(SubZ1);

SubZ1 BRANCH,1:
            IF,TimeNeed.LT.0,More1:
            IF,TimeNeed.GE.MaxGX,Fine1:
            ELSE,Chos1;
```

```
Chos1 BRANCH,1:
        IF,DurationT23.LT.DurationT22,DT22:
        ELSE, DT23;

DT22  ASSIGN:
        Prd=DurationT22:
        QL=NQ(PhaseTR2Q):
        ExtraTime=Green2-DurationT22: NEXT(Extr1);

DT23  ASSIGN:
        Prd=DurationT23:
        QL=NQ(PhaseTL2Q):
        ExtraTime=Green2-DurationT23;

;Calculating the Saturation Flow Rate of Next Competing Phase - Direct Observation Method

Extr1 BRANCH,1:
        IF,Prd.LE.(Green2-ChkTime-TimeNeed),Dur1:
        ELSE,Grn1;

Grn1  ASSIGN: Prd=Green2-ChkTime-TimeNeed;

Dur1  ASSIGN: Satur=(QL*(3600/(CycleLength-(Green2-TimeNeed))))/
                    ((3600/(Prd/QL))*((Green2-TimeNeed)/CycleLength));

Extr1 BRANCH,1:
        IF,Satur.GT.MaxSat,Fine1:
        IF,ExtraTime.GT.TimeNeed,Extd1:
        ELSE,Prior1;

;Calculating the Average Stopped Delay Observed in the Next Competing Phase
;       Total Delay Up to Now=(T-t1)+(T-t2)+...+(T-tn)
;                            =nT-(t1+t2+...+tn)
;AvgDel()=(Total Delay Up to Now)/n.
;where
;T = current simulation time,
;t's = arrival time of vehicles joining the next competing queue, and
;n = number of observed vehicles.

Prior1 ASSIGN:
        Vol(3)=NC(ArrTrk2)-NC(DepTrk2):
        Vol(4)=(NC(ArrAll2)-NC(DepAll2))-Vol(3);
```

```
        BRANCH,1:
                IF,Vol(3)==0,Cg1:
                ELSE,Dn1;

Cg1     ASSIGN: Vol(3)=.0000001;

Dn1     ASSIGN:
                AvgDel(3)=((Vol(3)*TNOW)-TimeInT(3))/Vol(3):
                AvgDel(4)=((Vol(4)*TNOW)-TimeInT(4))/Vol(4):

;Delay Index

                DI=(AvgDel(3)*W(1))+(AvgDel(4)*W(2));

        BRANCH,1:
                IF,DI.LT.DIValue,Extd1:
                ELSE,Fine1;

Extd1   WRITE, TypeFile1: K,TNOW,TimeNeed,ExtraTime,DI;

      BRANCH,1:
                IF,TimeNeed.GT.TimeNeedT,Loop1:
                ELSE,Lop1;

Loop1   ASSIGN:
                TimeNeedT=TimeNeed+.001:
                K=K+1: NEXT(More1);

Lop1    ASSIGN:
                K=K+1: NEXT(More1);

Fine1   DELAY: ChkTime+TimeNeedT;

        ASSIGN:
                DI=0:
                JR2a=1:
                JL2a=1:
                JR2b=1:
                JL2b=1:
                TagR=0:
                TagL=0;
```

```
        QUEUE, Dum2Q;
        PREEMPT: PhaseTR1;                  North, T_R Red--All Red Condition
        PREEMPT: PhaseTL1;                  North, T_L Red--All Red Condition

        DELAY: LostTime/4;

        RELEASE: PhaseTR2: PhaseTL2;        East Green, all the rest Red
        DELAY: Green2-ChkTime-TimeNeedT;

        ASSIGN:
               NewG1=NewG1+TimeNeedT:
               NewG2=Green2-TimeNeedT:
               TimeNeedT=0;

        BRANCH,1:
               IF,PARTII==2.AND.Option==0,PII2:
               ELSE,Fine2;

PII2    ASSIGN: K=1;

More2   BRANCH,1:
               IF,NQ(PhaseTR1Q).GT.440.OR.NQ(PhaseTL1Q).GT.440.OR.
               NQ(PhaseTR3Q).GT.440.OR.NQ(PhaseTL3Q).GT.440.OR.
               NQ(PhaseTR4Q).GT.440.OR.NQ(PhaseTL4Q).GT.440,Fine2:
               IF,NQ(PhaseTR2Q).GE.JR2a,QSrch22:
               IF,NQ(PhaseTL2Q).GE.JL2a,QSrch23:
               ELSE,Cond2;

QSrch22 SEARCH, PhaseTR2Q,JR2a: VehType==3;
        BRANCH,1:
               IF,J.GT.0,J2a22:
               IF,NQ(PhaseTL2Q).GE.JL2a,QSrch23:
               ELSE,Cond2;

J2a22   ASSIGN: JR2a=J+1: NEXT(QTrk22);

QSrch23 SEARCH, PhaseTL2Q,JL2a: VehType==3;
        BRANCH,1:
               IF,J.GT.0,J2a23:
               ELSE,Cond2;
```

```
J2a23  ASSIGN: JL2a=J+1: NEXT(QTrk23);

Cond2 BRANCH,1:
            IF,NR(ReaderTR2)==1.AND.TypArLog22(2)==3
              .AND.TagR==0,Busy22:
            IF,NR(ReaderTL2)==1.AND.TypArLog23(2)==3
              .AND.TagL==0,Busy23:
            IF,NQ(ReaderTR2Q).GE.JR2b,QSch22:
            IF,NQ(ReaderTL2Q).GE.JL2b,QSch23:
            ELSE,Fine2;

QSch22 SEARCH, ReaderTR2Q,JR2b: VehType==3
                         .AND.ArrTime.LE.TNOW-TravTime+R2b;
       BRANCH,1:
            IF,J.GT.0,J2b22:
            IF,NQ(ReaderTL2Q).GE.JL2b,QSch23:
            ELSE,Fine2;

J2b22  ASSIGN: JR2b=J+1: NEXT(QTk22);

QSch23 SEARCH, ReaderTL2Q,JL2b: VehType==3
                         .AND.ArrTime.LE.TNOW-TravTime+R2b;
       BRANCH,1:
            IF,J.GT.0,J2b23:
            ELSE,Fine2;

J2b23  ASSIGN: JL2b=J+1: NEXT(QTk23);

QTrk22 ASSIGN:
            TimeNeed=((Discharge(3)+((J-1)*Discharge(4)))+
                   (Discharge(TypeLog22(2))-(TNOW-TimeLog22)))-
                   ChkTime: NEXT(SubZ2);

QTrk23 ASSIGN: TimeNeed=((Discharge(3)+((J-1)*Discharge(4)))+
                   (Discharge(TypeLog23(2))-(TNOW-TimeLog23)))-
                   ChkTime: NEXT(SubZ2);

Busy22 BRANCH,1:
            IF,DurationT22.GT.((ArrLog22(2)+TravTime)-TNOW),Bsy22:
            ELSE,Bs22;

Bs22   ASSIGN:
```

```
          TagR=1:
          TimeNeed=((ArrLog22(2)+TravTime)-TNOW)-
                 ChkTime: NEXT(SubZ2);

Bsy22  ASSIGN:
          TagR=1:
          TimeNeed=DurationT22+Discharge(3)-ChkTime: NEXT(SubZ2);

Busy23 BRANCH,1:
          IF,DurationT23.GT.((ArrLog23(2)+TravTime)-TNOW),Bsy23:
          ELSE,Bs23;

Bs23   ASSIGN:
          TagL=1:
          TimeNeed=((ArrLog23(2)+TravTime)-TNOW)-
                 ChkTime: NEXT(SubZ2);

Bsy23  ASSIGN:
          TagL=1:
          TimeNeed=DurationT23+Discharge(3)-ChkTime: NEXT(SubZ2);

QTk22 BRANCH,1:
          IF,(DurationT22+((J-1)*Discharge(4))).GT.
            ((ArrLog22(J+2)+TravTime)-TNOW),Qk22:
          ELSE,Qkk22;

Qk22   ASSIGN:
          TimeNeed=(DurationT22+((J-1)*Discharge(4)))+
                 Discharge(3)-ChkTime: NEXT(SubZ2);

Qkk22 ASSIGN:
          TimeNeed=((ArrLog22(J+2)+TravTime)-TNOW)-
                 ChkTime: NEXT(SubZ2);

QTk23 BRANCH,1:
          IF,(DurationT23+((J-1)*Discharge(4))).GT.
            ((ArrLog23(J+2)+TravTime)-TNOW),Qk23:
          ELSE,Qkk23;

Qk23   ASSIGN:
          TimeNeed=(DurationT23+((J-1)*Discharge(4)))+
                 Discharge(3)-ChkTime: NEXT(SubZ2);
```

```
Qkk23 ASSIGN:
            TimeNeed=((ArrLog23(J+2)+TravTime)-TNOW)-
                 ChkTime: NEXT(SubZ2);

SubZ2 BRANCH,1:
            IF,TimeNeed.LT.0,More2:
            IF,TimeNeed.GE.MaxGX,Fine2:
            ELSE,Chos2;

Chos2 BRANCH,1:
            IF,DurationT33.LT.DurationT32,DT32:
            ELSE,DT33;

DT32  ASSIGN:
            Prd=DurationT32:
            QL=NQ(PhaseTR3Q):
            ExtraTime=Green3-DurationT32
            :NEXT(Extr2);

DT33  ASSIGN:
            Prd=DurationT33:
            QL=NQ(PhaseTL3Q):
            ExtraTime=Green3-DurationT33;

Extr2  BRANCH,1:
            IF,Prd.LE.(Green3-ChkTime-TimeNeed),Dur2:
            ELSE,Grn2;

Grn2   ASSIGN: Prd=Green3-ChkTime-TimeNeed;

Dur2   ASSIGN: Satur=(QL*(3600/(CycleLength-(Green3-TimeNeed))))/
            ((3600/(Prd/QL))*((Green3-TimeNeed)/CycleLength));

Extr2  BRANCH,1:
            IF,Satur.GT.MaxSat,Fine2:
            IF,ExtraTime.GT.TimeNeed,Extd2:
            ELSE,Prior2;

Prior2 ASSIGN:
            Vol(5)=NC(ArrTrk3)-NC(DepTrk3):
            Vol(6)=(NC(ArrAll3)-NC(DepAll3))-Vol(5);
```

```
        BRANCH,1:
                IF,Vol(5)==0,Cg2:
                ELSE,Dn2;

Cg2     ASSIGN: Vol(5)=.0000001;

Dn2     ASSIGN:
                AvgDel(5)=((Vol(5)*TNOW)-TimeInT(5))/Vol(5):
                AvgDel(6)=((Vol(6)*TNOW)-TimeInT(6))/Vol(6):
                DI=(AvgDel(5)*W(1))+(AvgDel(6)*W(2));

        BRANCH,1:
                IF,DI.LT.DIValue,Extd2:
                ELSE,Fine2;

Extd2   WRITE, TypeFile2: K,TNOW,TimeNeed,ExtraTime,DI;

        BRANCH,1:
                IF,TimeNeed.GT.TimeNeedT,Loop2:
                ELSE,Lop2;

Loop2   ASSIGN:
                TimeNeedT=TimeNeed+.001:
                K=K+1: NEXT(More2);

Lop2    ASSIGN: K=K+1: NEXT(More2);

Fine2   DELAY: ChkTime+TimeNeedT;

        ASSIGN:
                DI=0:
                JR2a=1:
                JL2a=1:
                JR2b=1:
                JL2b=1:
                TagR=0:
                TagL=0;

        QUEUE, Dum3Q;
        PREEMPT: PhaseTR2;      East, T_R Red--All Red Condition
        PREEMPT: PhaseTL2;      East, T_L Red--All Red Condition
```

```
        DELAY: LostTime/4;

        RELEASE: PhaseTR3: PhaseTL3;          South Green, all the rest Red
        DELAY: Green3-ChkTime-TimeNeedT;

        ASSIGN:
                NewG2=NewG2+TimeNeedT:
                NewG3=Green3-TimeNeedT:
                TimeNeedT=0;

        BRANCH,1:
                IF,PARTII==2.AND.Option==0,PII3:
                ELSE,Fine3;

PII3    ASSIGN: K=1;

More3   BRANCH,1:
                IF,NQ(PhaseTR1Q).GT.440.OR.NQ(PhaseTL1Q).GT.440.OR.
                NQ(PhaseTR2Q).GT.440.OR.NQ(PhaseTL2Q).GT.440.OR.
                NQ(PhaseTR4Q).GT.440.OR.NQ(PhaseTL4Q).GT.440,Fine3:
                IF,NQ(PhaseTR3Q).GE.JR2a,QSrch32:
                IF,NQ(PhaseTL3Q).GE.JL2a,QSrch33:
                ELSE,Cond3;

QSrch32 SEARCH, PhaseTR3Q,JR2a: VehType==5;
        BRANCH,1:
                IF,J.GT.0,J2a32:
                IF,NQ(PhaseTL3Q).GE.JL2a,QSrch33:
                ELSE,Cond3;

J2a32   ASSIGN: JR2a=J+1: NEXT(QTrk32);

QSrch33 SEARCH, PhaseTL3Q,JL2a: VehType==5;
        BRANCH,1:
                IF,J.GT.0,J2a33:
                ELSE,Cond3;

J2a33   ASSIGN: JL2a=J+1: NEXT(QTrk33);

Cond3   BRANCH,1:
                IF,NR(ReaderTR3)==1.AND.TypArLog32(2)==5
```

```
                .AND.TagR==0,Busy32:
              IF,NR(ReaderTL3)==1.AND.TypArLog33(2)==5
                .AND.TagL==0,Busy33:
              IF,NQ(ReaderTR3Q).GE.JR2b,QSch32:
              IF,NQ(ReaderTL3Q).GE.JL2b,QSch33:
              ELSE,Fine3;

QSch32 SEARCH, ReaderTR3Q,JR2b: VehType==5
                              .AND.ArrTime.LE.TNOW-TravTime+R2b;
       BRANCH,1:
              IF,J.GT.0,J2b32:
              IF,NQ(ReaderTL3Q).GE.JL2b,QSch33:
              ELSE,Fine3;

J2b32  ASSIGN: JR2b=J+1: NEXT(QTk32);

QSch33 SEARCH, ReaderTL3Q,JL2b: VehType==5
                              .AND.ArrTime.LE.TNOW-TravTime+R2b;

       BRANCH,1:
              IF,J.GT.0,J2b33:
              ELSE,Fine3;

J2b33  ASSIGN: JL2b=J+1: NEXT(QTk33);

QTrk32 ASSIGN:
              TimeNeed=((Discharge(5)+((J-1)*Discharge(6)))+
                     (Discharge(TypeLog32(2))-(TNOW-TimeLog32)))-
                     ChkTime: NEXT(SubZ3);

QTrk33 ASSIGN: TimeNeed=((Discharge(5)+((J-1)*Discharge(6)))+
                     (Discharge(TypeLog33(2))-(TNOW-TimeLog33)))-
                     ChkTime: NEXT(SubZ3);

Busy32 BRANCH,1:
              IF,DurationT32.GT.((ArrLog32(2)+TravTime)-TNOW),Bsy32:
              ELSE,Bs32;

Bs32   ASSIGN:
              TagR=1:
              TimeNeed=((ArrLog32(2)+TravTime)-TNOW)-
                      ChkTime: NEXT(SubZ3);
```

```
Bsy32  ASSIGN:
          TagR=1:
          TimeNeed=DurationT32+Discharge(5)-ChkTime: NEXT(SubZ3);

Busy33 BRANCH,1:
          IF,DurationT33.GT.((ArrLog33(2)+TravTime)-TNOW),Bsy33:
          ELSE,Bs33;

Bs33   ASSIGN:
          TagL=1:
          TimeNeed=((ArrLog33(2)+TravTime)-TNOW)-
                ChkTime: NEXT(SubZ3);

Bsy33  ASSIGN:
          TagL=1:
          TimeNeed=DurationT33+Discharge(5)-ChkTime: NEXT(SubZ3);

QTk32 BRANCH,1:
          IF,(DurationT32+((J-1)*Discharge(6))).GT.
           ((ArrLog32(J+2)+TravTime)-TNOW),Qk32:
          ELSE,Qkk32;

Qk32   ASSIGN:
          TimeNeed=(DurationT32+((J-1)*Discharge(6)))+
                Discharge(5)-ChkTime: NEXT(SubZ3);

Qkk32 ASSIGN:
          TimeNeed=((ArrLog32(J+2)+TravTime)-TNOW)-
                ChkTime: NEXT(SubZ3);

QTk33 BRANCH,1:
          IF,(DurationT33+((J-1)*Discharge(6))).GT.
           ((ArrLog33(J+2)+TravTime)-TNOW),Qk33:
          ELSE,Qkk33;

Qk33   ASSIGN:
          TimeNeed=(DurationT33+((J-1)*Discharge(6)))+
                Discharge(5)-ChkTime: NEXT(SubZ3);

Qkk33 ASSIGN:
          TimeNeed=((ArrLog33(J+2)+TravTime)-TNOW)-
```

```
                ChkTime: NEXT(SubZ3);

SubZ3 BRANCH,1:
            IF,TimeNeed.LT.0,More3:
            IF,TimeNeed.GE.MaxGX,Fine3:
            ELSE,Chos3;

Chos3 BRANCH,1:
            IF,DurationT43.LT.DurationT42,DT42:
            ELSE, DT43;

DT42  ASSIGN:
            Prd=DurationT42:
            QL=NQ(PhaseTR4Q):
            ExtraTime=Green4-DurationT42
            :NEXT(Extr3);

DT43   ASSIGN:
             Prd=DurationT43:
            QL=NQ(PhaseTL4Q):
            ExtraTime=Green4-DurationT43;

Extr3  BRANCH,1:
            IF,Prd.LE.(Green4-ChkTime-TimeNeed),Dur3:
            ELSE,Grn3;

Grn3   ASSIGN: Prd=Green4-ChkTime-TimeNeed;

Dur3   ASSIGN: Satur=(QL*(3600/(CycleLength-(Green4-TimeNeed))))/
            ((3600/(Prd/QL))*((Green4-TimeNeed)/CycleLength));

Extr3  BRANCH,1:
            IF,Satur.GT.MaxSat,Fine3:
            IF,ExtraTime.GT.TimeNeed,Extd3:
            ELSE,Prior3;

Prior3 ASSIGN:
            Vol(7)=NC(ArrTrk4)-NC(DepTrk4):
            Vol(8)=(NC(ArrAll4)-NC(DepAll4))-Vol(7);

       BRANCH,1:
            IF,Vol(7)==0,Cg3:
```

```
          ELSE,Dn3;

Cg3    ASSIGN: Vol(7)=.0000001;

Dn3    ASSIGN:
          AvgDel(7)=((Vol(7)*TNOW)-TimeInT(7))/Vol(7):
          AvgDel(8)=((Vol(8)*TNOW)-TimeInT(8))/Vol(8):
          DI=(AvgDel(7)*W(1))+(AvgDel(8)*W(2));

       BRANCH,1:
          IF,DI.LT.DIValue,Extd3:
          ELSE,Fine3;

Extd3  WRITE, TypeFile3: K,TNOW,TimeNeed,ExtraTime,DI;

       BRANCH,1:
          IF,TimeNeed.GT.TimeNeedT,Loop3:
          ELSE,Lop3;

Loop3  ASSIGN:
          TimeNeedT=TimeNeed+.001:
          K=K+1: NEXT(More3);

Lop3   ASSIGN: K=K+1: NEXT(More3);

Fine3  DELAY: ChkTime+TimeNeedT;

       ASSIGN:
          DI=0:
          JR2a=1:
          JL2a=1:
          JR2b=1:
          JL2b=1:
          TagR=0:
          TagL=0;

       QUEUE, Dum4Q;
       PREEMPT: PhaseTR3;      South, T_R Red--All Red Condition
       PREEMPT: PhaseTL3;      South, T_L Red--All Red Condition

       DELAY: LostTime/4;
```

```
        RELEASE: PhaseTR4: PhaseTL4;            West Green, all the rest Red
        DELAY: Green4-ChkTime-TimeNeedT;

        ASSIGN:
          NewG3=NewG3+TimeNeedT:
          NewG4=Green4-TimeNeedT:
          TimeNeedT=0;

        BRANCH,1:
              IF,PARTII==2.AND.Option==0.AND.CPh.LT.Multi,PII4:
              ELSE,Fine4;

PII4   ASSIGN: K=1;

More4  BRANCH,1:
              IF,NQ(PhaseTR1Q).GT.440.OR.NQ(PhaseTL1Q).GT.440.OR.
                    NQ(PhaseTR2Q).GT.440.OR.NQ(PhaseTL2Q).GT.440.OR.
                    NQ(PhaseTR3Q).GT.440.OR.NQ(PhaseTL3Q).GT.440,Fine4:
              IF,NQ(PhaseTR4Q).GE.JR2a,QSrch42:
              IF,NQ(PhaseTL4Q).GE.JL2a,QSrch43:
              ELSE,Cond4;

QSrch42 SEARCH, PhaseTR4Q,JR2a: VehType==7;

        BRANCH,1:
              IF,J.GT.0,J2a42:
              IF,NQ(PhaseTL4Q).GE.JL2a,QSrch43:
              ELSE,Cond4;

J2a42  ASSIGN: JR2a=J+1: NEXT(QTrk42);

QSrch43 SEARCH, PhaseTL4Q,JL2a: VehType==7;
        BRANCH,1:
              IF,J.GT.0,J2a43:
              ELSE,Cond4;

J2a43  ASSIGN: JL2a=J+1: NEXT(QTrk43);

Cond4 BRANCH,1:
              IF,NR(ReaderTR4)==1.AND.TypArLog42(2)==7
                .AND.TagR==0,Busy42:
              IF,NR(ReaderTL4)==1.AND.TypArLog43(2)==7
```

```
                .AND.TagL==0,Busy43:
              IF,NQ(ReaderTR4Q).GE.JR2b,QSch42:
              IF,NQ(ReaderTL4Q).GE.JL2b,QSch43:
              ELSE,Fine4;

QSch42 SEARCH, ReaderTR4Q,JR2b: VehType==7
                          .AND.ArrTime.LE.TNOW-TravTime+R2b;
       BRANCH,1:
              IF,J.GT.0,J2b42:
              IF,NQ(ReaderTL4Q).GE.JL2b,QSch43:
              ELSE,Fine4;

J2b42  ASSIGN: JR2b=J+1: NEXT(QTk42);

QSch43 SEARCH, ReaderTL4Q,JL2b: VehType==7
                          .AND.ArrTime.LE.TNOW-TravTime+R2b;
       BRANCH,1:
              IF,J.GT.0,J2b43:
              ELSE,Fine4;

J2b43  ASSIGN: JL2b=J+1: NEXT(QTk43);

QTrk42 ASSIGN:
              TimeNeed=((Discharge(7)+((J-1)*Discharge(8)))+
                    (Discharge(TypeLog42(2))-(TNOW-TimeLog42)))-
                    ChkTime: NEXT(SubZ4);

QTrk43 ASSIGN: TimeNeed=((Discharge(7)+((J-1)*Discharge(8)))+
                    (Discharge(TypeLog43(2))-(TNOW-TimeLog43)))-
                    ChkTime: NEXT(SubZ4);

Busy42 BRANCH,1:
              IF,DurationT42.GT.((ArrLog42(2)+TravTime)-TNOW),Bsy42:
              ELSE,Bs42;

Bs42   ASSIGN:
              TagR=1:
              TimeNeed=((ArrLog42(2)+TravTime)-TNOW)-
                     ChkTime: NEXT(SubZ4);

Bsy42  ASSIGN:
              TagR=1:
```

```
        TimeNeed=DurationT42+Discharge(7)-ChkTime: NEXT(SubZ4);

Busy43 BRANCH,1:
        IF,DurationT43.GT.((ArrLog43(2)+TravTime)-TNOW),Bsy43:
        ELSE,Bs43;

Bs43   ASSIGN:
        TagL=1:
        TimeNeed=((ArrLog43(2)+TravTime)-TNOW)-
                ChkTime: NEXT(SubZ4);

Bsy43  ASSIGN:
        TagL=1:
        TimeNeed=DurationT43+Discharge(7)-ChkTime: NEXT(SubZ4);

QTk42  BRANCH,1:
        IF,(DurationT42+((J-1)*Discharge(8))).GT.
          ((ArrLog42(J+2)+TravTime)-TNOW),Qk42:
        ELSE,Qkk42;

Qk42   ASSIGN:
        TimeNeed=(DurationT42+((J-1)*Discharge(8)))+
               Discharge(7)-ChkTime: NEXT(SubZ4);

Qkk42 ASSIGN:
        TimeNeed=((ArrLog42(J+2)+TravTime)-TNOW)-
               ChkTime: NEXT(SubZ4);

QTk43 BRANCH,1:
        IF,(DurationT43+((J-1)*Discharge(8))).GT.
          ((ArrLog43(J+2)+TravTime)-TNOW),Qk43:
        ELSE,Qkk43;

Qk43   ASSIGN:
        TimeNeed=(DurationT43+((J-1)*Discharge(8)))+
               Discharge(7)-ChkTime: NEXT(SubZ4);

Qkk43 ASSIGN:
        TimeNeed=((ArrLog43(J+2)+TravTime)-TNOW)-
               ChkTime: NEXT(SubZ4);

SubZ4 BRANCH,1:
```

```
              IF,TimeNeed.LT.0,More4:
              IF,TimeNeed.GE.MaxGX,Fine4:
              ELSE,Chos4;

Chos4 BRANCH,1:
              IF,DurationT13.LT.DurationT12,DT12:
              ELSE, DT13;

DT12  ASSIGN:
               Prd=DurationT12:
              QL=NQ(PhaseTR1Q):
              ExtraTime=Green1-DurationT12
              :NEXT(Extr4);

DT13  ASSIGN:
               Prd=DurationT13:
              QL=NQ(PhaseTL1Q):
              ExtraTime=Green1-DurationT13;

Extr4  BRANCH,1:
              IF,Prd.LE.(Green1-ChkTime-TimeNeed),Dur4:
              ELSE,Grn4;

Grn4   ASSIGN: Prd=Green1-ChkTime-TimeNeed;

Dur4   ASSIGN: Satur=(QL*(3600/(CycleLength-(Green1-TimeNeed))))/
              ((3600/(Prd/QL))*((Green1-TimeNeed)/CycleLength));

Extr4  BRANCH,1:
              IF,Satur.GT.MaxSat,Fine4:
              IF,ExtraTime.GT.TimeNeed,Extd4:
              ELSE,Prior4;

Prior4 ASSIGN:
              Vol(1)=NC(ArrTrk1)-NC(DepTrk1):
              Vol(2)=(NC(ArrAll1)-NC(DepAll1))-Vol(1);

       BRANCH,1:
              IF,Vol(1)==0,Cg4:
              ELSE,Dn4;

Cg4    ASSIGN: Vol(1)=.0000001;
```

```
Dn4    ASSIGN:
              AvgDel(1)=((Vol(1)*TNOW)-TimeInT(1))/Vol(1):
              AvgDel(2)=((Vol(2)*TNOW)-TimeInT(2))/Vol(2):
              DI=(AvgDel(1)*W(1))+(AvgDel(2)*W(2));

       BRANCH,1:
              IF,DI.LT.DIValue,Extd4:
              ELSE,Fine4;

Extd4  WRITE, TypeFile4: K,TNOW,TimeNeed,ExtraTime,DI;

       BRANCH,1:
              IF,TimeNeed.GT.TimeNeedT,Loop4:
              ELSE,Lop4;

Loop4  ASSIGN:
              TimeNeedT=TimeNeed+.001:
              K=K+1: NEXT(More4);

Lop4   ASSIGN: K=K+1: NEXT(More4);

Fine4  DELAY: ChkTime+TimeNeedT;

       ASSIGN:
              DI=0:
              JR2a=1:
              JL2a=1:
              JR2b=1:
              JL2b=1:
              TagR=0:
              TagL=0;

       QUEUE, Dum5Q;
       PREEMPT: PhaseTR4;                South, T_R Red--All Red Condition
       PREEMPT: PhaseTL4;                South, T_L Red--All Red Condition

       DELAY: LostTime/4;

       RELEASE: PhaseTR1: PhaseTL1;
       RELEASE: PhaseTR2: PhaseTL2;
       RELEASE: PhaseTR3: PhaseTL3;
```

```
        RELEASE: PhaseTR4: PhaseTL4;

        ASSIGN:
                NewG4=NewG4+TimeNeedT:
                NewCycle=NewG1+NewG2+NewG3+NewG4+LostTime;

        ASSIGN:
                AvgDel(1)=0:
                AvgDel(2)=0:
                AvgDel(3)=0:
                AvgDel(4)=0:
                AvgDel(5)=0:
                AvgDel(6)=0:
                AvgDel(7)=0:
                AvgDel(8)=0;

        ASSIGN:
                CPh=CPh+1:
                ChangeTime=TNOW-StartTime;

        BRANCH,1:
                IF,ChangeTime.GE.Period2,Print:
                ELSE,Cycle;

;Calculating Average Stopped Delay at the End of Each Period

Print   ASSIGN:
                Vol(1)=NC(DepTrk1)-OVol(1):
                Vol(2)=(NC(DepAll1)-NC(DepTrk1))-OVol(2):
                Vol(3)=NC(DepTrk2)-OVol(3):
                Vol(4)=(NC(DepAll2)-NC(DepTrk2))-OVol(4):
                Vol(5)=NC(DepTrk3)-OVol(5):
                Vol(6)=(NC(DepAll3)-NC(DepTrk3))-OVol(6):
                Vol(7)=NC(DepTrk4)-OVol(7):
                Vol(8)=(NC(DepAll4)-NC(DepTrk4))-OVol(8):

                OVol(1)=NC(DepTrk1):
                OVol(2)=NC(DepAll1)-NC(DepTrk1):
                OVol(3)=NC(DepTrk2):
                OVol(4)=NC(DepAll2)-NC(DepTrk2):
                OVol(5)=NC(DepTrk3):
                OVol(6)=NC(DepAll3)-NC(DepTrk3):
```

```
OVol(7)=NC(DepTrk4):
OVol(8)=NC(DepAll4)-NC(DepTrk4):

LapVol(1)=NC(ArrTrk1)-NC(DepTrk1):
LapVol(2)=(NC(ArrAll1)-NC(DepAll1))-LapVol(1):
LapVol(3)=NC(ArrTrk2)-NC(DepTrk2):
LapVol(4)=(NC(ArrAll2)-NC(DepAll2))-LapVol(3):
LapVol(5)=NC(ArrTrk3)-NC(DepTrk3):
LapVol(6)=(NC(ArrAll3)-NC(DepAll3))-LapVol(5):
LapVol(7)=NC(ArrTrk4)-NC(DepTrk4):
LapVol(8)=(NC(ArrAll4)-NC(DepAll4))-LapVol(7):

LapDel(1)=(LapVol(1)*TNOW)-TimeInT(1):
LapDel(2)=(LapVol(2)*TNOW)-TimeInT(2):
LapDel(3)=(LapVol(3)*TNOW)-TimeInT(3):
LapDel(4)=(LapVol(4)*TNOW)-TimeInT(4):
LapDel(5)=(LapVol(5)*TNOW)-TimeInT(5):
LapDel(6)=(LapVol(6)*TNOW)-TimeInT(6):
LapDel(7)=(LapVol(7)*TNOW)-TimeInT(7):
LapDel(8)=(LapVol(8)*TNOW)-TimeInT(8):

Vol(1)=Vol(1)+LapVol(1):
Vol(2)=Vol(2)+LapVol(2):
Vol(3)=Vol(3)+LapVol(3):
Vol(4)=Vol(4)+LapVol(4):
Vol(5)=Vol(5)+LapVol(5):
Vol(6)=Vol(6)+LapVol(6):
Vol(7)=Vol(7)+LapVol(7):
Vol(8)=Vol(8)+LapVol(8):

;Calculating Total Stopped Delay at the End of Simulation Run Time

AvgStopT(1)=AvgStopT(1)+AvgStop(1):
AvgStopT(2)=AvgStopT(2)+AvgStop(2):
AvgStopT(3)=AvgStopT(3)+AvgStop(3):
AvgStopT(4)=AvgStopT(4)+AvgStop(4):
AvgStopT(5)=AvgStopT(5)+AvgStop(5):
AvgStopT(6)=AvgStopT(6)+AvgStop(6):
AvgStopT(7)=AvgStopT(7)+AvgStop(7):
AvgStopT(8)=AvgStopT(8)+AvgStop(8);

BRANCH,1:
```

```
            IF,Vol(1)==0,Chg1:
            IF,Vol(3)==0,Chg3:
            IF,Vol(5)==0,Chg5:
            IF,Vol(7)==0,Chg7:
            ELSE,Don;

Chg1  ASSIGN: Vol(1)=.0000001;

      BRANCH,1:
            IF,Vol(3)==0,Chg3:
            ELSE,Ck3;

Chg3  ASSIGN: Vol(3)=.0000001;

Ck3   BRANCH,1:
            IF,Vol(5)==0,Chg5:
            ELSE,Ck5;

Chg5  ASSIGN: Vol(5)=.0000001;

Ck5   BRANCH,1:
            IF,Vol(7)==0,Chg7:
            ELSE,Don;

Chg7  ASSIGN: Vol(7)=.0000001;

Don   ASSIGN:
            AvgStop(1)=(AvgStop(1)+LapDel(1)-OLapDel(1))/Vol(1):
            AvgStop(2)=(AvgStop(2)+LapDel(2)-OLapDel(2))/Vol(2):
            AvgStop(3)=(AvgStop(3)+LapDel(3)-OLapDel(3))/Vol(3):
            AvgStop(4)=(AvgStop(4)+LapDel(4)-OLapDel(4))/Vol(4):
            AvgStop(5)=(AvgStop(5)+LapDel(5)-OLapDel(5))/Vol(5):
            AvgStop(6)=(AvgStop(6)+LapDel(6)-OLapDel(6))/Vol(6):
            AvgStop(7)=(AvgStop(7)+LapDel(7)-OLapDel(7))/Vol(7):
            AvgStop(8)=(AvgStop(8)+LapDel(8)-OLapDel(8))/Vol(8):

            OLapDel(1)=LapDel(1):
            OLapDel(2)=LapDel(2):
            OLapDel(3)=LapDel(3):
            OLapDel(4)=LapDel(4):
            OLapDel(5)=LapDel(5):
            OLapDel(6)=LapDel(6):
```

```
        OLapDel(7)=LapDel(7):
        OLapDel(8)=LapDel(8):

        TrkDelay=((AvgStop(1)*Vol(1))+(AvgStop(3)*Vol(3))+
                   (AvgStop(5)*Vol(5))+(AvgStop(7)*Vol(7)))/
                  (Vol(1)+Vol(3)+Vol(5)+Vol(7)):
        CarDelay=((AvgStop(2)*Vol(2))+(AvgStop(4)*Vol(4))+
                  (AvgStop(6)*Vol(6))+(AvgStop(8)*Vol(8)))/
                 (Vol(2)+Vol(4)+Vol(6)+Vol(8)):
        IntDelay=((AvgStop(1)*Vol(1))+(AvgStop(2)*Vol(2))+
                 (AvgStop(3)*Vol(3))+(AvgStop(4)*Vol(4))+
                 (AvgStop(5)*Vol(5))+(AvgStop(6)*Vol(6))+
                 (AvgStop(7)*Vol(7))+(AvgStop(8)*Vol(8)))/
                 (Vol(1)+Vol(2)+Vol(3)+Vol(4)+Vol(5)+Vol(6)+Vol(7)+Vol(8)):

        AvgStop(1)=0:
        AvgStop(2)=0:
        AvgStop(3)=0:
        AvgStop(4)=0:
        AvgStop(5)=0:
        AvgStop(6)=0:
        AvgStop(7)=0:
        AvgStop(8)=0;

    Delay: Period3-ChangeTime;

    BRANCH,1:
        IF,Option==0,Wr:
        ELSE,Rd;

Rd  READ, F,FREE: ActFlow1,ActFlow2,ActFlow3,ActFlow4,
                  Truck%1,Truck%2,Truck%3,Truck%4;

Wr  WRITE, Output: TNOW,ActFlow1,ActFlow2,ActFlow3,ActFlow4,
                  Truck%1,Truck%2,Truck%3,Truck%4,
                  TrkDelay,CarDelay,IntDelay;

;Calculate the Overall Average Stopped Delay at the End of the Simulation Run

    BRANCH,1:
        IF,TNOW.GT.Period4,Print1:
        ELSE,Goo;
```

```
Print1  ASSIGN:
                Vol(1)=NC(DepTrk1):
                Vol(2)=NC(DepAll1)-NC(DepTrk1):
                Vol(3)=NC(DepTrk2):
                Vol(4)=NC(DepAll2)-NC(DepTrk2):
                Vol(5)=NC(DepTrk3):
                Vol(6)=NC(DepAll3)-NC(DepTrk3):
                Vol(7)=NC(DepTrk4):
                Vol(8)=NC(DepAll4)-NC(DepTrk4):

;Used in Validation of Results

;               TT(1)=(AvgStopT(1)+AvgStopT(2))/3600:
;               TT(2)=(AvgStopT(3)+AvgStopT(4))/3600:
;               TT(3)=(AvgStopT(5)+AvgStopT(6))/3600:
;               TT(4)=(AvgStopT(7)+AvgStopT(8))/3600:

                AvgStopT(1)=AvgStopT(1)/Vol(1):
                AvgStopT(2)=AvgStopT(2)/Vol(2):
                AvgStopT(3)=AvgStopT(3)/Vol(3):
                AvgStopT(4)=AvgStopT(4)/Vol(4):
                AvgStopT(5)=AvgStopT(5)/Vol(5):
                AvgStopT(6)=AvgStopT(6)/Vol(6):
                AvgStopT(7)=AvgStopT(7)/Vol(7):
                AvgStopT(8)=AvgStopT(8)/Vol(8):

;Used in Validation of Results

;               TT(1)=((AvgStopT(1)*Vol(1))+(AvgStopT(2)*Vol(2)))/(Vol(1)+Vol(2)):
;               TT(2)=((AvgStopT(3)*Vol(3))+(AvgStopT(4)*Vol(4)))/(Vol(3)+Vol(4)):
;               TT(3)=((AvgStopT(5)*Vol(5))+(AvgStopT(6)*Vol(6)))/(Vol(5)+Vol(6)):
;               TT(4)=((AvgStopT(7)*Vol(7))+(AvgStopT(8)*Vol(8)))/(Vol(7)+Vol(8)):

                TrkDelay=(((AvgStopT(1)*Vol(1))+(LapVol(1)*LapDel(1)))+
                         ((AvgStopT(3)*Vol(3))+(LapVol(3)*LapDel(3)))+
                         ((AvgStopT(5)*Vol(5))+(LapVol(5)*LapDel(5)))+
                         ((AvgStopT(7)*Vol(7))+(LapVol(7)*LapDel(7))))/
                         (Vol(1)+Vol(3)+LapVol(1)+LapVol(3)+
                          Vol(5)+Vol(7)+LapVol(5)+LapVol(7)):
                CarDelay=(((AvgStopT(2)*Vol(2))+(LapVol(2)*LapDel(2)))+
                         ((AvgStopT(4)*Vol(4))+(LapVol(4)*LapDel(4)))+
```

```
                    ((AvgStopT(6)*Vol(6))+(LapVol(6)*LapDel(6)))+
                    ((AvgStopT(8)*Vol(8))+(LapVol(8)*LapDel(8))))/
                    (Vol(2)+Vol(4)+LapVol(2)+LapVol(4)+
                     Vol(6)+Vol(8)+LapVol(6)+LapVol(8)):
        IntDelay=(((AvgStopT(1)*Vol(1))+(LapVol(1)*LapDel(1)))+
                    ((AvgStopT(2)*Vol(2))+(LapVol(2)*LapDel(2)))+
                    ((AvgStopT(3)*Vol(3))+(LapVol(3)*LapDel(3)))+
                    ((AvgStopT(4)*Vol(4))+(LapVol(4)*LapDel(4)))+
                    ((AvgStopT(5)*Vol(5))+(LapVol(5)*LapDel(5)))+
                    ((AvgStopT(6)*Vol(6))+(LapVol(6)*LapDel(6)))+
                    ((AvgStopT(7)*Vol(7))+(LapVol(7)*LapDel(7)))+
                    ((AvgStopT(8)*Vol(8))+(LapVol(8)*LapDel(8))))/
                    (Vol(1)+Vol(3)+LapVol(1)+LapVol(3)+
                     Vol(5)+Vol(7)+LapVol(5)+LapVol(7)+
                     Vol(2)+Vol(4)+LapVol(2)+LapVol(4)+
                     Vol(6)+Vol(8)+LapVol(6)+LapVol(8));

        WRITE, Output: TrkDelay,CarDelay,IntDelay;
;       WRITE, Output: TT(1),TT(2),TT(3),TT(4);
;       WRITE, Output: Delay1,Delay2,Delay3,Delay4,WebDelay;

Goo     ASSIGN: StartTime=TNOW;

        BRANCH,1:
                IF,Option==0,AVI:
                ELSE,Cycle;
```

;Traffic Flow Module

```
;Phase 1 - North Bound

        CREATE;

ReadNx1 READ, F11,FREE: ArrTime,VehType,VehDir;

        ASSIGN: InterArr=ArrTime-TNOW;

;Adjusting the Short Headways

        ASSIGN: V1=V1+1;
```

```
        BRANCH,1:
                IF,V1==1,OK1:
                ELSE, Chk1;

Chk1            ASSIGN: Headway=(Clearance(VehType)+VehLen(LeadVeh))/Spd;

                BRANCH,1:
                        IF, InterArr.LT.Headway, Shift1:
                        ELSE, OK1;

Shift1  ASSIGN: InterArr=Headway;

OK1     ASSIGN: LeadVeh=VehType;

        DELAY: InterArr;

        Duplicate: 1,ReadNx1;

        BRANCH,1:
                IF,(VehDir==1.AND.NQ(ReaderTR1Q).LE.NQ(ReaderTL1Q)).OR.
                  VehDir==2,RR1:
                ELSE, LL1;

RR1     ASSIGN:
                V12=V12+1:
                Direction=2;

        BRANCH,1:
                IF,V12==1,IA12:
                ELSE, TV12;

TV12    BRANCH,1:
                IF,ArrTime.LT.(OAT12+TravTime),TT12:
                ELSE, IA12;

TT12    ASSIGN: TravTime=ArrTime-OAT12;


IA12    ASSIGN:
                OAT12=ArrTime:
                TVA12=TVA12+1:
```

```
          TypArLog12(TVA12)=VehType:
          ArrLog12(TVA12)=ArrTime;

     QUEUE, ReaderTR1Q;
     SEIZE: ReaderTR1;
     DELAY: TravTime;
     RELEASE: ReaderTR1;

;Renumbering the TypArLog() and ArrLog() array by shifting strategy.

     ASSIGN: TVD12=2;

     BRANCH,1:
          IF,TVA12==2,Frel2:
          ELSE, Chkk12;

Frel2 ASSIGN:
          TypArLog12(TVA12)=0:
          ArrLog12(TVA12)=0;

Chkk12 BRANCH,1:
          IF,TVD12.LT.TVA12,Shf12:
          ELSE, Upt12;
Shf12 ASSIGN:
          TypArLog12(TVD12)=TypArLog12(TVD12+1):
          ArrLog12(TVD12)=ArrLog12(TVD12+1):
          TVD12=TVD12+1: NEXT(Chkk12);

Upt12 ASSIGN: TVA12=TVA12-1: NEXT(Dch1);

LL1   ASSIGN:
          V13=V13+1:
          Direction=3;

     BRANCH,1:
          IF,V13==1,IA13:
          ELSE, TV13;

TV13 BRANCH,1:
          IF,ArrTime.LT.(OAT13+TravTime),TT13:
          ELSE, IA13;
```

```
TT13   ASSIGN: TravTime=ArrTime-OAT13;

IA13   ASSIGN:
              OAT13=ArrTime:
              TVA13=TVA13+1:
              TypArLog13(TVA13)=VehType:
              ArrLog13(TVA13)=ArrTime;

       QUEUE, ReaderTL1Q;
       SEIZE: ReaderTL1;
       DELAY: TravTime;
       RELEASE: ReaderTL1;

       ASSIGN: TVD13=2;

       BRANCH,1:
              IF,TVA13==2,Fre13:
              ELSE, Chkk13;

Fre13  ASSIGN:
              TypArLog13(TVA13)=0:
              ArrLog13(TVA13)=0;

Chkk13 BRANCH,1:
              IF,TVD13.LT.TVA13,Shf13:
              ELSE, Upt13;

Shf13  ASSIGN:
              TypArLog13(TVD13)=TypArLog13(TVD13+1):
              ArrLog13(TVD13)=ArrLog13(TVD13+1):
              TVD13=TVD13+1: NEXT(Chkk13);

Upt13  ASSIGN: TVA13=TVA13-1;

Dch1   BRANCH,1:
              IF,(NR(PhaseTR1)==0.AND.NQ(PhaseTR1Q)==0.AND.
                     Direction==2).OR.
                     (NR(PhaseTL1)==0.AND.NQ(PhaseTL1Q)==0.AND.
                     Direction==3),NoDel1:
              ELSE,Del1;

NoDel1 ASSIGN: Duration=0: NEXT(Dup1);
```

```
Del1    ASSIGN: Duration=Discharge(VehType);

Dup1    ASSIGN: StopDel=TNOW;

;Count Trucks and All Vehicles

        BRANCH,1:
                IF,VehType==1,Trk1:
                ELSE,All1;

Trk1    COUNT: ArrTrk1;
All1    COUNT: ArrAll1;

                ASSIGN:
                TimeIn1=TNOW:
                TimeInT(VehType)=TimeInT(VehType)+TimeIn1;

        BRANCH,1:
                IF,Direction==2,R1:
                IF,Direction==3,L1;

L1      ASSIGN:
                CA13=CA13+1:
                TypeLog13(CA13)=VehType:
                DurationT13=DurationT13+Duration: NEXT(TrL1);

R1      ASSIGN:
                CA12=CA12+1:
                TypeLog12(CA12)=VehType:
                DurationT12=DurationT12+Duration;

        COUNT: Arr12;

        QUEUE, PhaseTR1Q;
        SEIZE: PhaseTR1;
        DELAY: Duration;
        RELEASE: PhaseTR1;

        COUNT: Dep12;

;Renumbering Variable TypeLog() Array
```

```
        ASSIGN: CD12=2;

        BRANCH,1:
                IF,CA12==2,Free12:
                ELSE, Chck12;

Free12 ASSIGN: TypeLog12(CA12)=0;

Chck12 BRANCH,1:
                IF,CD12.LT.CA12,Shft12:
                ELSE, Updt12;
Shft12 ASSIGN:
                TypeLog12(CD12)=TypeLog12(CD12+1):
                CD12=CD12+1: NEXT(Chck12);

Updt12  ASSIGN:
                CA12=CA12-1:
                DurationT12=DurationT12-Duration:
                TimeLog12=TNOW: NEXT(PunOut1);

TrL1    COUNT: Arr13;

        QUEUE, PhaseTL1Q;
        SEIZE: PhaseTL1;
        DELAY: Duration;
        RELEASE: PhaseTL1;

        COUNT: Dep13;

        ASSIGN: CD13=2;

        BRANCH,1:
                IF,CA13==2,Free13:
                ELSE, Chck13;

Free13 ASSIGN: TypeLog13(CA13)=0;

Chck13 BRANCH,1:
                IF,CD13.LT.CA13,Shft13:
                ELSE, Updt13;
Shft13 ASSIGN:
```

```
            TypeLog13(CD13)=TypeLog13(CD13+1):
            CD13=CD13+1: NEXT(Chck13);

Updt13 ASSIGN:
            CA13=CA13-1:
            DurationT13=DurationT13-Duration:
            TimeLog13=TNOW;

PunOut1 ASSIGN: Timeout=TNOW;

      BRANCH,1:
            IF,VehType==1,TrkD1:
            ELSE,AllD1;

TrkD1 COUNT: DepTrk1;
AllD1 COUNT: DepAll1;

;Accumulate the Time Each Vehicle Spends in the system.

      ASSIGN:
            TimeInT(VehType)=TimeInT(VehType)-TimeIn1:  !Updating TimeInT()
            AvgStop(VehType)=AvgStop(VehType)+(TNOW-TimeIn1);

      TALLY: VehType, INT(StopDel):
              DISPOSE;

;Phase 2 - West Bound

      CREATE;

ReadNx2 READ, F12,FREE: ArrTime,VehType,VehDir;

      ASSIGN: InterArr=ArrTime-TNOW;

      ASSIGN: V2=V2+1;

      BRANCH,1:
            IF,V2==1,OK2:
            ELSE, Chk2;

Chk2  ASSIGN: Headway=(Clearance(VehType)+VehLen(LeadVeh))/Spd;
```

```
        BRANCH,1:
                IF, InterArr.LT.Headway, Shift2:
                ELSE, OK2;

Shift2  ASSIGN: InterArr=Headway;

OK2     ASSIGN: LeadVeh=VehType;

        DELAY: InterArr;


         Duplicate: 1,ReadNx2;

        BRANCH,1:
                IF,(VehDir==1.AND.NQ(ReaderTR2Q).LE.NQ(ReaderTL2Q)).OR.
                  VehDir==2,RR2:
                ELSE, LL2;

RR2     ASSIGN:
                V22=V22+1:
                Direction=2;

        BRANCH,1:
                IF,V22==1,IA22:
                ELSE, TV22;

TV22    BRANCH,1:
                IF,ArrTime.LT.(OAT22+TravTime),TT22:
                ELSE, IA22;

TT22    ASSIGN: TravTime=ArrTime-OAT22;

IA22    ASSIGN:
                OAT22=ArrTime:
                TVA22=TVA22+1:
                TypArLog22(TVA22)=VehType:
                ArrLog22(TVA22)=ArrTime;

        QUEUE, ReaderTR2Q;
        SEIZE: ReaderTR2;
        DELAY: TravTime;
        RELEASE: ReaderTR2;
```

```
        ASSIGN: TVD22=2;

        BRANCH,1:
                IF,TVA22==2,Fre22:
                ELSE, Chkk22;

Fre22   ASSIGN:
                TypArLog22(TVA22)=0:
                ArrLog22(TVA22)=0;

Chkk22 BRANCH,1:
                IF,TVD22.LT.TVA22,Shf22:
                ELSE, Upt22;

Shf22   ASSIGN:
                TypArLog22(TVD22)=TypArLog22(TVD22+1):
                ArrLog22(TVD22)=ArrLog22(TVD22+1):
                TVD22=TVD22+1: NEXT(Chkk22);

Upt22   ASSIGN: TVA22=TVA22-1: NEXT(Dch2);

LL2     ASSIGN:
                V23=V23+1:
                Direction=3;

        BRANCH,1:
                IF,V23==1,IA23:
                ELSE, TV23;

TV23    BRANCH,1:
                IF,ArrTime.LT.(OAT23+TravTime),TT23:
                ELSE, IA23;

TT23    ASSIGN: TravTime=ArrTime-OAT23;

IA23    ASSIGN:
                OAT23=ArrTime:
                TVA23=TVA23+1:
                TypArLog23(TVA23)=VehType:
                ArrLog23(TVA23)=ArrTime;
```

```
        QUEUE, ReaderTL2Q;
        SEIZE: ReaderTL2;
        DELAY: TravTime;
        RELEASE: ReaderTL2;

        ASSIGN: TVD23=2;

        BRANCH,1:
                IF,TVA23==2,Fre23:
                ELSE, Chkk23;

Fre23   ASSIGN:
                TypArLog23(TVA23)=0:
                ArrLog23(TVA23)=0;

Chkk23 BRANCH,1:
                IF,TVD23.LT.TVA23,Shf23:
                ELSE, Upt23;
Shf23   ASSIGN:
                TypArLog23(TVD23)=TypArLog23(TVD23+1):
                ArrLog23(TVD23)=ArrLog23(TVD23+1):
                TVD23=TVD23+1: NEXT(Chkk23);

Upt23   ASSIGN: TVA23=TVA23-1;

Dch2    BRANCH,1:
                IF,(NR(PhaseTR2)==0.AND.NQ(PhaseTR2Q)==0.AND.
                        Direction==2).OR.
                        (NR(PhaseTL2)==0.AND.NQ(PhaseTL2Q)==0.AND.
                        Direction==3),NoDel2:
                ELSE,Del2;

NoDel2 ASSIGN: Duration=0: NEXT(Dup2);

Del2    ASSIGN: Duration=Discharge(VehType);

Dup2    ASSIGN: StopDel=TNOW;

;Count Trucks and All Vehicles

        BRANCH,1:
                IF,VehType==3,Trk2:
```

```
        ELSE,All2;

Trk2  COUNT: ArrTrk2;
All2  COUNT: ArrAll2;

      ASSIGN:
            TimeIn2=TNOW:
            TimeInT(VehType)=TimeInT(VehType)+TimeIn2;


      BRANCH,1:
            IF,Direction==2,R2:
            IF,Direction==3,L2;

L2    ASSIGN:
            CA23=CA23+1:
            TypeLog23(CA23)=VehType:
            DurationT23=DurationT23+Duration: NEXT(TrL2);

R2    ASSIGN:
            CA22=CA22+1:
            TypeLog22(CA22)=VehType:
            DurationT22=DurationT22+Duration;

      COUNT: Arr22;

      QUEUE, PhaseTR2Q;
      SEIZE: PhaseTR2;
      DELAY: Duration;
      RELEASE: PhaseTR2;

      COUNT: Dep22;

      ASSIGN: CD22=2;

      BRANCH,1:
            IF,CA22==2,Free22:
            ELSE, Chck22;

Free22 ASSIGN: TypeLog22(CA22)=0;

Chck22  BRANCH,1:
```

```
            IF,CD22.LT.CA22,Shft22:
            ELSE, Updt22;

Shft22 ASSIGN:
            TypeLog22(CD22)=TypeLog22(CD22+1):
            CD22=CD22+1: NEXT(Chck22);

Updt22  ASSIGN:
            CA22=CA22-1:
            DurationT22=DurationT22-Duration:
            TimeLog22=TNOW: NEXT(PunOut2);

TrL2   COUNT: Arr23;

       QUEUE, PhaseTL2Q;
       SEIZE: PhaseTL2;
       DELAY: Duration;
       RELEASE: PhaseTL2;

       COUNT: Dep23;

       ASSIGN: CD23=2;

       BRANCH,1:
            IF,CA23==2,Free23:
            ELSE, Chck23;

Free23 ASSIGN: TypeLog23(CA23)=0;

Chck23  BRANCH,1:
            IF,CD23.LT.CA23,Shft23:
         ELSE, Updt23;

Shft23 ASSIGN:
            TypeLog23(CD23)=TypeLog23(CD23+1):
            CD23=CD23+1: NEXT(Chck23);

Updt23  ASSIGN:
            CA23=CA23-1:
            DurationT23=DurationT23-Duration:
            TimeLog23=TNOW;
```

```
PunOut2  ASSIGN: Timeout=TNOW;

         BRANCH,1:
                  IF,VehType==3,TrkD2:
                  ELSE,AllD2;

TrkD2 COUNT: DepTrk2;
AllD2             COUNT: DepAll2;

         ASSIGN:
                  TimeInT(VehType)=TimeInT(VehType)-TimeIn2:
                  AvgStop(VehType)=AvgStop(VehType)+(TNOW-TimeIn2);

         TALLY: VehType, INT(StopDel):
                     DISPOSE;

;Phase 3 - South Bound

         CREATE;

ReadNx3 READ, F13,FREE: ArrTime,VehType,VehDir;

         ASSIGN: InterArr=ArrTime-TNOW;

         ASSIGN: V3=V3+1;

         BRANCH,1:
                  IF,V3==1,OK3:
                  ELSE, Chk3;

Chk3     ASSIGN: Headway=(Clearance(VehType)+VehLen(LeadVeh))/Spd;

         BRANCH,1:
                  IF, InterArr.LT.Headway, Shift3:
                  ELSE, OK3;

Shift3   ASSIGN: InterArr=Headway;

OK3      ASSIGN: LeadVeh=VehType;

         DELAY: InterArr;
```

```
        Duplicate: 1,ReadNx3;

        BRANCH,1:
                IF,(VehDir==1.AND.NQ(ReaderTR3Q).LE.NQ(ReaderTL3Q)).OR.
                  VehDir==2,RR3:
                ELSE, LL3;

RR3     ASSIGN:
                V32=V32+1:
                Direction=2;

        BRANCH,1:
         IF,V32==1,IA32:
         ELSE, TV32;

TV32    BRANCH,1:
                IF,ArrTime.LT.(OAT32+TravTime),TT32:
                ELSE, IA32;

TT32    ASSIGN: TravTime=ArrTime-OAT32;

IA32    ASSIGN:
                OAT32=ArrTime:
                TVA32=TVA32+1:
                TypArLog32(TVA32)=VehType:
                ArrLog32(TVA32)=ArrTime;

        QUEUE, ReaderTR3Q;
        SEIZE: ReaderTR3;
        DELAY: TravTime;
        RELEASE: ReaderTR3;

        ASSIGN: TVD32=2;

        BRANCH,1:
                IF,TVA32==2,Fre32:
                ELSE, Chkk32;

Fre32   ASSIGN:
                TypArLog32(TVA32)=0:
                ArrLog32(TVA32)=0;
```

```
Chkk32 BRANCH,1:
            IF,TVD32.LT.TVA32,Shf32:
            ELSE, Upt32;

Shf32  ASSIGN:
            TypArLog32(TVD32)=TypArLog32(TVD32+1):
            ArrLog32(TVD32)=ArrLog32(TVD32+1):
            TVD32=TVD32+1: NEXT(Chkk32);

Upt32  ASSIGN: TVA32=TVA32-1: NEXT(Dch3);

LL3    ASSIGN:
            V33=V33+1:
            Direction=3;

       BRANCH,1:
            IF,V33==1,IA33:
            ELSE, TV33;

TV33    BRANCH,1:
        IF,ArrTime.LT.(OAT33+TravTime),TT33:
        ELSE, IA33;

TT33   ASSIGN: TravTime=ArrTime-OAT33;

IA33   ASSIGN:
            OAT33=ArrTime:
            TVA33=TVA33+1:
            TypArLog33(TVA33)=VehType:
            ArrLog33(TVA33)=ArrTime;

       QUEUE, ReaderTL3Q;
       SEIZE: ReaderTL3;
       DELAY: TravTime;
       RELEASE: ReaderTL3;

       ASSIGN: TVD33=2;

       BRANCH,1:
            IF,TVA33==2,Fre33:
            ELSE, Chkk33;
```

```
Fre33  ASSIGN:
               TypArLog33(TVA33)=0:
               ArrLog33(TVA33)=0;

Chkk33 BRANCH,1:
               IF,TVD33.LT.TVA33,Shf33:
               ELSE, Upt33;
Shf33  ASSIGN:
               TypArLog33(TVD33)=TypArLog33(TVD33+1):
               ArrLog33(TVD33)=ArrLog33(TVD33+1):
               TVD33=TVD33+1: NEXT(Chkk33);

Upt33  ASSIGN: TVA33=TVA33-1;

Dch3   BRANCH,1:
               IF,(NR(PhaseTR3)==0.AND.NQ(PhaseTR3Q)==0.AND.
                      Direction==2).OR.
                      (NR(PhaseTL3)==0.AND.NQ(PhaseTL3Q)==0.AND.
                      Direction==3),NoDel3:
               ELSE,Del3;

NoDel3 ASSIGN: Duration=0: NEXT(Dup3);

Del3   ASSIGN: Duration=Discharge(VehType);

Dup3   ASSIGN: StopDel=TNOW;

;Count Trucks and All Vehicles

       BRANCH,1:
               IF,VehType==5,Trk3:
               ELSE,All3;

Trk3   COUNT: ArrTrk3;
All3   COUNT: ArrAll3;

       ASSIGN:
               TimeIn3=TNOW:
               TimeInT(VehType)=TimeInT(VehType)+TimeIn3;

       BRANCH,1:
               IF,Direction==2,R3:
```

```
          IF,Direction==3,L3;

L3     ASSIGN:
          CA33=CA33+1:
          TypeLog33(CA33)=VehType:
          DurationT33=DurationT33+Duration: NEXT(TrL3);

R3     ASSIGN:
          CA32=CA32+1:
          TypeLog32(CA32)=VehType:
          DurationT32=DurationT32+Duration;

       COUNT: Arr32;

       QUEUE, PhaseTR3Q;
       SEIZE: PhaseTR3;
       DELAY: Duration;
       RELEASE: PhaseTR3;

       COUNT: Dep32;

       ASSIGN: CD32=2;

       BRANCH,1:
          IF,CA32==2,Free32:
          ELSE, Chck32;

Free32 ASSIGN: TypeLog32(CA32)=0;

Chck32  BRANCH,1:
          IF,CD32.LT.CA32,Shft32:
          ELSE, Updt32;

Shft32 ASSIGN:
          TypeLog32(CD32)=TypeLog32(CD32+1):
          CD32=CD32+1: NEXT(Chck32);

Updt32  ASSIGN:
          CA32=CA32-1:
          DurationT32=DurationT32-Duration:
          TimeLog32=TNOW: NEXT(PunOut3);
```

```
TrL3    COUNT: Arr33;

        QUEUE, PhaseTL3Q;
        SEIZE: PhaseTL3;
        DELAY: Duration;
        RELEASE: PhaseTL3;

        COUNT: Dep33;

        ASSIGN: CD33=2;

        BRANCH,1:
                IF,CA33==2,Free33:
                ELSE, Chck33;

Free33 ASSIGN: TypeLog33(CA33)=0;

Chck33  BRANCH,1:
                IF,CD33.LT.CA33,Shft33:
                ELSE, Updt33;

Shft33 ASSIGN:
                TypeLog33(CD33)=TypeLog33(CD33+1):
                CD33=CD33+1: NEXT(Chck33);

Updt33  ASSIGN: CA33=CA33-1:
                DurationT33=DurationT33-Duration:
                TimeLog33=TNOW;

PunOut3 ASSIGN: Timeout=TNOW;

        BRANCH,1:
                IF,VehType==5,TrkD3:
                ELSE,AllD3;

TrkD3 COUNT: DepTrk3;
AllD3  COUNT: DepAll3;

        ASSIGN:
                TimeInT(VehType)=TimeInT(VehType)-TimeIn3:
                AvgStop(VehType)=AvgStop(VehType)+(TNOW-TimeIn3);
```

```
        TALLY: VehType, INT(StopDel):
                DISPOSE;

;Phase 4 - East Bound

        CREATE;

ReadNx4 READ, F14,FREE: ArrTime,VehType,VehDir;

        ASSIGN: InterArr=ArrTime-TNOW;

        ASSIGN: V4=V4+1;

        BRANCH,1:
                IF,V4==1,OK4:
                ELSE, Chk4;

Chk4    ASSIGN: Headway=(Clearance(VehType)+VehLen(LeadVeh))/Spd;

        BRANCH,1:
                IF, InterArr.LT.Headway, Shift4:
                ELSE, OK4;

Shift4  ASSIGN: InterArr=Headway;

OK4     ASSIGN: LeadVeh=VehType;

        DELAY: InterArr;

        Duplicate: 1,ReadNx4;

        BRANCH,1:
                IF,(VehDir==1.AND.NQ(ReaderTR4Q).LE.NQ(ReaderTL4Q)).OR.
                   VehDir==2,RR4:
                ELSE, LL4;

RR4     ASSIGN:
                V42=V42+1:
                Direction=2;

        BRANCH,1:
                IF,V42==1,IA42:
```

```
        ELSE, TV42;

TV42  BRANCH,1:
        IF,ArrTime.LT.(OAT42+TravTime),TT42:
        ELSE, IA42;

TT42  ASSIGN: TravTime=ArrTime-OAT42;

IA42  ASSIGN:
        OAT42=ArrTime:
        TVA42=TVA42+1:
        TypArLog42(TVA42)=VehType:
        ArrLog42(TVA42)=ArrTime;

      QUEUE, ReaderTR4Q;
      SEIZE: ReaderTR4;
      DELAY: TravTime;
      RELEASE: ReaderTR4;

      ASSIGN: TVD42=2;

      BRANCH,1:
        IF,TVA42==2,Fre42:
        ELSE, Chkk42;

Fre42  ASSIGN:
        TypArLog42(TVA42)=0:
        ArrLog42(TVA42)=0;

Chkk42 BRANCH,1:
        IF,TVD42.LT.TVA42,Shf42:
        ELSE, Upt42;

Shf42  ASSIGN:
        TypArLog42(TVD42)=TypArLog42(TVD42+1):
        ArrLog42(TVD42)=ArrLog42(TVD42+1):
        TVD42=TVD42+1: NEXT(Chkk42);

Upt42  ASSIGN: TVA42=TVA42-1: NEXT(Dch4);

LL4    ASSIGN:
        V43=V43+1:
```

```
        Direction=3;

    BRANCH,1:
        IF,V43==1,IA43:
        ELSE, TV43;

TV43  BRANCH,1:
        IF,ArrTime.LT.(OAT43+TravTime),TT43:
        ELSE, IA43;

TT43  ASSIGN: TravTime=ArrTime-OAT43;

IA43  ASSIGN:
        OAT43=ArrTime:
        TVA43=TVA43+1:
        TypArLog43(TVA43)=VehType:
        ArrLog43(TVA43)=ArrTime;

    QUEUE, ReaderTL4Q;
    SEIZE: ReaderTL4;
    DELAY: TravTime;
    RELEASE: ReaderTL4;

    ASSIGN: TVD43=2;

    BRANCH,1:
        IF,TVA43==2,Fre43:
        ELSE, Chkk43;

Fre43 ASSIGN:
        TypArLog43(TVA43)=0:
        ArrLog43(TVA43)=0;

Chkk43 BRANCH,1:
        IF,TVD43.LT.TVA43,Shf43:
        ELSE, Upt43;

Shf43 ASSIGN:
        TypArLog43(TVD43)=TypArLog43(TVD43+1):
        ArrLog43(TVD43)=ArrLog43(TVD43+1):
        TVD43=TVD43+1: NEXT(Chkk43);
```

```
Upt43  ASSIGN: TVA43=TVA43-1;

Dch4   BRANCH,1:
              IF,(NR(PhaseTR4)==0.AND.NQ(PhaseTR4Q)==0.AND.
              Direction==2).OR.
              (NR(PhaseTL4)==0.AND.NQ(PhaseTL4Q)==0.AND.
              Direction==3),NoDel4:
              ELSE,Del4;

NoDel4 ASSIGN: Duration=0: NEXT(Dup4);

Del4   ASSIGN: Duration=Discharge(VehType);

Dup4   ASSIGN: StopDel=TNOW;

;Count Trucks and All Vehicles

       BRANCH,1:
              IF,VehType==7,Trk4:
              ELSE,All4;

Trk4   COUNT: ArrTrk4;
All4   COUNT: ArrAll4;

       ASSIGN:
              TimeIn4=TNOW:
              TimeInT(VehType)=TimeInT(VehType)+TimeIn4;

       BRANCH,1:
              IF,Direction==2,R4:
              IF,Direction==3,L4;

L4     ASSIGN:
              CA43=CA43+1:
              TypeLog43(CA43)=VehType:
              DurationT43=DurationT43+Duration: NEXT(TrL4);

R4     ASSIGN:
              CA42=CA42+1:
              TypeLog42(CA42)=VehType:
              DurationT42=DurationT42+Duration;
```

```
        COUNT: Arr42;

        QUEUE, PhaseTR4Q;
        SEIZE: PhaseTR4;
        DELAY: Duration;
        RELEASE: PhaseTR4;

        COUNT: Dep42;

        ASSIGN: CD42=2;

        BRANCH,1:
              IF,CA42==2,Free42:
              ELSE, Chck42;

Free42 ASSIGN: TypeLog42(CA42)=0;

Chck42  BRANCH,1:
              IF,CD42.LT.CA42,Shft42:
              ELSE, Updt42;

Shft42 ASSIGN:
              TypeLog42(CD42)=TypeLog42(CD42+1):
              CD42=CD42+1: NEXT(Chck42);

Updt42  ASSIGN:
              CA42=CA42-1:
              DurationT42=DurationT42-Duration:
              TimeLog42=TNOW: NEXT(PunOut4);

TrL4    COUNT: Arr43;

        QUEUE, PhaseTL4Q;
        SEIZE: PhaseTL4;
        DELAY: Duration;
        RELEASE: PhaseTL4;

        COUNT: Dep43;

        ASSIGN: CD43=2;

        BRANCH,1:
```

```
            IF,CA43==2,Free43:
            ELSE, Chck43;

Free43 ASSIGN: TypeLog43(CA43)=0;

Chck43  BRANCH,1:
            IF,CD43.LT.CA43,Shft43:
            ELSE, Updt43;

Shft43 ASSIGN:
            TypeLog43(CD43)=TypeLog43(CD43+1):
            CD43=CD43+1: NEXT(Chck43);

Updt43  ASSIGN:
            CA43=CA43-1:
            DurationT43=DurationT43-Duration:
            TimeLog43=TNOW;

PunOut4 ASSIGN: Timeout=TNOW;

       BRANCH,1:
            IF,VehType==7,TrkD4:
            ELSE,AllD4;

TrkD4 COUNT: DepTrk4;
AllD4       COUNT: DepAll4;

       ASSIGN:
            TimeInT(VehType)=TimeInT(VehType)-TimeIn4:
            AvgStop(VehType)=AvgStop(VehType)+(TNOW-TimeIn4);

       TALLY: VehType, INT(StopDel):
             DISPOSE;
END;
```

Simulation Model - EXPERIMENT

Lincoln Way and Duff Avenue Intersection

```
BEGIN;

PROJECT,     4_Phase Intersection,Ali;

VARIABLES:
             Option,0:                  !If 0, AVSI is engaged
             PARTII,2:                  !If 2, PARTII is engaged

             Truck%1,.30:
             Truck%2,.30:
             Truck%3,.30:
             Truck%4,.30:

             ActFlow1,600:
             ActFlow2,600:
             ActFlow3,600:
             ActFlow4,600:

             W(2),.75,.25:
             DIValue,15:

             Period1,299.999:
             Period2,299.995:
             Period3,299.9999:

             Period4,7199:

             Decel,10:
             MaxGX,10:
             MaxSat,1:
             R2b,20:

             LostTime,8:
             Xc,1:
             Multi:
             Adj:
             SatFlow1: SatFlow2: SatFlow3: SatFlow4:
             CycleLength: Green1: Green2: Green3: Green4:
```

```
ActCl:
VoverS1: VoverS2: VoverS3: VoverS4:
NewCycle:
NewG1: NewG2: NewG3: NewG4:

Spd,44:
V1: V2: V3: V4:

Discharge(8),3,2,3,2,3,2,3,2:
;        Discharge(8),4,2,4,2,4,2,4,2:

Clearance(8),15,10,15,10,15,10,15,10:
VehLen(8),60,15,60,15,60,15,60,15:

TypeLog12(50): TimeLog12:
TypeLog13(50): TimeLog13:
TypeLog22(50): TimeLog22:
TypeLog23(50): TimeLog23:
TypeLog32(50): TimeLog32:
TypeLog33(50): TimeLog33:
TypeLog42(50): TimeLog42:
TypeLog43(50): TimeLog43:

CA12,1: CD12:
CA13,1: CD13:
CA22,1: CD22:
CA23,1: CD23:
CA32,1: CD32:
CA33,1: CD33:
CA42,1: CD42:
CA43,1: CD43:

V12: V13:
V22: V23:
V32: V33:
V42: V43:

OAT12: OAT13:
OAT22: OAT23:
OAT32: OAT33:
OAT42: OAT43:
```

TVA12,1: TVD12:
TVA13,1: TVD13:
TVA22,1: TVD22:
TVA23,1: TVD23:
TVA32,1: TVD32:
TVA33,1: TVD33:
TVA42,1: TVD42:
TVA43,1: TVD43:

ArrLog12(20): ArrLog13(20):
ArrLog22(20): ArrLog23(20):
ArrLog32(20): ArrLog33(20):
ArrLog42(20): ArrLog43(20):

TypArLog12(20): TypArLog13(20):
TypArLog22(20): TypArLog23(20):
TypArLog32(20): TypArLog33(20):
TypArLog42(20): TypArLog43(20):

JR2a,1: JL2a,1:
JR2b,1: JL2b,1:
TagL: TagR:

DurationT12: DurationT13:
DurationT22: DurationT23:
DurationT32: DurationT33:
DurationT42: DurationT43:

ExtraTime:
DI:
TimeInT(8):
AvgDel(8):
TimeNeed:
TimeNeedT:
BusyNum:
AvgStop(8): AvgStopT(8):
TT(4):
ChkTime:
Satur:
QL: Prd:
Tm: ChgTm: OV(4): Q(4);
K: CPh:

```
                LapDel(8): OLapDel(8):
                LapVol(8):
                Vol(8): OVol(8):
                IntDelay:
                TrkDelay:
                CarDelay:
                StartTime:
                ChangeTime:

;               X1: X2: X3: X4:
;               Cap1: Cap2: Cap3: Cap4:
;               Delay1: Delay2: Delay3: Delay4:
;               WebDelay;

TABLES:
                HV%,0,.05,1,.975,.95,.93,.91,.89,.87,
                    .85,.83,.81,.79,.77,.75;

ATTRIBUTES:
                TravTime,60:
                ArrTime:
                VehType:
                VehDir:
                Headway:
                LeadVeh:
                StopDel:
                InterArr:
                Duration:
                Timeout:
                TimeIn1: TimeIn2: TimeIn3: TimeIn4;

QUEUES:
                PhaseTR1Q: PhaseTL1Q:
                PhaseTR2Q: PhaseTL2Q:
                PhaseTR3Q: PhaseTL3Q:
                PhaseTR4Q: PhaseTL4Q:

                ReaderTR1Q: ReaderTL1Q:
                ReaderTR2Q: ReaderTL2Q:
                ReaderTR3Q: ReaderTL3Q:
                ReaderTR4Q: ReaderTL4Q:
```

Dum1Q: Dum2Q: Dum3Q: Dum4Q: Dum5Q;

RESOURCES:

PhaseTR1: PhaseTL1:
PhaseTR2: PhaseTL2:
PhaseTR3: PhaseTL3:
PhaseTR4: PhaseTL4:

ReaderTR1: ReaderTL1:
ReaderTR2: ReaderTL2:
ReaderTR3: ReaderTL3:
ReaderTR4: ReaderTL4;

COUNTERS:

ArrTrk1: DepTrk1:
ArrAll1: DepAll1:
ArrTrk2: DepTrk2:
ArrAll2: DepAll2:
ArrTrk3: DepTrk3:
ArrAll3: DepAll3:
ArrTrk4: DepTrk4:
ArrAll4: DepAll4:

Arr12: Dep12:
Arr13: Dep13:
Arr22: Dep22:
Arr23: Dep23:
Arr32: Dep32:
Arr33: Dep33:
Arr42: Dep42:
Arr43: Dep43;

TALLIES:

Truck Delay in Phase 1:
Car Delay in Phase 1:
Truck Delay in Phase 2:
Car Delay in Phase 2:
Truck Delay in Phase 3:
Car Delay in Phase 3:
Truck Delay in Phase 4:
Car Delay in Phase 4;

```
DSTATS:
            NQ(PhaseTR1Q),Phase 1 Queue T_R:
            NQ(PhaseTL1Q),Phase 1 Queue T_L:
            NQ(PhaseTR2Q),Phase 2 Queue T_R:
            NQ(PhaseTL2Q),Phase 2 Queue T_L:
            NQ(PhaseTR3Q),Phase 3 Queue T_R:
            NQ(PhaseTL3Q),Phase 3 Queue T_L:
            NQ(PhaseTR4Q),Phase 4 Queue T_R:
            NQ(PhaseTL4Q),Phase 4 Queue T_L;

FILES:
            Output,"o",SEQ,"(5(f7.2,1x),4(f3.2,1x),3(f5.2,1x))",IGN:
            TypeFile1,"tf1",SEQ,"(3(f7.2,1x),8f7.2)",IGN:
            TypeFile2,"tf2",SEQ,"(3(f7.2,1x),8f7.2)",IGN:
            TypeFile3,"tf3",SEQ,"(3(f7.2,1x),8f7.2)",IGN:
            TypeFile4,"tf4",SEQ,"(3(f7.2,1x),8f7.2)",IGN:
            F11,"t11",SEQ,FRE,IGN:
            F12,"t12",SEQ,FRE,IGN:
            F13,"t13",SEQ,FRE,IGN:
            F14,"t14",SEQ,FRE,IGN:
            F,"t",SEQ,FRE,IGN;

REPLICATE, 1,0,7260;
;REPLICATE,1,0,3660;

END;
```

APPENDIX III - PRE-GENERATOR PROGRAM LISTINGS

Pre-Generator Model - MODEL

Lincoln Way and Duff Avenue Intersection

```
BEGIN;

;Create vehicles--Phase 1 - North Bound

        CREATE,1,Off: EXPO(3600/vph1,1);

        ASSIGN: ArrTime=TNOW;

;Vehicle Type Assignment

        ASSIGN: VehType=DISC(Truck%1,1,1,2,2);

;Vehicle Direction Assignment--DirL1% left turn, DirR1% right turn, rest through.

        ASSIGN: VehDir=DISC(DirL1,3,DirR1,2,1,1,3);

;Count Trucks and All Vehicles

        BRANCH,1:
                IF,VehType==1,Trk1:
                ELSE,All1;

Trk1    COUNT: ArrTrk1;
All1    COUNT: ArrAll1;

        WRITE, F11,FREE: ArrTime,VehType,VehDir: DISPOSE;

;Create vehicles--Phase 2 - West Bound

        CREATE,1,Off: EXPO(3600/vph2,1);

        ASSIGN: ArrTime=TNOW;

;Vehicle Type Assignment

        ASSIGN: VehType=DISC(Truck%2,3,1,4,2);
```

```
;Vehicle Direction Assignment

        ASSIGN: VehDir=DISC(DirL2,3,DirR2,2,1,1,3);

;Count Trucks and All Vehicles

        BRANCH,1:
                IF,VehType==3,Trk2:
                ELSE,All2;

Trk2    COUNT: ArrTrk2;
All2    COUNT: ArrAll2;

      WRITE, F12,FREE: ArrTime,VehType,VehDir: DISPOSE;

;Create vehicles--Phase 3 - South Bound

        CREATE,1,Off: EXPO(3600/vph3,1);

        ASSIGN: ArrTime=TNOW;

;Vehicle Type Assignment

        ASSIGN: VehType=DISC(Truck%3,5,1,6,2);

;Vehicle Direction Assignment

        ASSIGN: VehDir=DISC(DirL3,3,DirR3,2,1,1,3);

;Count Trucks and All Vehicles

        BRANCH,1:
                IF,VehType==5,Trk3:
                ELSE,All3;

Trk3    COUNT: ArrTrk3;
All3    COUNT: ArrAll3;

        WRITE, F13,FREE: ArrTime,VehType,VehDir: DISPOSE;

;Create vehicles--Phase 4 - East Bound
```

```
        CREATE,1,Off: EXPO(3600/vph4,1);
        ASSIGN: ArrTime=TNOW;

;Vehicle Type Assignment

        ASSIGN: VehType=DISC(Truck%4,7,1,8,2);

;Vehicle Direction Assignment

        ASSIGN: VehDir=DISC(DirL4,3,DirR4,2,1,1,3);

;Count Trucks and All Vehicles

        BRANCH,1:
                IF,VehType==7,Trk4:
                ELSE,All4;

Trk4    COUNT: ArrTrk4;
All4    COUNT: ArrAll4;

      WRITE, F14,FREE: ArrTime,VehType,VehDir: DISPOSE;

;Record Traffic Flow Rates and Truck Percentages for Each Period

        CREATE;
Rep     DELAY: Period;

        ASSIGN:
                VehperInt1=NC(ArrAll1)-CountVeh1:
                CountVeh1=NC(ArrAll1):
                ActFlow1=VehperInt1*(3600/Period):
                TrkperInt1=NC(ArrTrk1)-CountTrk1:
                CountTrk1=NC(ArrTrk1):
                Truck%1=TrkperInt1/VehperInt1;

        ASSIGN:
                VehperInt2=NC(ArrAll2)-CountVeh2:
                CountVeh2=NC(ArrAll2):
                ActFlow2=VehperInt2*(3600/Period):
                TrkperInt2=NC(ArrTrk2)-CountTrk2:
                CountTrk2=NC(ArrTrk2):
```

```
            Truck%2=TrkperInt2/VehperInt2;

        ASSIGN:
            VehperInt3=NC(ArrAll3)-CountVeh3:
            CountVeh3=NC(ArrAll3):
            ActFlow3=VehperInt3*(3600/Period):
            TrkperInt3=NC(ArrTrk3)-CountTrk3:
            CountTrk3=NC(ArrTrk3):
            Truck%3=TrkperInt3/VehperInt3;

        ASSIGN:
            VehperInt4=NC(ArrAll4)-CountVeh4:
            CountVeh4=NC(ArrAll4):
            ActFlow4=VehperInt4*(3600/Period):
            TrkperInt4=NC(ArrTrk4)-CountTrk4:
            CountTrk4=NC(ArrTrk4):
            Truck%4=TrkperInt4/VehperInt4;

        WRITE, F,FREE: ActFlow1,ActFlow2,ActFlow3,ActFlow4,
                     Truck%1,Truck%2,Truck%3,Truck%4;

        BRANCH,1:
;           IF,TNOW.GT.Period1,Wrap:
            Else,Rep;

Wrap    WRITE, F,FREE: vph1,vph2,vph3,vph4;

;       WRITE, F11,FREE: 3600+61,2;
;       WRITE, F12,FREE: 3600+61,4;
;       WRITE, F13,FREE: 3600+61,6;
;       WRITE, F14,FREE: 3600+61,8: DISPOSE;

        WRITE, F11,FREE: 7200+1,2;
        WRITE, F12,FREE: 7200+1,4;
        WRITE, F13,FREE: 7200+1,6;
        WRITE, F14,FREE: 7200+1,8: DISPOSE;

END;
```

Pre-Generator Model - EXPERIMENT

Lincoln Way and Duff Avenue Intersection

```
BEGIN;

PROJECT,    4_phase Intersection,Ali;

ATTRIBUTES:
            vph1,600:
            vph2,600:
            vph3,600:
            vph4,600:

            Period,300:
;           Period,3600:

            Truck%1,.30:
            Truck%2,.30:
            Truck%3,.30:
            Truck%4,.30:

            DirL1,.30: DirR1,.45:
            DirL2,.30: DirR2,.45:
            DirL3,.30: DirR3,.45:
            DirL4,.30: DirR4,.45:

;Used in Validation of Results
;
;           vph1,665:
;           vph2,633:
;           vph3,665:
;           vph4,630:
;           Truck%1,.009:
;           Truck%2,.008:
;           Truck%3,.015:
;           Truck%4,.029:
;           DirL1,.296: DirR1,.409:
;           DirL2,.313: DirR2,.572:
;           DirL3,.202: DirR3,.331:
;           DirL4,.363: DirR4,.522:
```

```
          Off,0:
          VehDir:
          ArrTime:
          VehType:
          Period1, 7199:

          ActFlow1: ActFlow2: ActFlow3: ActFlow4:
          VehperInt1: VehperInt2: VehperInt3: VehperInt4:
          TrkperInt1: TrkperInt2: TrkperInt3: TrkperInt4:
          CountVeh1: CountVeh2: CountVeh3: CountVeh4:
          CountTrk1: CountTrk2: CountTrk3: CountTrk4;

COUNTERS:
          ArrTrk1: ArrAll1:
          ArrTrk2: ArrAll2:
          ArrTrk3: ArrAll3:
          ArrTrk4: ArrAll4;

FILES:
          F11,"t11",SEQ,FRE:
          F12,"t12",SEQ,FRE:
          F13,"t13",SEQ,FRE:
          F14,"t14",SEQ,FRE:
          F,"t",SEQ,FRE;

REPLICATE, 1,0,7260;
;REPLICATE,1,0,3660;

END;
```

APPENDIX IV - DELAY STUDY WORKSHEETS

Intersection Delay Worksheet

Number of Stopped Vehicles

Sec. / Min.	0	20	40					
00:01								
00:02								
00:03								
00:04								
00:05								
00:06								
00:07								
00:08								
00:09								
00:10								
00:11								
00:12								
00:13								
00:14								
00:15								
00:16								
00:17								
00:18								
00:19								
00:20								
Totals								

$V_s =$ _______

$Volume, V =$ _______

$Delay = \frac{\sum V_s \times I}{V} =$ _______

Traffic Volume Counts Field Sheet

Intersection : Lincoln Way and Duff Avenue

Date: Wednesday 10/26/94

Time: 4:30-5:30 PM

North Bound Approach

END TIME	RIGHT		THROUGH		LEFT	
(minute)	Car	Truck	Car	Truck	Car	Truck
00:05						
00:10						
00:15						
00:20						
00:25						
00:30						
00:35						
00:40						
00:45						
00:50						
00:55						
01:00						