DEVELOPMENT OF A HARDWARE-IN-THE-LOOP ANALYSIS FRAMEWORK FOR ADVANCED ITS APPLICATIONS

A Thesis
Presented to
The Academic Faculty

by

Matthew Roe

In Partial Fulfillment
of the Requirements for the Degree
Civil Engineering and City and Regional Planning in the
School of Civil Engineering and City Planning, Georgia Institute of Technology

Georgia Institute of Technology
May 2009

DEVELOPMENT OF A HARDWARE-IN-THE-LOOP ANALYSIS FRAMEWORK FOR ADVANCED ITS APPLICATIONS

Approved by:

Dr. Michael Hunter, Advisor
School of Civil Engineering
Georgia Institute of Technology

Dr. Jorge Laval
School of Civil Engineering
Georgia Institute of Technology

Dr. Michael Rodgers
School of Civil Engineering
Georgia Institute of Technology

Dr. Jiawen Yang
School of City and Regional Planning
Georgia Institute of Technology

Date Approved: April 6, 2009

To My Dad, who first encouraged me to attend the Georgia Institute of Technology.

ACKNOWLEDGEMENTS

First I would like to thank my advisor, Dr. Michael Hunter who first interested me in transportation research and has helpfully supported and encouraged my work ever since. I would also like to thank the faculty and staff of the school of Civil Engineering and the school of City and Regional Planning, who are too numerous to name but helpful nonetheless.

I would like to thank my parents, John and Tricia, and my sister Catherine for the support and guidance they have given me in my pursuit of academic endeavors. I would like to thank my close friends and fraternity brothers without whom I would not have survived this place.

I would like to thank my gracious proofreaders: Andy Jones, Anna Dorminey, Daniel O'Shaughnessy, Dustin Hipp, Elise Barrella, James Wagner, John Carroll, and Katie Lawrence.

Finally I would like to thank my Lord and Savior Jesus Christ who has given me the grace and mercy to persevere through my academic career, and whose world I have had the pleasure to describe here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

2070	Traffic Signal Controller Standards Specification
ACTRA	Traffic Signal Control System Provided by Siemens
CID	Controller Interface Device
HILS	Hardware-in-the-Loop
ITS	Intelligent Transportation System
SCATS	Sydney Coordinated Adaptive Traffic System
VISSIM	Traffic Simulation Software Manufactured by PTV AG

SUMMARY

As Intelligent Transportation Systems (ITS) become more prevalent, there is an increasing need for a system capable of the rigorous evaluation of new ITS strategies for a wide variety of applications. Pre-deployment testing and fine-tuning of the system, performance evaluation, and alternatives analysis are all potential benefits that could be gained through the evaluation of ITS. Simulation, an increasingly popular tool for transportation analysis, would seem an ideal solution to this problem as it allows for the consideration of many scenarios that may be improbable or impossible to observe in the field. Also, simulation provides a framework that allows for the application of rigorous analysis techniques to the output data, providing an accurate and statistically significant conclusion.

The difficulty is that many ITS strategies are difficult or impossible to implement in a simulated environment. The rapid nature of technology development and the complicated nature of many ITS solutions are difficult to emulate in simulation models. Furthermore, the emulation of a particular ITS solution is not guaranteed to provide the same result that the physical system would, were it subject to the same inputs.

This study seeks to establish a framework for the analysis of advanced ITS applications through the use of Hardware-in-the-Loop Simulation (HILS), which provides a procedure for interfacing simulation models with real-world hardware to conduct analysis. This solution provides the benefits of both advanced ITS evaluation and simulation for powerful and accurate analysis. A framework is established that includes all the steps of the modeling process including construction, validation, calibration, and

output analysis. This ensures that the process surrounding the HILS implementation is valid so that the results of the evaluation are accurate and defendable.

Finally, a case study of the application of the developed framework to the evaluation, a real-world implementation of an advanced ITS application (SCATS in this case) is considered. The effectiveness of the framework in creating and evaluating a corridor using a simulation model wed to real-world hardware is shown. The results of the analysis show the power of this method when correctly applied and demonstrate where further analysis could expand upon the proposed procedure.

CHAPTER 1

INTRODUCTION

Over the years, simulation has developed into a powerful tool for planners and engineers to use in the analysis of transportation systems. Simulation models are desirable for traffic analysis because they are relatively cheap to create and provide a wide range of analysis possibilities. They also generate attractive and easily recognized renderings of traffic conditions that are helpful in the presentation of traffic impact analysis to decision makers and the public at large. Additionally, as simulations have increased in importance for their analysis capabilities, many different Intelligent Transportation Systems (ITS) have been developed as potential solutions to transportation issues across a wide range of applications. ITS is rapidly becoming an integral part of most transportation networks and is seen as a potential solution to many present transportation problem. ITS solutions are very attractive because they propose relatively cheap solutions that often make better use of existing capacity in solving transportation bottlenecks. Simulation provides a relatively low cost evaluation method and allows for the scientific evaluation of various traffic phenomena not possible with the mere observation of field conditions. The positive traits of both ITS applications and simulation thus make the combination of the two in an analysis framework an attractive proposition.

An ideal solution would see proposed or existing ITS strategies implemented in a simulation framework where they can be rigorously evaluated and refined so that their implementation in the field leads to significant improvements for users and a justifiable return on investment for the implementing jurisdiction. Simulation would allow ITS developers to "field-test" their systems at a fraction of the cost of an actual implementation, as well as allowing the ITS to be exposed to scenarios that would be difficult to impossible to manifest on demand during short field tests. These could include accident scenarios, future growth traffic pattern changes, and special event traffic impacts.

Study Need

While it is certainly desirable to make use of simulation in the evaluation of ITS strategies, not all ITS strategies are easily modeled within the existing simulation frameworks. While emulation can provide some representation of ITS elements within a simulated network, emulators are not available for all ITS strategies. Furthermore, the quality of the emulators in representing the true behavior of the ITS cannot always be guaranteed. The only way to resolve both issues is to make use of Hardware-in-the-Loop Simulation (HILS) where the real-world elements of the ITS are married to a simulation model for testing and evaluation purposes.

Study Objective

This study will outline the process by which a HILS system can be developed from the ground up, including the model construction, validation, calibration, and combination with a real-world ITS. The pitfalls and challenges in implementing the system will be explored. Additionally, a framework for the statistically rigorous analysis of the results generated by the HILS system will be proposed.

Study Overview

The purpose of this study is to create a framework for the rigorous statistical analysis of advanced intelligent transportation systems through the use of Hardware-in-the-Loop simulation. A summary of the sections in this report is presented below.

Background

This section gives a basic overview of VISSIM, the selected simulation package for this project. Details about network elements and the model representation of the physical network, driver behavior parameters, and baseline input data and distributions used in the creation of a model are reviewed. The facilities for data output for further analysis are also discussed. A basic overview of HILS and a conceptual framework for its implementation with VISSIM is presented.

Literature Review

The literature review covers a wide variety of articles on many topics relevant to the simulation process. Particular attention is paid to the initial calibration and final statistical analysis portions of the simulation process. They both have significant impact on the validity of the results, yet these are the points many simulation frameworks fail to adequately address. Documentation on the concept of HILS as well as its implementation in previous studies is reviewed. Other topics including distributed computing, federation of models, and data output visualization are also reviewed as they are relevant to various phases of the framework development or potential future research.

General Procedure

The procedure for the implementation of the HILS framework is presented here. The following sections detail each step of the process:

1. HILS Setup
2. Model Construction
3. Validation
4. Calibration
5. Simulation Management
6. Analysis Tools
7. Statistical Analysis

Case Study: Cobb Parkway

A case study of an 11-intersection corridor in Cobb County, GA is presented in this chapter. The framework for HILS study is applied to compare an ACTRA time-of-day traffic signal control system against a SCATS adaptive control system. First, the implementation of the HILS system is discussed, including technical hurdles cleared in creating an operational system. Next, effectiveness of the validation and calibration routines in bringing the ACTRA and SCATS models in line with the field data is considered. Finally, the statistical analysis framework is applied to compare the two systems against each other under baseline AM and PM peak hour conditions and under a highway diversion scenario for both the AM and PM peak hours. The initial results of these studies are presented.

CHAPTER 2

BACKGROUND

VISSIM Overview

VISSIM is a microscopic continuous traffic simulation tool with advanced modeling and evaluation capabilities produced by Planung Transport Verker AG (PTV) in Karlsruhe, Germany [1]. Because of its unique modeling capabilities and functionality, VISSIM is effective for the evaluation of a wide range of transportation issues. This section seeks to provide a brief overview of VISSIM as a background for the more in-depth discussions of the implementation of the software in the evaluation framework. VISSIM uses two separate components to simulate traffic flow within a network. The relationship between the two modules is shown in Figure 1.

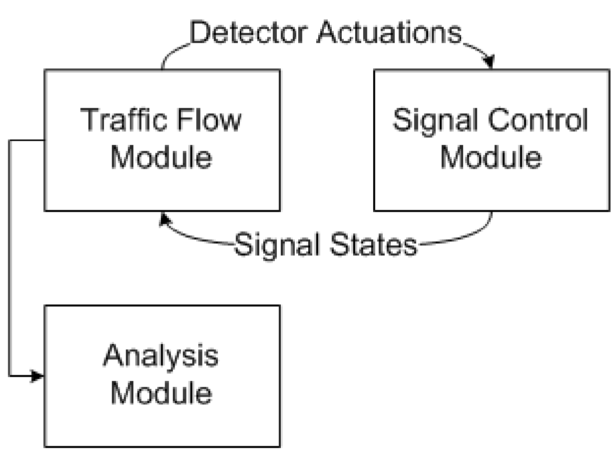

Figure 1. VISSIM Program Structure
(Figure Credit: VISSIM 4.30 User Manual [1])

The Traffic flow module implements the driving behavior for the vehicles in the network and the traffic control module provides the signal states based on available detector information from the flow module [1]. One of the desirable features of VISSIM for this project is the ability to "swap out" the traffic control module from the built-in module that comes with VISSIM to any one of many alternative control systems including hardware-in-the-loop. Additionally, one should note the analysis module. While not providing any input data for the performance of the traffic simulation, the analysis module allows for the capture of performance data from the VISSIM model for later use in validation, calibration, and analysis of alternatives.

Network Representation

Links and Connectors

VISSIM uses a link/connector structure to represent the transportation network under consideration. Links and connectors can have multiple lanes. Links are generally used to represent roadway segments and connectors are typically used to model the turning movements available at junctions or nodes. The link/connector structure allows VISSIM to be used to analyze complex geometries. Typically the construction of a VISSIM model starts with the definition of links and connectors drawn on an appropriately scaled background aerial of the study corridor (VISSIM has support for background images [1]).

Figure 2 shows the link/connector structure in VISSIM. The links are blue and the connectors are pink. On the right side of the figure a typical intersection with turning movements defined by the connectors is shown. To the left is demonstrated an example of more complex geometry with a connector branching off of a link to provide access to the left turn bay.

Figure 2. VISSIM Link/Connector Example

Volume Inputs

In order to place traffic on the network, volume inputs are used. Because all links are one-way, volume inputs are defined for a link and vehicles enter the link at the appropriate end. To generate vehicles in the middle of a link one must define a short "source" link and connect it to the main link at the desired input location point. Volume inputs are defined in terms of vehicles per hour and can vary depending on the simulation time to simulate the rise and ebb of traffic over the course of a given time period. This is discussed further in the section on traffic inputs on page 141.

Speeds

Vehicles in VISSIM enter the network with a desired speed that is defined as part of their initial properties. This desired speed is the speed that the vehicle will travel if they are not constrained by other vehicles or network controls. Vehicle speed can be affected in one of two ways. The first is through the use of reduced speed areas. These areas have no effect on the desired speed of a vehicle, but temporarily cause the speed of a vehicle to be reduced to a specified level. These are useful for reducing speed for turning movements and in other areas where temporary reduced speeds are desired. When approaching a reduced speed area a vehicle will attempt to slow such that it will reach the reduced speed as it enters the portion of the link covered by the reduced speed area. The vehicle will then return to its initially desired speed after exiting the area [1]. The second method for altering the speed of vehicles in the network involves the permanent change of the vehicle's desired speed through the use of desired speed decisions. These are typically used to model permanent or long-term speed changes seen when speed limits change, vehicles enter or exit a freeway, and other applicable situations.

Routing Decisions and Routes

Routing decisions are the method by which the vehicles traveling on a network are directed to their final destination. Because VISSIM does not have a typical link/node structure, the provided routing framework must be flexible enough to allow for vehicles to be accurately routed through complex geometry. Routes are defined as a fixed sequence of links and connectors [1]. These routes can be defined for an entire system of origins and destinations, or to simply direct vehicles on an approach through the upcoming intersection. VISSIM models routes by defining two points. The first point is called the "Decision" and the second the "Destination". For most routes the destination will be a link or connector, but there are also routes used to direct vehicles to parking.

These are defined in the section dealing with parking lots in the next section. When a vehicle in VISSIM passes over a decision point, provided it is not already following a route, it selects a destination point based on the percentage split assigned to each destination point associated with that particular route. After passing a decision point a vehicle will attempt to maneuver into the lane on its current link in order for it to make the next movement on its assigned route. The distance at which the vehicle begins to attempt this change is defined for the connector onto which the vehicle is attempting to enter and is called the "Lane Change Distance". Like volume inputs, the distribution of traffic following various routing decisions can also vary over the course of time to reflect fluctuations in traffic patterns.

Parking

Parking lots in VISSIM allow for the modeling of on-street or off-street parking operations and for the complex modeling of origins and destinations when using advanced dynamic routing. Vehicles are routed to a parking lot using a special type of routing decision; once there, it will seek to find a spot into which it will fit. After parking, the vehicle will remain in the space for a specified dwell time and then leave. A more thorough discussion on the topic of parking lots can be found in [1].

Non-Signalized Control

VISSIM provides network elements for the non-signalized control of intersections through the use of stop signs, priority rules, and conflict areas. In this sense "non-signalized" control occurs not only at an intersection without a signal, but some elements of the non-signalized control set may also be used to manage vehicles at a signalized intersection as well. One example of the application of unsignalized intersection control elements at a signalized intersection would be governing the behavior of vehicles making a right turn on red.

Stop Signs

These basic control devices cause vehicles in VISSIM to come to a stop before proceeding further along their current link they are presently traveling on. Stop controls do not explicitly cause vehicles to yield to traffic on cross movements; instead, this function falls on priority rules and conflict areas. Stop signs can also be defined in a way so that they are only active when a particular phase of a signal is red in order to allow for the modeling of the stop portion of right turn on red behavior.

Priority Rules/Conflict Areas

Typically VISSIM vehicles on a link or connector are ambivalent to vehicles on other links or connectors, even if the links or connectors are coincident. This allows for the modeling of grade-separated crossings as one might find at a freeway interchange or similar. However, when modeling intersections it is desirable to have vehicles consider the conflicting movements of vehicles on other links. To accomplish this VISSIM provides priority rules and conflict areas.

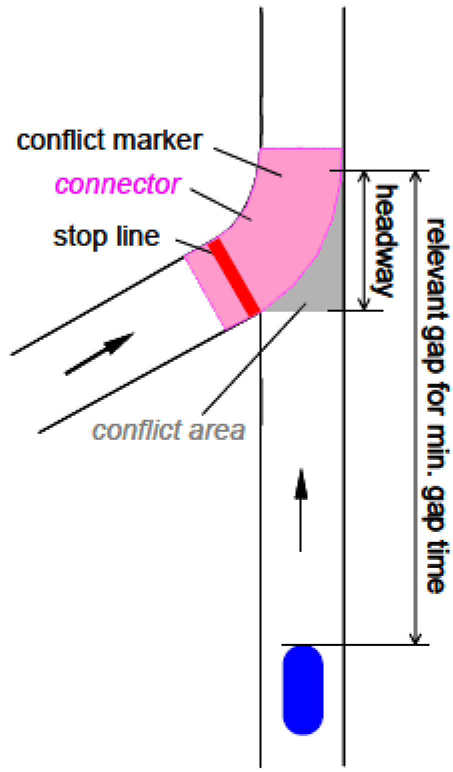

Figure 3. VISSIM Conflict Areas
(Figure Credit: VISSIM 4.30 User Manual [1])

Functionally equivalent to the yield control, priority rules have two elements. The first is the stop line and the second is the conflict area both shown in Figure 3 [1]. Priority rules cause vehicles on the minor approach to yield to vehicles traveling on the major approach (i.e. the approach to which priority is given). The vehicle approaching the stop bar on the minor approach will consider the conditions on the major approach and determine if there is an acceptable gap for entry into the traffic stream on the major approach. If not, the vehicle will stop at the stop line and wait for an acceptable gap. These network elements are generally used to model right turns on red and all conflicting movements at un-signalized junctions. Priority rules can also be defined in a way that they only become active when a particular signal group is red, allowing them to be useful in preventing queues from spilling back and blocking intersections. This is known as a “Keep Clear Area”, the functionality of which is demonstrated in Figure 4 [1].

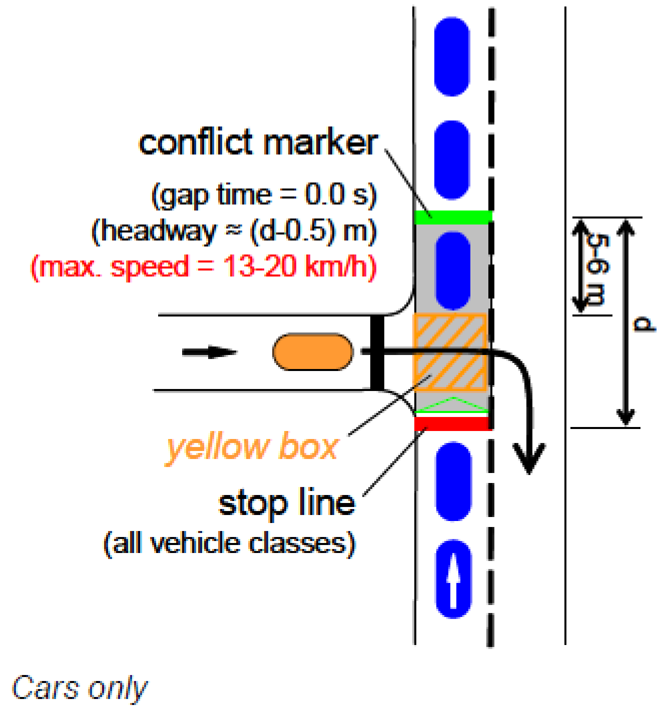

Figure 4. VISSIM Conflict "Keep Clear" Area
(Figure Credit: VISSIM 4.30 User Manual [1])

Though powerful, priority rules are complicated and require a great deal of effort to code correctly. Conflict areas were added to VISSIM to ease the process of defining conflicting movements. Conflict areas intelligently identify when two links or connectors overlap and allow the modeler to define the desired behavior (i.e. which link should yield). Logic is built into VISSIM that causes vehicles approaching a defined conflict area to plan their movement through the conflict area depending on the prevailing traffic conditions [1]. Most of the functionality of priority rules is encapsulated in the functionality of conflict areas, but the selection of the configuration parameters is handled internally making the definition of conflicting movements easier. Despite easing the network coding process significantly, there is still a need for both conflict areas and priority rules as some situations are too complex to be accurately modeled with conflict areas.

Signalized Control

VISSIM provides for many types of signalized control. Signalized control is accomplished in the traffic flow module through the use of signal heads and detectors which interact with the traffic control module.

Signal Heads

Signal heads are the representations of the signals in the VISSIM model. Unlike in the real world, signal heads are located on the same side of the intersection as the traffic they are intended to control (i.e. at the stop bar). Signal heads belong to signal groups which are usually associated with a particular phase in the traffic control module. One signal head may belong to multiple signal groups to allow for the implementation of protected/permissive phasing. Signal heads can also be defined so that they only apply to a specific class of vehicles. This allows for the definition of complex traffic control systems. One example might make use of separate signal heads for transit vehicles and regular traffic.

Detectors

Detectors provide feedback for the traffic control module. They can be of any length, but are defined separately for each lane of a given connector or link. However, detectors with the same channel number are considered to be the same detector, thus making it possible to define one detection zone for an entire multilane movement. Much like signal heads, detectors can be defined to only interact with particular vehicle types, which is useful in the modeling of complex control mechanisms as exemplified earlier. VISSIM implements detection such that when a vehicle's front end enters the detector an impulse is sent to the controller; a separate impulse is sent when the vehicle's rear end exits the detector. This leaves the specific handling of the detector actuation up to the control emulator [1, 2].

Signal Controllers

While VISSIM provides for a wide variety of signal control systems, discussion will be limited to the three most helpful to this study: Fixed, NEMA, and VAP. Fixed time controllers are the most straightforward and provide a simple progression of the defined phases based on the selected intervals, there is no consideration of detector data or actuations in this control system. NEMA control is an external control system that allows for semi-actuated control and fully actuated control that emulates a standard NEMA controller. VAP is a protocol that allows for VISSIM to interact with external programs including custom control emulators [1, 3].

Input Base Data

Driving Behavior

"The traffic flow model in VISSIM is a discrete, stochastic, time step based, microscopic model with driver-vehicle-units as single entities" [1]. The model used in VISSIM is based on the work of Wiedmann who proposed a vehicle following model that asserts drivers can be described as being in one of the following modes [1]:

- Free Driving
- Approaching
- Following
- Braking

VISSIM provides many driver behavior parameters that can be used to alter the behavior of the simulated drivers in these four categories. The definition of these parameters has been shown to have a significant effect on the model performance [4, 5]. Additionally, VISSIM provides inputs to control the behavior of drivers in lane changing, lateral behavior, and signal control. The details of these parameters are discussed extensively in other studies [4-6] and will be omitted here for brevity.

Vehicles

VISSIM provides a hierarchical organizational structure for the classification and grouping of vehicles as follows:

- Vehicle Type – This is a group of vehicles with similar characteristics. This could be defined as cars, trucks, busses and so on. Vehicles with the same type share similar driving behavior and technical characteristics such as acceleration and deceleration functions. In VISSIM, acceleration and deceleration are defined as functions to capture the stochastic nature of vehicle performance present in the real world.

- Vehicle Class – Vehicle classes are the organizational structure by which vehicles are differentiated by the elements of the VISSIM network. For example, if a detector was intended to only recognize transit busses, those busses would need to be placed in a separate class from other vehicles.
- Vehicle Category – Vehicle types used to define a group of vehicles that have a similar interaction with the network.

Figure 5 demonstrates the framework described above.

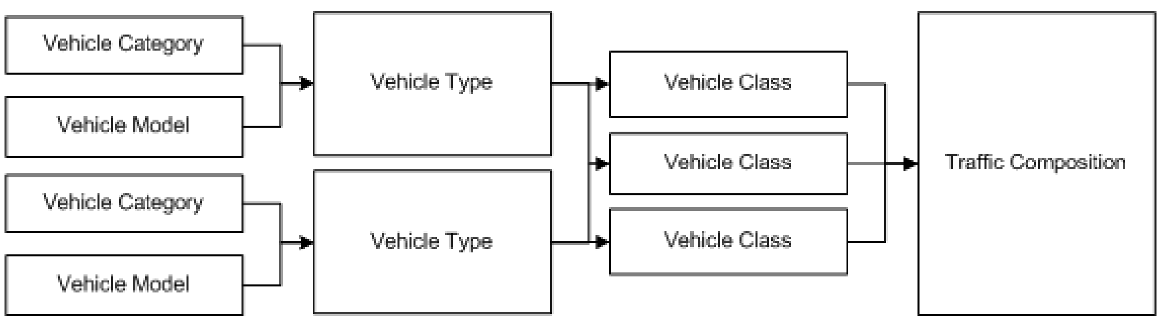

Figure 5. VISSIM Vehicle Definition Framework

Distributions

VISSIM incorporates the stochastic nature of many phenomena seen in real world traffic systems through the use of distributions. A distribution allows for some variation in the input values for a simulation model. It is important to note, though, that the distribution implementation is dependent on the random seed value. This means that for the same random seed value, a model will produce the exact same results from a given distribution. The following are the distributions relevant to this project:

- Desired Speed Distribution – This is used to select the desired speed of vehicles when they initially enter the network or when they pass over a desired speed decision. This distribution can have a significant effect on the calculation of delay in the model as the delay value is calculated based on the driver's desired speed versus their actual speed. This means that if the band of potential desired speeds is sufficiently wide delay can be artificially increased and conversely, if the band is narrow the delay can be artificially decreased. For this reason internal VISSIM delay is not used in this study and the delay is calculated separately based on travel times as detailed in later sections covering the evaluation process.
  Color Distribution – This defines a color distribution for a particular vehicle type. While this has no impact on the simulation it does allow for some visual variation in the color of vehicles in the network. In this simulation different colors are used to differentiate the purpose of vehicles in the model:
    - Blue: Base Traffic
    - Orange: Probe Vehicles
    - Green: Scenario Traffic
- Vehicle Model Distribution – Determines the 2D and 3D model used to represent the vehicle and can impact the simulation based on the vehicle's dimensions.
- Dwell Time Distribution – This defines the range of times that a vehicle will spend stopped at parking lots, stop signs, and transit stops. The distribution may be normal or empirical. A normal distribution with a standard deviation of 0

functions as a constant dwell time. The empirical distribution allows for the creation of a graph similar to the speed distribution, thereby generating any desired distribution.

- Traffic Composition –The composition of traffic defines the mix of vehicle types created for each vehicle input. This could be used to model a traffic flow with 95% cars and 5% heavy vehicles or similar.

Output Data

Performance Measures

Because VISSIM allows for complex network modeling, the evaluation tools provided can be somewhat more complicated than those. VISSIM provides a wide range of model performance evaluation and data collection functions, including travel times, queue lengths, delay, lane change records, signal change records and many more. Through the proper placement and configuration of these analysis measures, the traditional transportation metrics can be obtained. Also, there are several non-standard analysis methods that can be employed to gain a better understanding of model performance. This study will review data outputs available in VISSIM that can be collectively used to gather data from a simulation run to make important conclusions about model performance.

Vehicle Record

This is the most basic data output VISSIM provides. At specified times this data collection function will output data about every vehicle in the model including, but not limited to, location, speed, acceleration, and routing information. The data recorded by this output function could potentially be used to reconstruct any of the other metrics, but performance limitations prevent that level of data collection from being efficient. Typically, when this feature is used, a filter is defined to ensure that the function only records data for a specific class of vehicle (transit buses for example) to reduce the data collection overhead. One benefit of the vehicle record output is that it does not require the placement of any network elements to collect data.

Travel Times

Travel time segments are defined in the model with a simple two-point start/end structure. VISSIM can record the travel time for each vehicle traveling along the segment individually or aggregate the data into user-defined bins. When recording the output data, VISSIM also provides the length of the segment so that delay can be manually calculated based on the speed limit(s) on the segment. Typically, segments are defined for each movement at each intersection to get movement and intersection LOS. Segments are also often defined for the entire length of a study corridor to get mainline travel time.

Data Collection Points

Data collection points allow for the gathering of data at a particular cross section in the model, as opposed to a section as with travel time segments. This is particularly helpful for gathering count data for the various movements at an intersection but is not limited to that. Data collection points can also record other vehicle attributes like speed and occupancy. While data collection points must be defined individually for each lane on a link or connector, they can be grouped together to combine the data. One example of this would be to obtain a movement count on a multi-lane approach.

Queue Counters

Queue counters record the average and maximum queue length from the point of definition back along all links on which the queue extends. The longest queue is reported as the maximum queue length regardless of what link the on which the final vehicle is contained. The number of stops within the queue is also recorded. The queue length is monitored as long as the last vehicle in the queue is still in the queue condition, even if some of the vehicles further ahead no longer meet the queue condition [1].

<u>Output Format</u>

VISSIM provides three types of data output:

- Window Representation
- Text Files
- Database Tables

While graphical representation in windows is helpful when directly observing the model and troubleshooting problems, it is not useful when conducting replicate runs to generate data for statistical analysis. The majority of the data collection is done with flat text files and database tables. Text files are the default output and the easiest to implement for experiment runs as long as the volume of data collected is not too great. These files are created in the same directory as the VISSIM network file when the simulation is run unless otherwise specified. VISSIM also supports connecting to a database (Microsoft Office and Microsoft SQL Server are two examples) when the data collection task is too great for flat text files. In both cases one must use care when conducting replicate runs as VISSIM will overwrite previous output files or database tables when a new run is started.

Therefore it is important to archive the results files and data after each run to ensure that the recorded data is not lost.

<u>Error Files</u>

During the course of a simulation run VISSIM will record any warnings or minor errors to a flat text file in the same way that it records simulation performance measures to text files. These warnings might include notices about routing decisions that have been placed too close to downstream connectors, vehicle removal from the network for various reasons, or the failure of a defined input to produce all its vehicles during the simulation due to link congestion. These messages often indicate problems with the model that need to be addressed and so it is important to archive the error file (if produced) along with the network performance data for later analysis when conducting simulation runs.

It bears mention here that effective simulation requires far more than the information briefly detailed here. Validation and calibration procedures detailed later in the paper are critical to ensure that the model is behaving as expected and consistent with real-world data. Furthermore, this overview merely touches the surface of the VISSIM software. For further information and details on the functionality available see [1, 2, 7]

HILS Overview

Hardware-in-the-Loop Simulation (HILS) involves creating a hybrid system that combines real-world elements with simulated elements to conduct analysis not possible with a purely real-world or purely simulated environment. In traffic engineering the concept of HILS typically involves combining a microscopic traffic simulation program with a real-world Intelligent Transportation System (ITS). In assessing the effectiveness of the system, simulation is often preferable to field-testing for the evaluation of ITS because it allows for controlled experiments to be conducted and statistically strong conclusions to be drawn [8-10]. Simulation also allows for the evaluation of scenarios that would not be possible in the field such as accident lane closures or traffic growth patterns. Traffic simulations rely on the software emulation of real-world objects including vehicles, roadway geometry, and traffic controllers. This approach works well for many applications but faces two problems when used to evaluate cutting-edge or proprietary ITS implementations. First, no software emulation of the ITS in question may be available; second, the software emulation of the ITS, if available, is not guaranteed to be an accurate representation of its real-world counterpart. HILS can overcome both of these issues by enabling the ITS to interact with the simulation model, thereby gaining the benefit of simulation analysis while retaining the full functionality of the real-world system [11, 12].

VISSIM is an ideal candidate for HILS because of its structure for interaction between the traffic flow model and the traffic control module shown in Figure 6. As mentioned in the previous section, VISSIM allows for custom control modules to be utilized providing an avenue for convenient communication between VISSIM and a real-world traffic controller. Figure 1presents a modification of Figure 6 to demonstrate this new arrangement of VISSIM when making use of HILS.

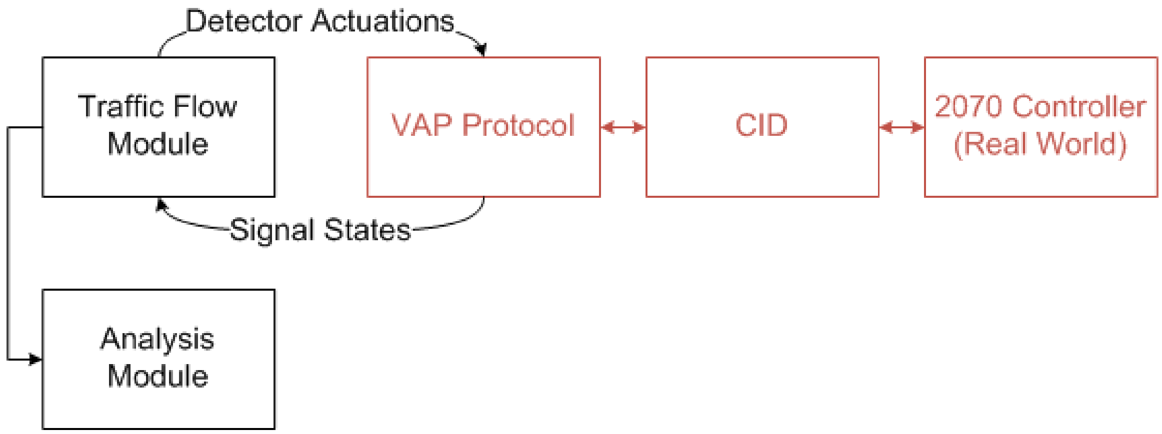

Figure 6. VISSIM-CID Program Structure

In this implementation all corridor geometry, vehicle traffic, detectors and signal heads are coded into the simulation model. The usual internal logic for signal control is bypassed and detector actuations from vehicles in the model are sent out to the real-world controller. Meanwhile signal state information is accepted by the model from the real-world controller. Between the model and the real-world controller is a hardware device that facilitates the communication between the two, called a Controller Interface Device (CID).

It must be noted that while the HILS architecture provides a powerful analysis framework for the evaluation of ITS strategies paired with simulation models there are still some drawbacks. Any inaccuracies in the simulation model can have a significant effect on the performance of the "real-world" ITS which is typically subject to, and designed for, implementation in the real world.

While the overview of the HILS as described seems rather straightforward, the actual implementation is somewhat more complicated. The complexity is further compounded when implementing HILS on a larger scale as in this study. The problems encountered and their resolutions are detailed in later sections.

Conclusion

VISSIM is a helpful traffic analysis tool flexible enough for the modeling and evaluation of complex geometry. Because of this flexibility, VISSIM also contains a fair amount of complexity and requires careful implementation to ensure that the results gained from the modeling effort are consistent with the real-world. This underlying complexity in the model becomes even more important when VISSIM is used in HILS as inconsistencies in the model can potentially cause significant inconsistencies in the performance of the real-world hardware. In the next chapter relevant documentation about the proper implementation and analysis of transportation models in VISSIM will be reviewed.

CHAPTER 3

LITERATURE REVIEW

The establishment of a framework for the evaluation of advanced traffic control systems through the use of hardware-in-the-loop (HILS) requires a wide breadth of knowledge. This is not only limited to the implementation of HILS, but also regarding the setup, validation, and calibration of simulation models, and the effective analysis of their output. Many studies have been published on each of these topics but few have been done on the complete end-to-end application of the best practices for simulation. Many of the studies tend to focus on one of the specific steps or on the general implementation of a simulation model without specific concern for the important steps in the proper construction and evaluation of that model. The studies below have been categorized where appropriate in a rough chronological order in terms of model development.

Validation and Calibration

Statistical Validation of Traffic Simulation Models (Toledo, T. and H.N. Koutsopoulos, 2004) [13]

This article lays out a procedure for conducting statistically rigorous analysis of the performance of simulation models as they relate to the real world. The validation procedure is identified as taking two steps. First, the validation procedure calls for the initial estimation of the individual models (route choice, driving behavior) that serve as components of the larger simulation model. The second step involves calibration and validation of the simulation as a whole, based on the results of the first step.

When it comes to validation there are two potential methods that are used, visual and statistical [13]. The visual analysis involves the graphical representation of model performance through charts and graphs, as well as direct observation of the model's operation. These simulated observations are then compared to the real world to determine the fitness of the models performance. Statistical analysis is far more objective and involves the application of various tests to determine model fitness [8-10, 13]. The article proposes a two-level statistical test for model validation, one that compares the respective means and another that compares the distributions of the selected measures of performance (MOP).

The importance of the generation of inputs is discussed [8, 13]. When recording data from the field for comparison against the model, the inputs must be recorded in addition to the outputs. This enables the simulated model to be run under the same conditions as the field and allows for effective analysis of the selected MOP. The choice of MOP is important and based on the context of application, independence of the measure, error sources, traffic dynamics and most importantly the level of effort required for data collection [8-10, 13]. Often insufficient time or budgets limit the collection of some of the more expensive measures of performance in practice even though they are theoretically the most effective.

The statistical validation of the model as defined in the article takes three approaches [13]:

1. Goodness-of-Fit Measures
2. Hypothesis Testing and Confidence Intervals
3. Test of Underlying Structure

While the first two are fairly straightforward in regards to simulation statistical analysis, the third is more unique. The test of underlying structure is a method of analysis that can get around the common limitations of collected data by creating and comparing meta-models that represent the underlying relationships between various traffic characteristics [13].

For a case study the article demonstrates the application of both traditional statistical tests and the creation and evaluation of meta-models to evaluate a model of the M27 Freeway in Southampton, England. The paper shows that while the statistical tests are effective, they are unable to define the ability of the model to reproduce the same traffic behavior in the field. The power of meta-models in this context is then effectively demonstrated using the Pipes-Munjal model [13].

A Framework for the Calibration of Microscopic Traffic Flow Models (Cuffo, B.F,, V. Punzo, and V.Torrieri, 2007) [14]

This paper, among others, proposes a two-phase model calibration approach based on parameter selection and calibration. The first step is the determination and setup of the calibration procedure. The second step demonstrates the application of the previously determined procedure in the calibration of an example simulation model [5, 6, 14-16]. In the first step, measures of performance (MOP) on which the model will be evaluated during the calibration phase are established. Based on these measures, the optimization function can then be determined, and the likelihood of calibrating the model can be evaluated.

The MOP selection is based largely on the context of the model and the data available to the researchers [5, 14]. The measures of performance chosen will have a significant impact on the model evaluation as integral parts of the objective function, or the function by which the simulation is optimized.

Also, the simulation parameters must be evaluated to determine which parameters have the greatest control over the variance in the model performance. This “optimal” set of parameters is best determined using a sensitivity analysis. The recommended sensitivity analysis procedure in this case, and several others, is ANOVA [5, 6, 13-16].

The next step is to determine the optimization algorithm and the objective function for use in the calibration phase. The objective function serves as a measure of the difference between the simulated MOP and the observed MOP [14]. In the calibration phase the simulation is run repeatedly in a series of steps, each with different parameters. The optimization algorithm is run at each step in the calibration process and makes use of the objective function to evaluate the performance of the model for that particular set of parameters. Based on the results, the optimization algorithm then determines the set of parameters for use in the next step of the simulation process.

The presented procedure then proposes a test to determine the effectiveness of the calibration procedure, the sensitivity analysis, and the optimization algorithm using laboratory data. In this case a baseline is established as the objective, the parameter values are randomized and the calibration is run. The calibration framework is considered good if objective function and the optimization algorithm lead back to the baseline. If the framework holds up after the laboratory testing, it can be similarly applied to calibrate the model using properly validated real-world data as the baseline [14].

The article demonstrates the previously detailed procedure through the calibration of an AIMSUN 5 model of Freeway E32 from Naples to Salerno in southern Italy. The

MOP's were defined as the traffic counts and speed as this data was readily available from the field. It was determined using ANOVA that for the speed MOP, 94% of the variance could be explained by the following parameters [14]:

1. Reaction Time
2. Max Desired Speed
3. Max Acceleration

Similarly for the count MOP, 94% of the variance was described by [14]:

1. Reaction Time
2. Max Desired Speed

Because the two parameter sets overlap, only one MOP needs to be used in the objective function, which will simplify the optimization procedure. The optimization was performed using a computer software package called LINDO API. The results were promising, but not conclusive, as the provided field data was not clean enough to facilitate effective calibration [14]. However even using inconsistent field data, the calibration procedure demonstrated some effectiveness at reaching the optimum values for the critical parameters. Further analysis using validated data will be required to confirm this result.

A Genetic Algorithm-Based Approach to the Calibration of VISSIM Using GPS Data (Yu, L., X. Li, and W. Zhuo, 2004) [16]

Traffic simulations, VISSIM in this particular case, have a large number of driver behavior parameters that need to be properly calibrated before any particular model can be used to evaluate the traffic conditions for a real-world setting. This article, in agreement with several others, demonstrates that the use of the default or average values for the parameters can result in large errors in analysis, and shows how extensive calibration against field data is required to ensure that the model accurately represents field conditions [5, 6, 8, 14-17].

Because VISSIM has so many driving behavior parameters, many of which have a wide range of possible values, it would be computationally prohibitive to evaluate every possible combination. The use of Genetic Algorithm (GA) is proposed as an effective solution for searching the solution space [5, 6, 15, 16, 18]. Genetic Algorithm provides a parameter value selection process that, much like real-world natural selection, seeks to preserve the best values from step to step while seeking to optimize the defined measure of performance. In the case presented in the paper, a GPS-equipped vehicle was used to collect average speed data on the road network around the Intercontinental Airport of Houston (IAH), which was then compared with the instantaneous speeds recorded at various points within the VISSIM model [16].

The model was developed using geometric and traffic data collected in the field. A computer program was likewise developed to automatically run the VISSIM model and generate velocity data. This data could then be used to calculate the Sum of Squared Error for each of the velocity cross-sections within VISSIM compared with their real-world counterpart. This data was then evaluated using the Genetic Algorithm Toolbox within MATLAB, and the next step (or generation as defined in GA nomenclature) of the calibration process was defined. The results of 20 generations showed that the SSE for the model was decreased my nearly 50% [16].

Development and Evaluation of a Calibration and Validation Procedure for Microscopic Simulation Models (Park, B., and J. Won, 2006) [5]

In this paper the authors laid out a framework for the calibration of simulation models using a rigorous statistical method to demonstrate its effectiveness in multiple scenarios with two case studies using multiple simulation tools. The process involves the following steps consistent with those in several other papers shown in Figure 7 [5, 6, 15].

The procedure selects the relevant parameters using ANOVA and a Latin Hypercube Design algorithm in MATLAB to reduce the total number of parameters to the set that could be calibrated in a reasonable amount of time while still covering the parameter surface [5]. The parameters are then calibrated with a Genetic Algorithm procedure based on a selected measure of effectiveness [5, 6, 15, 16]. The effectiveness of the procedure was then demonstrated in two case studies. The first involved a signalized intersection, while the second centered on a freeway segment. The importance of input data consistency was stressed as the calibration process would be worthless without proper reference data, no matter how sound the procedure. In both cases the GA procedure was able to calibrate the model to within a reasonable and statistically significant range of the values measured in the field [5].

In addition to the presentation of the calibration procedure, the paper presents a detailed analysis of the features and limitations of various traffic simulation software packages. CORSIM, VISSIM and PARAMICS are all evaluated. The important distinction for all models in regards to calibration is that they each possess the facilities for remote programmatic control, a requirement for any type of intensive software-based calibration [5].

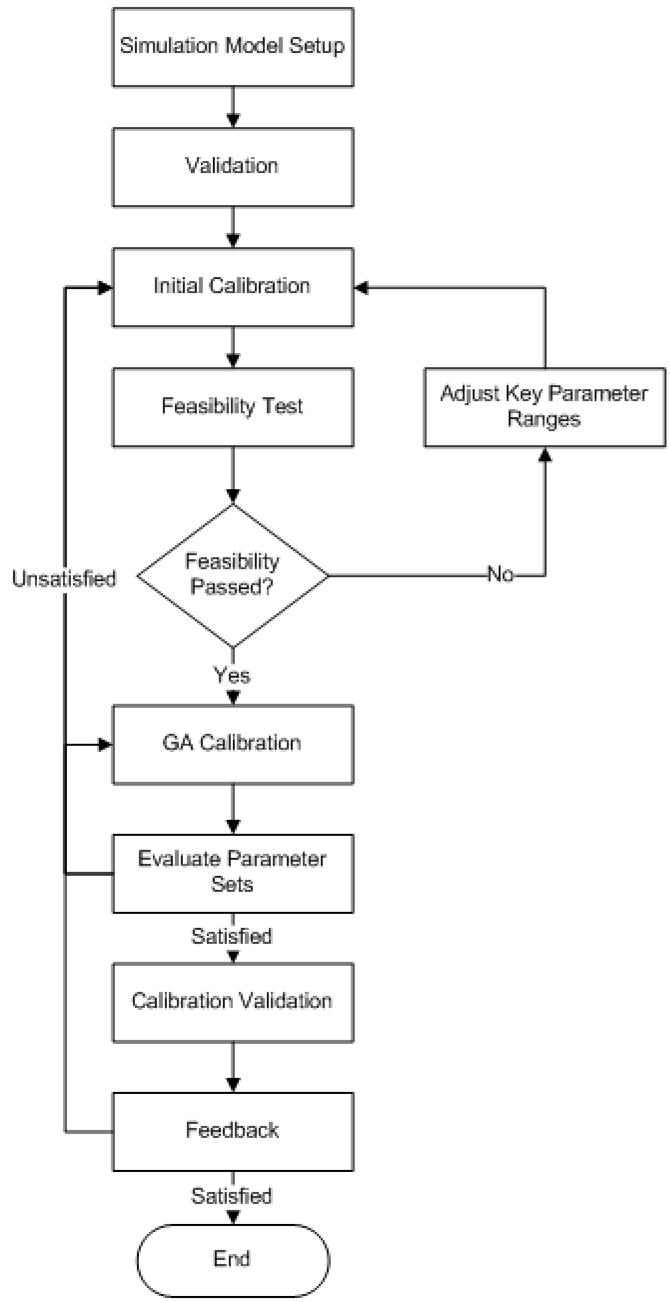

Figure 7. Genetic Algorithm Calibration Procedure

(Figure Credit: Development and Evaluation of a Calibration and Validation Procedure for Microscopic Simulation Models [5])

Some Topics for Simulation Optimization (Fu, M.C., C.H.Chen, and L. Shi, 2008) [19]

A challenge with simulation optimization is that a trade-off always exists between the allocation of computing resources for searching the solution space and evaluating the fitness of a particular solution using simulation [5, 14, 16, 19]. This paper proposes three methods to address this trade-off when running simulations to search for an optimal solution for a given system. The first method is optimal computing budget allocation which involves the allocation of resources (simulation runs) based on the initially demonstrated potential of a model based on variance and mean values. The second method is stochastic gradient estimation, which involves a local gradient search based on a particular performance measure. The final method uses nested partitions whereby the sample space is divided into separate partitions and analyzed with the most effective partition being further divided and analyzed recursively until an acceptable solution is reached. These methods are each detailed and demonstrated for further consideration in simulation optimization [19].

Distributed/Federated Simulation Framework

Large-Scale Traffic Simulation through Distributed Computing of Paramics (Liu, H.X., et al., 2004) [20]

The limitations of existing hardware performance and effects on the micro simulation of larger networks are discussed in this article. One possible solution is demonstrated through distributed computing, or the process of breaking large models up into smaller networks that can be run semi-independently on several machines at much greater speed than running the entire network at once [20]. The three primary benefits are demonstrated as (1) the estimated time taken for computing hardware to catch up with the desired large-scale model size is more than one decade; (2) low to mid-range hardware systems are cheaper and more readily available than high end systems; and (3) the development of a framework for distributed simulation will also contribute to more effective simulation data management [20].

The overall organization of the distributed system involves a client-server architecture, where one master computer controls various clients, each running a portion of the overall network [20]. Various methods of control are discussed, but the implementation revolves around a light global control/independent subnets framework to reduce overall communications load and efficiency. The synchronizing mechanism for vehicles is detailed and a load balancing concept proposed to ensure that the faster computers in the distributed environment carry more of the simulation workload for effective resource utilization. Two case studies demonstrate the effectiveness of the proposed framework for corridors and large interconnected networks [20].

A Hybrid Mesoscopic – Microscopic Traffic Simulation Mode: Design, Implementation and Computational Analysis (Ziliaskopoulos, A., J. Zhang, and H. Shi, 2006) [21]

The need for large scale simulation can also be met using federations of mesoscopic models with microscopic models where the efficiency of mesoscopic models can be used for large scale simulation while still maintaining the rich data collection and analysis techniques available in microscopic models [21]. This paper proposes a framework for the marriage of these two model types and details the potential problems posed in the combination. The main issues are the transfer of vehicles from one model to another and the establishment of boundary conditions that ensure that the effects of congestion are properly propagated across the interfaces [20, 21]. This not only involves the development and spillback of queues, but also the movement of shock waves within the traffic stream. The boundary conditions are defined as (1) the available free space on the downstream link and (2) the number of vehicle seeking to move across the interface [21]. Important to note in the development of a federated simulation is the difference in update time period between the two models [20, 21]. While the microscopic model is updated every second, the mesoscopic model is only updated every 6 seconds. This makes the transfer of vehicles and traffic conditions from one model to another more difficult. A solution for the effective implementation of these conditions within a federated model are detailed and demonstrated in a simple abstract case study [21].

General Simulation Development

Framework for Building Intelligent Simulation Models of Construction Operations (Mohamed, Y. and S.M. Abourizk, 2005) [22]

As the effectiveness of simulation as an analysis tool increases, the development of simulation models becomes more important in many different fields of study. Many of the intended users of simulation models are not skilled in the high level programming that is required to effectively build and revise many simulation models [22]. In response to this problem a framework for developing a simulation model for construction operations is proposed in this paper. The objectives of this framework are as follows [22]:

- Functional View
- SPS Representation (Modeling elements represent their real-world counterparts.)
- Autonomy

- Flexible Topological Changes

To accomplish the first two the framework divides the simulation development process into two levels. The first level is designed for non-programmers and allows experts in the field in which the simulation is being used for analysis to visually construct simulations using blocks that appear familiar to them. The second level is targeted at users with more technical knowledge and exposes the underlying framework of the simulation process for the development and modification of the complex features that support the operation of the front end objects presented on the first level [22].

To address the second two objectives the framework proposes the development of a simulation management agent. This agent is an element of the simulation model that evaluates various performance measures and makes logical and functional changes to the model to improve efficiency. In this manner the simulation can capture the impacts of on-the-spot operational decisions often made in construction [22]. One example might be the number of dump trucks employed in an earth moving operation which varies based on the wait time of the loader at the excavation site.

The effectiveness of the framework is demonstrated in a case study of a tunneling operation. The agent-based framework proposed in the paper is able to make functional adjustments and arrives at the optimal result much faster than the traditional simulation model [22].

Some Guidelines for Selecting Microsimulation Models for Interchange Traffic Operational Analysis (Fang, F.C. and L. Elefteriadou, 2005) [23]

While many simulation packages are available for use in traffic simulation the underlying functions required for effective traffic simulation are relatively constant. The features of a particular simulation system can be split into two categories: input and output. For input, a simulation package needs to provide functionality to manipulate and control the (1) geometric characteristics; (2) signal control; (3) traffic; and (4) any other additional features that affect simulation performance such as driver behavior [8, 23]. Output is discussed in terms of performance measures which exclusively center on control delay in this study [23]. A case study demonstrates the use of these functions in the development of a model and also demonstrates the importance of the calibration procedure on model performance.

Object Oriented Methodology Based on UML for Urban Traffic System Modeling (Chabrol, M. and D. Sarramia, 2000) [24]

A new framework for the development of traffic models is proposed that uses a flexible, object-oriented approach to represent the various network elements and the interaction between them. A traffic model is presented as two separate representations, the knowledge model that functions as a natural or graphic representation and the action model that translates the knowledge model into a formal mathematical representation [24]. UML is used to describe the traffic system as a product of three separate subsystems (1) the logical system; (2) the physical system; and (3) the decision system [24]. Each subsystem contains various parts of the overall model. For example the logical system contains the moving vehicles, the physical system represents the geometric alignment of the links and the decision system contains the control and routing logic. These subsystems are defined for the particular scenario under consideration and then communicate with each other to execute the simulation and provide the results for analysis [24].

Output Visualization

Development of Interactive Visualization Tool for Effective Presentation of Traffic Impacts to Nonexperts (Prevedouros, P., D. Brauer, and R.J. Sykes, 1994) [25]

It is important that every enterprise employ effective marketing and transportation planning is no exception. In this paper the authors detail a system designed to effectively communicate technical details to non-experts in a variety of settings [22, 25]. Often the reports and data generated by engineers is highly effective in describing the optimal solution to a given problem, but cannot be easily conveyed to decision makers or the public. Decision makers may have some expertise, but lack the time required to assimilate all the engineering data, and cannot make use of the raw data effectively in the political realm. Member of the public often lack the technical proficiency required to assimilate the data and put it into a context that would allow them to contribute to the planning process [25]. The interactive visualization of traffic impacts (IVTI) concept detailed in the paper solves this problem by integrating the technical data into a visual framework that is easily used by non-experts. IVTI integrates the work of several experts including public affairs officers, traffic engineers, and others. The result is a system that can easily be displayed at meetings or run on a kiosk for individual interaction. The visuals integrate the engineering data with multimedia voice, video, and picture data that helps contextualize the data in a way that everyone can understand [25].

Three Dimensional Transportation Analysis: Planning and Design (Easa, S.M., et al., 2002) [26]

The development of 3-D transportation models and analysis tools has drastically altered and improved the methods used to conduct simulation and analysis for transportation planning and facility design. 3-D means different things for different analysts. In engineering it involves the level of mathematics that must be used in analytical models while in planning it may refer more to the rendering of the environment in visualization models [26]. The use of 3-D data in transportation facility engineering improves route planning, geometric design and structural design by providing data allowing for the more accurate calculation of sight distance, elevation change, and other factors. Similarly, in the discipline of transportation planning 3-D data impacts system planning, corridor/project planning and environmental analysis [25, 26]. 3-D data allows planners to more effectively present data to the public and visually demonstrate the effects that changes will have on both the built environment and the social or demographic environment. The use of the data also informs the analytical equations used to calculate various aspects of system performance. For example, the calculation of particulate matter transport based on the prevailing weather conditions and the surrounding topography [26].

Output Analysis

Introduction to Simulation (Goldsman, D. and G. Tokol, 2000) [8]

This paper, consistent with several other studies, presents an overview of simulation development from formulation of random numbers and the input variables to analysis of the simulation output. It is the stochastic nature of these simulations that requires an extensive discussion of those topics [8-10, 27-29].

The generation of random variables is critical to the development of effective input data for simulation models. It is shown that most common distributions can be derived from uniform(0,1) numbers through a variety of transforms including Inverse Transform, Box-Muller, Central Limit, Convolution and Acceptance-Rejection [8]. Determining the correct input distribution is critical, as an incorrectly configured input could render all the simulation data unusable for analysis. A framework is proposed for the analysis of input data including the following steps [8]:

- Determine what input variables are required.
- Collect as much real-world data as possible.
- Make reasonable assumptions about the nature of the data as needed.
- Perform statistically rigorous goodness-of-fit tests on the data.
- Try to determine if more complex approaches are warranted.

The output analysis also poses some challenges as simulation data is typically not independent, identically distributed or normally distributed [8-10, 27-29]. Seeing this it is difficult to apply traditional statistical analysis techniques, and so other manipulative techniques are needed to analyze the raw data provided from simulations. Analysis for terminating and non-terminating simulation models is proposed, but the focus of this study is on the non-terminating simulation analysis portion as that is most relevant to the present work. Non-terminating simulations often suffer from initialization bias which must be accounted for, or eliminated, before effective output analysis can be conducted. This can be done though data truncation or by running the model for a sufficient time period to ensure that the initialization inconsistencies are not significant when compared with the rest of the data [8, 9, 27].

Two methods of parameter estimation for continuous simulations are proposed. Each is designed to account for the typical interdependence of many performance measures. The first method proposed is called batch means [8, 9, 27, 30]. It involves dividing the simulation run into several smaller time periods and calculating the value of the performance measure as an average across each of those smaller periods. The second method is called independent replications and involves conducting several sequential runs to generate values of the performance measure and re-starting the simulation between each run [8, 9, 27]. The comparison of two systems is discussed. Several methods are proposed including classical confidence intervals, common random numbers, antithetic random numbers and selecting the best system. Methods for obtaining the variance for the sample mean and confidence intervals are also detailed. These include spectral estimation, regeneration and standardized time series [8].

Advanced Output Analysis for Simulation (Seila, A.F., 1992) [28]

This paper expands on previous work in the area of simulation output analysis by detailing some alternative methods for the analysis of simulation data. These include the sequential methods, standardized time series, methods for estimating quantiles and, multivariate estimation [28]. The sequential methods procedure stipulates that the analyst specify the desired level of accuracy sought from the simulation run, and then the procedure runs the simulation until that specified accuracy is reached. The standardized time series method shows that if the process is allowed to run for a sufficient period of time, the present performance measures will be functionally independent of the past values of that measure. Methods for estimating quantiles details some of the procedures used to estimate these values, something that is typically difficult to do for a non-terminating simulation. Multivariate estimation demonstrates that if more than one variable is estimated, the confidence interval will be a product of the confidence intervals for each of the parameters to be estimated [28].

To Batch or Not to Batch (Alexopoulos, C and D. Goldsman, 2003) [30]

This article discusses the logic behind the selection between one of two data analysis methods for non-terminating simulations. Batch means (BM) involves separating the simulation run into several discrete increments and calculating the value of the performance measure over the course of each of those periods [8, 30]. Independent replication (IR) collects several samples for the performance measure by running the simulation several times [8, 30]. There are inherent drawbacks to each approach. The IR method suffers from significant initialization errors for transient conditions. BM, however, suffers from a lack of true independence between measurements. The results of the analysis show that the estimators for the mean and standard deviation of the sample population are practically equivalent for both BM and IR given a large enough sample size. However for smaller samples IR is more effective for steady-state evaluation and BM is more effective when used in the presence of an initial transient [30].

Nonparametric Techniques in Simulation Analysis: A Tutorial (Ycuesan, E., 1994) [29]

This paper details the weakness of traditional statistical analysis of simulation results and proposes alternative analysis methods that do not attach "extraneous, but analytically convenient, conditions to the null hypothesis" [29]. These conditions would include the assumption of normality, independence and common variance required for the use of statistical tests such as the Student's t-Test. Several non-parametric analysis methods are proposed including the Permutation Test, Rank Test and Bootstrapping. While often more computationally intensive than other methods, none of the proposed methods are beyond what could be run on a standard personal computer [29]. And these methods have the added benefit of being distribution agnostic when it comes to the sample under consideration [29]. Applications are presented to demonstrate the flexibility of these tests when used to analyze simulation outputs.

Hardware-in-the-Loop Simulation

Use of Hardware-in-the-Loop Traffic Simulation in a Virtual Environment (Smadi, A, and S. Birst, 2006) [11]

This paper proposes the development of a Controller Interface Device (CID) that can be used to integrate real world controllers into simulation models. The proposed simulation model for integration in this case is VISSIM. The CID is designed to communicate with the simulation over an Ethernet connection which provides location flexibility although speed and performance may become an issue if the connection speed is not sufficient [11]. The VISSIM signal control interface is contained in a single DLL that enables communication over the network with all the CID's for all the controlled intersections in a model. The CID itself operates as a standalone device with its own operating system (FreeDOS) and services that facilitate the communication between VISSIM and the controller [11]. The CID also possesses the power to control the time progression within VISSIM to ensure that the model and the real-world controller remain in sync. This is important as factors on the PC may otherwise lead to slowdowns and cause errors in signal control [11].

US-95 ACS-Lite System Evaluation (Chatila, H. and Z. Li, 2007) [12]

Researchers conducted a small-scale evaluation of the ACS-Lite signal control system on a four intersection test corridor to determine the performance of the system as compared with a traditional time-of-day (TOD) system. Unlike the TOD system, ACS-Lite is able to adapt to the traffic flows making it an effective option for day to day usage and a potential solution for the need to constantly re-time corridors to respond to traffic pattern growth and change in the long run [12]. Four scenarios were developed to demonstrate the performance improvements of ACS-Lite over TOD control. They included [12]:

- Improvement of existing signal timings.
- Correction of poor splits
- Correction of poor offsets.
- Adaption to volume changes.

The evaluation made use of hardware-in-the-loop to attach physical controllers running the signals to a VISSIM model generating the traffic. The performance measure was the mainline corridor travel time and the aggregate travel time including the side streets. In this report the ACS-Lite out-performed the traditional TOD control in all scenarios [12]. Additionally ACS-Lite was compared at a high level with SCATS and SCOOT, two other adaptive control systems on the market. No objective simulation comparison was conducted, but the control systems were compared based on functionality, install base and cost.

The logic and technical requirements of ACS-Lite were also demonstrated. ACS-Lite does not have the ability to modify the cycle lengths, but it can modify phase splits and controller offsets to find the optimal performance configuration [12]. Selection is made based on the “utilization” of each phase which is based on the vehicle to capacity (v/c) ratio. Detectors are used to obtain this data, and ACS-Lite is designed to work with the typical detector configuration present at any fully-actuated intersection. However, if offset optimization is desired then an upstream detector for each of the coordinated approaches must be provided [12].

Summary

While several studies [12, 24, 31] discussed the creation of models for the purpose of evaluating a particular real-world system, few of them endeavored to discuss the details of the validation, calibration, and output analysis process used to generate the conclusions. The only complete guide to simulation development was found [17] However, there is still a lack of specific detail about the best procedures to use in the calibration and evaluation process. Other studies [5, 6, 8, 13-16, 19] extensively covered the process of validation or calibration and showed its application in the modeling process with case studies.

Validation, while not as rigorous a process as calibration or output analysis, should be done consistent with the findings in [13]. It would appear that Genetic Algorithm is a good candidate for use in the calibration procedure as it is relatively easy to implement and generated extremely effective results when applied to a wide variety of models [5]. Output analysis was discussed extensively in [8-10, 27-30] all of which provide helpful advice as to the development of an accurate statistical analysis procedure for the model results. The output analysis procedures will also be important in the other steps in the model development as the analysis components in the validation and calibration procedures should be done consistently with the previously mentioned studies.

Visualization of the generated data is another key area. Research studies [25, 26] show that the development of effective visualization tools is critical both to the effective development of the model as well as the presentation of the findings to other potential non-experts. The development of federated and distributed models was discussed in [20, 21, 31] where some of the pitfalls of multi-system models were discussed. The most prevalent seemed to be the synchronization of data and time across the various systems to ensure that the simulated environment was continuous. This concept was further expanded in the hardware-in-the-loop studies [11, 12] which detail the complications of using HIS in the modeling environment and discuss the potential analysis benefits.

CHAPTER 4

GENERAL PROCEDURE

HILS System Architecture/Implementation

Simulation is a common method of traffic analysis for situations where it would not be feasible to evaluate the scenario or scenarios in question using field data or analytical analysis. Some of these scenarios might be lane or parallel route closures, large events or future growth, and development impacts. Simulation will allow us to evaluate several scenarios that are of interest on a particular corridor, Cobb Parkway in Cobb County, Georgia. However, many times the analysis is on the performance of the signal control system itself, where the usual practice of emulating the controllers with software within the simulation is not feasible. Therefore, it was decided that the simulation model would be implemented using a Hardware in the Loop System (HILS) involving real-world physical signal controllers and associated hardware attached to a computerized simulation model.

In this system, there are two primary connections to consider. The first is a relatively simple connection between the Controller Interface Device (CID) and the 2070 controller which is implemented using the proprietary connection cable provided with the CID. The second connection between the simulation computer and the CID is more complex. The physical connection is implemented with a Universal Serial Bus (USB) connection, requiring a driver to be installed on the computer. This driver software only enables the computer to communicate with the CID, but additional software (CID.exe) is required to facilitate the transfer of simulation data. This software uses the VISSIM VAP protocol to mediate between the CID driver and VISSIM. Figure 8 details this interaction and shows how other software on the computer is designed to interact with the controller. This software includes a suitcase tester emulator and a hardware tester. Both additional tools were useful in troubleshooting efforts carried out in the debugging phase of the system development.

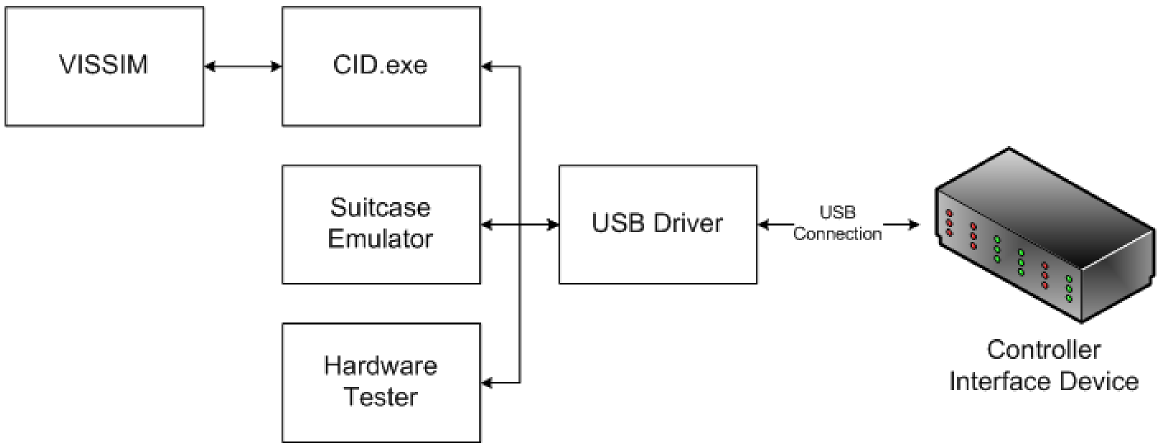

Figure 8. Hardware-in-the-Loop Software-CID Connection

From this one intersection example, an entire simulated corridor of signalized intersections can be implemented using HILS. Consider the real-world network demonstrated in Figure 9.

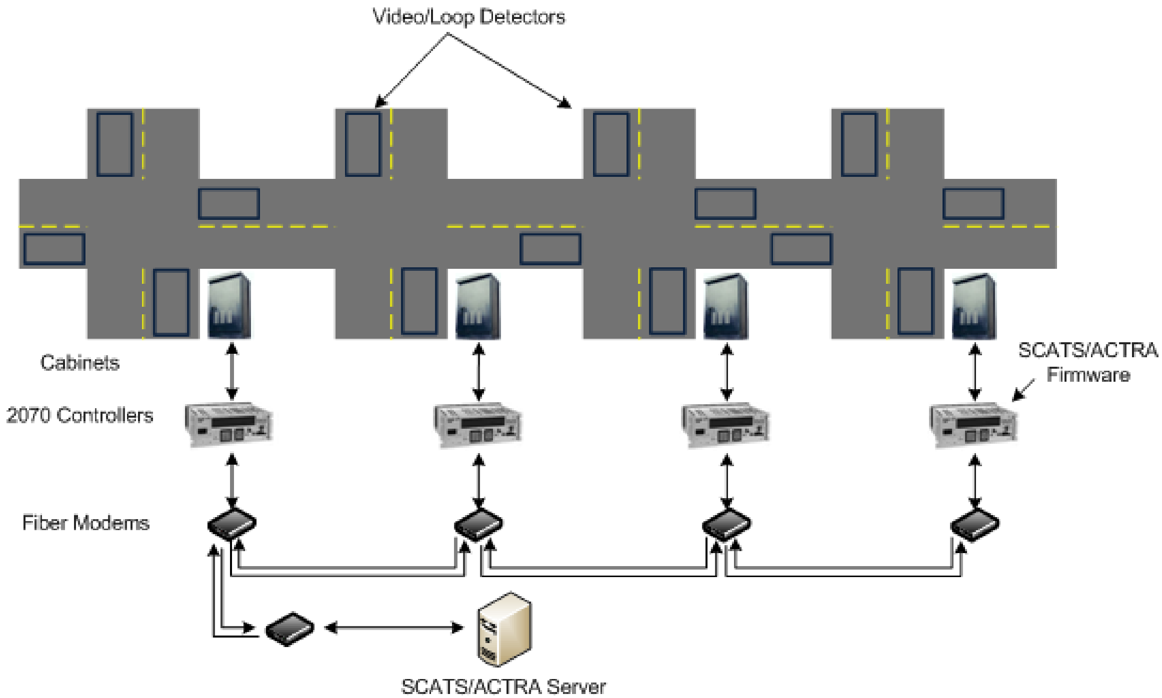

Figure 9. Hardware-in-the-Loop Field Implementation

(Figure Credit: Cobb Parkway ATMS Evaluation [32])

The schematic in Figure 9 roughly outlines the field implementation of SCATS and ACTRA as implemented on Cobb Parkway. In this case, each controller resides in the cabinet at the intersection and is directly attached to the intersection detectors and signal heads. The controller communicates with the central server at the TOC through a daisy-chain of fiber optic cables and modems. For evaluation purposes the schematic shown in Figure 10 was implemented using HILS to replace the physical network with a simulated one.

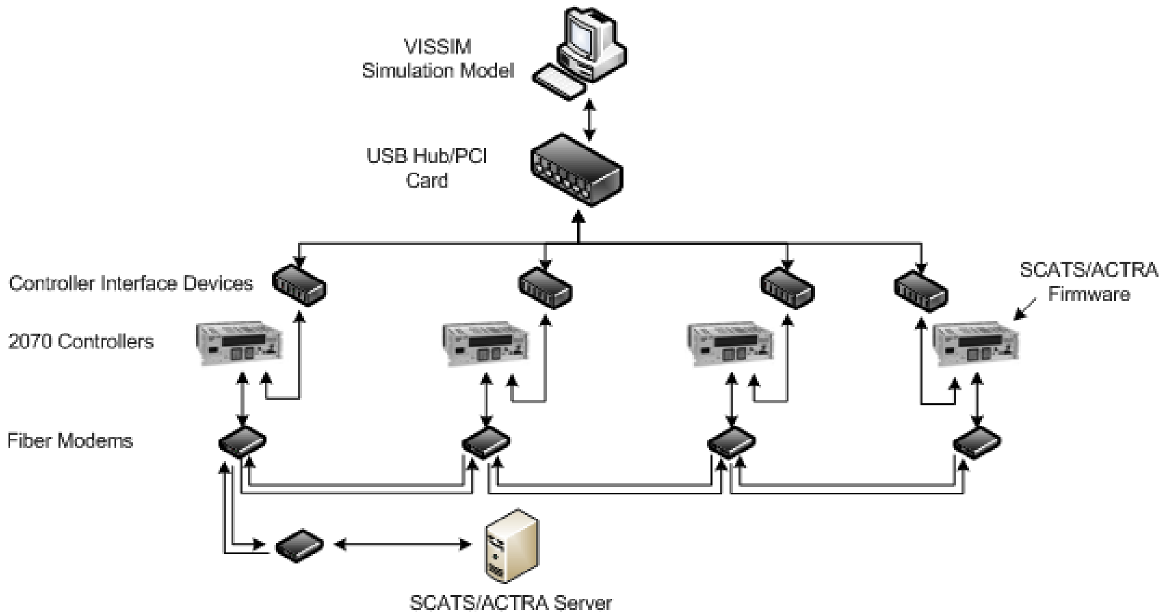

Figure 10. Hardware-in-the-Loop Simulated Implementation
(Figure Credit: Cobb Parkway ATMS Evaluation [32])

In this implementation, the simulation provides the required detector actuations and receives the signal head data. Most importantly, the system architecture “south” of the CID’s is functionally identical to the field implementation, allowing for an analysis of the control system’s effectiveness under various scenarios. While this system works well for analysis purposes, the HILS implementation requires the simulation run in real time (i.e. at 1 simulated second per second). Simulations with emulated controllers can usually be run at much faster speeds, facilitating rapid gathering of simulation results and providing further incentive for simulation analysis. The real-time run constraint generally increases the time required for experiment runs and analysis.

Issues

HILS is a relatively new technology that has been successfully implemented on a large scale by only a few groups. One of the challenges of a HILS system test bed is that each system tends to have a unique set of architecture components, i.e. computers, operating systems, controllers, simulation versions, etc. Thus, a new HILS test bed can, and likely will, have a number of implementation hurdles that must be overcome. The following is a brief discussion of some of the issue that arose with the development of the HILS test bed for this research and the software and firmware updates required for their ultimate resolution.

Simulation Run Time Issues

Typically VISSIM models are run at a speed much greater than 1 simulation second per 1 real-world second. This saves time by allowing several simulation runs of many simulated hours to be run in a relatively short period of time. However, as stated, in HILS the simulation must be run in real time, that is, one simulation second per real-world second. In the initial HLS implementation it was noticed that the simulation would hang every 10 seconds for about half a second. As the controllers were expecting continuous data input this slight pause was not consistent with expected field data, potentially influencing any study results. Upgrading the VISSIM (version 4.1) software service pack to service pack 14 successfully resolved this issue.

Detector Issues

In the initial HILS system setup only one intersection was implemented with HILS to allow for a hopefully rapid and efficient debugging. The remaining intersections in the Cobb Parkway model were run with the integrated VISSIM NEMA controllers. During this initial test the issue arose that, when sending detector actuations to the controller, the CID would only send a continuous signal if the vehicle actuating the detector was stationary. Otherwise, if the vehicle was moving while on the detector, the signal from the CID was a short “ping” each second the vehicle occupied the detector. As with the simulation pause issue this did not represent a reasonable reflection of real-world detector behavior, where the detection signal is continuous (if detecting vehicle presence) whenever a vehicle is within the detection zone.

Preliminary investigation of the issue indicated that the reason for this behavior could either be with the hardware and software related to the CID, or with the VISSIM model itself. Initial debugging efforts focused on the model as this was the most straightforward to troubleshoot. It should be noted that PTV was instrumental in providing guidance in this troubleshooting stage. Numerous troubleshooting measures were taken including upgrading VISSIM to version 4.20 (the present installation was running version 4.10) as detectors in version 4.20 had additional settings that might resolve the issue. VISSIM was upgraded and additional settings became accessible that allowed for the setting of the type of detector in the model. After investigation of the new settings it was determined that Presence, Pulse and Standard detection within VISSIM 4.2 could be applied to the model however these did seem to correct the detection issue.

Once the VISSIM troubleshooting efforts were exhausted efforts focused on the CID. Efforts to resolve this issue (and many subsequent issues) involved assistance from both the CID distributor (McCain Traffic Solutions) and the primary system developers (The University of Idaho). Based on initial investigations one of the original CID software developers (Mr. Zhen Li) was able to successfully assist in updating the CID software through updates to the CID.exe file. The process of updating the CID.exe proved to be iterative as it was necessary to test the CID.exe on the HILS test bed configuration being used in this research effort to determine if all issues had been resolved. This was something that could not be accomplished remotely by the HILS developers and required interaction with the research team. For example, in one iteration when a detector was actuated, it remained so for a full second, even if the vehicle that had caused the actuation had exited the detector already. This had the potential to make successive actuations by several vehicles appear as a continuous actuation. Fortunately, each of the issues that arose was able to be resolved with the guidance and direct involvement of the HILS developers.

<u>CID Connection</u>

The next major issue encountered came when the number of intersections included in the HILS test bed was expanded from one test intersection 11, the number required for the analysis. To expand the HILS test bed intersections were added sequentially, with the system tested upon the addition of each intersection. When the 9^{th} intersection was added to the system the VISSIM-CID connection software was unable to recognize the CID was attached to the computer. Addressing this issue involved considerable coordination between the CID distributor, developers, and the research team. Numerous troubleshooting measures were tried including different computer hardware, several computer operating systems, updated USB drivers, updated USB hubs, and updated CID drivers. Ultimately, through a significant effort including the developers, distributor, and research team the connection issue was resolved through the use of updated firmware and the physical installation of new chips containing the firmware in the CIDs, along with updated CID drivers.

SCATS Access Errors

As stated this research effort involves testing the performance of SCATS and ACTRA under numerous demand scenarios. To accomplish this it is necessary to be able to operate the HILS test bed under both the SCATS and ACTRA operating systems in the controllers. Switching between SCATS and ACTRA implementations for model runs ads a significant level of complexity to the system. In one instance there were several alarms recorded in SCATS Access for an intersection. The subject intersection continually returned to Fallback regardless of communication status. After coordination with TransCore (the firm handling the Cobb County SCATS implementation) it was determined that the controller needed a ram update. This procedure could be accomplished through SCATS Access and resolved the issues with the controller.

High Demand Detector Actuation

With 11 intersections incorporated into the HILS test bed the hardware testing began in earnest. In the course of this testing it was noticed that in SCATS there were a large number of invalid signal change errors being logged.

VISSIM will note when a signal head changes in an erratic manner, for example: rapidly or from one color to another unexpected color, such as green to red or amber to green. Also, when communications between the CID and the model is lost the signal heads in the model will be set to an ambiguous amber state which vehicles in the model treat as an all red.

Through investigation it was determined that the errors were being logged because at random points in the model run one or more simulated intersections were losing communications with the controller. Further investigation revealed that the intermittent (<0.5 sec) communications loss would only occur when 16 or more separate detector channels were all actuated simultaneously at the same intersection. The randomness of the communications failure was due to the fact that the conditions for the failure were only met at large intersections with 16 or more detectors and only when heavily congested. Additionally, this problem only affected SCATS because the SCATS system uses a separate detector channel for each lane, while ACTRA aggregates detectors by movement on each approach. It is not unusual for a larger SCATS intersection to have 23-25 separate detector channels while ACTRA has a maximum of 8 separate channels. This issue was successfully addressed by new updates to the CID.exe by the software developers.

Intersection Error Files

After the completion of each simulation run it was noted that there were several error files created for each intersection in the simulation directory. Inspection of these files showed that they were created because the VAP protocol for that intersection was attempting to communicate with non –existent signal groups. Interaction with the software developer and testing of the model has lead to the conclusion that these files are an artifact of the system and there is not direct impact on simulation performance.

Model Verification

Prior to calibration of a given simulation model a relatively simple model verification procedure must be conducted to ensure that that the model is consistent with itself. That is to say, that the verification process compares the coded performance defined by the model developer with the actual behavior of the model. For example, verification would ensure that a volume input actually creates the number of vehicles that it was coded to create within the timeframe of the model run. In this sense the evaluation of the model is not an extensive determination of behavioral parameters (achieved later through calibration), but rather it is a rough comparison of the model behavior against expected behavior. This verification process typically has two steps 1) the visual verification of the model and 2) the statistical validation of the model [13]. Visual verification is helpful in that it can often identify some problems with the model (e.g. "It does not look like that in the field"), but often underlying problems with the model cannot be identified by visual observation alone [13]. Several statistical frameworks for evaluation of model behavior against the expected behavior are shown in [9, 10, 13] and should be applied to the model to smooth out the rough errors before calibration is undertaken. A miscoded volume input or routing decision could lead to statistically inaccurate and wasted calibration effort.

Calibration

One of the most important issues facing the effective use of microscopic simulation models in analyzing traffic conditions is that of the proper calibration of model parameter values. Calibration of traffic simulation models is the process of determining values for the various behavioral parameters within the models that most closely match the behavior of traffic observed in the real world. This is a challenging process because of the great amount of data and time required to arrive at an effective solution, but a critical step because the default parameter values are often inadequate in describing prevailing field traffic conditions [5, 6, 13, 15, 16].

Several previous studies were evaluated in order to determine the best calibration procedure to use. All reviewed procedures had the same basic progression as that shown in Figure 11. The process can be split into essentially two parts. The first involves the selection of the appropriate calibration parameters, a set that is often unique to the particular model under consideration [5, 14]. Also in this phase the appropriate measures of performance (MOP) are selected to provide feedback for the optimization algorithm to find the best values for the previously selected parameters. The MOP are selected based on the model as well as on the availability and quality of real-world data with which to compare the simulation results [14]. Often, ANOVA is used to conduct a sensitivity analysis on the model to objectively choose the best parameter set for calibration [6, 14], though other algorithms like Latin Hypercube can also be useful in the process [6]. The second phase of the calibration involves the application of an optimal solution search algorithm to the model. In this phase, the simulation is run recursively and the parameters varied according to the results of the previous runs until the optimal solution is reached. There was considerable variation in the search algorithm used in across the different calibration frameworks that were studied. In some cases a particular objective function is formulated that uses the MOP to define the quality of the solution provided for a particular set of parameter values. This objective function can then be optimized using a third party software package or through manual or spreadsheet-aided calculation [14, 17].

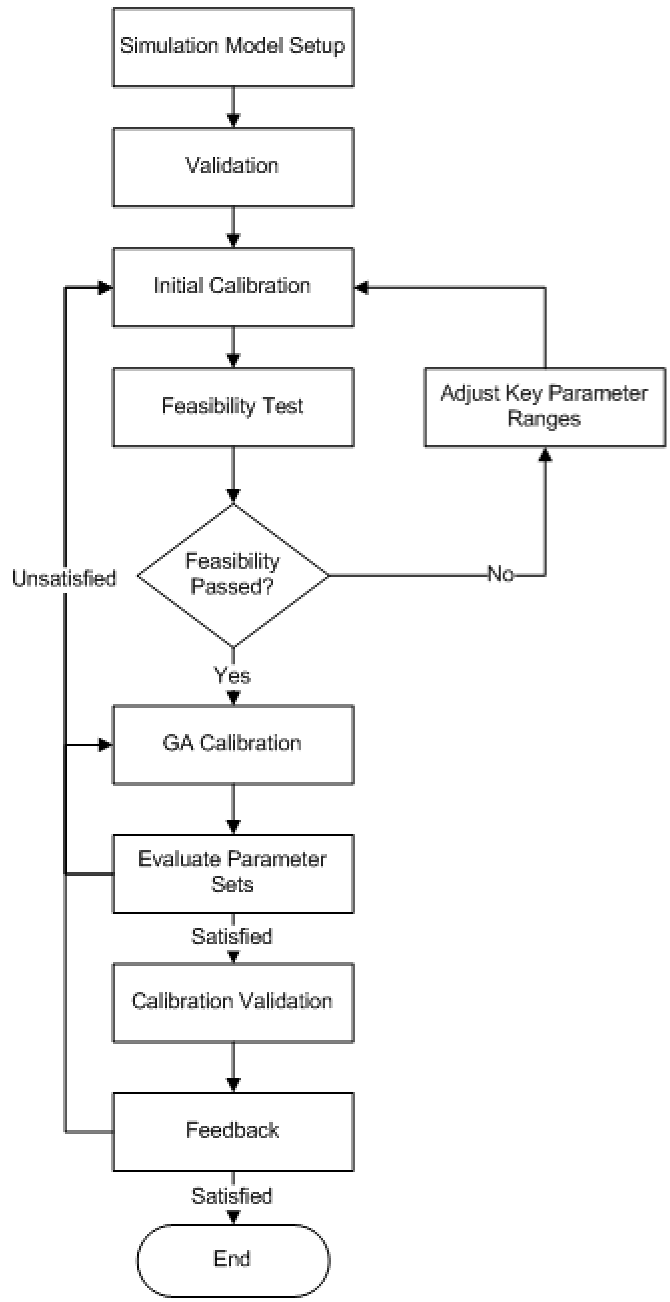

Figure 11. Genetic Algorithm Calibration Procedure
(Figure Credit: Development and Evaluation of a Calibration and Validation Procedure for Microscopic Simulation Models [5])

Several studies have proposed the use of Genetic Algorithm (GA) for the location of the optimal solution [5, 6, 15, 16]. Since GA is relatively simple to apply and has some significant benefits over the other methods, it was selected as the optimization procedure for this project. GA optimizes the computing power used in the calibration process as described in [19], functioning as an implementation of the nested partitions method of searching the solution space.

The selected calibration process used in this study is shown in Figure 11 and uses ANOVA sensitivity analysis to determine the optimal parameter set for calibration. The initial values for each of those parameters and the range of possible values to be evaluated were selected based on engineering judgment informed by the work of previous researchers [4]. After the selection of parameter ranges and initial values, GA was used to calibrate the model. GA uses a procedure that does not rely on calculus to determine the derivative in creating a gradient for the potential solution space, and thus it is less prone to return only a local optimum [18]. The GA procedure is also model agnostic as long as an acceptable MOP is available for comparison. In addition GA has an excellent reputation for providing effective search and optimization for artificial intelligence systems [5]. While many other GA calibration procedures rely on external software packages such as MATLAB to perform the optimization functions [5, 16], a Visual Basic software program was constructed to conduct the GA procedure and ensure that control of every step of the optimization process was retained [32]. There are three basic steps in the GA procedure: reproduction, crossover and mutation [5]. All the possible values for each of the selected parameters is represented in binary form (called a gene) with an entire set of parameters appearing as a string of binary digits called a “chromosome”. A group of chromosomes, otherwise known as a population, make up a generation. The three steps mentioned previously are run for each generation. First each chromosome is evaluated to determine its fitness value. Because each chromosome represents an entire set of model driving behavior parameters this means that there is a separate model for each chromosome that must be evaluated. Because of the stochastic nature of the models, the evaluation of each model requires a number of independent runs [13]. In this case five independent runs are used with different random seed values to determine the average MOP value for each chromosome. This MOP value is then evaluated against the field value to determine the fitness of the chromosome. The inverse (to minimize the value) of the mean square error between the observed and simulated travel time was used in this evaluation [32]. The GA function then selects the best chromosomes from the generation based on the fitness value. This is the reproduction phase. In the crossover phase the GA function exchanges genes between the chromosomes selected in the reproduction phase to form new chromosomes for evaluation. The likelihood of a particular gene being switched is determined by the crossover probability which is recommended to be around 6% [18]. The mutation phase changes one binary digit based on the mutation probability to induce some randomness into the search and escape any local optimum values [32]. The mutation probability is very low to avoid making the GA procedure a random search, typically the value is 1/(population size) [18]. The previously described procedure is demonstrated in Figure 12.

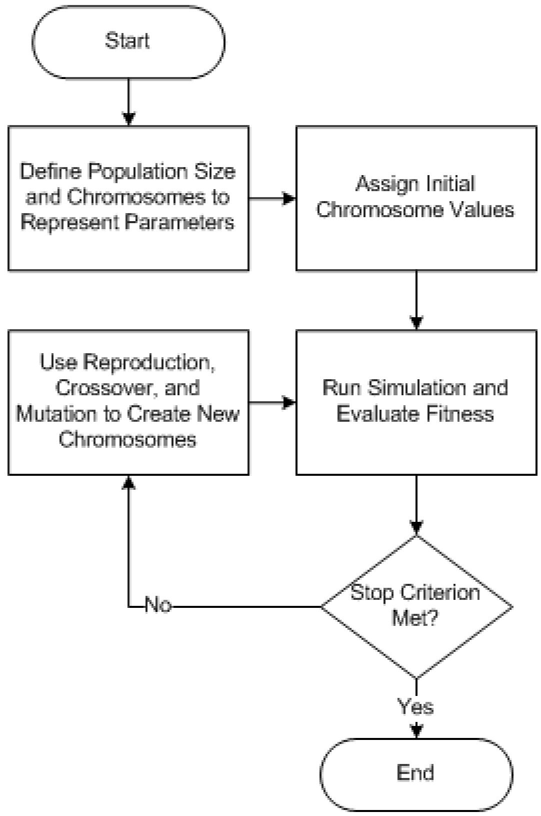

Figure 12. GA Procedure

(Figure Credit: Cobb Parkway ATMS Evaluation [32])

Simulation Management

Throughout the simulation evaluation process many different scenarios and implementations were run. It became difficult to keep track of all the data and requirements associated with each particular evaluation. Therefore a tool was developed that would aid in this task. Initially simulations were conducted using the VBA COM interface to VISSIM [2] through Excel. The process entailed starting Excel and entering in the appropriate parameters in particular spreadsheet cells. A call would be placed to a particular VBA macro which would then run the VISSIM model and collect the data as the simulation ran. While this process worked, there were several limitations. First, there was no separation between the runtime simulation control and the data collection. If the simulation crashed or if the Excel sheet was closed without saving, all of the data for that run and all previous runs were lost Also, even after the simulation was run the excel sheet retained both the data and the runtime code, which made it both part of the simulation framework and a product of it. Second, VBA had some limitations in running the simulation. For example it was not possible to interact with databases without significant effort. Finally, Excel was setup to do some data processing in the collection process and so it was impossible to return to the raw data if needed after the run was completed.

To resolve these issues a progression was proposed as demonstrated in Figure 13.

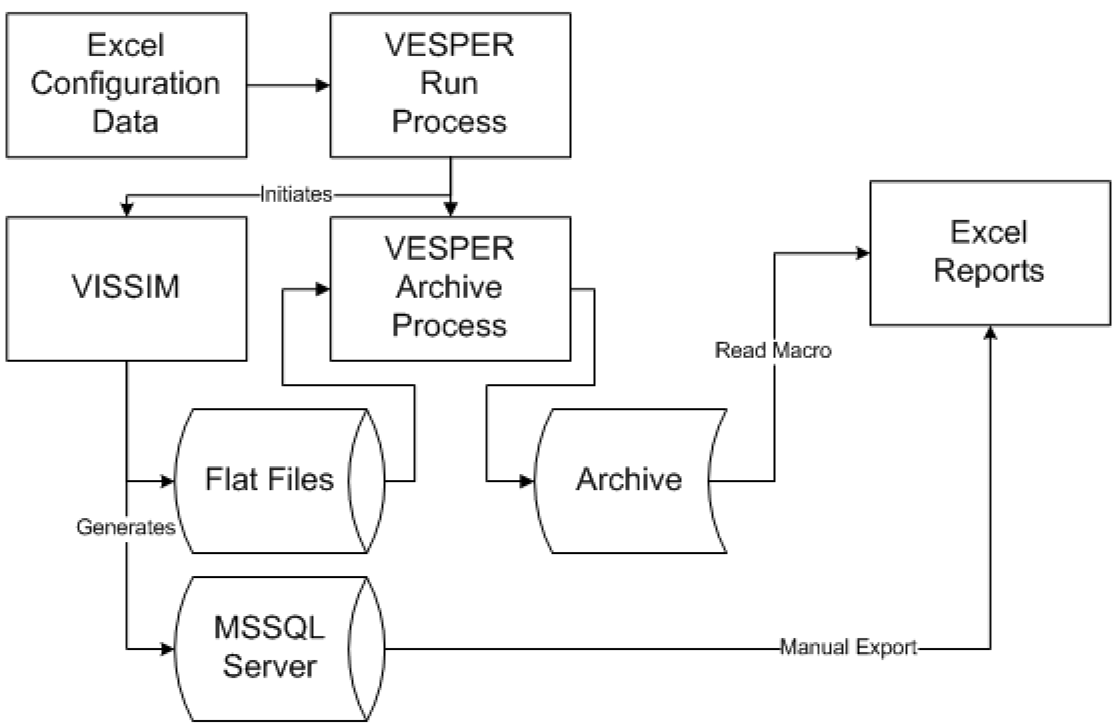

Figure 13. VESPER Runtime Framework

Excel would remain an important part of the analysis and visualization steps; the function of Excel in this role is demonstrated on page 66.

VESPER

VESPER or the VISSIM External Simulation Performance Evaluation Routine was created to automate the simulation run and data collection portions of the process shown in Figure. VESPER was written in C#.NET and contains the functionality for conducting VISSIM runs previously contained in VBA in Excel. VESPER controls the VISSIM model through COM conducting runs and automatically archiving the data for later evaluation. VESPER is designed to work with the folder structure shown in Figure 14.

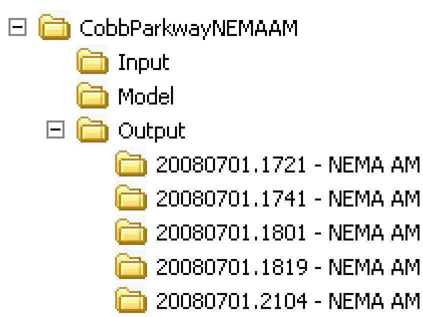

Figure 14. VESPER Folder Structure

VESPER reads in the configuration data from the Excel file which included number of replications, run length, seeding period and so on. The runs are conducted and each time a run is completed VESPER copies the data into an archive folder in the data directory. To uniquely identify each model run data set, a folder naming system was developed. Each results folder was named with the date in the following format:

<YEAR><MONTH><DAY>.<HOUR><MINUTES> - <SCENARIO NAME>

As an example, an evaluation run of SCATS AM that completed on March 19, 2009 at 10:52 PM the folder name would read:

20090319.2252 – SCATS AM

With this system, not only was each run uniquely identified, but the folders for all the runs would appear in sequential order when viewed with the default settings in Windows.

Another benefit of this tool is that VESPER allows for the sequential evaluation of multiple scenarios. Up to three configuration files may be input for a single run and VESPER sequentially processes each configuration file and associated simulation model.

While an effective tool for simulation management there are some improvements that could be integrated into a future version. The first improvement is that VESPER should be integrated with the data visualization tools to completely automate the process of simulation evaluation. In this way the analysis and visualization tools would be called after the model run was completed. The automatic generation of the simulation results analysis would make the process easier for non-experts to conduct and would speed the evaluation process for researchers. The second improvement is that VESPER could be expanded to allow for distributed evaluation of simulation models as detailed in [20]. This would allow both for the evaluation of large models and for the concurrent evaluation of multiple small models. Multiple scenarios could be evaluated quickly, and large-scale networks could be divided up and evaluated on multiple systems [20].

Data Analysis Tools

To analyze the vast amount of raw data generated in the model runs a set of Excel spreadsheets were developed using Visual Basic for Applications (VBA) to develop macros that would speed repetitive data analysis tasks. The analysis process typically contained three steps. The first was to import the raw data into a form that could be easily analyzed. Turning movement counts, travel times, and queue data records were all output from the model into semi-colon delimited text files which could be imported into the analysis tool using Excel's data import functions. The second step involved refining the raw data to identify errors and convert units and values into those appropriate for eventual analysis. The final step was to present the data in an easily readable format. While seemingly trivial, this step posed the most challenges. The volume of data was difficult to process and often times there were many different ways to visualize the data based on the end goal of the analysis. A few standard visual data outputs were created with an option available for custom views to be created as needed.

Figures

Figures, like shown in Figure 15, form the basis for most transportation-related data visualizations. They show the numbers that represent delay, turning volumes, but other metrics but also account for the spatial nature of the data, making it relatively easy for an engineer to ascertain what is occurring at the various study intersections along the corridor.

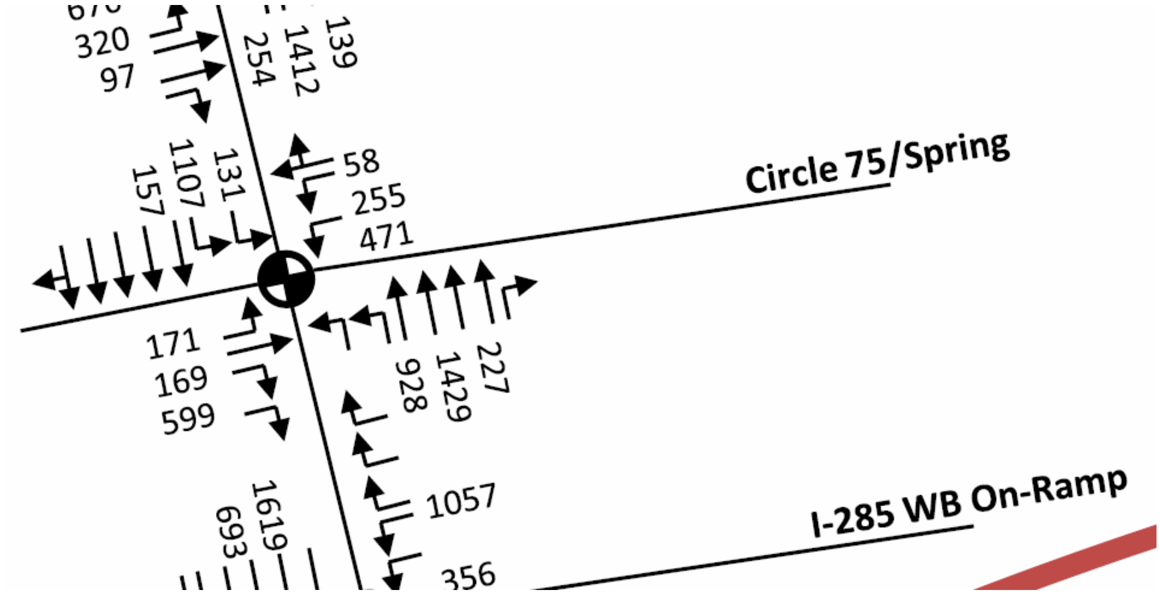

Figure 15. Example of Volume Figure

Often, figures are created by hand, but the size of the study corridor and the volume of data to display made the time to generate the figures demonstrated above by hand prohibitive. Instead custom figure generation code was developed in VBA in Excel. The processed data could be entered into the figure generation spreadsheet and the VBA macros would generate 11x17 format figures for printout and analysis. A template figure with placeholders for the various values was the only portion of the process that had to be constructed by hand. After the template is created, it can be used by the VBA macro to create all subsequent figures saving considerable time and making the generation of this form of data visualization feasible for alternative analysis.

Travel Time Data

Although figures are useful for the evaluation of volumes and delays, they are not as effective as representations of travel time data. To display the travel times a set of graphs was developed to provide a better visualization of the travel times along the corridor. Excel was again employed to create several template graphs that showed the travel time along the entire corridor as a function of time as well as the travel time between various points in the network. The processed data could then be inserted into the graphs using a VBA macro and displayed against the appropriate field data for comparison.

For the northbound and southbound segments a travel time/simulation time chart was used to show the variation of the travel time over the course of the simulation run. This allowed one to watch for the development of steady state and determine the fitness of the results when compared with the recorded field data. An example of this output is shown in below in Figure 16 (repeated from later in the results section):

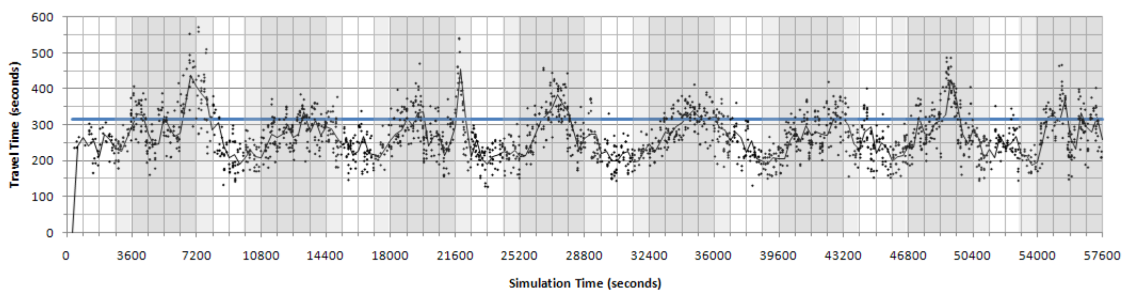

Figure 16. SCATS PM Northbound Base Travel Time

For the incremental segments showing the time from either end of the network to each intersection time/distance charts were developed to show the portion of total travel time contributed by each segment. These were particularly helpful data visualizations as they showed the contribution of each intersection to total delay along the corridor and helped determine what (if any) simulated intersections were not performing consistently with the observed field conditions.

Queues

Excel was also used to analyze queues which were represented with length/time graphs that showed the maximum length of the queue at various times during the simulation. This helped determine if the queues were clearing with each cycle or if a standing queue was developing, indicating a cycle failure. The plots also showed whether the model was reaching steady state. Linear or exponential queue growth would indicate problems, preventing the model from reaching steady-state.

Velocity Profiles

As the calibration and validation process wore on, it was determined that velocity profiles would assist in the interpretation of the model performance. The per-vehicle record data recorded for each of the probe vehicles was generated and a custom Excel file was built to plot the velocity profile of each individual vehicle as it traveled through the corridor. The link and position-on-link data could be used to determine the position of a vehicle along the corridor and when combined with the vehicle speed these could be compared against similar data from the field to determine if the model accurately reflected field conditions. In particular these graphs showed where most vehicles were stopping for signals, demonstrated the maximum speed vehicles were achieving between intersections, and gave an impression of the average overall speed of traffic along the corridor.

Traffic Visualization

During development of the modeling process a tool was required that would allow for the review of the model run so that users could go back and each the traffic performance at any point. Several attempts were made to implement this type of visualization including screen capture software and VISSIM's internal video recording. None of the attempted solutions provided the level of detail desired, specifically the ability to zoom and pan through the model. Both the screen capture script and video recording had to be conducted from a fixed perspective, and would not allow for the close examination of multiple parts of the model. In seeking to provide this functionality, a tool was created using VB.NET that had the ability to read specially recorded vehicle record data and play it back at various speeds over the course of the entire model run. This tool required that the model be set to record, by way of the individual vehicle information, the x and y coordinates for every vehicle at a user defined time step [1]. Because of the great volume of data generated in this process VISSIM was configured to record the vehicle records to an MSSQL database installed on the simulation workstation. Additionally, the data was only recorded once every few time steps to reduce the volume and workload for the simulation workstation. After the conclusion of the simulation the data was indexed to improve performance and was then displayed using the visualizer program. The visualizer allows the user to spool to any point during the simulation run and watch the playback at any speed desired. Additionally the program has zoom and pan functionality so that any point in the network can be viewed at any time. This greatly contributed to the analysis of the model runs.

R Statistical Package

R is an open source statistical package that provides powerful statistical analysis functionality above and beyond what could be accomplished with Excel and VBA. After initial data analysis was complete in Excel refined data was exported for analysis in R. R provided two key components to the statistical analysis of the simulation data. First R allowed for the generation of a graphical representation of the distribution of the data including normal Q-Q plots and histograms. Second R provided computerized statistical tests which enabled advanced statistical analysis methods to be carried out efficiently. This included the Shapiro-Wilk test of normality and the Wilcoxon Ranked Sum Test to compare populations.

The overall structure of the data analysis and visualization process is demonstrated in Figure 17.

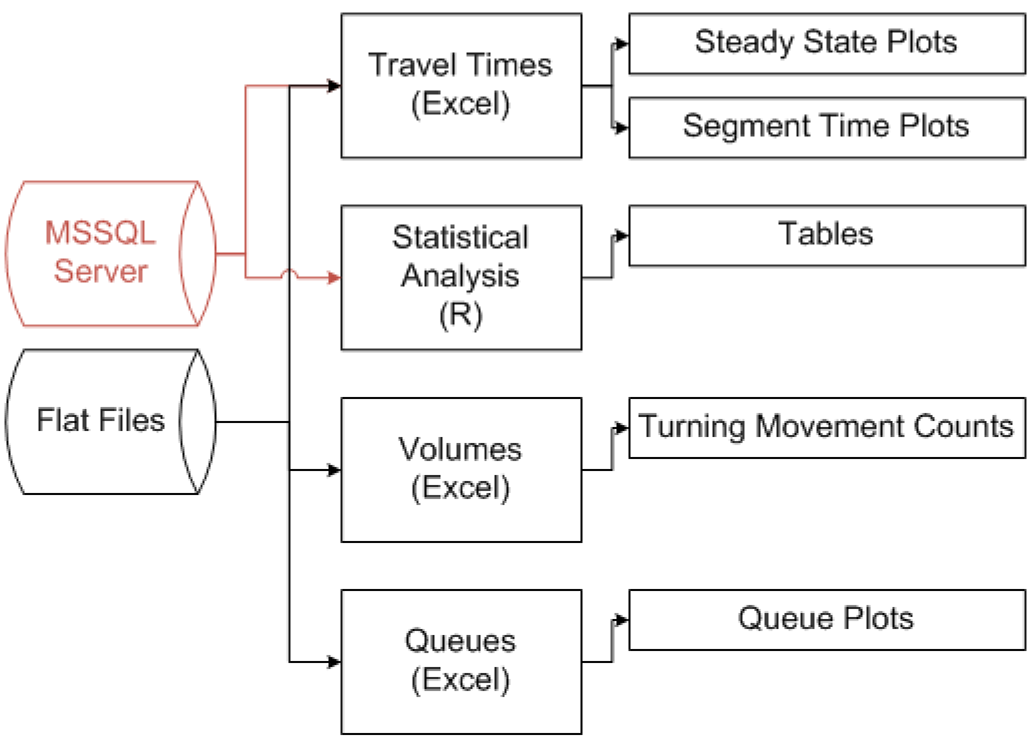

Figure 17. Data Analysis Framework

The most important result of any data visualization tool however must be the effective presentation of the simulation results in a manner that is easily understood by the intended audience [25]. To that end the tools developed for the visualization of the data output from this simulation are effective for expert analysis and interpretation, but fall short of making the impacts of the simulation real for non-experts. Given the flexibility of this system in evaluating the performance of various ITS implementations, the results of analysis using this system are bound to be important for many outside the transportation engineering profession. Policy makers and the general public desire the ability to understand what the simulation is telling them [25]. To that end further work must be done to expand and enhance the functionality of the existing tools so that they are effective in conveying the results of the model to a wide audience.

Statistical Analysis of Simulation Results

The accurate analysis of simulation results is arguably the most important part of the simulation process. Correct interpretation of the simulation data and the ensuing conclusions are what make simulation such an enticing solution for the evaluation of various scenarios. There are two problems with the typical analysis of simulation results. First, most useful simulations are stochastic in nature which means that the resulting raw outputs are subject to sampling error [8, 10]. The second problem arises from the first in that traditional statistical techniques, which would normally be employed to mitigate the sampling error, are hampered by assumptions about the data that do not apply to the output from most simulations. Specifically, simulations produce output data that is not independent, not identically distributed, and not normally distributed [8, 10, 27]. These two issues dictate that special care be used in the evaluation of simulation data to ensure that the resulting conclusions are statistically valid and, more practically, an accurate representation of real-world results.

The discussion of the statistical analysis of simulation results must start at the beginning with a brief overview of the simulation input. Input analysis is an important step in the simulation process as the misspecification of a particular simulation input could lead to invalid results [8]. A procedure in [8] indicates that the following steps are important for simulation input analysis:

1. Determine what input variables are required.
2. Collect as much field data as possible for the formulation of the input data.
3. Critically evaluate the collected data and make assumptions about the underlying distributions as appropriate.
4. Estimate the parameters from any parametric univariate distributions that are observed.
5. Subject the assumed distributions to a formal goodness-of-fit test to ensure that the fit of the distribution is sufficient.
6. Consider the possibility that more-sophisticated distributions may be appropriate (poisson or auroregressive).

The overall objective of input analysis is to ensure input data consistency. One example relevant to this study is the volume and turning movement inputs. The nature of field data collection will often lead to inconsistency in traffic counts taken at consecutive intersections or counts that were taken on different days or during different seasons. This can result in significant discontinuity between the number of vehicles exiting the first intersection and the number of vehicles entering the second. In this case a reasonable assumption must be made about the nature of the traffic flow to ensure that the input data for the model is consistent. Perhaps there is a source or a sink in between the two intersections, or perhaps one of the field counts was done on a day with slightly lower volume and the numbers need to be adjusted by a reasonable amount. As will be shown in the case study, there is a great deal of art and science that go into the selection of proper input data. However engineering judgments must be made carefully as the selection of input data will have a significant impact on the simulation outputs [8].

Also, input analysis seeks to establish a consistent set of inputs across all of the simulated scenarios under consideration. As will be shown in the section detailing the case study data comparisons, when comparing two systems with different logical or functional configurations it is important to ensure that all other inputs remain equal [8, 10]. This holds true both for the raw inputs specific to the function of the model (traffic volumes, turning movement distributions) as well as to the underlying parameters of the model itself (random seeds). This is important for the use of common random numbers in the evaluation of two or more systems, a concept that will be discussed further in the section dealing with the case study [8].

Output analysis is the process by which the raw output of the simulation is considered in a manner consistent with its nature so that conclusions can be drawn and applied in the real-world. Because the output data is not independent, identically distributed (IID) normal elementary statistical techniques cannot be simply applied in its evaluation [8]. In this case there must be a focus on overcoming the limitations of the data as part of the analysis process. Traffic simulation is what is known as a nonterminating (or steady-state) simulation as opposed to terminating (or transient). Unlike a terminating simulation where there is a finite amount of time allotted to a run (a store with a specific closing time for example), this traffic simulation represents a continuously running system. This poses a unique set of problems. The first is initialization error which is the result of the arbitrary determination of initial input values [8, 9, 27]. In this model the initialization error is clearly demonstrated in that the entire corridor is empty at the start of the simulation run so any resulting evaluations of performance would be misleading until the model reaches a state of congestion consistent with the field during the evaluation period under consideration. There are two potential solutions to initialization bias. The first is to truncate the output and eliminate the biased data by giving the model a period of time in which to warm-up. While effective, this method requires that the point for output truncation is relatively clear, an observation that can be rather difficult [8]. The second method for overcoming initialization bias is to conduct a simulation run of sufficient length as to ensure that the effects of the initialization bias are negated by the sheer volume of steady-state data [8, 9, 27]. For some models this method may prove impractical though as it may require an excessively long run to provide the necessary volume of data [8]. In this study truncation was used to overcome the effects of initialization bias. As demonstrated in the case study it was observed that the nature of the prevailing traffic conditions was such that one hour was more than sufficient time to allow the model to reach steady state. This warm-up period, also known as the seeding period, provided sufficient time for traffic on the corridor to reach appropriate congestion levels while also providing sufficient history for the adaptive control mechanisms to make accurate decisions.

The collection and aggregation of data from the simulation model was also an important step in the process of output analysis. There were several methods proposed for accurate data collection in the literature. These included methods such as spectral estimation [10, 27], standardized time series [10, 27], and sequential methods [28] among others. While all methods are at least somewhat effective in the analysis of simulation output, the following methods were determined to be the most effective at meeting the needs of this analysis.

Data Collection Methods

Batch Means

The method of batch means involves the combination (hence "batching") of several independent observations over a fixed time interval into one averaged estimator. The result is that each of the batched estimators is approximately IID normal [8, 10, 28, 30]. This method is particularly useful in combining the individual values from a simulation run or series of runs in a way that they can be used with traditional statistical analysis methods. This method is helpful in comparing two simulated models against each other in combination with the regeneration method detailed below. However, because there was only a limited amount of field data collected, this method was not useful in comparing simulated observations against real-world data as there were not enough real-world data points to effectively employ batch means.

Independent Replications

The method of independent replications is intended to address the concern of correlation between the estimators calculated using batch means. This method uses several runs of the simulation to calculate a series of estimators that are each completely independent as long as the replicate runs were each initiated with a different random seed value [8, 10]. This method overcomes the issues associated with sample size by comparing the simulation models against real world data by providing a set of independent observations for direct comparison without the averaging effects of the batch means method. This method, used in conjunction with the regeneration method below, is also preferred when possible given that independent replications provide better coverage for the confidence intervals than batch means when the initial transients are removed with truncation [30].

Regeneration

The method of regeneration finds that a long simulation run can be broken into IID blocks at particular "regeneration points" [10, 27]. While it can be difficult to determine these points naturally, they can be established using input variation as demonstrated in the case study. The conduct of one long run broken into several blocks allowed mitigation of some of the external limitations of the simulation environment while still providing independent replications and data for batch means for further analysis.

Data Analysis

Given that the output data can be collected and processed in a way that makes it consistent with IID normal data the following statistical analysis for the comparison of two systems can be applied [8, 27]:

Let $Z_{i,j}$ be the measure of performance from a simulated model. In this case, there may be two strategies represented with i=1,2 and for each strategy there are several replications represented by j=1,2, … ,b_i. It can be assumed that $Z_{i,1}$, $Z_{i,2}$, … , $Z_{i,bi}$ are IID normal based on the collection methods described in the previous section. Based on that assumption a 100(1-α)% confidence interval (CI) for the difference between the means of the distribution of the performance measure for the two strategies (μ_1 - μ_2) can be created. It can now be defined:

$$\bar{Z}_{i,b_i} = \frac{1}{b_i}\sum_{j=1}^{b_i} Z_{i,j}$$

and:

$$S_i^2 \equiv \frac{1}{b_i - 1}\sum_{j=1}^{b_i}\left(Z_{i,j} - \bar{Z}_{i,b_i}\right)^2$$

So an approximate 100(1-α)% CI would be defined as:

$$\mu_1 - \mu_2 \in \bar{Z}_{i,b_1} - \bar{Z}_{i,b_2} \pm t_{\alpha/2,v}\sqrt{\frac{S_1^2}{b_1} + \frac{S_2^2}{b_2}}$$

Where the degrees of freedom are defined:

$$v \equiv \frac{\left(\frac{S_1^2}{b_1} + \frac{S_2^2}{b_2}\right)^2}{\frac{(S_1^2/b_1)^2}{b_1 + 1} - \frac{(S_2^2/b_2)^2}{b_2 + 1}} - 2$$

The interpretation follows that if the CI is entirely to the right of zero then system 2 has a higher mean value, but if the CI is located entirely to the left of zero then system 1 has a higher mean value. If the interval contains zero then it can be assumed that the systems are statistically similar [8, 27].

Also, the procedure can be extrapolated to conduct a t-test based on the previous values and equations from [33]. This involves calculating a t-statistic (Considering Null Hypothesis H_0: μ_A-μ_B = δ):

$$t = \frac{\bar{Z}_{i,b_1} - \bar{Z}_{i,b_2} - \delta}{\sqrt{\frac{S_1^2}{b_1} + \frac{S_2^2}{b_2}}}$$

A hypothesis test for α accepts the null hypothesis if:

$$|t| \leq t_{\alpha/2,v}$$

And rejects the null hypothesis when:

$$|t| > t_{\alpha/2,v}$$

A paired-t test given that the same number of replications were run for each strategy can also be conducted. This procedure is found in [8]. The differences $D_j \equiv Z_{1,j}$-$Z_{2,j}$ for all replications j=1,2, … ,b can be found. Then the mean and standard deviation for the differences can be calculated:

$$\bar{D}_b \equiv \frac{1}{b}\sum_{j=1}^{b} D_j$$

$$S_D^2 \equiv \frac{1}{b-1}\sum_{j=1}^{b}\left(D_j - \bar{D}_b\right)^2$$

Which yields a 100(1-α)% CI:

$$\mu_1 - \mu_2 \in \bar{D}_b \pm t_{\alpha/2,b-1}\sqrt{S_D^2/b}$$

It turns out that if the two models representing each strategy were run independently the CI length is very small [8, 27]. This requires that each of the strategies sees the same input distributions, otherwise known as common random numbers. For example, it is ensured that the input traffic volumes and the turning distributions are the same for each strategy under consideration. Additionally it is ensured that the underlying structure of the model (i.e. the driving behavior parameters, random seed) is the same for each strategy. By presenting the models with identical conditions the performance of one against another can be determined even though the performance measure estimates are subject to sampling error [8, 27].

Nonparametric Data Analysis

Although the above method is good for the comparison of two models under identical experimental conditions, there are times when that requirement cannot be met and another analysis method must be used. In this case it's helpful to have an analysis method that is not hindered by the conditions of independence and distribution of the data for classical statistical analysis [29]. One application of this would be in the comparison of the field data against the simulated models. In this case it cannot be ensured that the two results are the product of identical inputs, and so nonparametric data analysis must be used. For this analysis a ranked sum test was recommended and is detailed below [29, 33]:

The rank sum test can be used to compare two populations regardless of the distribution of the data within the population [33]. The assumption is that the two populations have roughly the same distribution, with a potential location difference, δ, between the means. This is a reasonable assumption to make for many populations [33]. Each population must be independent and identically distributed, two assumptions that can be met using a traffic model given that either batch means or independent replication is used and the data is collected at steady state [8, 27, 33]. Often the rank sum test provides an analysis result as strong as the two-sample t-test without the requirement of normally distributed data [33].

Generally speaking the rank sum test evaluates the null hypothesis:

$$H_0: F_A(x) = F_B(x)$$

Where F_A and F_B are the two distributions being compared. The test also determines an approximate value for δ along with a confidence interval. Or, the test can be used to evaluate the likelihood that a given value of δ describes the magnitude of the distance between two distributions. The procedure behind the rank sum test is computationally extensive and generally requires the use of a computerized statistical analysis package. A detailed discussion of the procedure will be avoided here but can be found in [13] or any good statistical text dealing with nonparametric data analysis.

R Statistical Package

To conduct the bulk of the statistical analysis the computerized statistical package R was used. R is an open-source program with many powerful analysis tools that greatly simplified and sped the process of data analysis for this project. The specific implementation of R in the analysis process is discussed further on page 148.

CHAPTER 5

CASE STUDY: COBB PARKWAY MODEL

Model Construction

VISSIM Model Creation

The study area for this project is a segment of Cobb Parkway extending north and south of its intersection with I-285 and close to the junction of I-75 and I-285, as shown in Figure 18 and Figure 19. The model for this study was initially constructed by previous researchers based on geometric and vehicle data collected in the field. The initial model had 15 intersections starting at Lake Park Drive in the north and extending south to Paces Mill Road and used the built-in fixed-time controllers within VISSIM.

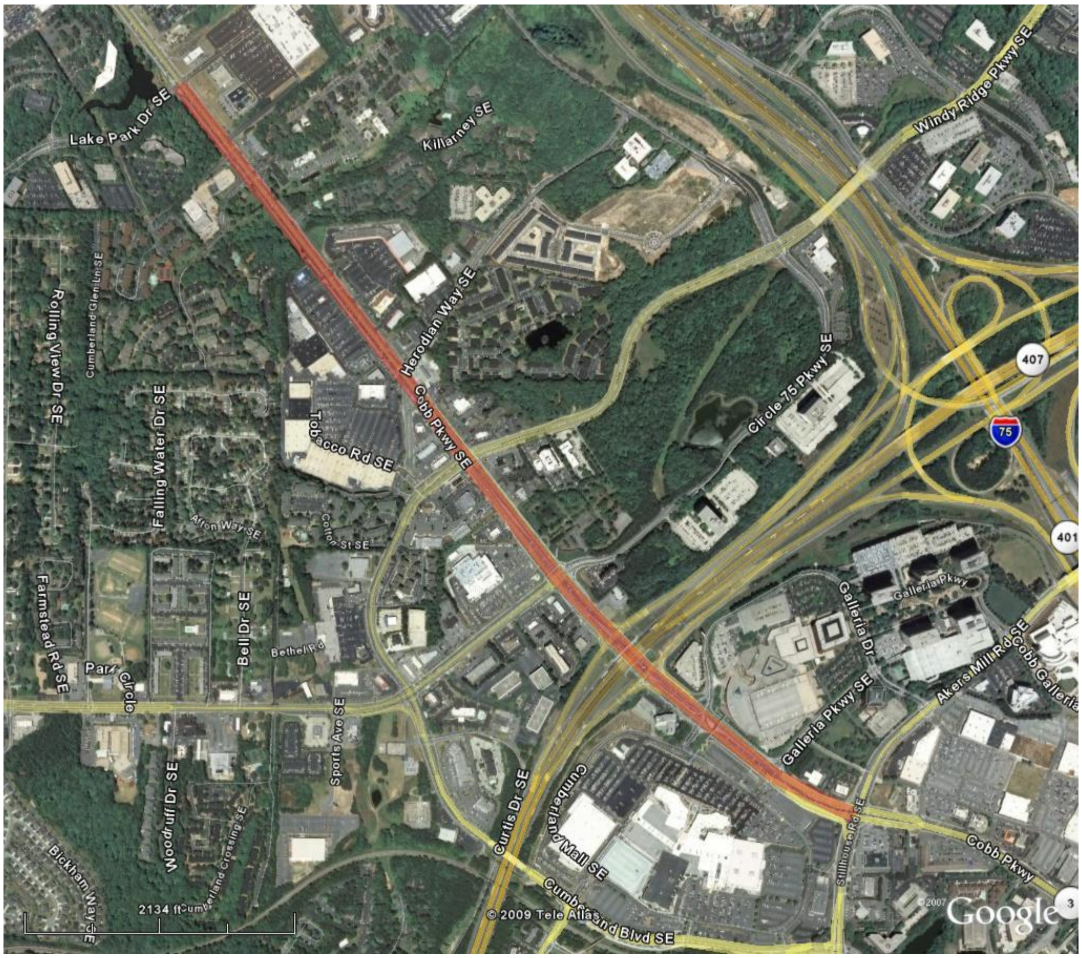

Figure 18. Study Corridor Aerial

(Figure Credit: Google Earth)

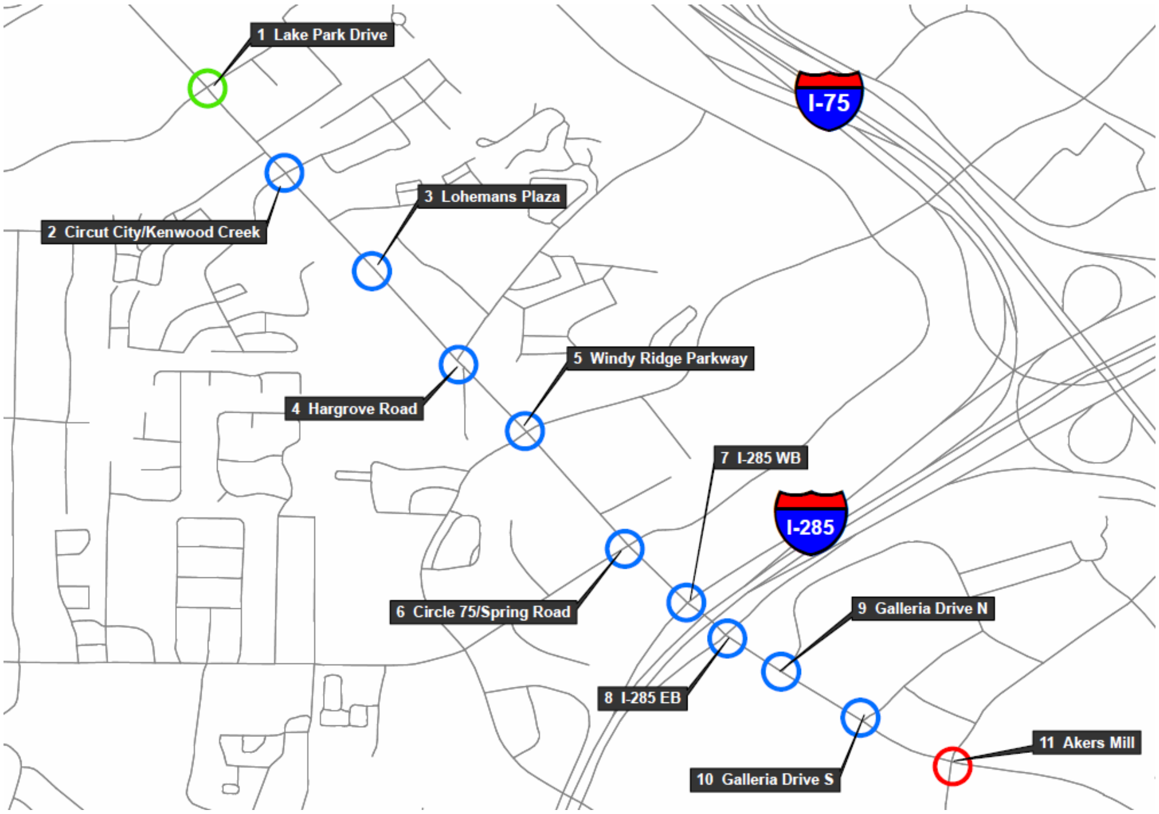

Figure 19. Study Intersections

General Implementation

There were several changes that had to be made to the initial model to ready it for evaluation within the proposed analysis framework. The requirements of SCATS and ACTRA dictated that there would need to be separate models to accommodate the detector and signal geometry specific to each of those systems. Additionally, the real-time limitations of the HILS created the need for a non-HILS system that could be operated at a speed greater than 1 simulated second/1 real-world second to allow for the conducting of effective model calibration routines.

Based on these requirements it was determined to create three models, one base model for calibration, and then one model for SCATS and one model for ACTRA. The only differences between these models were those detector and signal differences required to mirror the system as implemented in the field. The process started with the base model.

Beginning with the model provided by previous researchers, development began on a completely software-in-the-loop model for calibration purposes. The first step was to replace the fixed-time controllers in the original model with actuated NEMA controller emulators that would more accurately model the behavior of controllers as seen in the field. Not surprisingly, given the implementation of fixed-time controllers, the model as provided had no detectors at any of the modeled intersections. Working off of detailed plans provided by Cobb County each intersection in the model was modified to include detectors that closely approximated the field implementation.

In general, for the base model, at each model intersection along the study corridor left turn bays and side streets were the only actuated approaches. During the course of this work the signal heads as modeled in VISSIM were verified against the plans to ensure that they matched so that the programmed phases would correspond to the right signal heads.

The NEMA controllers were programmed using the provided interface within VISSM using timing sheets provided by Cobb County. The NEMA controller is an actuated ring/barrier controller that is similar to the ACTRA system deployed in the field. Using the NEMA controller, the behavior of the signals in the model should closely approximate the behavior of the signals in the field and should be accurate for validation and calibration.

Verification

After the proper control mechanism was implemented for the base model verification of the model was conducted. Unlike Calibration, which would follow later, the purpose of verification is to ensure that the elements of the model are performing as specified in the model coding process [5, 8, 10, 13, 14]. Several parts of the model were reviewed at in this phase of development.

The first evaluation of the model was a simple visual inspection of the model operation. This initial pass over the model involved running the model with visualization of vehicles and signal heads activated so a user could watch the behavior of the various elements and determine problem areas [13]. The next evaluation was based on analysis of travel times, turning movement counts and queue lengths. The model was outfitted with several VISSIM data-collection elements including data collection points, queue counters and travel time segments. These metrics would show if there were significant problems and would provide for the verification of factors to show if the input volumes, routing rules, and signal timings were behaving as expected. In the course of this evaluation there were several changes that had to be made to the model to ensure that it performed as expected [13].

Volume Changes

Based on the turning movement counts the number of vehicles making each movement for each intersection could be compared against the field data to ensure that the volume inputs coded into the model were performing as intended. Field measurements of turning movements had been provided for each intersection and from those a balanced set of turning movements for each intersection could be created. These balanced counts corrected various inconsistencies between the field counts and accounted for various sources and sinks not directly modeled at midblock locations. By comparing the results of the validation runs with this field data inputs were modified and appropriate sources and sinks were added to ensure that the model behaved as expected. These are shown in Table 1. As part of the validation it was noted in several places that the volume of vehicles making a specific movement at a given intersection was far less than the observed number of vehicles making the same movement in the field data. These problems were largely due to poor routing decision definitions. The identification and resolution of such issues are discussed at length in the next section.

Table 1. Sources and Sinks

Location	**AM**	**PM**
Lake Park to Circuit City	-200	1
Circuit City to Lake Park	0	-200
Hargrove to Windy Ridge	-50	0
Windy Ridge to Hargrove	-100	-75
Windy Ridge to Spring	0	-74
Spring to Windy Ridge	75	0
I-285 EB to Galleria Drive North	-150	-261

Erratic Lange Changes and Unreasonable Queue Lengths

In addition to problems with the volume inputs and sources/sinks unreasonable queue lengths and erratic lane changes were also noted. Unreasonable queue lengths usually resulted from a signal timing error or from an inaccurate routing decision volume. Erratic lane changes resulted when vehicles were unable to get in their desired lane in order to follow a routing decision and could have several causes. Incorrect routing decisions and inaccurate geometry were the leading causes. The effects of erratic lane changes on the model performance ranged from moderate to severe.

At times erratic lane changes could block all lanes of Cobb Parkway as vehicles attempted to change several lanes in the course of following a routing decision. There were two identified causes of this behavior. In the first case, the routing decision was directing too many vehicles onto a particular movement and overloaded the desired lane. In the second case, the route was defined in a way that did not allow adequate time for vehicles to change into the correct lane. Also, inaccurate geometry restricted lane changes in a way not present in the field and prevented vehicles from following their routes. In each case an extensive investigation was conducted to determine the cause of the erratic lane changes, and corrections were made.

Routing Decision Changes

Often times routing decisions were defined in a way that would not allow vehicles enough space/time to get into the correct lane to make their turning movement. In the Cobb Parkway model each intersection has routing decisions defined for each approach with destination points defined for each movement associated with each approach.

As changing routing decisions is relatively easier than changing the geometry of the model this was the initial and preferred method for resolving lane-change and turning movement volume issues. Many routing problems arose when the decision point was not placed sufficiently far upstream of the intersection for vehicles to make the appropriate lane changes.

In attempting to resolve lane change issues caused by short routing decisions the first action was to move the routing decision as far back as possible and define the lane change distance as least as far back as the decision point. While this resolved many of the routing issues, there was insufficient distance between some intersections to allow for the lane changes even when the decision point was placed as close to the upstream intersection as possible. In these cases the routing decision for a particular approach had to be defined for each of the approaches of the upstream intersection that resulted in vehicles being routed onto the downstream approach. Usually this resulted in three separate but identical routing decisions being defined on left, right, and through movements of the upstream intersection. This allowed vehicles more time to change lanes as they were "aware" of their destination even before they passed through the upstream intersection.

This was done for the following intersections:

- Cobb Parkway @ Hargrove/Herodian – Northbound
- Cobb Parkway @ Windy Ridge – Southbound
- Cobb Parkway @ Spring Road/Circle 75 – Northbound
- Cobb Parkway @ I-285 Westbound – Northbound (Pushed back through Cobb Parkway @ Galleria Drive North)
- Cobb Parkway @ I-285 Eastbound – Southbound (Pushed back through Cobb Parkway @ Spring Road/Circle 75)
- Cobb Parkway @ Galleria Drive North – Southbound
- Cobb Parkway @ Galleria Drive South – Southbound
- Cobb Parkway @ Aikers Mill – Eastbound

Geometric Changes

Unfortunately not all of the issues that arose in the course of the verification process could be resolved with routing changes. In a few cases geometric changes had to be made to the model either to correct turning movements and lane change issues or to more accurately reflect field behavior in the model.

Turn Bay Lengths

The initial model was constructed on top of an aerial photo of the corridor with many of the turn bay lengths matching exactly the turn bay length as painted on the corridor. Based on oil markings and field observations it was determined that often drivers on the study network were using the center two-way turn lane as additional left turn bay storage space. Thus, the model was modified to more accurately capture this where applicable.

I-285 Interchange

Extensive geometric modification occurred in the vicinity of the I-285/Cobb Parkway interchange and surrounding intersections, particularly the left-turn movement from Cobb Parkway southbound onto I-285 eastbound. The turn bay in this case starts as a one-lane conversion of a through lane upstream of the intersection of Cobb Parkway and the Spring/Circle 75 intersection. After passing through that intersection the turn bay widens to two lanes. As painted, the vehicles entering the turn bay upstream of Spring Road are directed into the inside lane of the turn bay. The added lane is open to additional vehicles seeking to make the left turn movement onto I-285 eastbound, particularly those turning right from eastbound Spring Road. In the field it was observed that many vehicles traveling southbound on Cobb Parkway would not enter the extended turn lane upstream of Spring Road, instead attempting to jump the queue by entering the second turn lane after passing through the Spring Road intersection. It was also noted that a significant portion of the right-turning traffic from Spring Road eastbound would turn across all lanes of Cobb Parkway to enter the left turn bay in preparation to turn left onto I-285 eastbound later.

To model this behavior separate links were created for the two-lane portion of the turn bays. Access to those links was provided using connectors consistent with field observations of traffic lane change behavior. Routing decisions were created that reflected the number of vehicles using the additional turn lane to jump the queue. Also, an additional connector was implemented for right-turns off Spring Road eastbound into the left turn bay to better reflect the field observations.

Shorten Intersections

Because there are only 11 real-world controllers to use to control intersections it was decided to eliminate 4 of the intersections from the original 15-intersection base model. The final simulation model is consistent with the study area shown in Figure 18 and Figure 19. It was also decided that, for analysis purposes, the northernmost intersections (Lake Park Drive to Aikers Mill) would be most useful. A careful analysis was conducted to determine what input volumes would be required to ensure that the proper northbound traffic input would be maintained and the southern four intersections were eliminated from the model.

Blocking Movements

At some intersections in the model it was observed that queues from downstream intersections would spill back and block side street movements. VISSIM is unable to model vehicle blocking across different links/connectors unless explicitly informed that vehicles on one link or connector would be blocking another. Additionally, field observations showed that when a queue would spill back vehicles would usually wait at the stop bar, even on a green, for appropriate clearance before entering the intersection. Particularly this behavior was noticed when the drivers were able to see the downstream signal for their desired movement and therefore appropriately judge if they could safely enter the intersection. This behavior was modeled using a combination of conflict areas to prevent vehicle "collisions" and priority rules. Depending on the geometry of the intersection the priority rules would only become active when the downstream signal head was red, preventing vehicles from entering the intersection until there was sufficient room to clear the conflicting movements.

Vehicle Distributions

In the course of this validation it was decided to modify the vehicle type distributions to exclude heavy vehicles. Based on the provided counts and field observations it was determined that there were not a significant number of heavy vehicles present on the study corridor during the analysis periods. Additionally, the inclusion of heavy vehicles had significant adverse effects on lane changes and navigation for other vehicles in the model and created traffic patterns that were not observed in the field.

Max Inhibit

There were a few adjustments that had to be made to the NEMA controller implementation to ensure that it accurately reflected the real-world. In the course of validation it was determined that the "Max Inhibit" option for the NEMA controllers did not perform as expected. Specifically, the option did not prevent the controller from limiting the length of time given to a particular phase based on the max green time. In this study, the analysis centered around the peak hour when the provided timing sheets specifically indicated that the field controllers were set to ignore the max greens. Additionally, it was noticed that the emulated controllers when failing to ignore the max green times performed very poorly, giving far too much green time to relatively low-volume side streets. Based on these findings it was determined that the max green times for each phase would need to be adjusted up to equal the total cycle length for each controller effectively forcing the controller into "Max Inhibit".

Per-Intersection Model Review

What follows is a detailed review of each of the 11 intersections studied along this corridor. For each intersection an aerial photo is provided along with a brief description of the modeling methodology used to represent the intersection in the simulation. Simplifications or departures from the observed conditions are identified and explained where they exist. Also figures showing the detector layout for SCATS and ACTRA are provided for each intersection.

Intersection #1: Cobb Parkway @ Lake Park Drive

The intersection of Cobb Parkway and Lake Park Drive is shown in Figure 20. Cobb Parkway runs north-south while Lake Park Drive runs east-west.

Figure 20. Cobb Parkway @ Lake Park Drive Aerial
(Figure Credit: Google Earth)

Southbound the intersection has one exclusive left turn lane, two through lanes, and one channelized free flow right turn lane. Northbound the intersection has one exclusive left turn lane, two through lanes, and one channelized free flow right turn lane. Eastbound the intersection has one exclusive left turn lane, one through lane, and one channelized free flow right turn lane. Although the through lane on this approach was marked in the field as a shared through/right this behavior did not seem logical given the geometry of the approach so it was modeled as a through only. Westbound the intersection has one exclusive left turn lane and one shared through/right turn lane.

The detector configuration for SCATS is shown in Figure 21 and the detector configuration for ACTRA is shown in Figure 22.

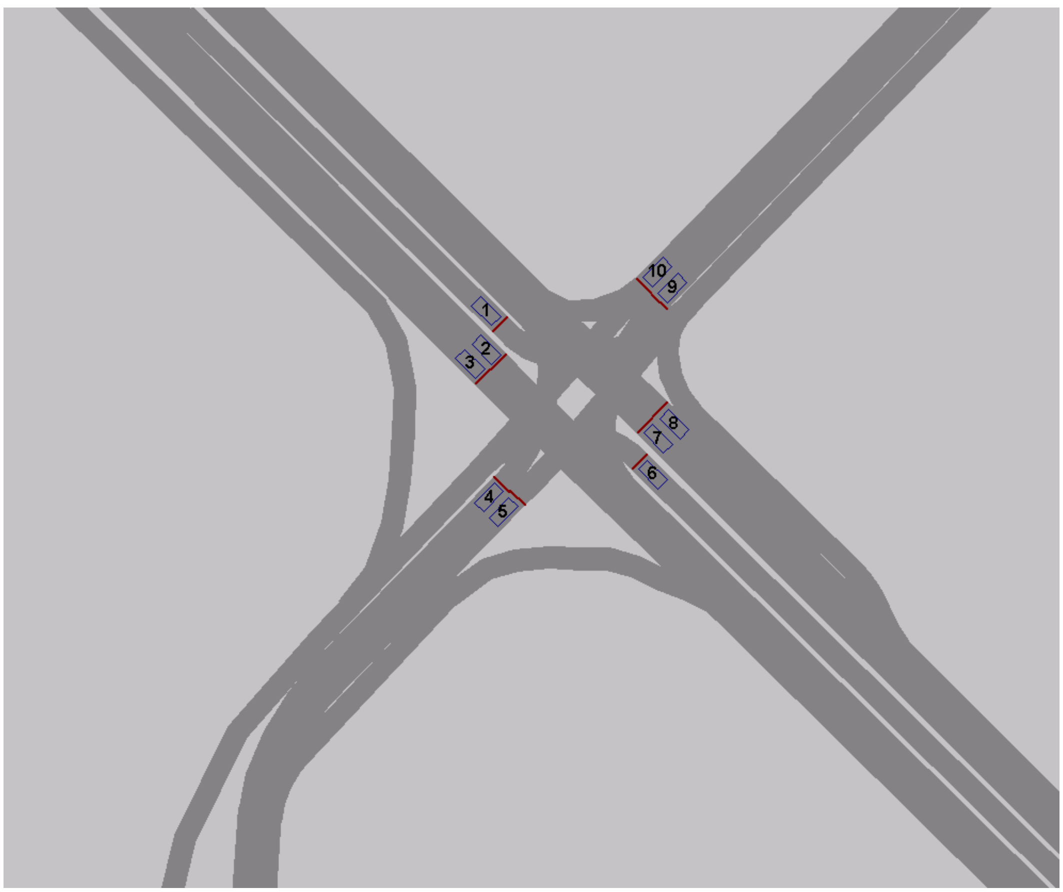

Figure 21. Cobb Parkway @ Lake Park Drive SCATS

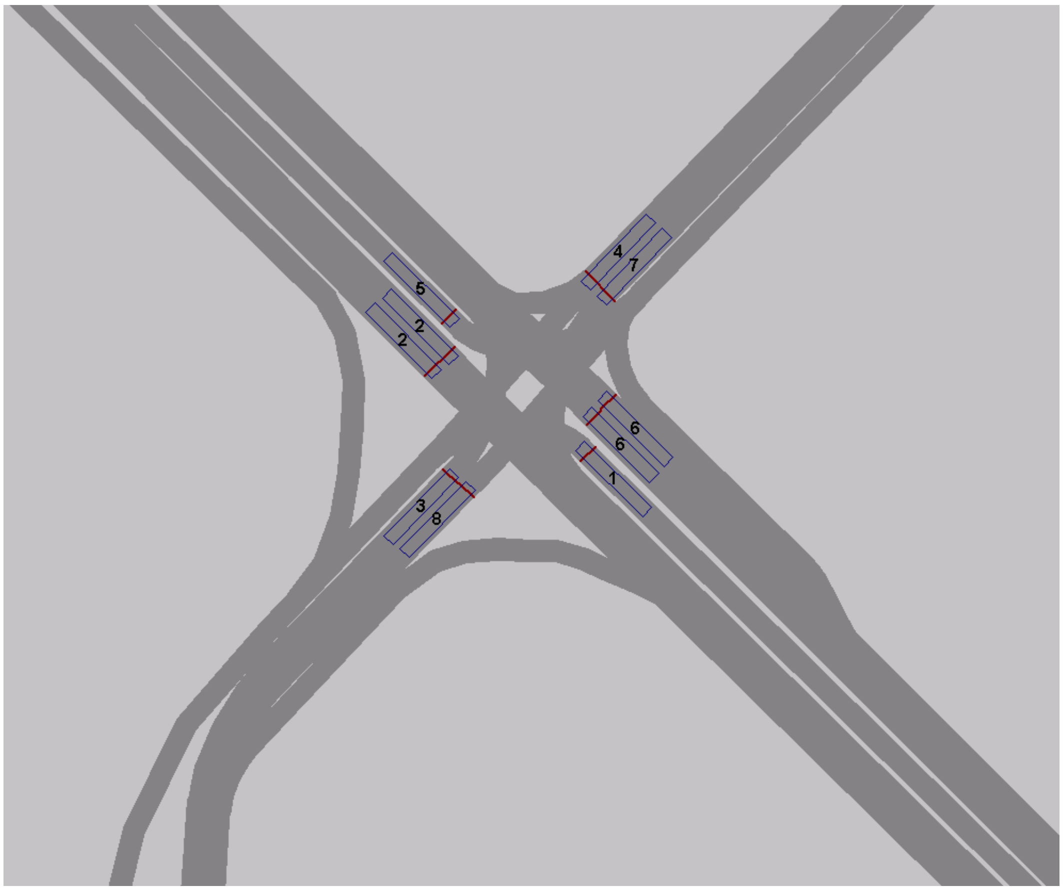

Figure 22. Cobb Parkway @ Lake Park Drive ACTRA

Intersection #2: Cobb Parkway @ Circuit City/Kenwood Creek

The intersection of Cobb Parkway and Circuit City/Kenwood Creek is shown in Figure 23. Cobb Parkway runs north-south while the Circuit City driveway approaches from the west and the Kenwood Creek driveway approaches from the east.

Figure 23. Cobb Parkway @ Circut City/Kenwood Creek Aerial (Figure Credit: Google Earth)

Southbound the intersection has one exclusive left turn lane, two through lanes, and one channelized free flow right turn lane. Northbound the intersection has one exclusive left turn lane, two through lanes, and one channelized free flow right turn lane.

Eastbound the intersection has one shared left turn/through lane and one channelized free flow right turn lane. Westbound the intersection has one shared left turn/through lane and one channelized free flow right turn lane.

The detector configuration for SCATS is shown in Figure 24 and the detector configuration for ACTRA is shown in Figure 25.

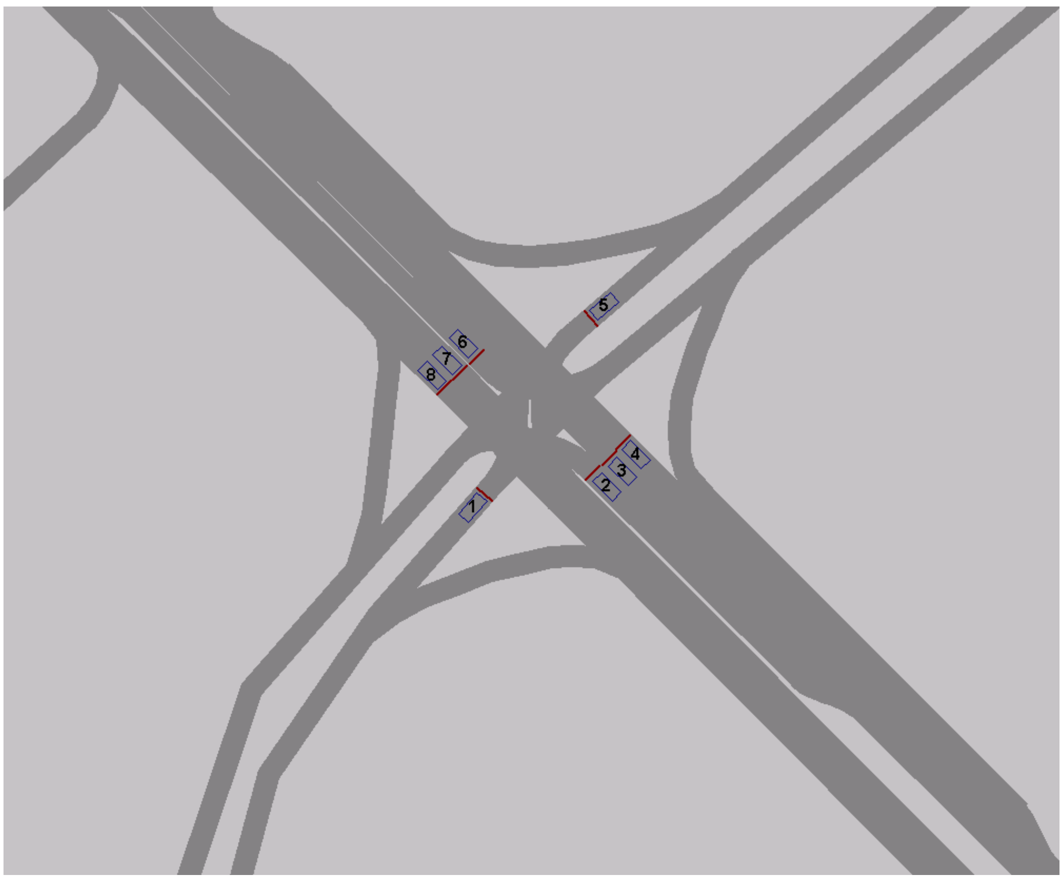

Figure 24. Cobb Parkway @ Circut City/Kenwood Creek SCATS

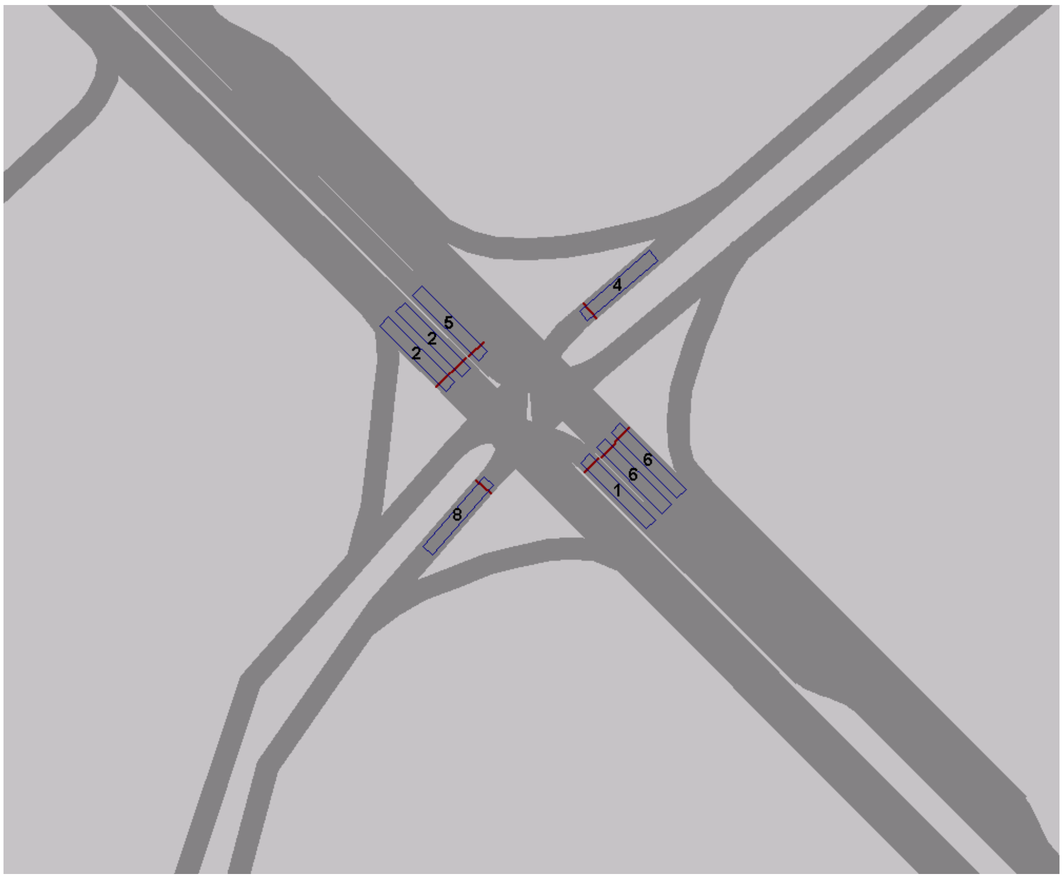

Figure 25. Cobb Parkway @ Circut City/Kenwood Creek ACTRA

Intersection #3: Cobb Parkway @ Lohemans Plaza

The intersection of Cobb Parkway and Lohemans Plaza is shown in Figure 26. Cobb Parkway runs north-south while the shopping center driveways (including Lohemans Plaza) run east-west.

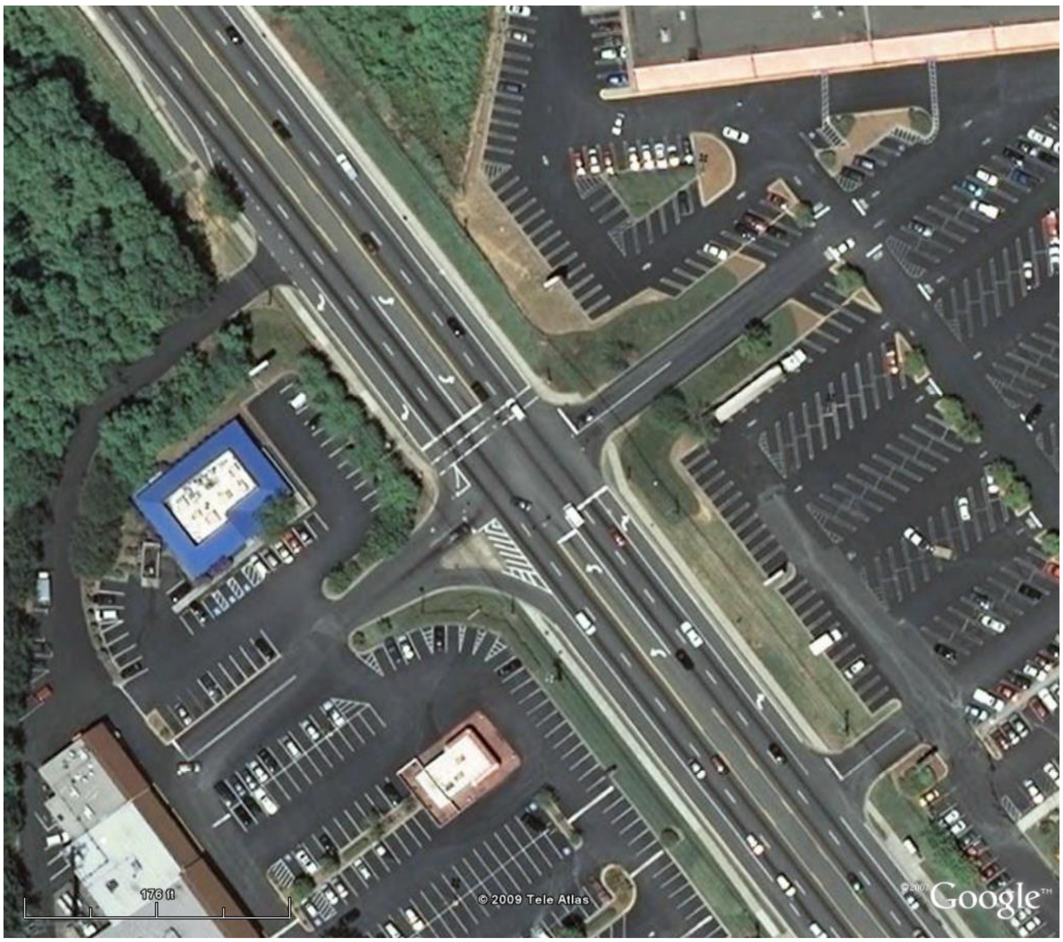

Figure 26. Cobb Parkway @ Lohemans Plaza Aerial
(Figure Credit: Google Earth)

Southbound the intersection has one exclusive left turn lane, two through lanes, and one exclusive free flow channelized right turn lane. Northbound the intersection has one exclusive left turn lane, two through lanes, and one exclusive free flow right turn lane. Even though the markings in the field indicate that there is a stop bar for the right turn, the right turn on red was modeled using a free flow connector with appropriate priority rules. Eastbound the intersection has one shared left turn/through lane, and one exclusive free flow channelized right turn lane. Westbound the intersection has one shared left turn/through/right turn lane.

The detector configuration for SCATS is shown in Figure 27 and the detector configuration for ACTRA is shown in Figure 28.

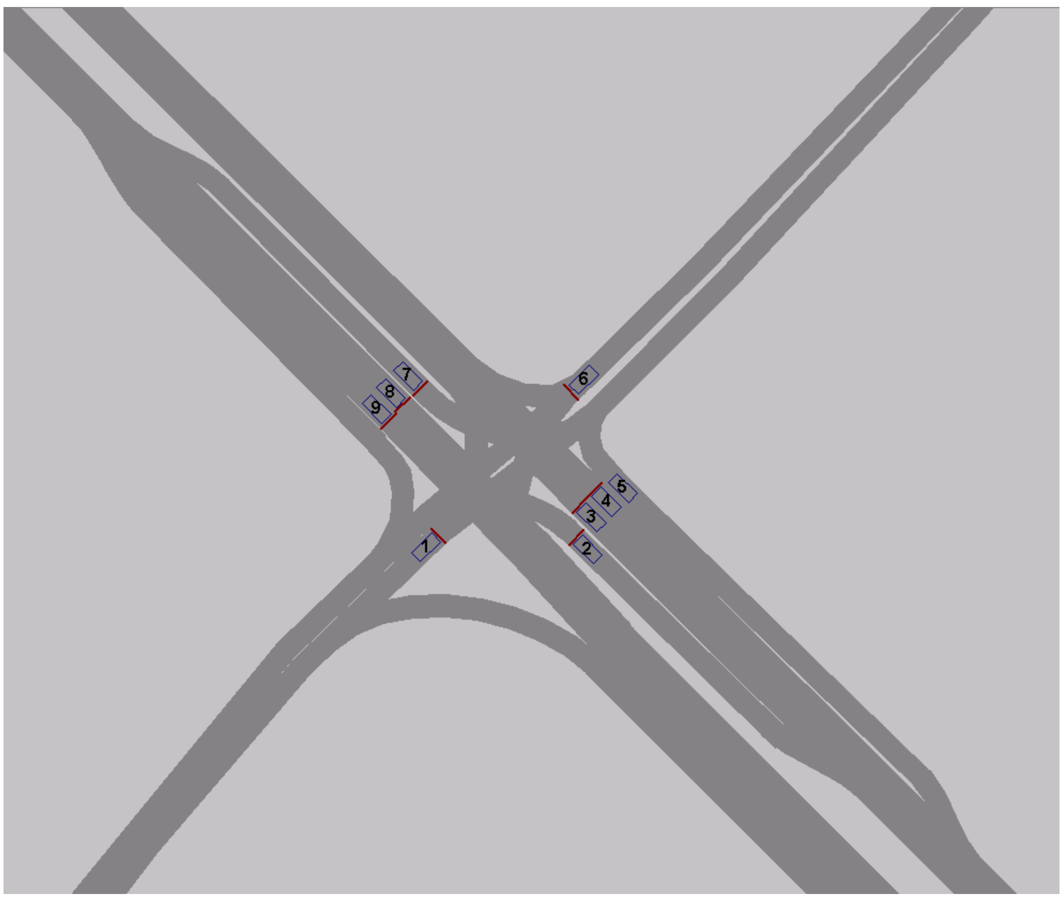

Figure 27. Cobb Parkway @ Lohemans Plaza SCATS

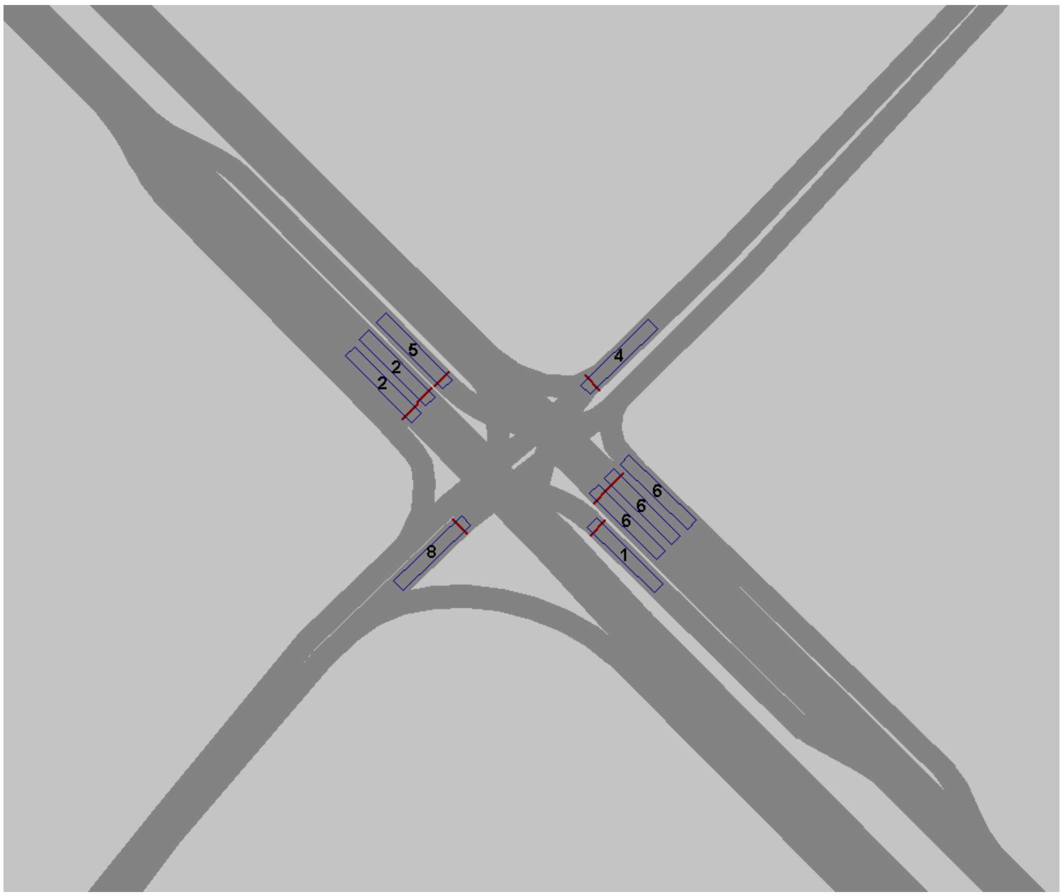

Figure 28. Cobb Parkway @ Lohemans Plaza ACTRA

Intersection #4: Cobb Parkway @ Hargrove Road/Herodian Way

The intersection of Cobb Parkway and Hargrove Road/Herodian Way is shown in Figure 29. Cobb Parkway runs north-south while Hargrove Road approaches from the west and Herodian Way approaches from the east.

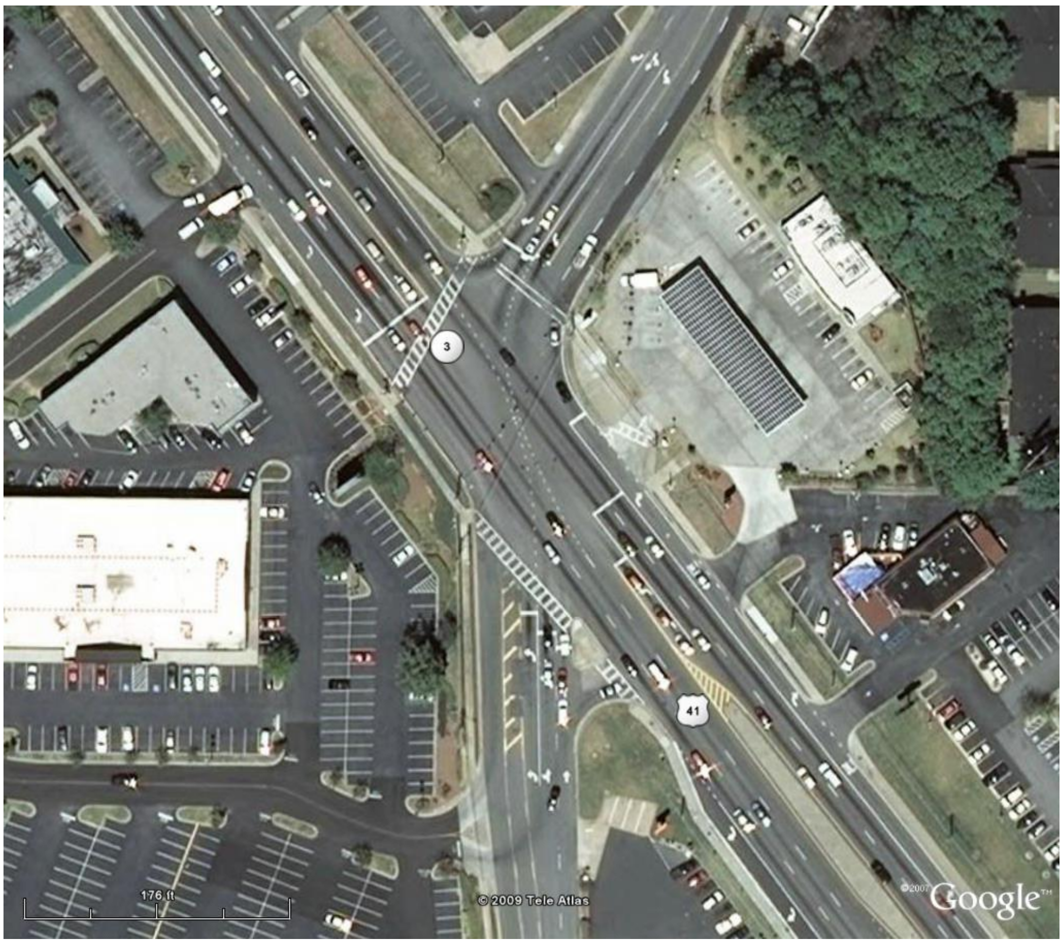

Figure 29. Cobb Parkway @ Hargrove Road/Herodian Way Aerial (Figure Credit: Google Earth)

Southbound the intersection has one exclusive left turn lane, two through lanes, and one exclusive free flow right turn lane. The southbound left-turn bay was extended north beyond what is marked in the field because vehicles queuing for that movement were observed waiting in the two-way left turn lane. Northbound the intersection has one exclusive left turn lane, two through lanes, and one exclusive free flow right turn lane. The northbound routing decision was pushed back through the upstream intersection to allow vehicles routed to the sink between the Windy Ridge intersection and this intersection enough room to make the appropriate movement. Even though the markings in the field indicate that there is a stop bar for the southbound and northbound right turns, the right turn on red was modeled using a free flow connector with appropriate priority rules. Eastbound the intersection has one exclusive left turn lane, two through lanes, and one exclusive channelized free flow right turn lane. Although the left most through lane is marked as a shared left turn/through it was not modeled that way because it did not appear to be a significant movement. Westbound the intersection has one exclusive left turn lane, one through lane, and one exclusive right turn lane. Although the through lane is marked as a shared left turn/through it was not modeled that way because it did not appear to be a significant movement.

The detector configuration for SCATS is shown in Figure 30 and the detector configuration for ACTRA is shown in Figure 31.

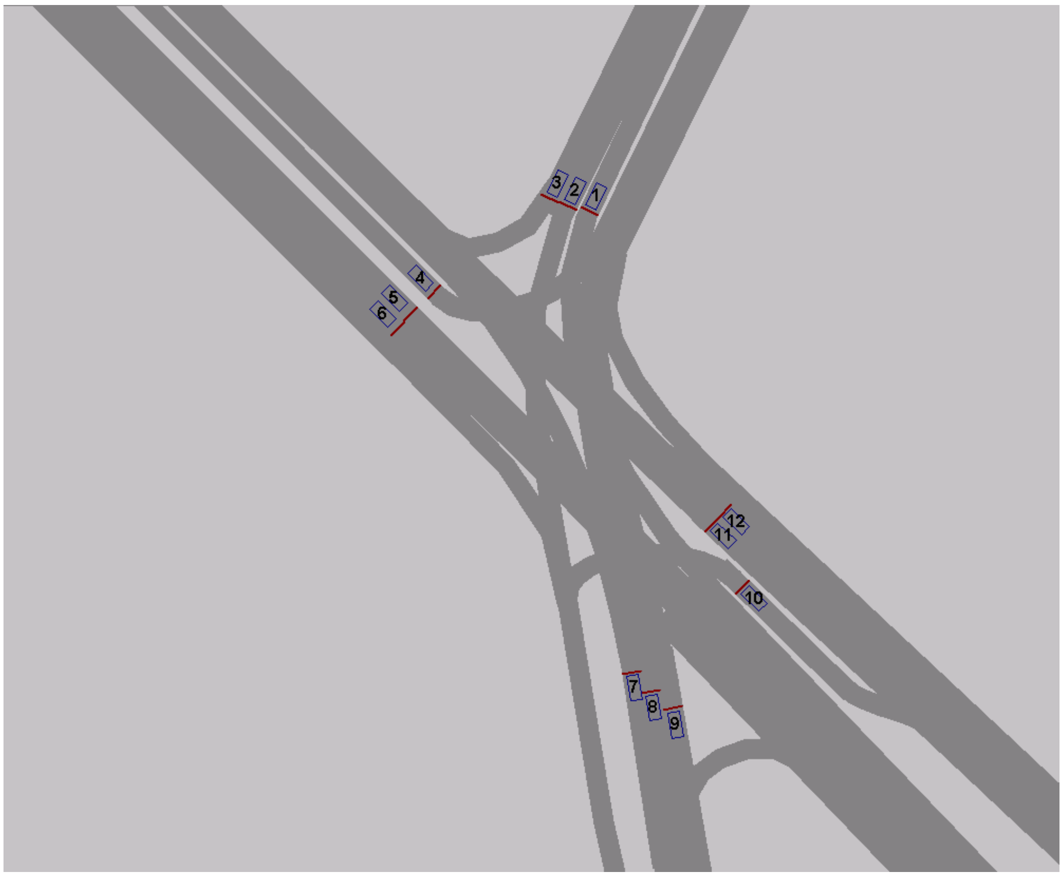

Figure 30. Cobb Parkway @ Hargrove Road/Herodian Way SCATS

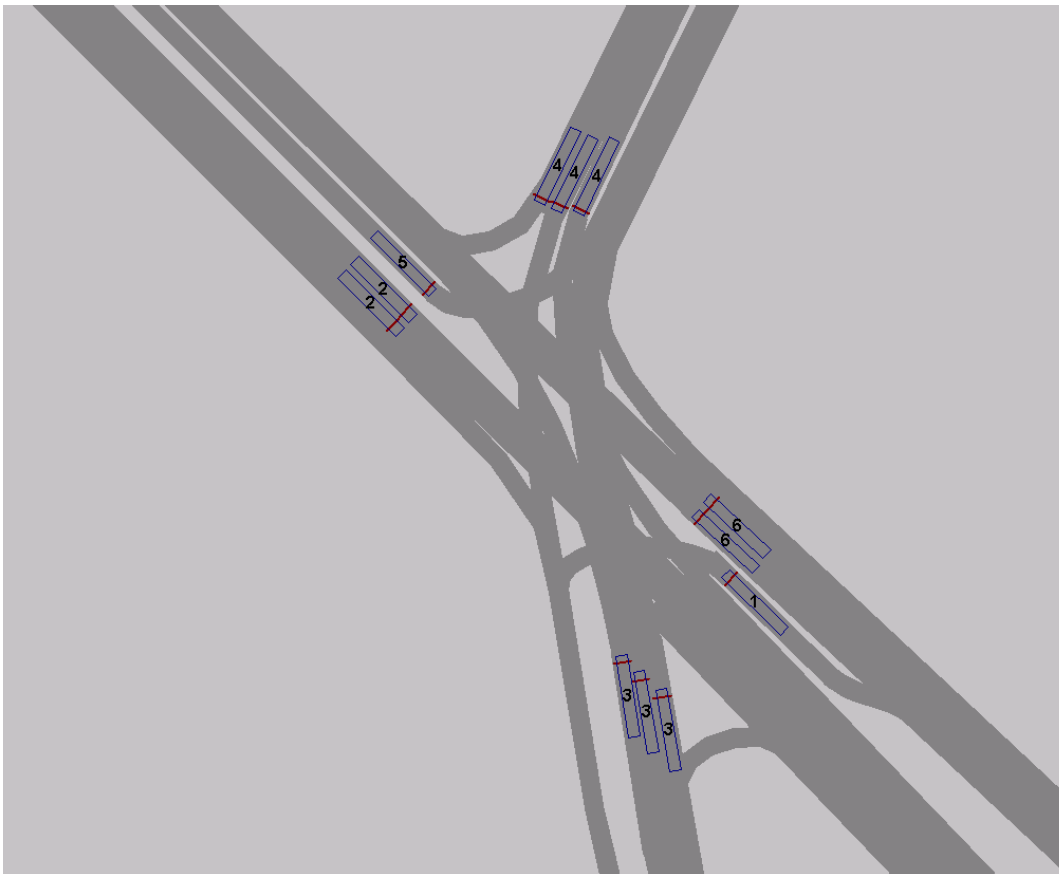

Figure 31. Cobb Parkway @ Hargrove Road/Herodian Way ACTRA

Intersection #5: Cobb Parkway @ Cumberland Boulevard/Windy Ridge Parkway

The intersection of Cobb Parkway and Cumberland Boulevard/Windy Ridge Parkway is shown in Figure 32. Cobb Parkway runs north-south while Cumberland Boulevard approaches from the west and Windy Ridge Parkway approaches from the east.

Figure 32. Cobb Parkway @ Cumberland Blvd./Windy Ridge Pkwy. Aerial (Figure Credit: Google Earth)

Southbound the intersection has one exclusive left turn lane, four through lanes, and one exclusive free flow right turn lane. The southbound routing decision was pushed back through the upstream intersection to accommodate vehicles routed onto the sink in between the intersection at Hargrove/Herodian and this intersection. Northbound the intersection has one exclusive left turn lane, three through lanes, and one exclusive free flow right turn lane. Eastbound the intersection has two exclusive left turn lanes, two through lanes, and one exclusive free flow right turn lane. Even though the markings in the field indicate that there is a stop bar for the southbound, northbound, and eastbound right turns, the right turn on red was modeled using a free flow connector with appropriate conflict areas. Westbound the intersection has one exclusive left turn lane, one through lane, and one shared through/right turn lane.

The detector configuration for SCATS is shown in Figure 33 and the detector configuration for ACTRA is shown in Figure 34.

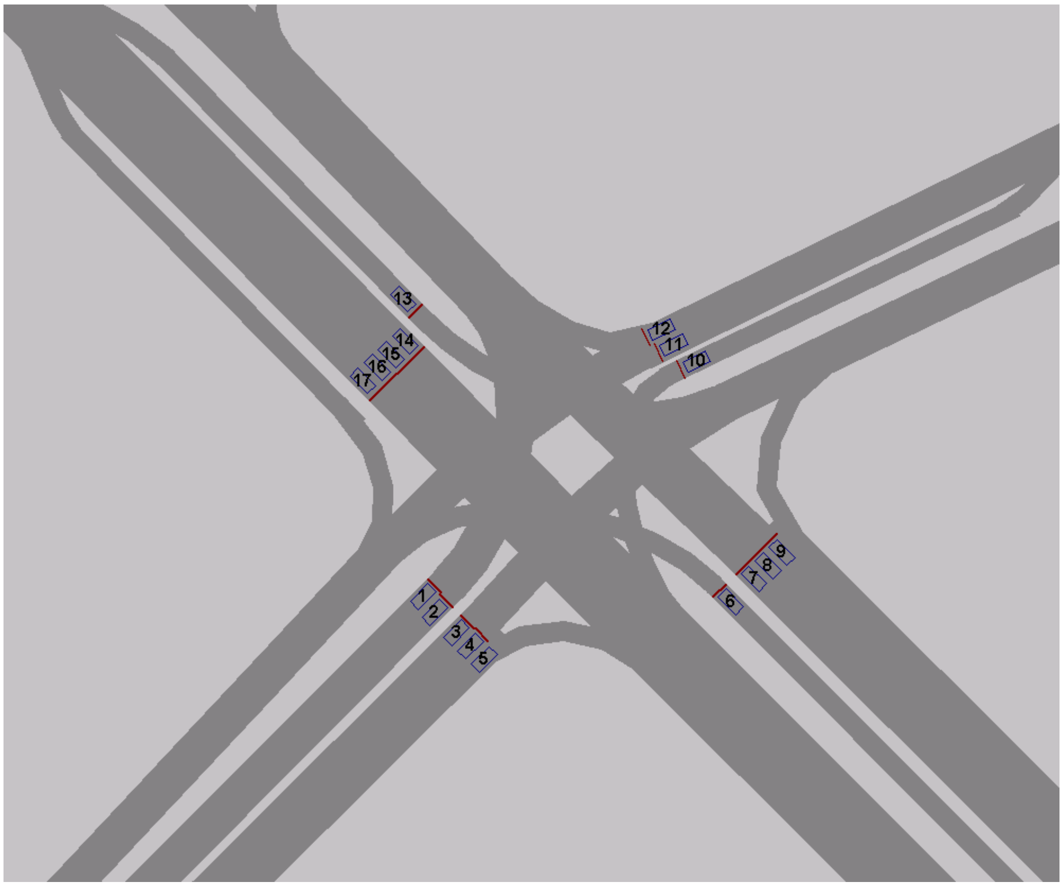

Figure 33. Cobb Parkway @ Cumberland Blvd./Windy Ridge Pkwy. SCATS

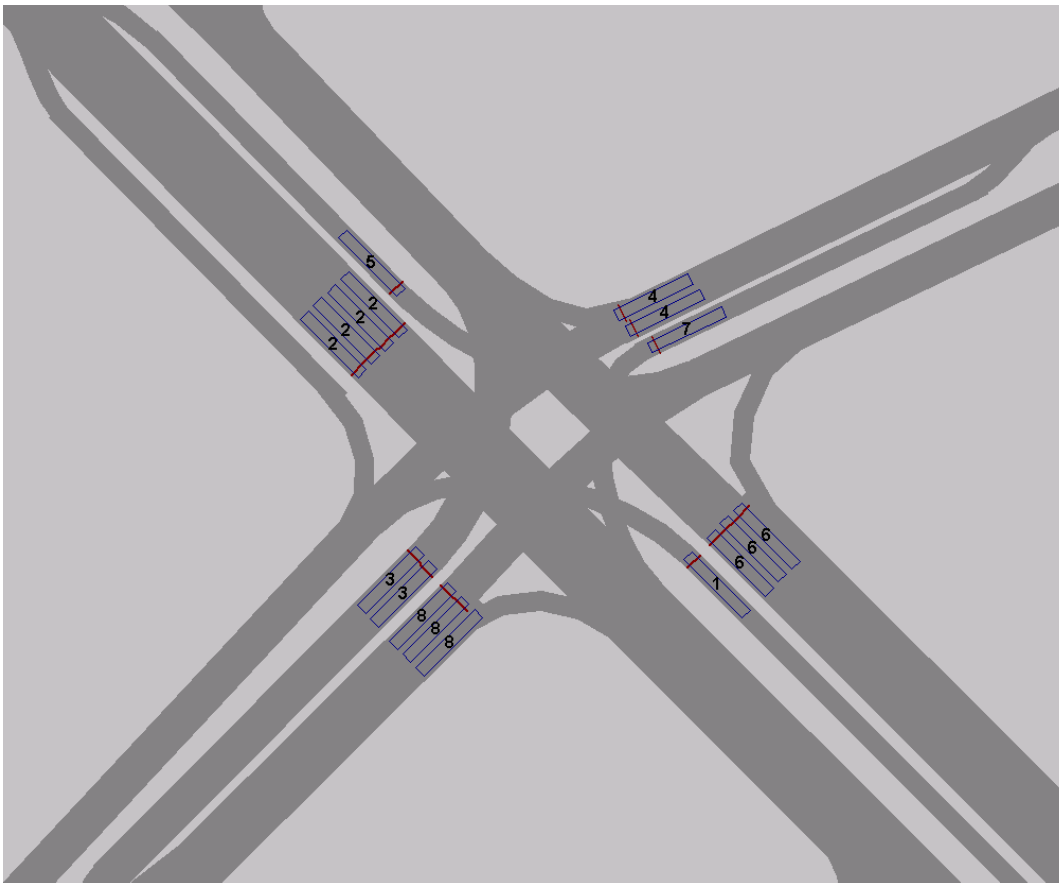

Figure 34. Cobb Parkway @ Cumberland Blvd./Windy Ridge Pkwy. ACTRA

Intersection #6: Cobb Parkway @ Spring Road/Circle 75 Parkway

The intersection of Cobb Parkway and Spring Road/Circle 75 Parkway is shown in Figure 35. Cobb Parkway runs north-south while Spring Road approaches from the west and Circle 75 approaches from the east.

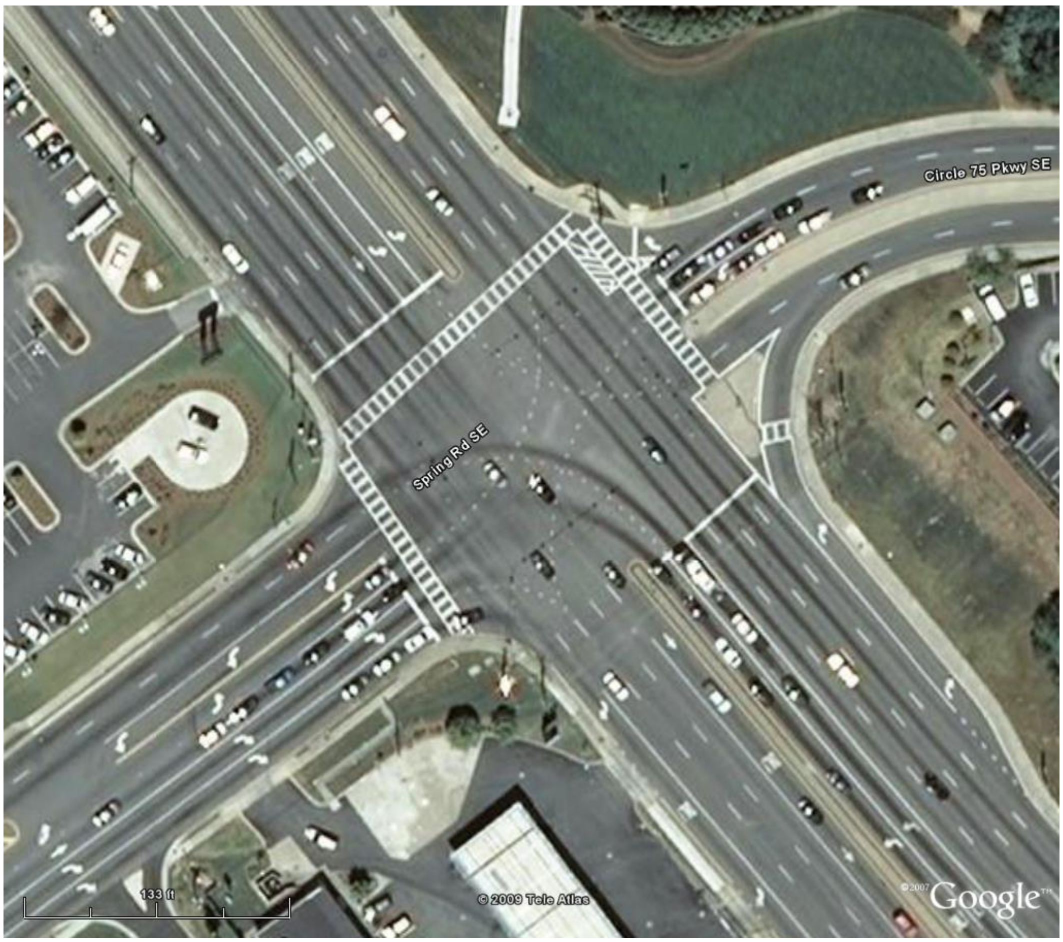

Figure 35. Cobb Parkway @ Spring Road/Circle 75 Pkwy. Aerial (Figure Credit: Google Earth)

Southbound the intersection has 2 exclusive left turn lanes, 4 through lanes, and one right/through lane. As can be seen in Figure 35 the leftmost southbound through lane is targeted at the downstream exclusive left turn for the I-285 eastbound on-ramp. This turn bay extension was initially modeled as a separate link, but observation of model performance and the field led to changes as detailed in the section dealing with geometric changes on page 91. Northbound there are 2 exclusive left turn lanes, 4 through lanes, and 1 free flow channelized right turn lane. The routing decision for this approach had to be pushed back through the upstream intersection because there was not enough room for vehicles passing through to enter the left turn bay for this intersection. Eastbound the intersection has 1 exclusive left turn lane, 1 left/through lane, and 1 free flow channelized right turn lane. It was observed that there were a significant number of vehicles turning right into the far left lanes of Cobb Parkway southbound to make the left turn onto I-285 eastbound. A link and routing decision was added to accommodate that movement. Also, the length of the right turn lanes was significant enough that they were modeled as a separate link from the other movements for this approach and have their own routing and volume input. Westbound the intersection has 1 exclusive left turn lane, 2 through lanes, and 2 exclusive right turn lanes.

The detector configuration for SCATS is shown in Figure 36 and the detector configuration for ACTRA is shown in Figure 37.

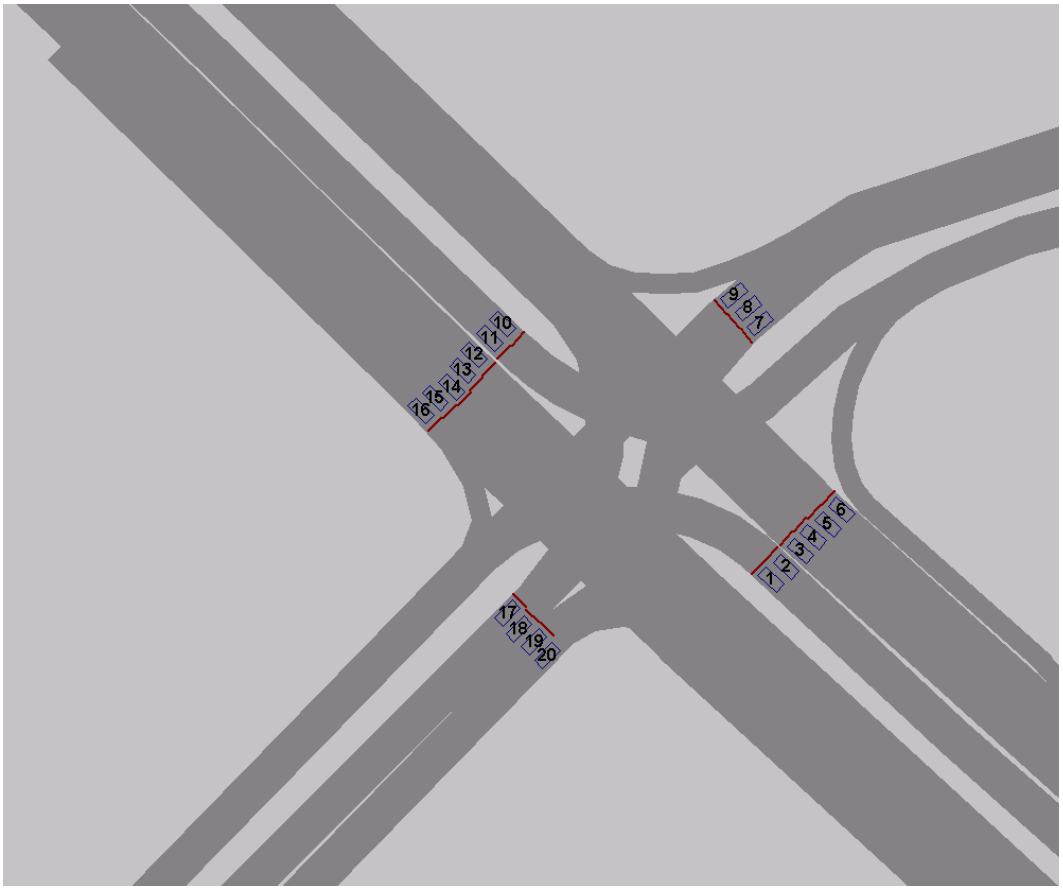

Figure 36. Cobb Parkway @ Spring Road/Circle 75 Pkwy. SCATS

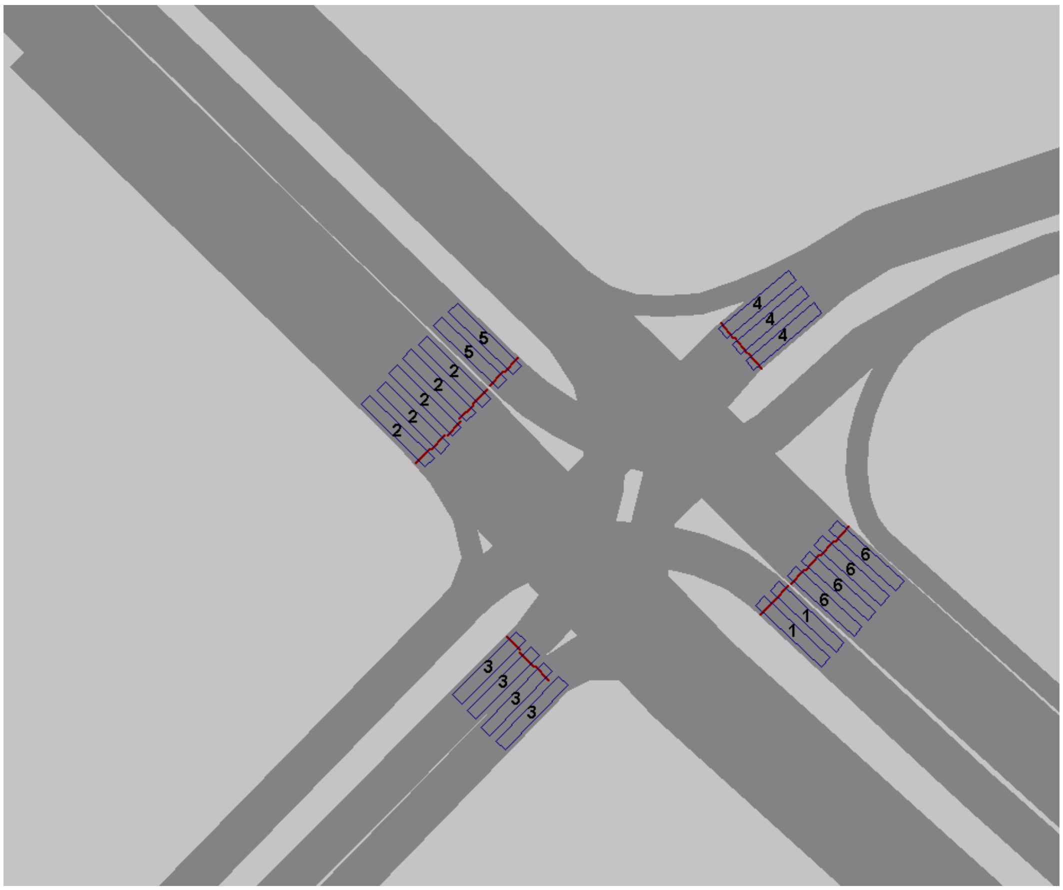

Figure 37. Cobb Parkway @ Spring Road/Circle 75 Pkwy. ACTRA

Intersection #7: Cobb Parkway @ I-285 Westbound

The intersection of Cobb Parkway and I-285 Westbound is shown in Figure 38. Cobb Parkway runs north-south while the I-285 westbound off ramp approaches from the east.

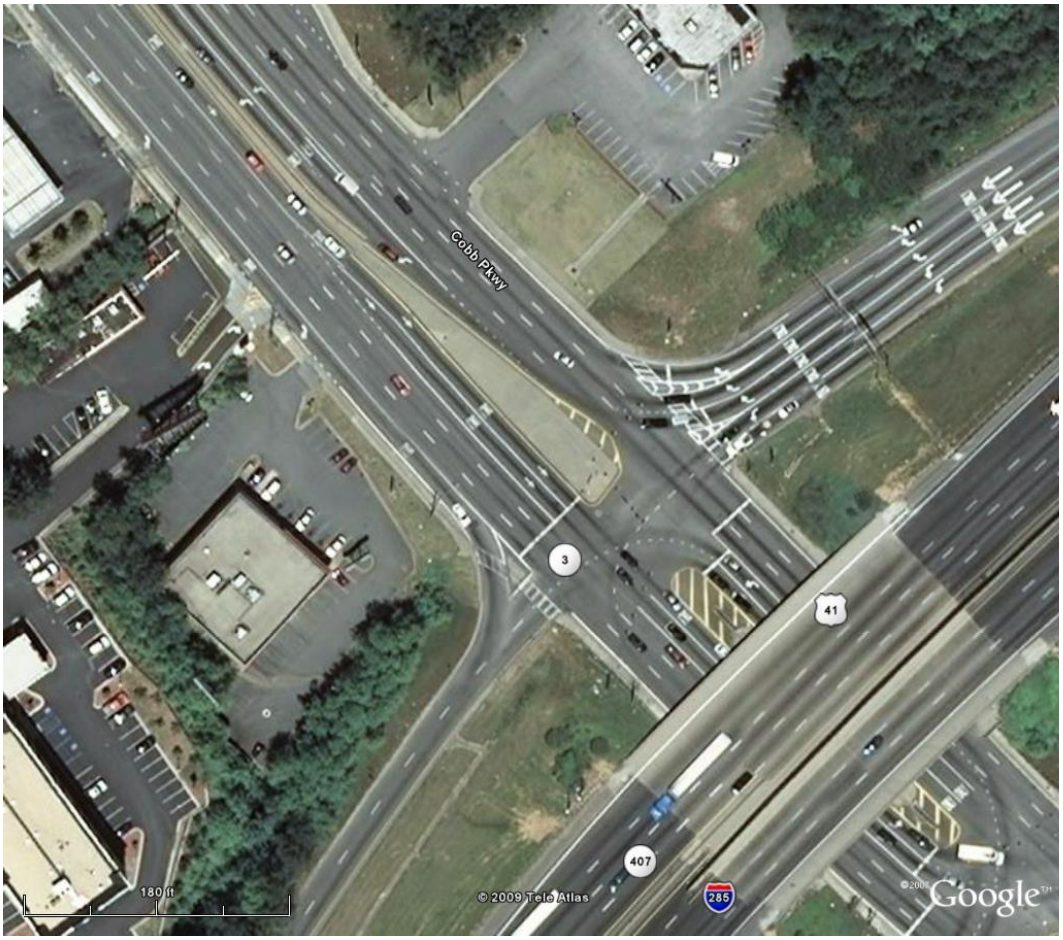

Figure 38. Cobb Parkway @ I-285 Westbound Aerial
(Figure Credit: Google Earth)

Southbound the intersection has five through lanes and one exclusive free flow channelized right turn lane. The leftmost two through lanes are extensions of the left turn bays for the downstream intersection at I-285 eastbound and are modeled as a separate link. This configuration is discussed further on page 91. Additionally there is a priority rule for this link that becomes active when the left turn signal at the I-285 eastbound turns red and prevents the queue for that movement from spilling back through this intersection. This behavior is consistent with what was observed in the field. The southbound right turn uses appropriate conflict areas to model the right turn on red behavior. Northbound the intersection has two exclusive left turn lanes and three through lanes. The routing decision for this movement was pushed back prior to the intersection at Galleria Drive North because the northbound left turn bays were modeled as separate links all the way back through that intersection. This modeling is consistent with markings in the field and observed driving behavior. Westbound the intersection has two exclusive left turn lanes and three exclusive right turn lanes. The left turn connector is modeled so all vehicles turn onto the through link for this intersection as it was assumed that a negligible number of these vehicles would be turning left at the I-285 eastbound intersection.

The detector configuration for SCATS is shown in Figure 39 and the detector configuration for ACTRA is shown in Figure 40.

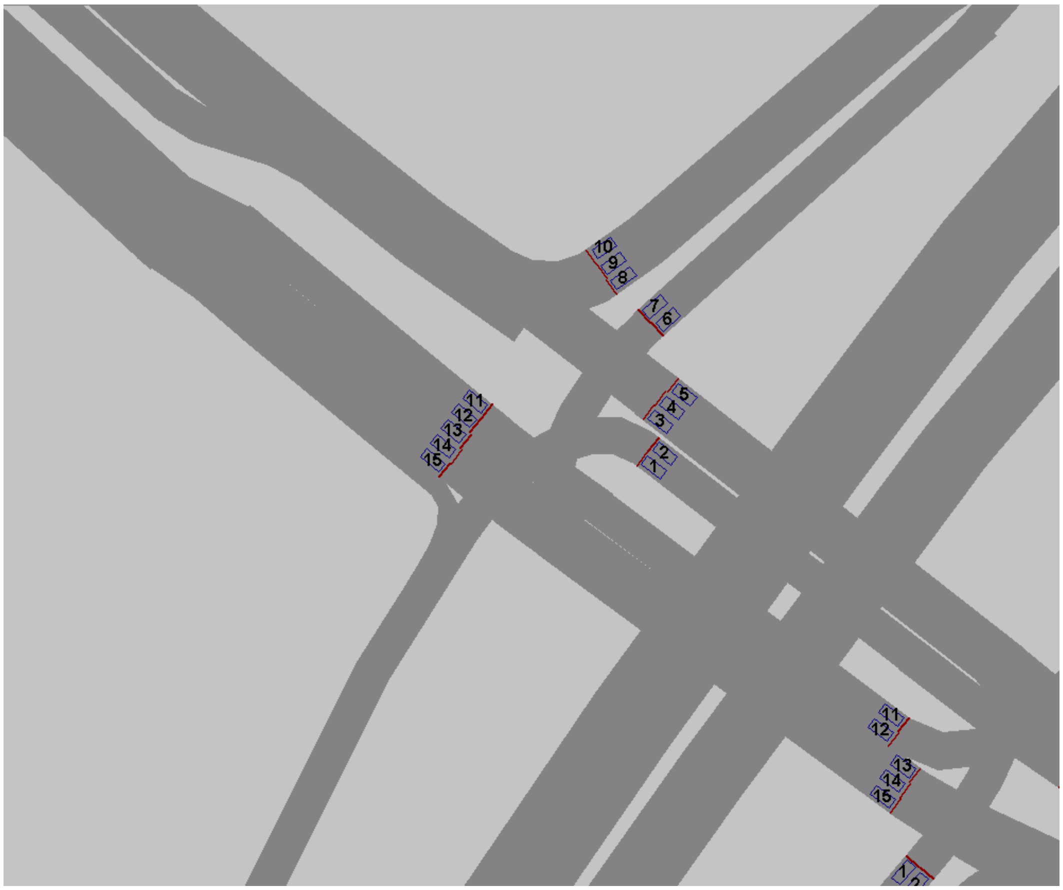

Figure 39. Cobb Parkway @ I-285 Westbound SCATS

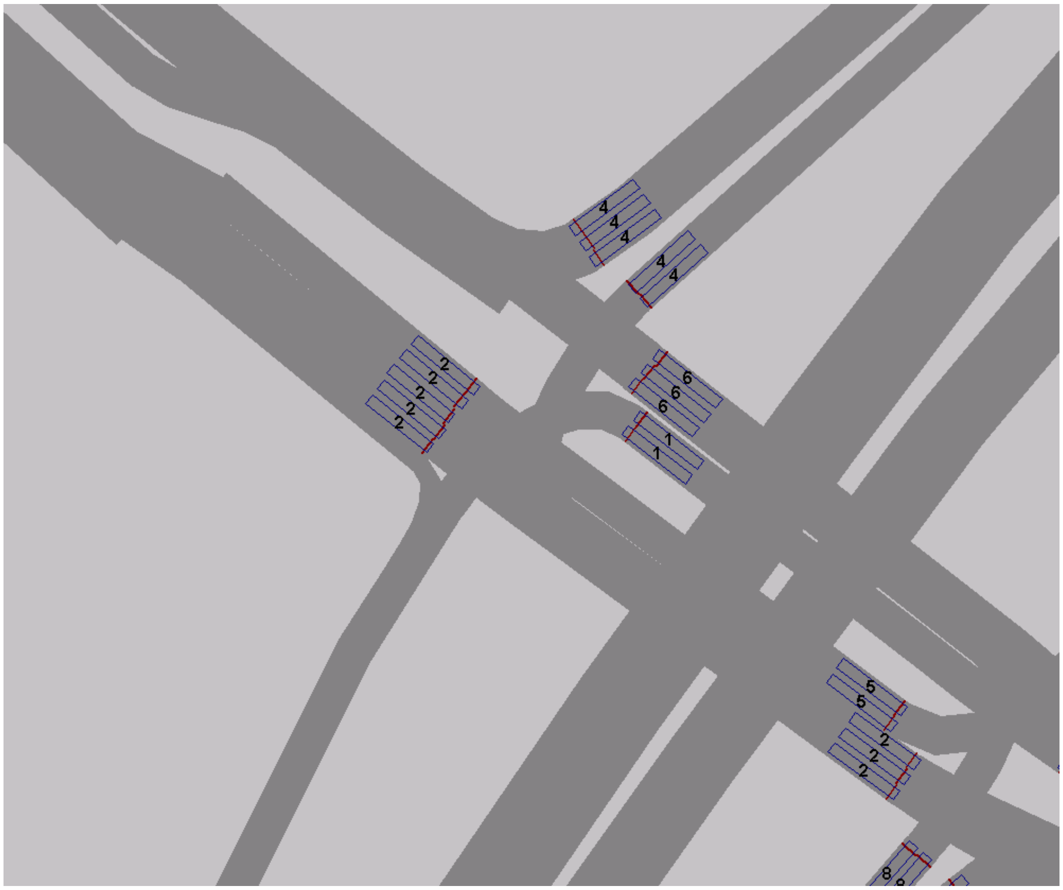

Figure 40. Cobb Parkway @ I-285 Westbound ACTRA

Intersection #8: Cobb Parkway @ I-285 Eastbound

The intersection of Cobb Parkway and I-285 Eastbound is shown in Figure 41. Cobb Parkway runs north-south while the I-285 eastbound off ramp approaches from the west.

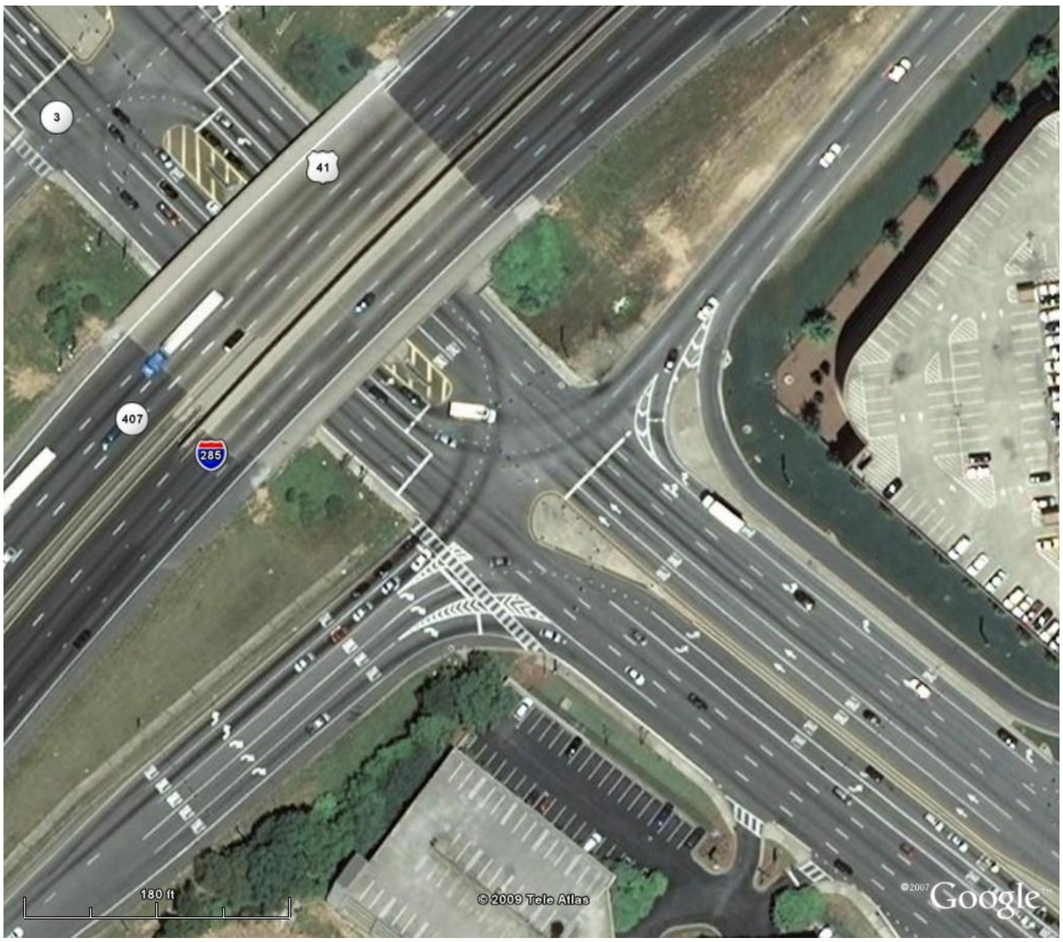

Figure 41. Cobb Parkway @ I-285 Eastbound Aerial
(Figure Credit: Google Earth)

Southbound the intersection has two exclusive left turn lanes and three through lanes. The routing decision for this movement was pushed back prior to the intersection at Spring Road/Circle 75 Pkwy. because the northbound left turn bays were modeled as separate links all the way back through that intersection. This modeling is consistent with markings on in the field and observed driving behavior. Northbound the intersection has five through lanes and one exclusive free flow channelized right turn lane. The leftmost two through lanes are extensions of the left turn bays for the downstream intersection at I-285 westbound and are modeled as a separate link. Although the rightmost through lane is marked as a shared through/right turn it is not modeled this way as that behavior was not significant to the model performance. The northbound exclusive right turn uses appropriate conflict areas to model the right turn on red behavior. Eastbound the intersection has two exclusive left turn lanes and three exclusive right turn lanes. The left turn connector is modeled so all vehicles turn onto the through link for this intersection as it was assumed that a negligible number of these vehicles would be turning left at the I-285 westbound intersection. The right turn movement has two connectors that allow some vehicles to turn directly into the left turn lane link for the southbound approach at Galleria Drive North, the rest of the right turning traffic turns on to the through/right turn link for that intersection. This is consistent with observed field behavior.

The detector configuration for SCATS is shown in Figure 42 and the detector configuration for ACTRA is shown in Figure 43.

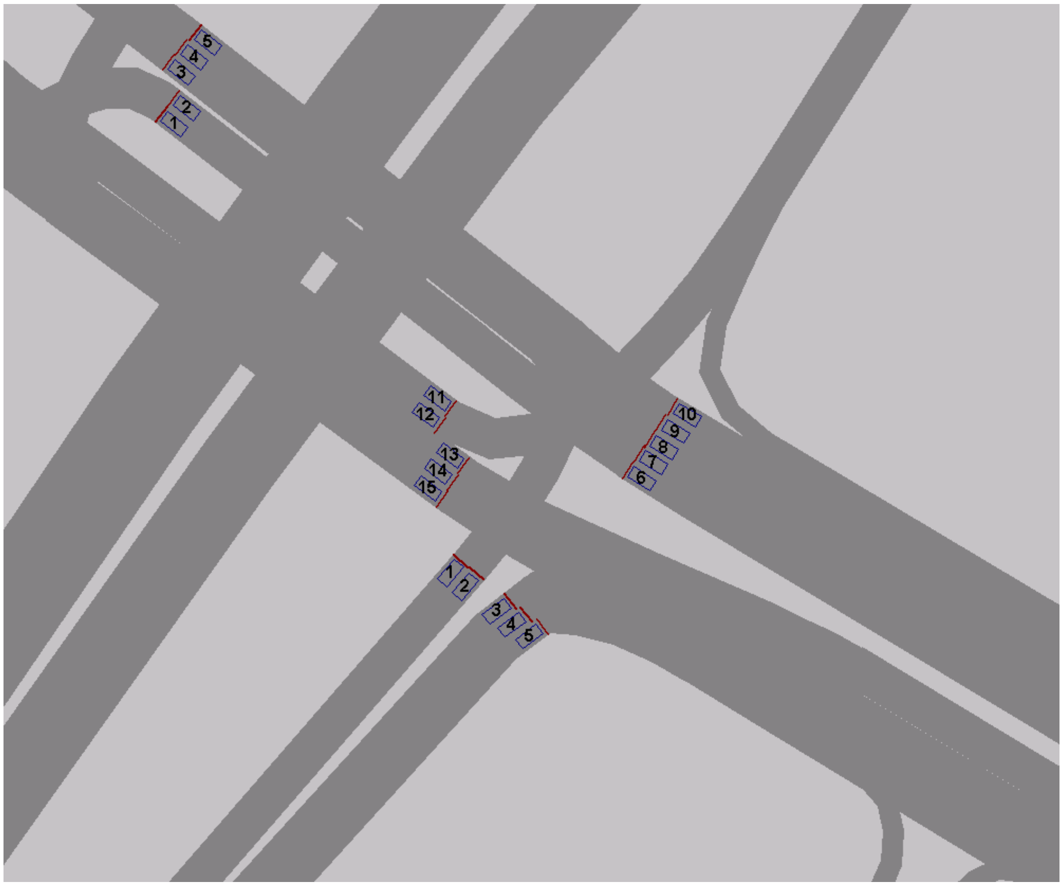

Figure 42. Cobb Parkway @ I-285 Eastbound SCATS

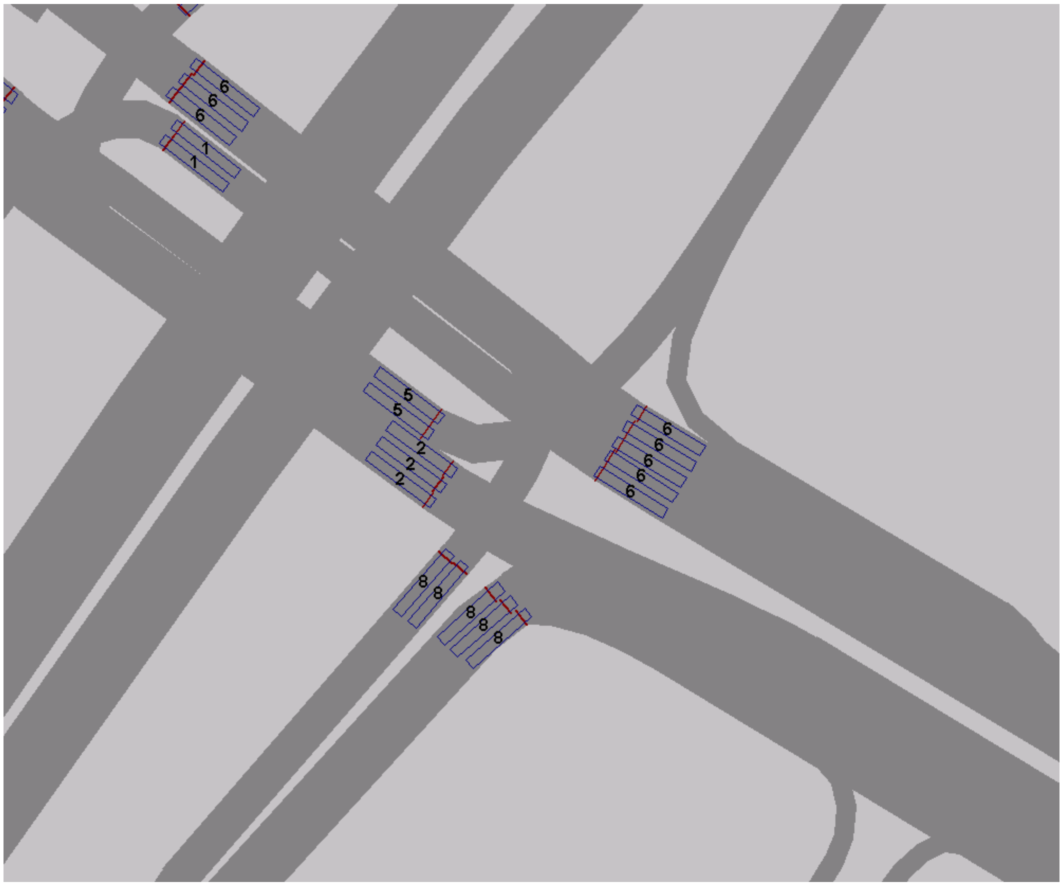

Figure 43. Cobb Parkway @ I-285 Eastbound ACTRA

Intersection #9: Cobb Parkway @ Galleria Drive North

The intersection of Cobb Parkway and Galleria Drive North is shown in Figure 44. Cobb Parkway runs north-south while Galleria Drive North runs east-west.

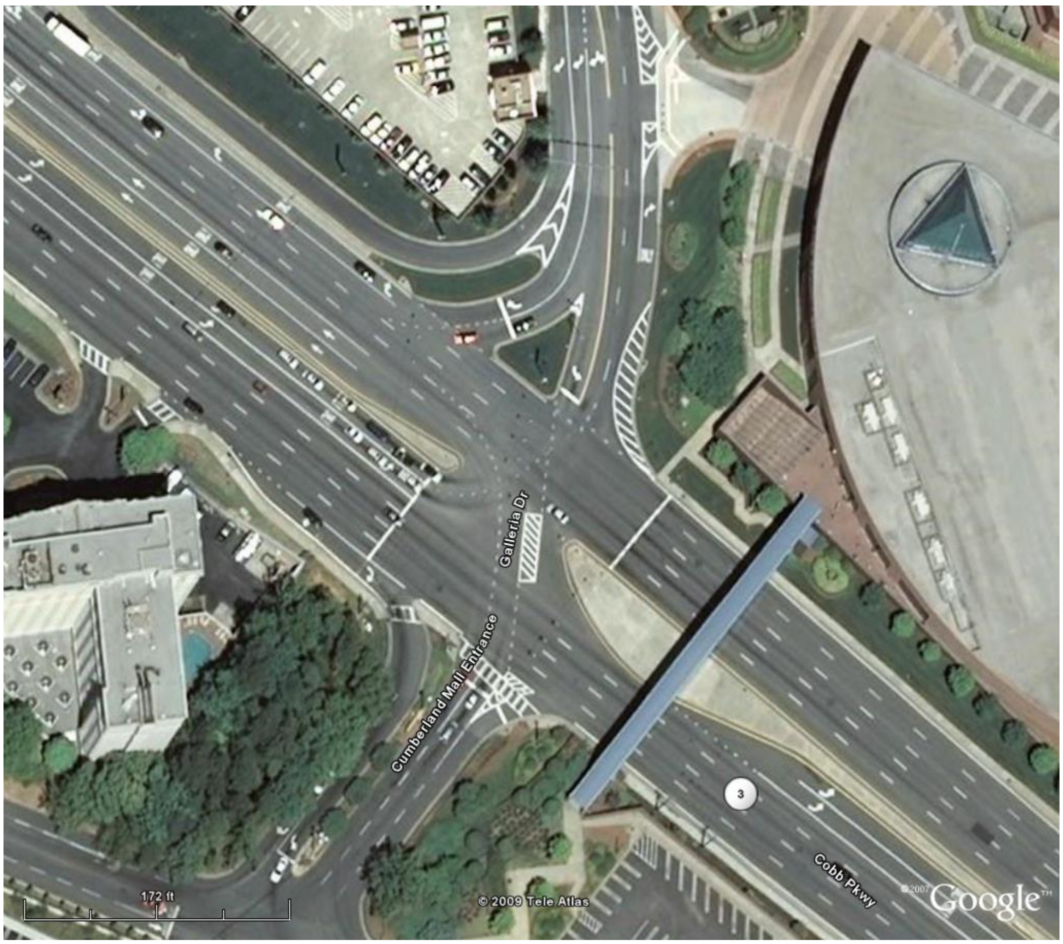

Figure 44. Cobb Parkway @ Galleria Drive North Aerial
(Figure Credit: Google Earth)

Southbound the intersection has two exclusive left turn lanes, four through lanes, and one exclusive right turn lane. The right turn lane is coded for right turns on red with the appropriate priority rules. The routing decision for this movement was pushed back through the intersection at I-285 eastbound because the left turn link for this approach is modeled as a separate link through that intersection. Northbound the intersection has four through lanes and one shared through/right turn lane. The leftmost two through lanes are linked to the turn bay links for the northbound left turn link at the I-285 westbound intersection. This is consistent with marking and observed behavior. Eastbound the intersection has two exclusive left turn lanes and one exclusive channelized free flowing right turn lane. The right turn lane has the appropriate priority rules to model right turn on red behavior. Westbound the intersection has one exclusive left turn lane and two exclusive right turn lanes. The right turn connectors are defined to allow the leftmost right turn lane to turn into either the through link or the left turn link for the downstream intersection at I-285 westbound. This is consistent with driver behavior in the field.

The detector configuration for SCATS is shown in Figure 45 and the detector configuration for ACTRA is shown in Figure 46.

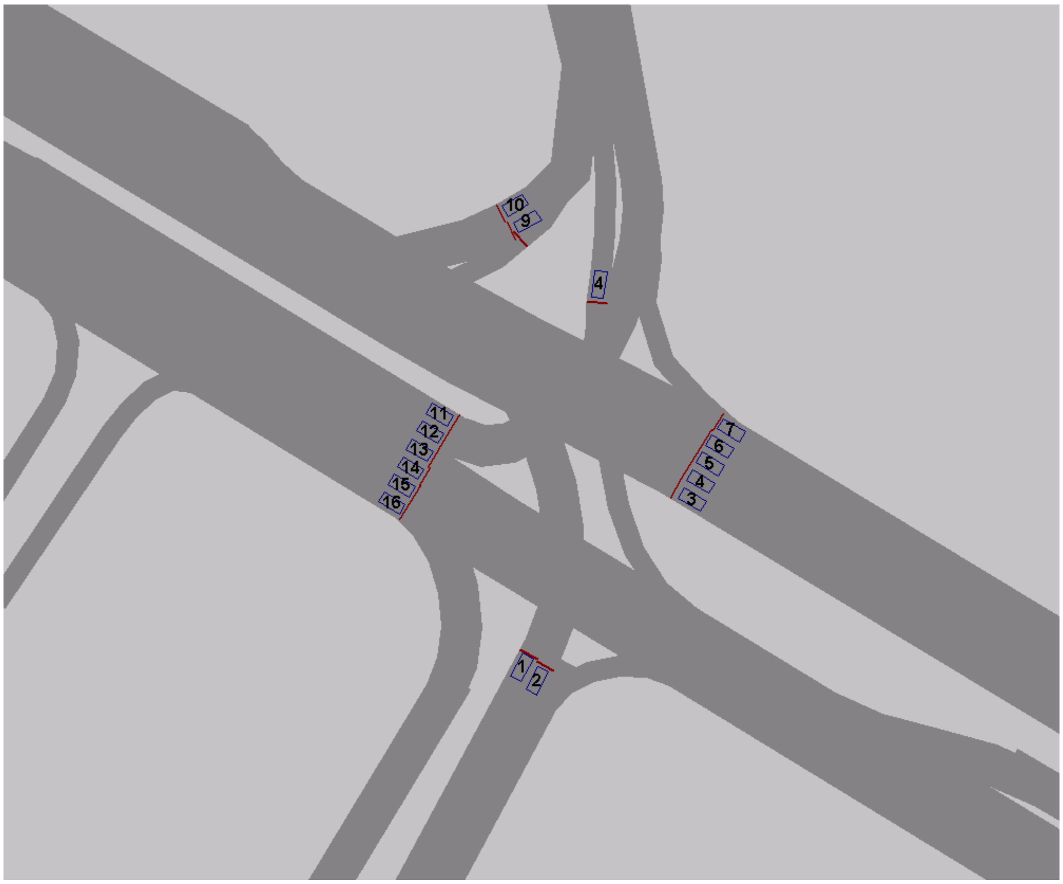

Figure 45. Cobb Parkway @ Galleria Drive North SCATS

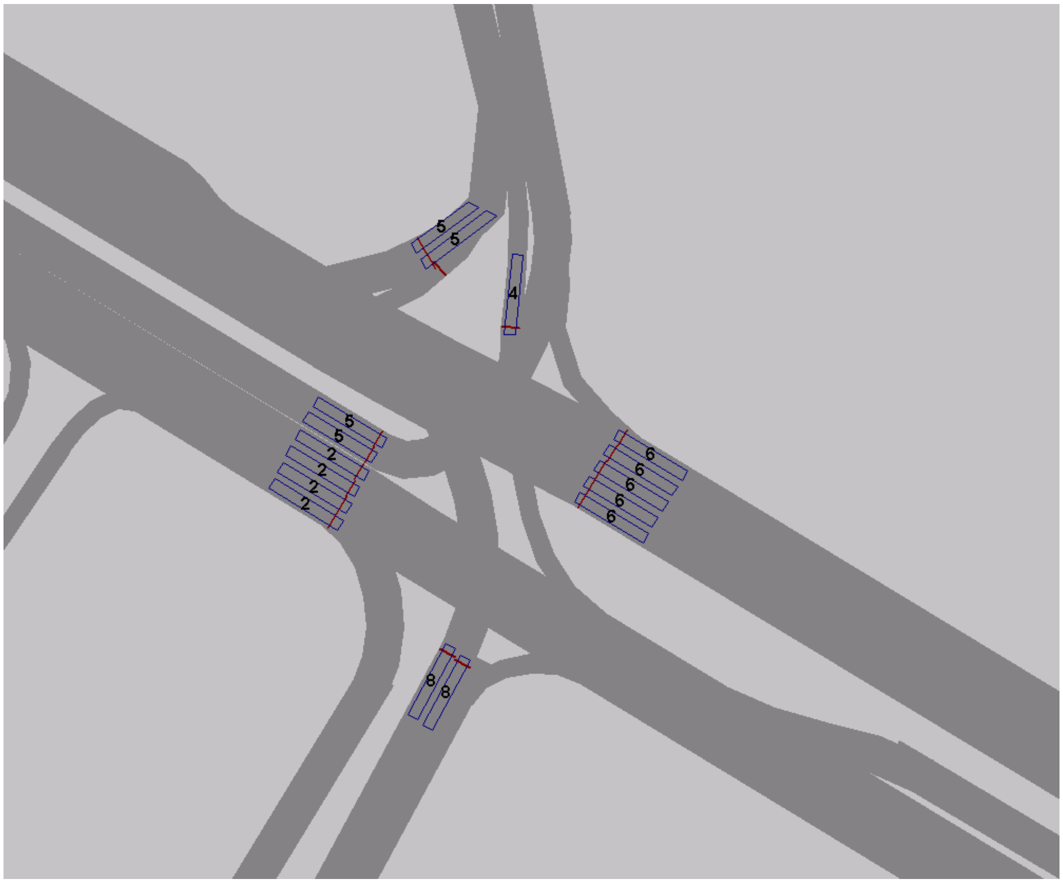

Figure 46. Cobb Parkway @ Galleria Drive North ACTRA

Intersection #10: Cobb Parkway @ Galleria Drive South

The intersection of Cobb Parkway and Galleria Drive South is shown in (Figure 47). Cobb Parkway runs north-south while Galleria Drive South runs east-west.

Figure 47. Cobb Parkway @ Galleria Drive South Aerial
(Figure Credit: Google Earth)

Southbound the intersection has two exclusive left turn lanes, three through lanes, and one shared through/right turn lane. The right turn is modeled with the appropriate priority rules for right turn on red. The routing decision for this movement was pushed back through the intersection at Galleria Drive North to give vehicles enough space to enter the left turn bays for this approach which start very close to the upstream intersection. Northbound the intersection has two exclusive left turn lanes, three though lanes, and one shared through/right turn link. The right turn is modeled with the appropriate priority rules for right turn on red. Eastbound the intersection has one exclusive left turn lane and one shared left turn/right turn lane. The right turn is modeled with the appropriate priority rules for right turn on red. Westbound the intersection has two right turn lanes. They do not allow right turn on red.

The detector configuration for SCATS is shown in Figure 48 and the detector configuration for ACTRA is shown in Figure 49.

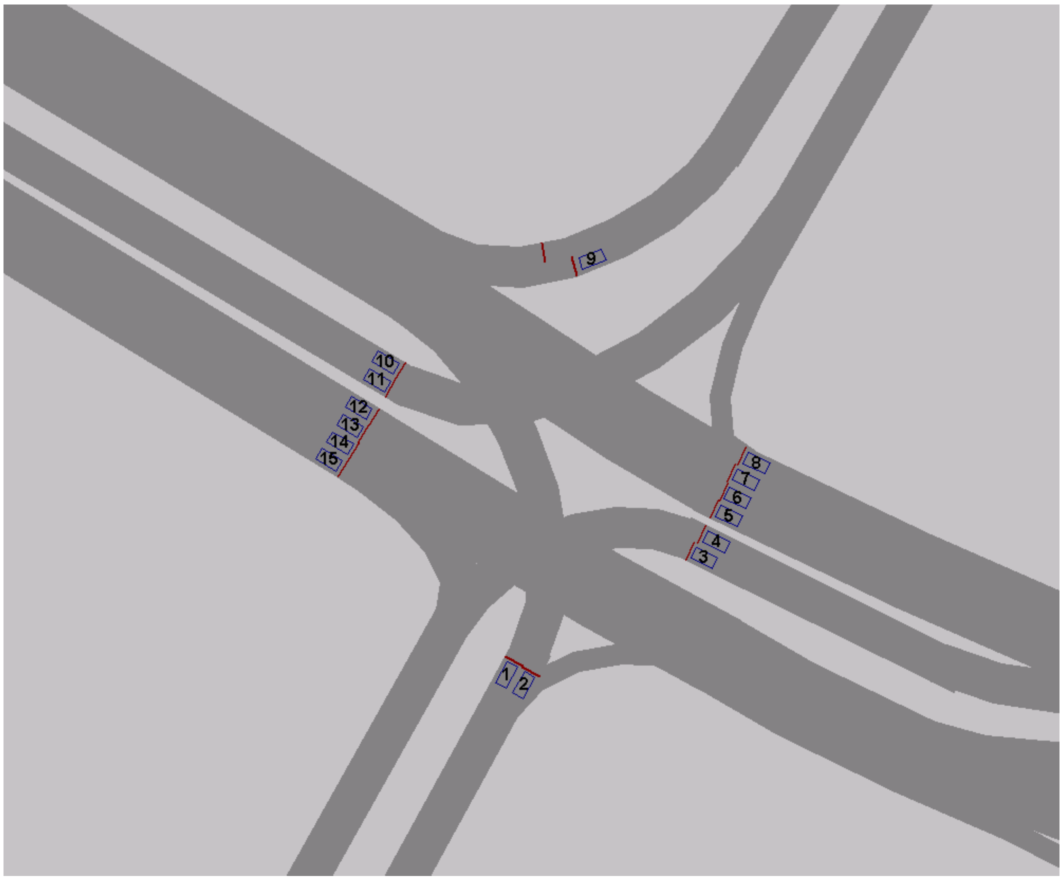

Figure 48. Cobb Parkway @ Galleria Drive South SCATS

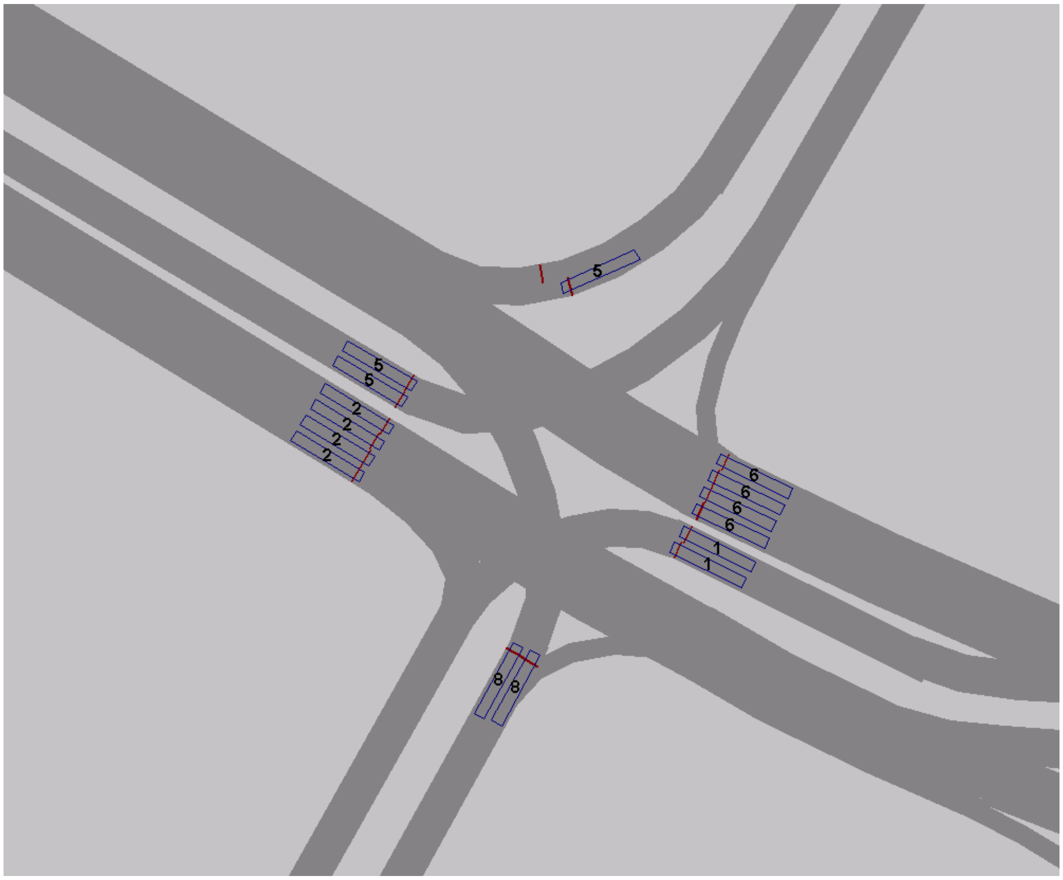

Figure 49. Cobb Parkway @ Galleria Drive South ACTRA

Intersection #11: Cobb Parkway @ Akers Mill Road

The intersection of Cobb Parkway and Akers Mill Road is shown in Figure 50. Cobb Parkway runs east-west while Akers Mill Road runs north-south.

Figure 50. Cobb Parkway @ Akers Mill Road Aerial
(Figure Credit: Google Earth)

Southbound the intersection has two exclusive left turn lanes, three through lanes, and one exclusive free flow channelized right turn lane. Northbound the intersection has two exclusive left turn lanes, three through lanes, and one exclusive free flow channelized right turn lane. Eastbound the intersection has two exclusive left turn lanes, three through lanes, and one exclusive free flow channelized right turn lane. The routing decision for this movement was pushed back through the intersection at Galleria Drive South so vehicles could make the connectors for the various movements on this approach which start very close to the upstream intersection. Westbound the intersection has two exclusive left turn lanes, four through lanes, and one exclusive free flow channelized right turn lane. Even though the markings in the field indicate that there is a stop bar for the right turn on all approaches, the right turn on red was modeled using a free flow connector with appropriate priority rules.

The detector configuration for SCATS is shown in Figure 51 and the detector configuration for ACTRA is shown in Figure 52.

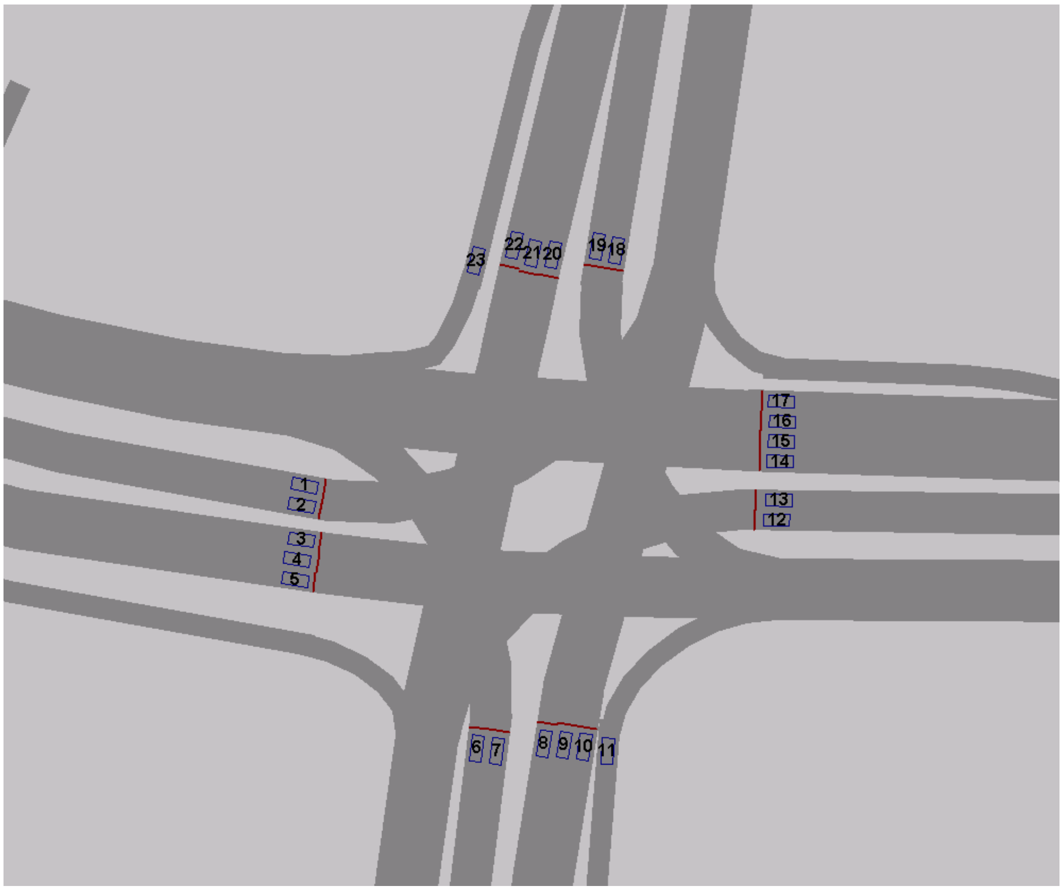

Figure 51. Cobb Parkway @ Akers Mill Road SCATS

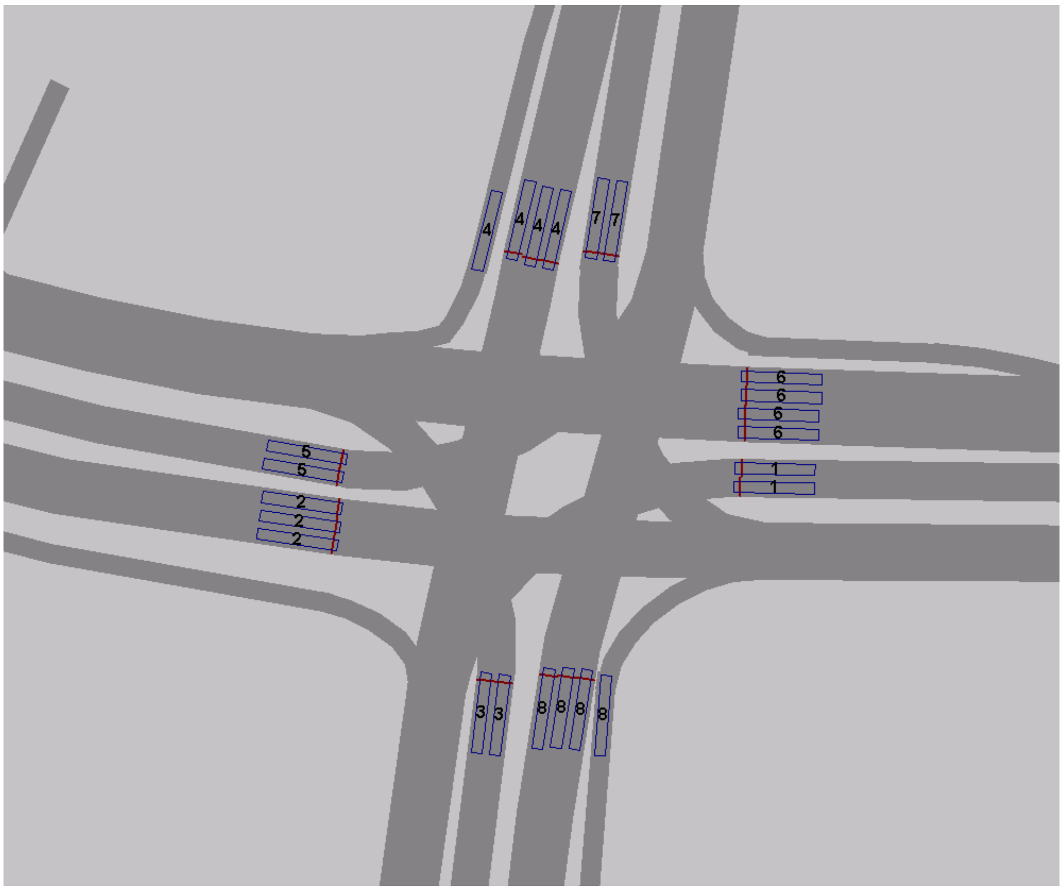

Figure 52. Cobb Parkway @ Akers Mill Road ACTRA

Initial Genetic Algorithm Procedure/Results

To conduct the final Genetic Algorithm analysis the most recent geometrically correct VISSIM models of the Cobb Parkway Study Corridor using the NEMA control systems for the AM and PM peak hours with the appropriate routing and volume inputs will be used. The AM and PM models were calibrated separately and the resultant parameter value sets compared to determine the final set of parameter values that most closely represents field conditions.

For the GA procedure 20 generations were run for each of the scenarios (AM and PM) to determine the optimum parameter values. For each chromosome (parameter value set) within each generation 5 separate runs were conducted varying only the random seed value. Travel time along the entire network was recorded and used in evaluating the fitness of each chromosome, and additional data (volumes, queue length, travel time along shorter segments) was recorded as necessary to validate the model's performance through the GA process.

The GA procedure itself was implemented in Visual Basic .NET and maintained copies of the model for each chromosome as well as the results of model runs as deemed necessary. Also the GA procedure kept logs of performance measures like run-time for analysis of the GA process itself. For each chromosome and for the final results the GA procedure presented a clear and accurate representation of parameter values. The final resulting parameter values are shown in Table 2.

Table 2. Final GA Calibration Results

#	Parameter	AM	PM
5	Average Standstill Distance, ax	4.49 ft	5.41 ft
6	Additive Part of Safety Distance, bx_{add}	1.66	2.14
7	Multiplicative Part of Safety Distance, bx_{mult}	2.50	2.36
8	Maximum Deceleration (own)	-16.14 ft/s^2	-8.40 ft/s^2
9	Maximum Deceleration (trailing)	-11.12 ft/s^2	-13.42 ft/s^2
15	Min. Headway (front/rear)	2.69 ft	2.56 ft
16	Safety Distance Reduction Factor	0.43	0.07
17	Max. Deceleration for Cooperative Braking	-24.80 ft/s^2	-13.19 ft/s^2
22	Lane Change Distance	Variable	Variable

HILS Specific Implementation

SCATS Implementation

The SCATS was the first system implemented and involved changes to the 2070 operating systems and the installation of a server component on the SCATS server. The initial setup of these components was performed by TransCore. They installed SCATS and implemented the Cobb County database and applicable configuration. They also installed the SCATS OS on the 2070 controllers. In addition to the SCATS OS, each controller also requires a “personality” which is the set of configuration settings that identify each controller to the SCATS server as the controller associated with a particular intersection. The implementation of each personality was accomplished by attaching to the controller using the serial port on a laptop and copying over the appropriate data files.

Communication between the SCATS server and the controllers was facilitated over a fiber/serial connection. Each controller is in constant contact with the server relaying detection information and receiving timing instructions.

Unlike the ACTRA system, the detectors in SCATS were significantly different than those from NEMA. SCATS makes use of one (typically 6ft) detector for each lane as it uses per-lane flow data in its timing calculations. In most cases there was a separate detector for each lane with the occasional exclusion of a free flow right-turn lane. In the field the SCATS system is fed by video detection that has two phases. When the signal associated with a particular movement is red the video uses one large detector to cover all lanes thereby avoiding detection errors. When the signal turns green the detector switches to per-lane detection as required for SCATS. While this functionality is not implemented using detectors in VISSIM there is confidence that the same effect is achieved as VISSIM provides flawless detection (i.e. the simulated detector does not miss simulated vehicles) [1]. Therefore each lane, for each approach, at each intersection was outfitted with a detector consistent with the observed field configuration.

To attach the model to the controllers using the CID a configuration file had to be created defining which CID input channels corresponded with the 2070 channel associated with a particular detector in SCATS. The identification of channels was made by using the hardware tester to send a signal on each CID input. Then the status screen for the SCATS intersection would indicate which input corresponded to each detector. The signal groups were more straightforward as each group number in SCATS corresponded directly with the number in VISSIM.

While SCATS does not explicitly choose timing plans based on the time of day, it does have some time-based elements. Specifically at 1 AM every weekday morning the system conducts a review of the previous day's detector day to prepare for selecting plans in the next 24-hour period. To ensure that each model run has a consistent background environment a baseline set of data was obtained from the field system to reset the SCATS system prior to any model runs. This also required that the SCATS server clock be set to 2 AM prior to any experimental runs to prevent the data review from occurring during the model run. Experiments showed that this time-shifting had no effect on the system performance as each of the controllers was automatically updated to match the server's time and the proper timing plans were selected for the volumes provided from the model.

ACTRA Implementation

The ACTRA implementation consisted of a single ACTRA server connected to the 11 2070 controllers through the same fiber/serial network used in the field in Cobb County. The server software was installed by an independent consultant, so the Cobb County ACTRA database had to be obtained separately to populate the system. ACTRA supports a hot data dump that can be used to create a copy of the active database without affecting the running system. Cobb County ran this procedure on their active system and provided us with the copy of their database. The database was then imported using the appropriate ACTRA utilities to duplicate the field system in the lab.

A few minor changes had to be made to the communications configuration within the Cobb County database to enable the server to function in this particular implementation. ACTRA can be configured to make use of a separate COM server to communicate with the controllers. This is the implementation in the field, however in the lab the COM server runs on the same machine as the ACTRA system. Modifications were made to the provided Cobb County database to recognize the local computer (localhost) as a valid COM server and each controller definition for the study intersections was updated to use the local COM server.

Also, provision had to be made for the transition between AM and PM timings. Specifically, a way was needed to force the controllers for the study intersection into a particular timing plan regardless of the real-world time of day. The dial, split, and offset for the AM and PM peak timing plans were determined based on the timing plans already configured in the database. The AM plan was 1,1,1 while the PM plan was 2,3,1. In the group configuration controls for the ACTRA group containing the study intersections the timing plan selection method was set to manual. When the AM or PM plan was desired the appropriate values would be entered in for the Manual Dial, Manual Split, and Manual Offset.

To use ACTRA on the controller side each of the 2070's was configured to run the original Eagle operating system. Each controller was then configured with the appropriate address that identified it to the server as the controller for a specific intersection. After this had been defined the timing plans stored within ACTRA could be pushed out through COM to each of the study controllers. Thereafter, any changes made to the ACTRA database would be pushed out to the controllers as long as they were running the Eagle OS.

On the model side the configuration was nearly identical to the NEMA setup. Detectors for each intersection were placed based on plans provided by the county. The

ACTRA system used 15ft loop detectors for the side street and any actuated movements, mostly left turns. The most difficult part was determining the configuration of the CID to convey detector actuations in the model to the proper 2070 channel. The mappings were determined by sending a signal on each of the CID inputs and watching for actuation recognition on the status display for the controller within ACTRA. This information was then recorded in the CID configuration file in addition to the more straightforward definitions for controllers and signal groups as previously described.

Data Collection

Measurement Metrics

To quantify the performance of the model it was decided that travel time and delay at a per-movement and per-intersection level would be collected. Because of the extensive field data collection effort there was ample real-world data to use in comparisons against the model runs to ensure accuracy and consistency. Travel times would be used to compare one model against another, and to compare all models against the real world. Delay measurements would be used largely to compare one model and control system against another.

Model Test Run Implementation

To generate results for analysis it was initially determined to take the average of each particular measurement across eight individual runs of the VISSIM model. Each run would vary the random seed, which is a configuration parameter in VISSIM that provides for variation in separate model runs. Initially it was decided that a 15 minute seeding period would be used to allow the model to reach steady-state before beginning data collection, however for SCATS 15 minutes was found to be insufficient, not allowing the SCATS adaptive control system to appropriately react to the traffic. The control system that SCATS employs relies heavily on historical volume data in determining future signal timings. At the start of each simulation run the model would be empty and presented an unrealistic condition to which SCATS would attempt to respond. After discussions with TransCore it was decided to allow SCATS a one-hour seeding period to give the adaptive control algorithms sufficient data to intelligently respond to the start up condition. After further experimentation it was observed that re-starting the model from scratch every two hours may be having an adverse effect on the SCATS algorithms. To more accurately reflect the field conditions a "long-run" model was developed. The long-run model runs for 16 hours, varying the traffic every other hour as would be consistent with entering and exiting the peak hour. An example of the fluctuation between the peak hours is shown in Table 3. In this example hours 2 and 4 are the peak hours while hour 3 is the transition period.

Table 3. Long Run Volume Fluctuation

Time Increment	2:00-3:00	3:00-3:15	3:15-3:45	3:45-4:00	4:00-5:00	…
Volume	100%	85%	75%	85%	100%	…

In this implementation, the first hour is the seeding period during which traffic is gradually increased from 0 to 100% in 15-minute increments. This helps SCATS develop the proper background settings, and eliminates transients that might affect the results analysis [8, 9, 27, 30]. The progression of volumes for this period is shown in Table 4.

Table 4. Warmup Volume Fluctuation

Time Increment	0:00-0:15	0:15-0:30	0:30-0:45	0:45-1:00	1:00-2:00
Volume	25%	50%	75%	100%	100%

Data is collected every other hour, during the peak. This provides eight hours of data and the randomness generated in the model is enough to sufficiently differentiate each of the individual hours as an independent run. This long-run simulation method was applied to both SCATS and ACTRA data collection for consistency.

Data Collection

To collect data from the model several measurement devices were used. All data was collected in five minute increments over the entire course of the run, including the seeding and "off" periods. This allowed for a more granular picture of what was going on without placing too much of a strain on the data recording elements within the model.

Turning Movements

To obtain turning movement count data each movement of each approach of each intersection was outfitted with a data collection point. Data collection points in VISSIM can be used for a variety of purposes but in this case the collection points were configured to simply record the number of vehicles that passed over that point.

Queues

Queue counters were likewise implemented on all controlled movements to track the length of the queue that forms due to the control device. Queue counters in VISSIM record both the maximum queue and the average queue. Maximum queues are calculated by counting all vehicles in the queue, even if it spills back through another intersection to find the maximum number/length of cars waiting due to the control device.

Travel Times

To obtain travel times several different travel time segments were implemented along the corridor. There are two segments that cover the entire length of the corridor and calculate end-to-end travel time. This particular measurement posed some difficulties. VISSIM calculated travel time for each five minute period by taking the average of the travel times of all vehicles completing the segment in that period. If no vehicles completed the segment in a given time period then no measurement data is provided for that time period. There were two issues that contributed to these data gaps. First, because of the high volumes turning onto I-285 eastbound, I-285 westbound and surrounding intersections most of the vehicles input at the ends of the Cobb Corridor were diverted and replaced with vehicles entering the network from side streets and the I-285 off-ramps. So although the volumes entering and exiting the corridor were relatively consistent, there were very few vehicles actually traveling the entire length. Also, because the data was collected in five minute increments, a problem arose in times of high congestion when there might not have been be a sufficient amount of time for vehicles to make the trip along the entire corridor. Because of the relatively low number of vehicles that end up traveling the entire length of the corridor, several probe vehicles were added and specifically routed to traverse the entire corridor and provide an adequate number of data points for measurement. The volumes are 40 probe vehicles per hour southbound and 20 probe vehicles per hour northbound. The vehicles were given a particular type so they could be easily identified and accounted for in the analysis, and given a unique color so they could easily be identified in visual analysis.

In addition to the segments covering the entire corridor, additional segments were added to provide incremental travel times both northbound and southbound. In each case the southernmost (or northernmost) intersection was used as a reference point and segments were created immediately after that intersection to a point immediately after each subsequent intersection in the model. In this manner travel times between intersections 1-2 and intersections 1-3 and so on could be obtained and the progression of vehicles along the corridor could be shown. This data had also been collected in the field, and so comparisons could be made.

Finally, segments were created that measured the travel times along each movement. VISSIM provides the length of each travel time segment which, when coupled with the recorded travel time, could be used to calculate the delay for that movement. Each segment in this case was defined so that the beginning of the segment was far enough back to capture reasonable queues forming for the movement, while still capturing all vehicles making that movement with the end defined immediately after the given control device.

Vehicle Records

VISSIM allows for the recording of nearly all vehicle behavior characteristics (speed, location, acceleration etc.) for each individual vehicle in the model at any time interval desired. While a useful tool, unfiltered individual vehicle data for a low time interval (every second for example) would generate massive amounts of data per model run. The volume of data generated could exceed several gigabytes depending on the specifics of the model. However, per-vehicle data is very useful in determining particular vehicle performance, and provides insights that cannot be obtained with other data collection methods. In this model per-vehicle record data was collected for the probe vehicles only during simulation runs to allow for more granular analysis of model performance.

One of the effective simulation performance visualizations was to plot vehicle velocity as a function of its position along the corridor. To collect this, data for the end-to-end trips per-vehicle data was collected for each probe vehicle in the model. The following data items were recorded every time step: Vehicle Number, Simulation Time, Link Number, Link Coordinate, Acceleration, Speed, and Desired Speed. From this the position/velocity plots could be created. The data load for this collection was rather low as the total number of probe vehicles is relatively low when compared with the total number of vehicles in the model.

Per-vehicle data was also collected whenever the model run was to be evaluated using the visualizer tool described on page 69. In this case, the x and y coordinates and the time were recorded for every vehicle in the model. Initially the collection was set to occur every time step, but the data volume and performance issues reduced that rate to collect every 5 simulation time steps which corresponded with one collection every second. Even with that reduced collection frequency there are still performance issues as the effort to collect the required data for all the vehicles in the model causes the simulation to run at a speed less than real-time. Further research will have to be conducted to determine a more optimal data collection procedure in this situation.

SCATS-Specific Data

For SCATS runs some data was collected from the SCATS system itself. Using the Traffic Reporter application historical SCATS performance data can be accessed. Because the computer clock on the SCATS server had to be adjusted to facilitate accurate data collection (see the SCATS Implementation section) the virtual time for each run had to be carefully recorded. Provided this, the historical data could be accessed using Traffic Reporter to determine nearly all aspects of SCATS performance for that run. Traffic Reporter provided an easily generated chart shows what cycle lengths, link plans and split plans SCATS was running.

Initial Results

This report will present the findings from the comparison of four different baseline simulated scenarios against the field data and against each other. The four simulated scenarios involved the analysis of the Cobb Parkway corridor using ACTRA control for both the AM and PM peak hours and the SCATS signal control for both the AM and PM peak hours. Initially, each scenario will be compared against the data collected in the field to determine the validity of the simulation and then the ACTRA scenarios will be compared against the SCATS scenarios to determine the performance of one relative to the other.

Statistical Analysis Methodology

Mainline travel time will form the basis for this phase of the analysis for two reasons. First, the collected field data provides a great deal about expected mainline travel times along the study corridor. And second, the mainline travel time is a good measurement of overall system performance.

To conduct a statistically rigorous analysis of the models individual vehicle travel times were collected for every vehicle in VISSIM that traveled the study corridor from one end to the other. Northbound and southbound traffic was differentiated for analysis. The travel time data includes not only the network traffic as determined by the turning movement counts collected in the field, but also the probe vehicles specifically added to ensure that there are a sufficient number of data points for meaningful analysis. While travel time data was collected for the entire 16-hour run in order to track trends while entering and exiting the peak hour, only the data collected during the 8 peak hours is used in the statistical analysis. Additionally, the collection of every vehicle's travel time data yielded a significant number of results which were randomly sampled at roughly 10% across the analysis period to ensure independence between the sampled travel time values.

The computerized statistical package R was used to conduct the statistical analysis of the travel time data. The first step was to examine the distribution of the travel time data so an appropriate comparison test could be selected to conduct a population comparison. Initially, it was assumed that the collected travel times were normally distributed and that a standard t-test could be used to compare the samples. However, upon analysis it was discovered that this was not the case.

Data Distribution Analysis

Figure 53 demonstrates the non-normality of the travel time data with a histogram and a Q-Q plot of the ACTRA PM Southbound travel times collected in the field. The corresponding data for the simulated ACTRA PM Southbound travel times is shown in Figure 54.

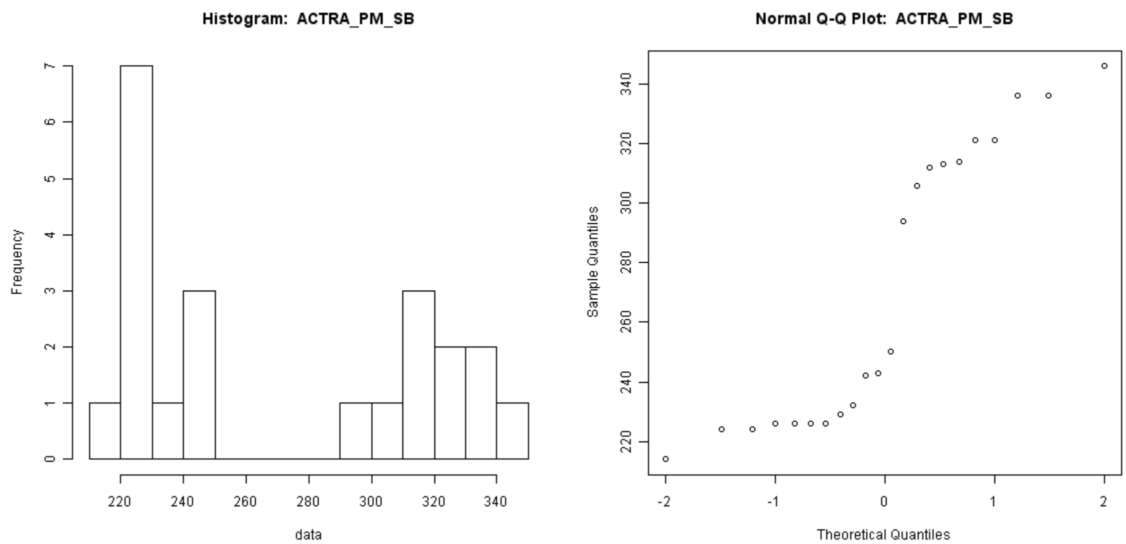

Figure 53. ACTRA PM Southbound Field Travel Time Distribution

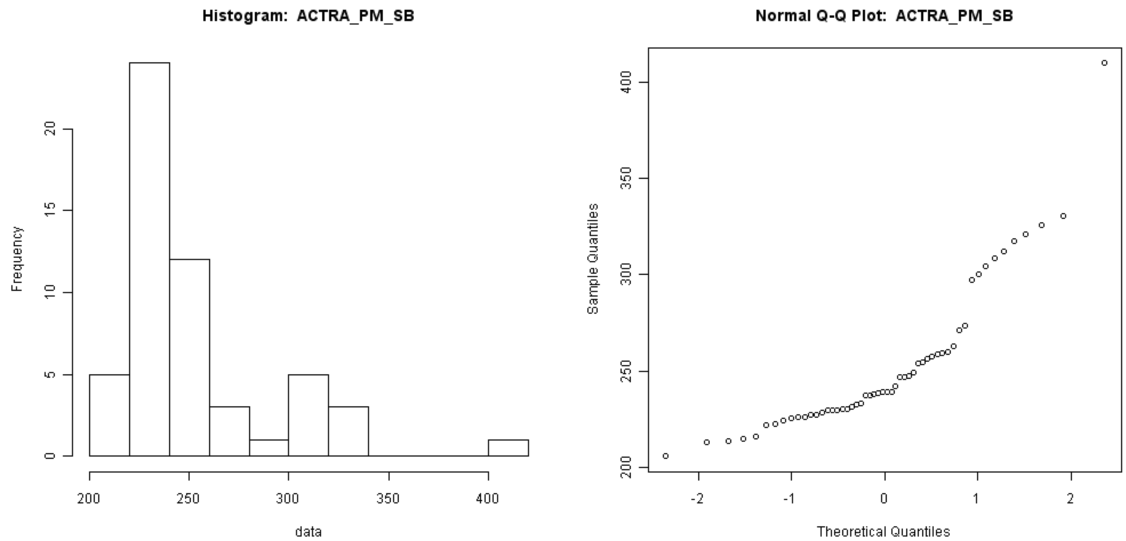

Figure 54. ACTRA PM Southbound Simulated Travel Time Distribution

Additional figures graphically demonstrating the distributions for all the scenario/control system combinations are shown in appendix A. It was noted that the SCATS travel times for all scenarios and control systems were closer to being normally distributed than the ACTRA travel times. Figure 55 (Field) and Figure 56 (Simulated) demonstrate the distribution of the SCATS PM Southbound travel times for comparison against the previous plots based on ACTRA data.

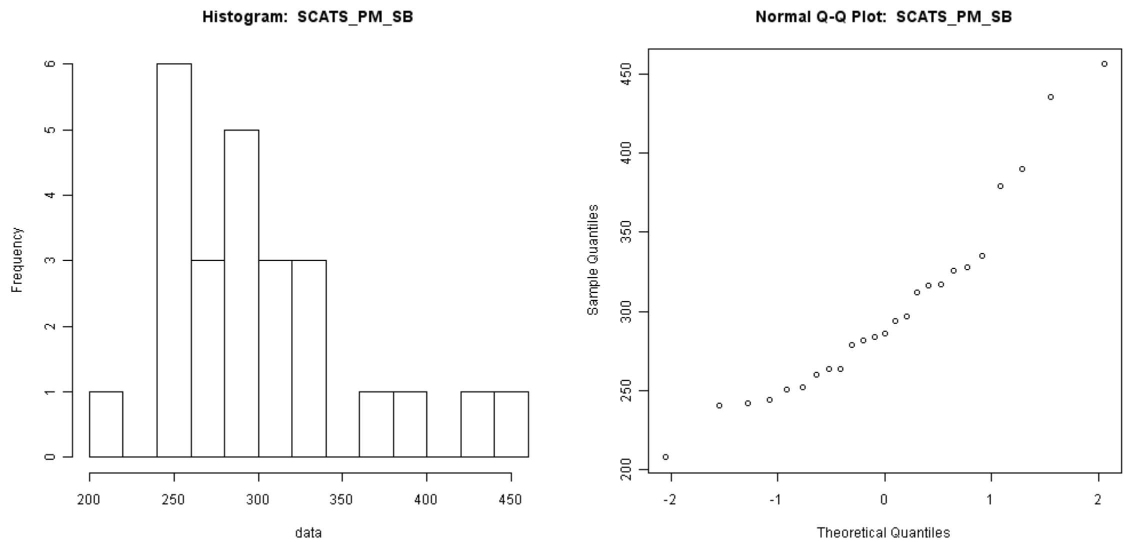

Figure 55. SCATS PM Southbound Field Travel Time Distribution

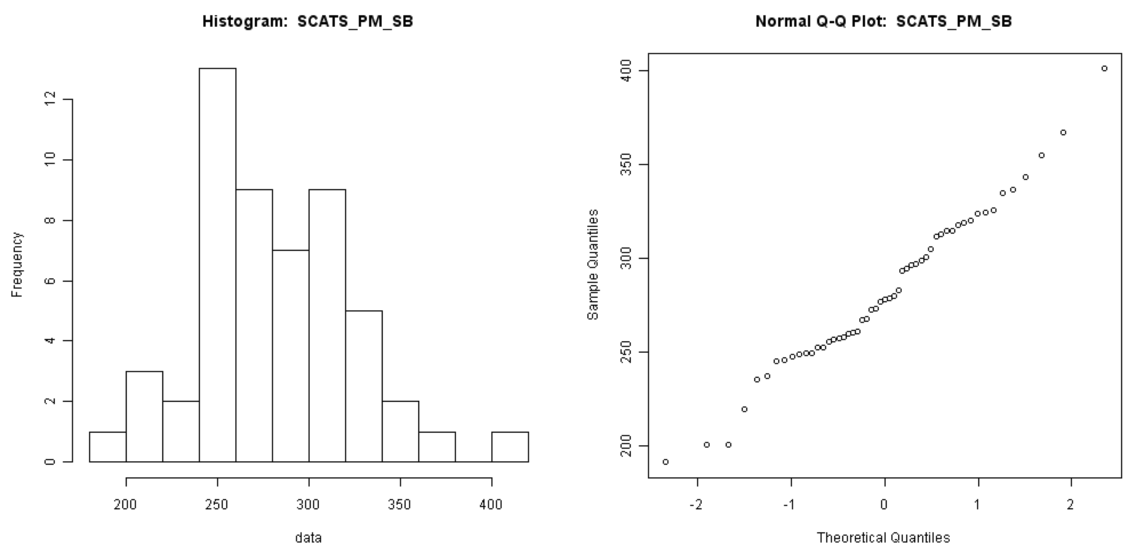

Figure 56. SCATS PM Southbound Simuated Travel Time Distribution

To rigorously analyze the normality of the datasets it was decided that a Shapiro-Wilk test should be conducted for each scenario/control system combination. The results of this test are shown in Table 5 for ACTRA and Table 6 for SCATS below along with the descriptive statistics relevant to each dataset.

Table 5. ACTRA Base Scenario Travel Time Results

	AM Field		**AM Simulated**		**PM Field**		**PM Simulated**	
	SB	NB	SB	NB	SB	NB	SB	NB
Min	116.0	163.0	127.2	149.9	214.0	248.0	205.5	231.4
Median	190.5	214.5	205.1	210.8	246.5	322.0	239.0	302.8
Mean	184.3	216.6	201.4	208.6	271.0	337.8	253.1	315.2
Max	237.0	366.0	297.1	341.8	246.0	535.0	410.0	431.3
W	0.91	0.85	0.96	0.88	0.83	0.88	0.82	0.96
p-value	0.11	0.002	0.001	0.0006	0.002	0.006	1.28e-06	0.0009

A p-value above 0.05 indicates that the distribution of the sample is normally distributed at a 95% confidence level. Of particular note is the difference between the SCATS and ACTRA data distributions. Based on the Shapiro-Wilk test it was found that only two SCATS samples from Table 6 (AM Field SB and PM Field SB) are not normally distributed while for ACTRA all samples, save one, from Table 5 (AM Field SB) are not normally distributed.

Table 6. SCATS Base Scenario Travel Time Results

	AM Field		AM Simulated		PM Field		PM Simulated	
	SB	NB	SB	NB	SB	NB	SB	NB
Min	122.0	142.0	136.0	162.5	208.0	144.0	191.4	179.3
Median	247.0	228.0	197.3	201.9	286.0	322.5	278.3	295.0
Mean	243.7	234.8	203.6	209.6	301.7	315.6	282.5	294.4
Max	421.0	364.0	286.5	278.2	456.0	494.0	400.9	465.7
W	0.93	0.97	0.99	0.96	0.91	0.98	0.98	0.99
p-value	0.04	0.50	0.20	0.29	0.04	0.97	0.61	0.24

The results of the Shapiro-Wilk test seem reactively intuitive and consistent with the nature of the signal control system. ACTRA, using a fixed time control system, damps variations in the natural traffic patterns while SCATS, using a variable timing control, has the flexibility to change along with the traffic patterns. This flexibility likely leads to more normally distributed travel times. More evidence of this result will be discussed later.

Also, it should be noted that while the distribution of the sample data varies from ACTRA to SCATS the differences in the means do not appear to be significant between simulated ACTRA and simulated SCATS scenarios. A more rigorous test of this observation will be presented later.

Scenario: ACTRA AM

The next analysis step was to compare the simulated runs against the field data to determine the closeness of fit. Because not all of the samples were normal it was decided that non-parametric data analysis should be used. Additionally, the data was plotted over the course of the entire simulation run against the data collected in the field for a visual representation of the relationship between the two datasets.

The figures below show this data for the ACTRA AM simulation run.

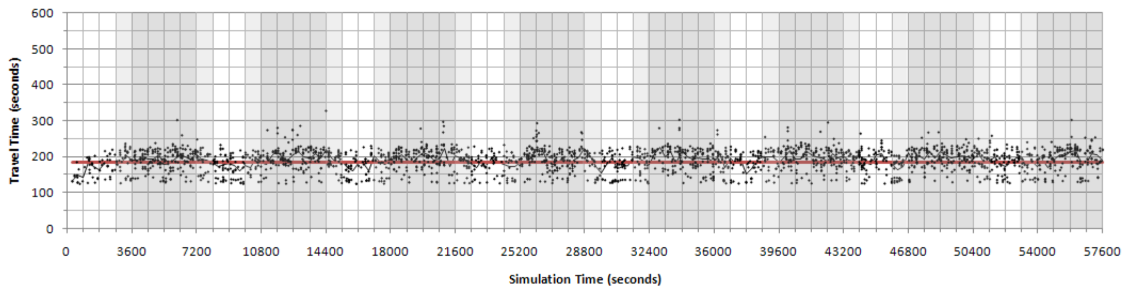

Figure 57. ACTRA AM Southbound Base Travel Time

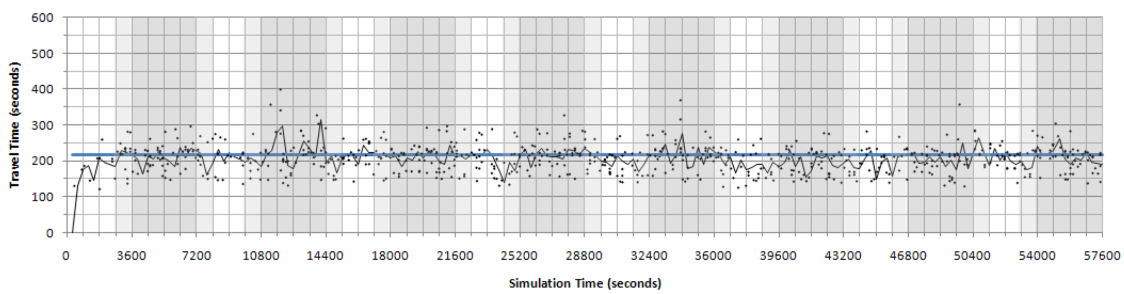

Figure 58. ACTRA AM Northbound Base Travel Time

In each figure, and in all subsequent travel time plots, a series of grey bands is seen. These indicate the state of traffic flow at that time as previously discussed in the methodology section. White represents 75% of total volume, light grey 85% and dark grey 100%. Evaluation data is collected during the 8 100% hours and measurements are averaged across each of the collection periods to ensure statistically relevant results.

In both figures, a close grouping of the collected travel times is observed around the expected mean value from the field which is indicated with a red line in Figure 57 and a blue line in Figure 58. Also, because this is the AM peak hour a much higher number of trips in the southbound direction is seen. To conduct a more statistically rigorous analysis of the data the field travel times and the simulated travel times were compared using a Wilcoxon Ranked Sum Test.

These tests assumed a null hypothesis that the difference in location between the two distributions (in this case the field data and the simulated data) is zero. If the hypothesis holds is likely that the two samples being compared were drawn from the same distribution. The alternative hypothesis is that the difference in location is not zero. This may indicated that the samples are being drawn from different distributions or from the same distributional form but offset to some degree (e.g. one may have a higher mean than the other). The Wilcoxon Ranked Sum Test was run using the R statistical package, the results of which are shown in Table 7 below.

Table 7. ACTRA AM Travel Time Wilcoxon Ranked Sum Results

Measurement	ACTRA AM Southbound	ACTRA AM Northbound
W	1177	502
p-value	0.3925	0.6354
Difference in Location	6.83	5.38
Confidence Interval (95%)	-9.13 – 25.70	-16.10 – 27.12

Based on the p-values shown in Table 7 the test fails to reject the null hypothesis for both directions with 95% confidence. This is further validated by the 95% confidence intervals for the difference in location between the two sample sets containing 0, which would indicate that it is likely that the two samples came from the same distribution.

While not statistically significant the positive values for the difference in location indicate that the field travel times are slightly shorter than those from the simulation. The calculated value for the difference in means from Table 5indicates the difference in travel time southbound is roughly 15 to 16 seconds and northbound is approximately 8 seconds. Along with the distribution not being statistically different these values are practically insignificant given the overall travel times of approximately 200 seconds and the wide tolerances with which traffic performance is usually calculated.

The processed network volumes were also evaluated to ensure that the travel time values generated were for traffic roughly consistent with the field. Figures showing the turning movement volumes for each intersection in the ACTRA AM scenario are available in Appendix B, Figure 89. Of particular note though, are the volumes entering the corridor from each end. These are shown in Figure 59 for vehicles entering southbound and in Figure 60 for vehicles entering northbound. Both figures show that the generated input volumes were roughly consistent with the expected volumes shown in red for southbound and blue for northbound.

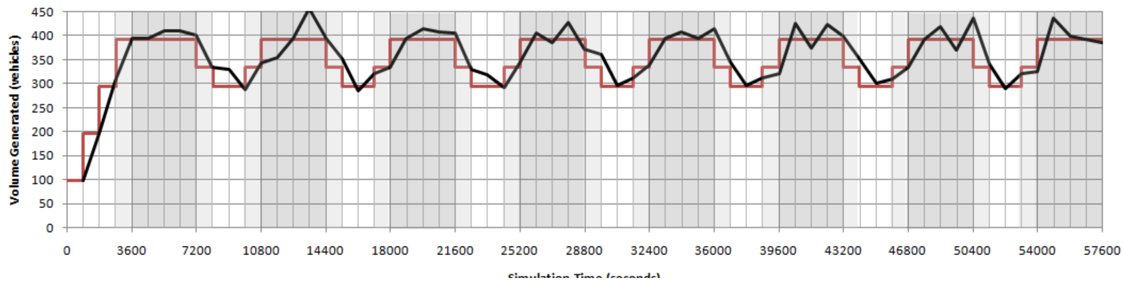

Figure 59. ACTRA AM Southbound Entering Volume

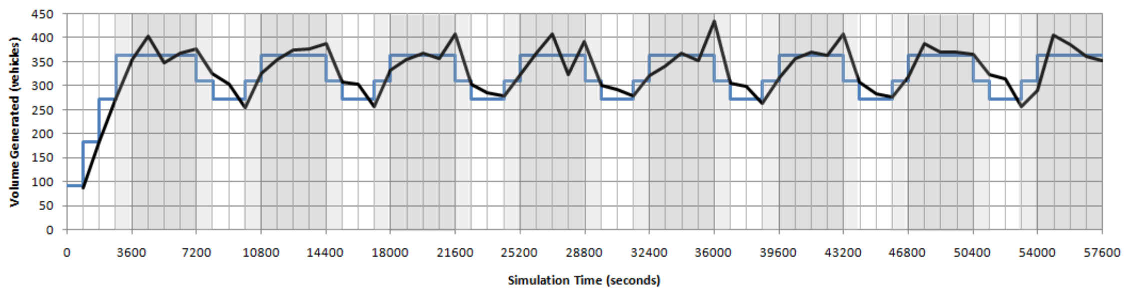

Figure 60. ACTRA AM Northbound Entering Volume

Scenario: ACTRA PM

The ACTRA PM data is shown in Figure 61 and Figure 62.

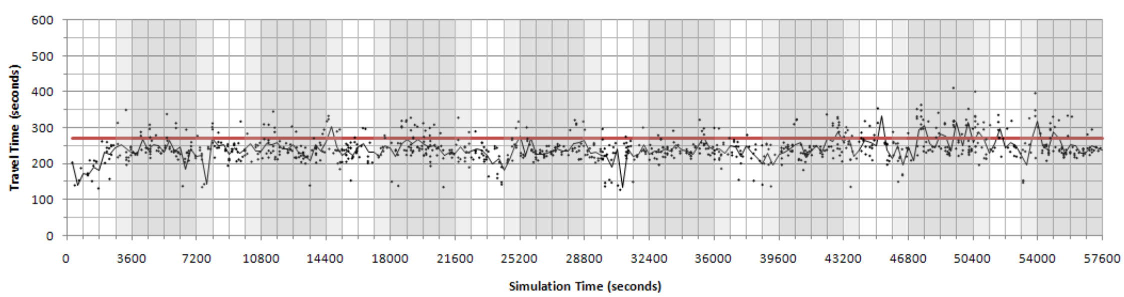

Figure 61. ACTRA PM Southbound Base Travel Time

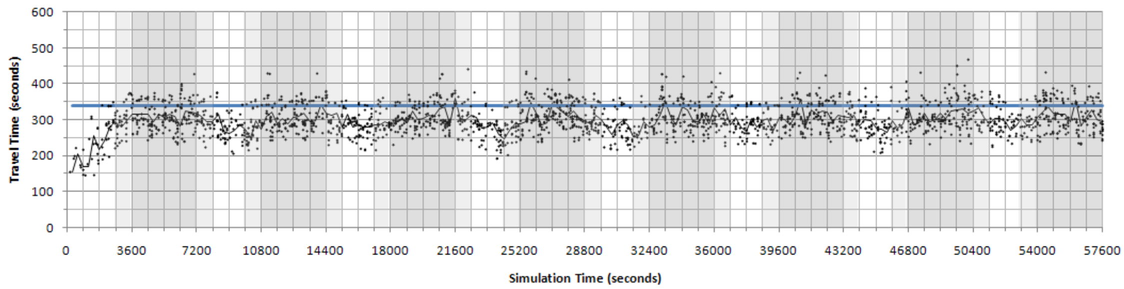

Figure 62. ACTRA PM Northbound Base Travel Time

In the PM scenario it may be observed that traffic in the northbound direction is much heavier than during the AM peak hour, although traffic tends to be heavier in the PM peak in general. Compared with the average travel time from the field shown in the red and blue lines as for the ACTRA AM scenario, the simulation data mean appears to be a bit lower than the field data. However, based on the results of the Wilcoxon test shown in Table 8 below the evaluation fails to reject the null hypothesis that the difference in the location between the two distributions is not zero. Given the narrowness of the calculated difference between the two distributions (roughly 18 seconds southbound and 22 seconds northbound) it can also be considered practically irrelevant.

Table 8. ACTRA PM Travel Time Wilcoxon Ranked Sum Results

Measurement	ACTRA PM Southbound	ACTRA PM Northbound
W	512	1340
p-value	0.3505	0.1826
Difference in Location	-6.58	-14.03
Confidence Interval (95%)	-62.75 – 7.10	-41.51 – 6.33

It can also be seen that there is wider variation in travel times in the PM scenario than was seen in the AM scenario. This is most likely a byproduct of congestion on the corridor due to the higher volumes seen during the PM peak. As the system approaches congestion, small changes that were easily absorbed in the excess capacity of the AM scenario have more significant impacts on the performance of the PM scenario.

The increased variation in the simulation data has also widened the confidence bounds for the difference in location significantly. Additionally, the calculated difference in location is negative and the confidence intervals for the difference in location is skewed towards the negative, supporting the earlier observation that in the PM peak the simulation appears to provide better performance than that measured in the field.

As with the AM scenario the input volumes and intersection turning counts were evaluated to ensure the model's performance was reasonable. The results of the per-intersection evaluations can be seen in Appendix B, Figure 90. Figure 63 and Figure 64 below show the southbound and northbound input evaluation. Both appear to be consistent with field observations.

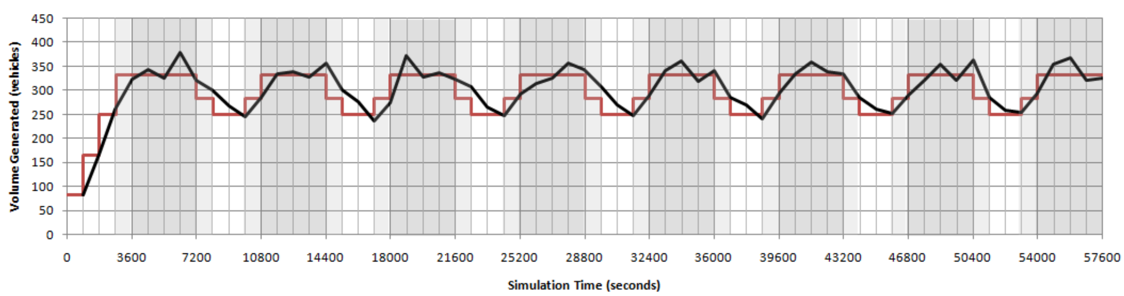

Figure 63. ACTRA PM Southbound Entering Volume

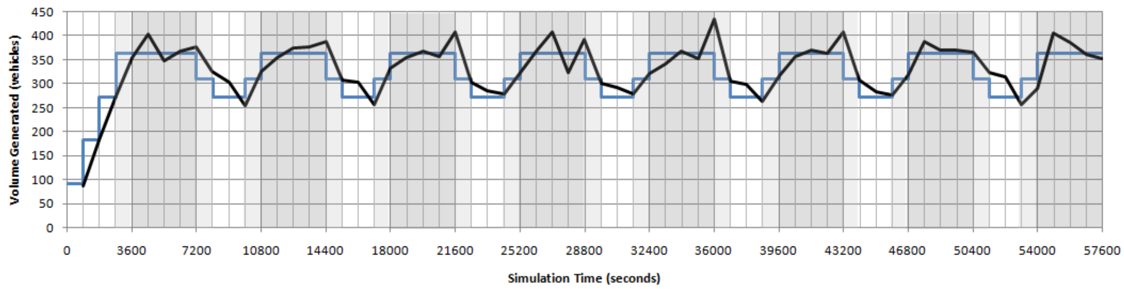

Figure 64. ACTRA PM Northbound Entering Volume

Scenario: SCATS AM

Figure 65 and Figure 66 below display the results from the SCATS AM scenario.

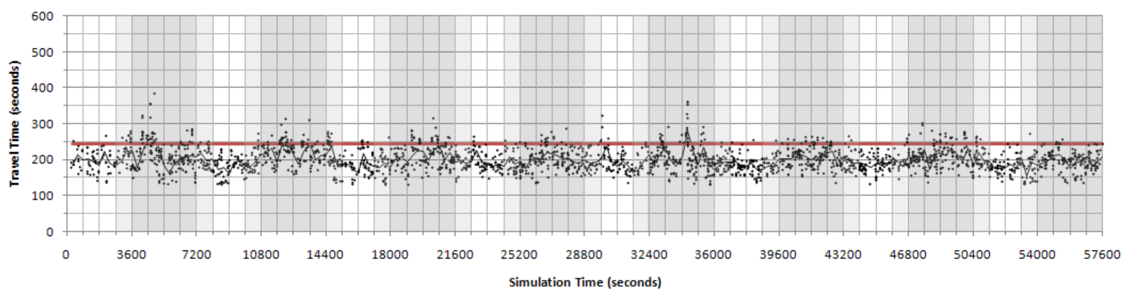

Figure 65. SCATS AM Southbound Base Travel Time

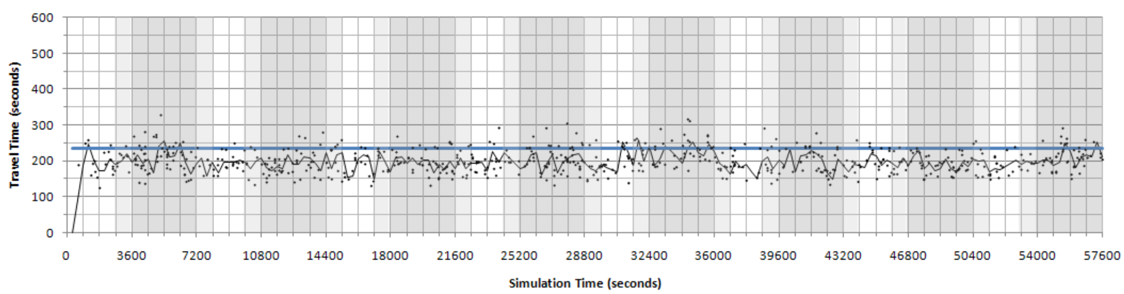

Figure 66. SCATS AM Northbound Base Travel Time

As in the ACTRA AM scenario, there are heavier volumes in the southbound direction. The same volume inputs were used in both the SCATS AM and ACTRA AM scenarios, so this is expected. It also appears that the simulated travel times are lower than the recorded field travel times.

Table 9. SCATS AM Travel Time Wilcoxon Ranked Sum Results

Measurement	SCATS AM Southbound	SCATS AM Northbound
W	1283	345
p-value	0.0002418	0.0008454
Difference in Location	-35.35	-36.34
Confidence Interval (95%)	-51.45 – -18.59	-55.93 – -16.45

Here there is sufficient evidence to reject the null hypothesis that the difference in location between the two distributions is zero. Visual observations from Figure 65 and Figure 66 when combined with the results of the statistical test shown in Table 9 clearly support this conclusion. Practically the calculated difference in means of approximately 40 seconds southbound and 25 seconds northbound is also significant. Given the average field measured corridor travel time of roughly 250 seconds the difference represents roughly 14% of the entire corridor travel time. This difference in travel times will be explored in more detail after examination of the data from the SCATS PM scenario, which exhibits the same traits.

As with the ACTRA scenarios a turning movement evaluation was conducted for SCATS AM. The results are shown in Appendix B, Figure 91. The southbound and northbound traffic volume inputs are shown in Figure 67 and Figure 68. From these it is clear that the provided inputs match the observed field volumes.

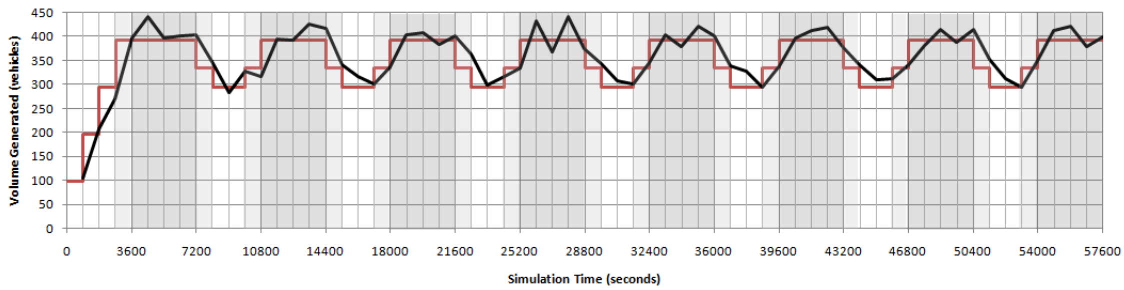

Figure 67. SCATS AM Southbound Entering Volume

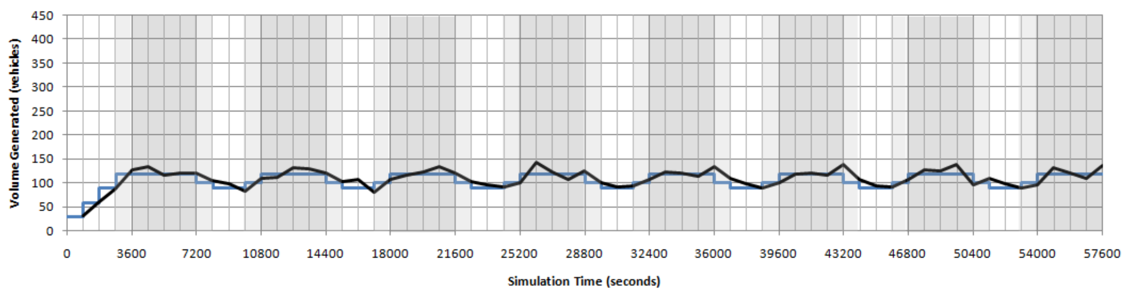

Figure 68. SCATS AM Northbound Entering Volume

Scenario: SCATS PM

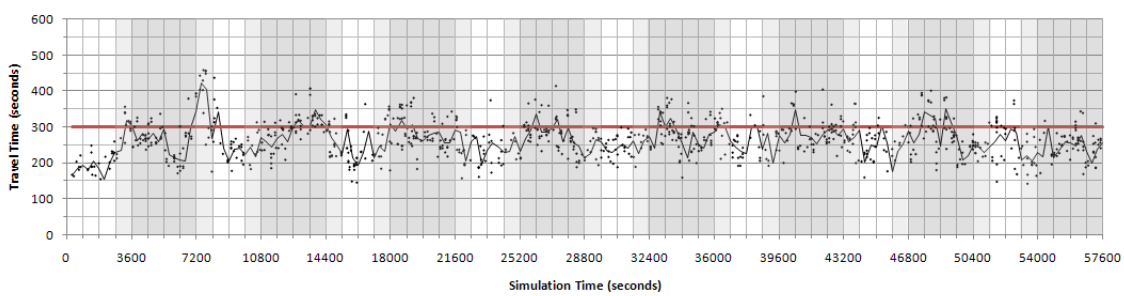

Figure 69. SCATS PM Southbound Base Travel Time

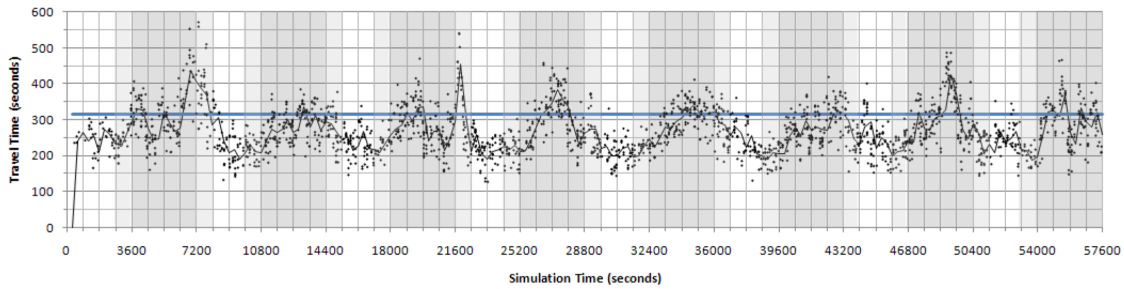

Figure 70. SCATS PM Northbound Base Travel Time

Figure 69 and Figure 70 show the SCATS PM travel time data. It is interesting to note that the travel time data from this scenario exhibits a significant amount of variability. The variability is significantly greater than the ACTRA PM data (Figure 61 and Figure 62). In considering possible explanations for this difference it is recalled that while ACTRA is forced to apply the same plan to the model regardless of the flow, SCATS is free to adapt to the prevailing traffic patterns. It becomes apparent that SCATS is able to reduce the travel time during the simulated lower demand periods while ACTRA provides little improvement. Also, SCATS is able to react to increasing traffic and scale up in an attempt to meet the higher demand when transitioning to the peak demand. However, it is also seen that there are several travel time spikes in the SCATS PM scenario not present under ACTRA control (Figure 62). There are several possible reasons for these spikes. The first is a potential modeling issue. It is possible that during the transition from high to low traffic demand that spillback issues are created that are not properly handled by the VISSIM model. These issues may be impacting the SCATS performance. Presently, corrections are being made to the model to ensure that traffic behavior is consistent with field observations. Future reports will reflect these updates. It is also possible that this behavior accurately reflects an artifact of SCATS control. That is, in the process of changing the signal timing to respond to the perceived demand changes travel time spikes could occur. That is, while SCATS provides a better PM mean travel time northbound (ACTRA at 315 seconds and SCATS at 294 seconds) ACTRA provides a more consistent performance. It is important to note that the travel time spikes are only seen during the high demand periods and that the SCATS improvement over ACTRA during the low demand periods of the simulation run is significant, on the order of 50+ seconds.

Table 10. SCATS PM Travel Time Wilcoxon Ranked Sum Results

Measurement	**SCATS PM Southbound**	**SCATS PM Northbound**
W	533	786
p-value	0.05733	0.06352
Difference in Location	-24.22	-37.84
Confidence Interval (95%)	-49.72 – 0.69	-72.21 – 2.31

Based on the statistical analysis shown in Table 10 the null hypothesis cannot be rejected, but only barely. The p-values for the SCATS PM Southbound are very close to 0.05 and it is seen that 95% confidence interval only barely contains zero. This is also true for SCATS PM Northbound, although the p-value is slightly higher. Also, the practical consideration of the 56 second southbound and 29 second northbound difference between the field data and the simulated data sample means must be considered. As with SCATS AM, while the difference between the two samples may not be statistically significant, the magnitude of the difference is worthy of future research efforts. Of particular interest in this study is the sensitivity of SCATS to changes in traffic demand. The volume counts used to generate the traffic demand in the model were generated based on data collected in 2004. Volume data was not collected during the SCATS travel time data collection completed in 2007 after the SCATS field implementation. The PM simulation results seem to indicate that SCATS is more responsive to changes in demand than ACTRA. Thus, the difference between the SCATS field travel time and the SCATS simulated travel time may be related to differences in the demand between the model (using the 2004 data) and the field (experiencing 2007 demand). Future research will seek to establish the performance effects of demand variation for both SCATS and ACTRA to determine if this may account for the results seen here.

As with all the previous scenarios it is important to evaluate the volumes processed at each intersection. The results of the eight individual runs are shown in Appendix B, Figure 92. Also, the inputs for traffic traveling northbound and southbound entering the corridor was recorded and is displayed in Figure 71 for southbound traffic and Figure 72 for the northbound traffic. In both cases the input volumes appear consistent with the observed field volumes.

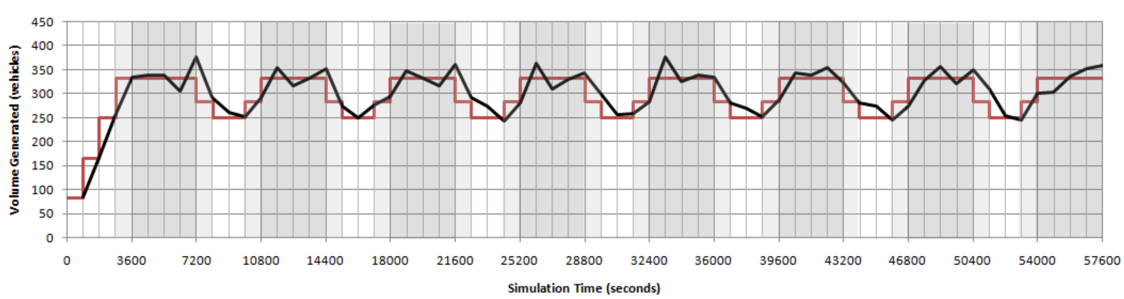

Figure 71. SCATS PM Southbound Entering Volume

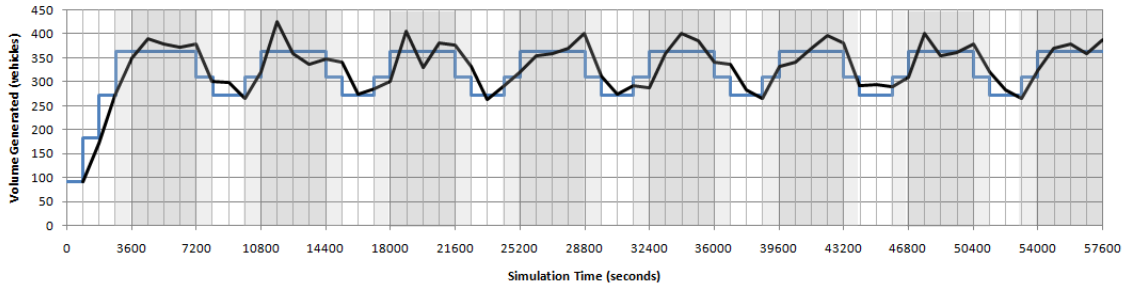

Figure 72. SCATS PM Northbound Entering Volume

Summary of Initial Findings

<u>Main Corridor Travel Time</u>

Previous reports based on the data collected in the field found that neither ACTRA nor SCATS had a significant advantage. However, given the discoveries regarding differences in the field data it may be possible to draw a different conclusion based on simulated comparisons where all other factors, save the control system, are held equal.

The initial analysis compared the performance of the ACTRA AM and PM models against their SCATS counterparts. While previous studies have compared these two control systems as they performed in the field, this is the first test where the input traffic conditions are exactly the same for both control systems. As shown in Table 9 and Table 10 and explained earlier the input traffic volumes and the distribution of turning movements can have a significant effect on the performance of the system.

Table 11 shows the results of the Wilcoxon Ranked Sum Test conducted to compare the travel time results of the ACTRA runs against the SCATS results. The null hypothesis in each of the tests was that the difference in the locations of the two samples was zero. The input data was identical for both the ACTRA and SCATS models and was representative of the traffic seen during the AM and PM peak hours respectively.

Table 11. ACTRA vs. SCATS Base Scenario Travel Time Results

Measurement	**ACTRA AM vs. SCATS AM**		**ACTRA PM vs. SCATS PM**	
	SB	NB	SB	NB
W	9068	647.5	1722.5	7891
p-value	0.3084	0.3734	0.4334	0.6143
Difference in Location	3.05	-8.06	3.25	-2.26
Confidence Interval (95%)	-2.76 – 9.26	-27.10 – 9.68	-5.25 –11.59	-10.91 – 6.45

From the p-values in Table 11 the test fails to reject the null hypothesis for all scenarios. Statistically, this would indicate the there is little or no observed difference between the two control methods under typical AM and PM peak hour conditions. However, as seen previously SCATS control is far more responsive to traffic demand fluctuations than ACTRA. Under SCATS during the peak hours, the focus of the statistical tests, travel time can increase up to the expected value of 300 seconds. Off-peak hours, however, enjoy a much lower travel time approaching 125 seconds. The ACTRA data shown in Figure 54 indicates that the travel time is much more consistent across the entire run. While the peak hour travel time nears and exceeds 300 seconds, the off-peak travel time only decreases to around 200 seconds, much higher than the value for SCATS under the same conditions.

Given these results, further analysis will seek to push traffic to levels beyond existing peak demand simulating volume changes that could be attributed to growth or incidents. Additionally, the turning movement distributions will be altered to simulate special event traffic and evaluate the system's ability to respond to changing traffic patterns.

Conclusions

The results of the case study have clearly demonstrated the power of the HILS framework in providing for the analysis of various ITS strategies. The framework executed the following steps to generate a model that could be attached to the real-world ITS to provide accurate data for evaluation:

1. HILS Setup
2. Model Construction
3. Validation
4. Calibration
5. Simulation Management
6. Analysis Tools
7. Statistical Analysis

The setup of the HILS overcame several technical difficulties to provide an effective system for communication between VISSIM and the real-world 2070 controllers. Further research should seek to implement other adaptive control systems on the existing HILS corridor for further comparative analysis or the system could be expanded to interface with other types of Intelligent Transportation Systems. The validation/verification and calibration of the model was done using Genetic Algorithm applied to a separate, non-HILS, model representative model that would eventually be used in HILS. This showed many of the computation and time-intensive aspects of the model development process can be conducted using far more time-efficient emulation, while still providing meaningful parameter values for use in the final HILS evaluation. Future research should seek to integrate the calibration process into the simulation management framework to make it easier to apply in the construction of a simulation model.

The simulation management and data analysis processes provided for the efficient generation and evaluation of simulation data using the HILS model. The analysis process itself could also be improved as there are several facets of the data that did not receive much attention in the current analysis process. The volume data for each intersection needs to be more rigorously evaluated particularly to ensure that both the SCATS and ACTRA systems are processing the same number of vehicles. Also, the delay information for the side streets needs to be carefully considered.

Overall, the framework proposed has significantly advanced the use of HILS in the evaluation of ITS strategies. While this work centered only on the analysis of SCATS and ACTRA, further research can build upon the findings here to create more effective analysis tools for the evaluation of a broader spectrum of ITS strategies. Additionally, these findings (based on the analysis of an arterial corridor) could be applied to the analysis of different network geometries including a limited-access interstate or a central business district grid. The application of HILS in transportation analysis has proven to be a powerful tool, and one that will hopefully benefit greatly from the research in this study.

APPENDIX A

DATA DISTRIBUTIONS

The following figures present the relevant data distributions for the SCATS and ACTRA travel time. The Travel time section covers both the data collected in the field as well as the data generated from the simulation.

Field Travel Time Distributions

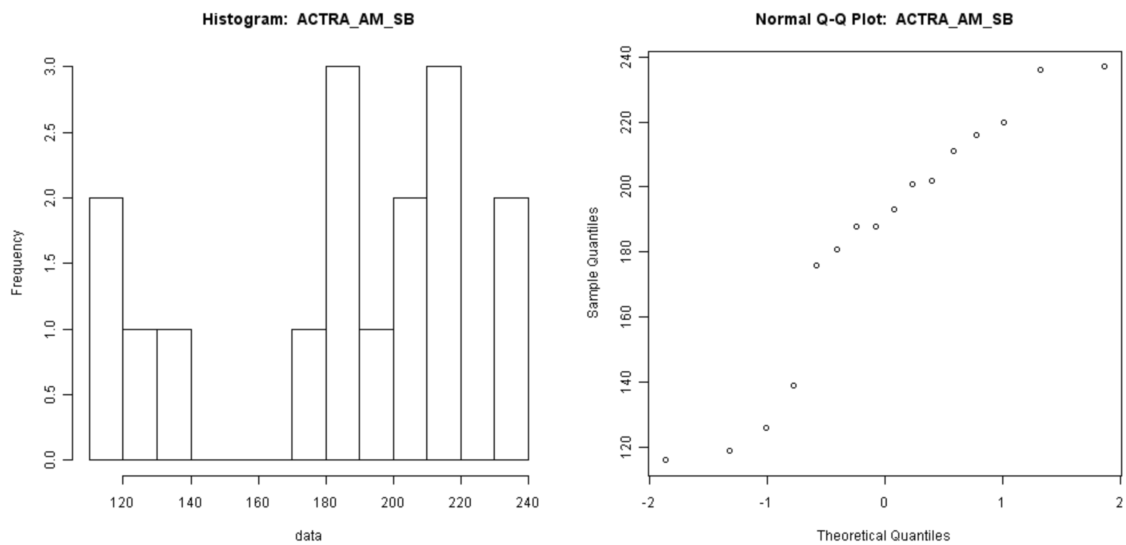

Figure 73. ACTRA AM Southbound Field Travel Time Distribution

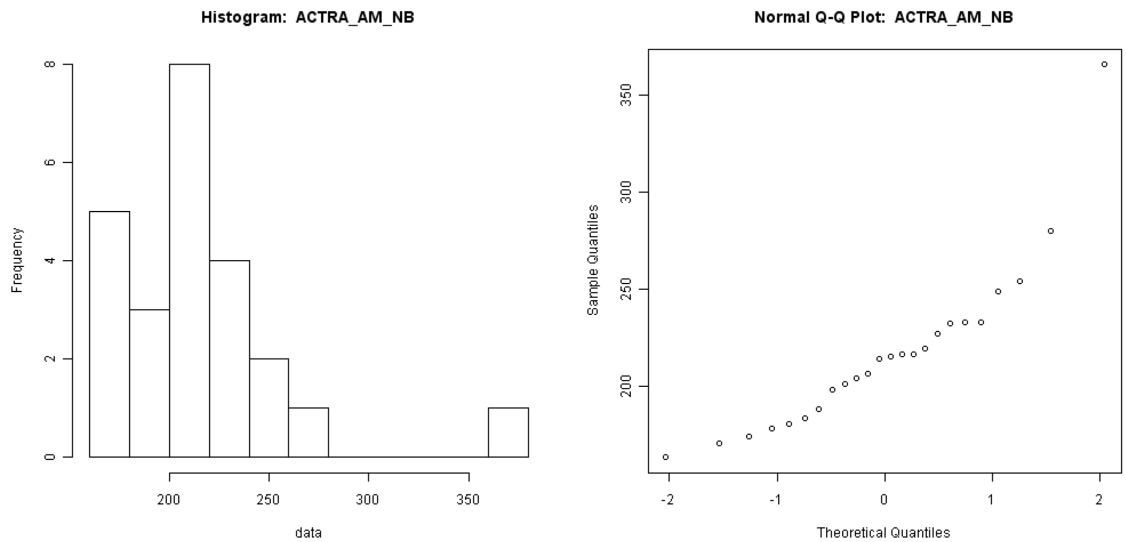

Figure 74. ACTRA AM Northbound Field Travel Time Distribution

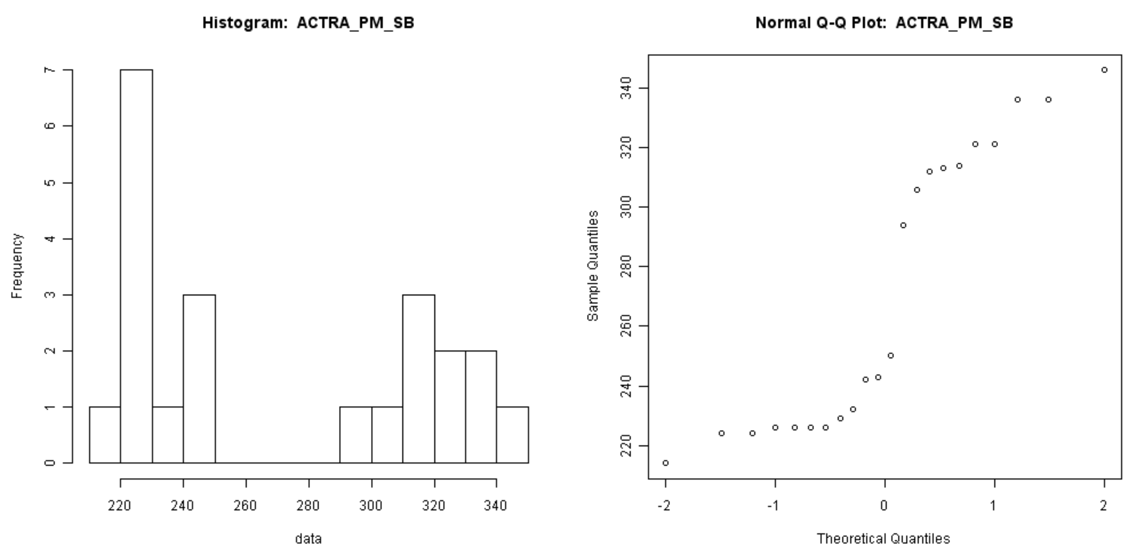

Figure 75. ACTRA PM Southbound Field Travel Time Distribution

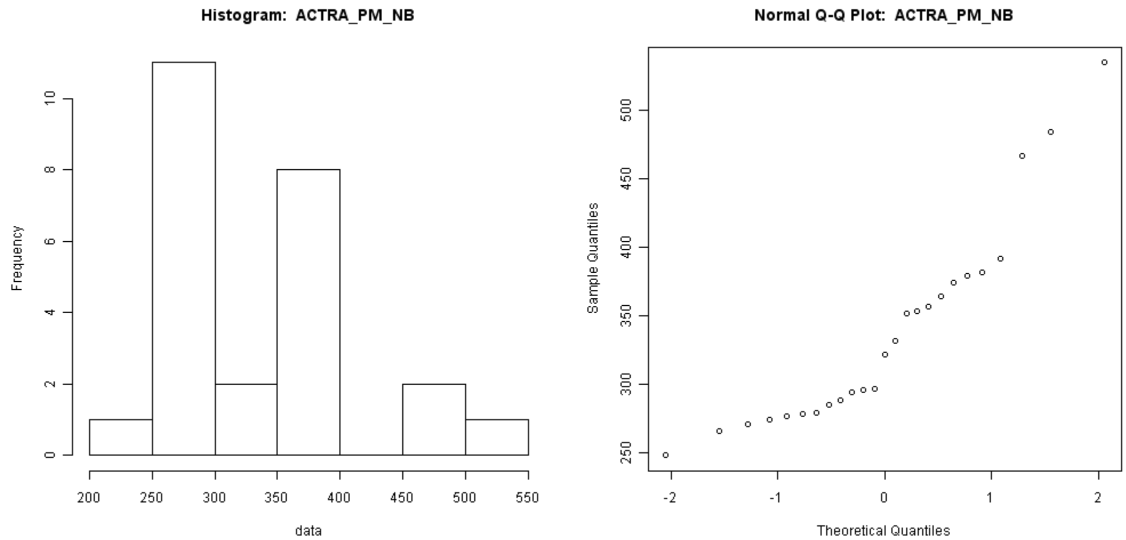

Figure 76. ACTRA PM Northbound Field Travel Time Distribution

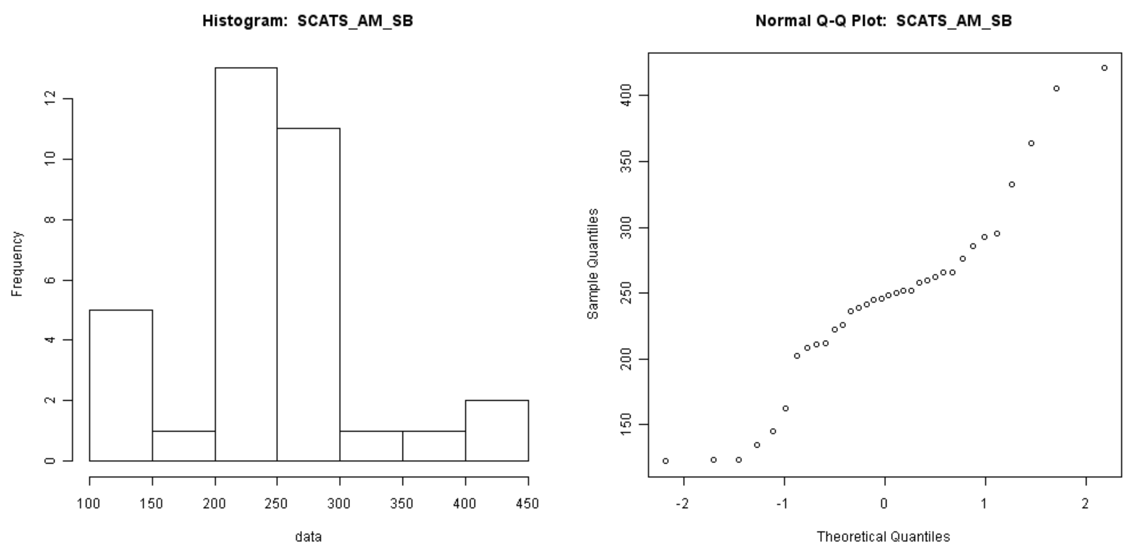

Figure 77. SCATS AM Southbound Field Travel Time Distribution

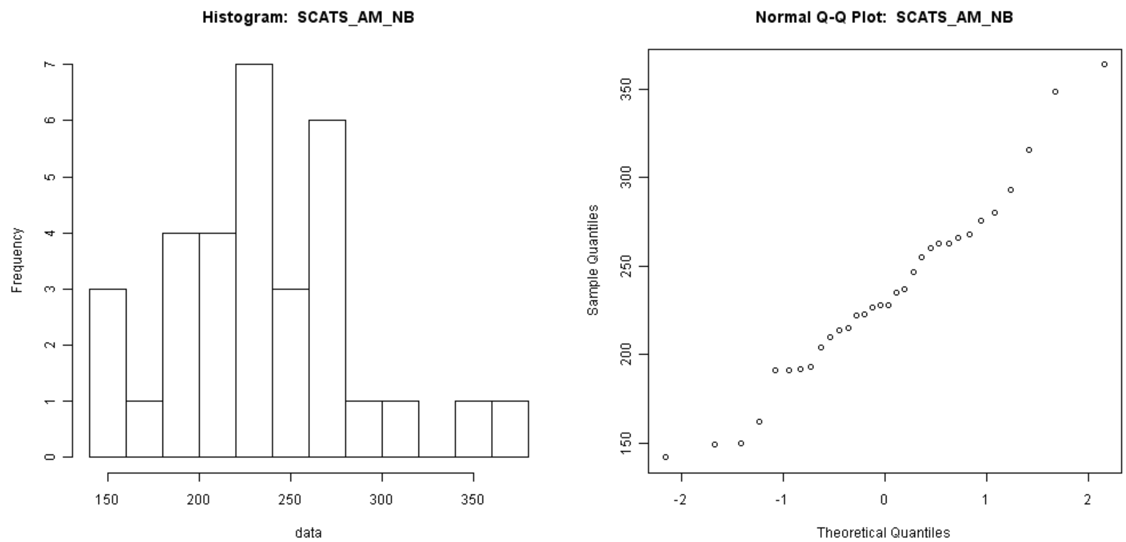

Figure 78. SCATS AM Northbound Field Travel Time Distribution

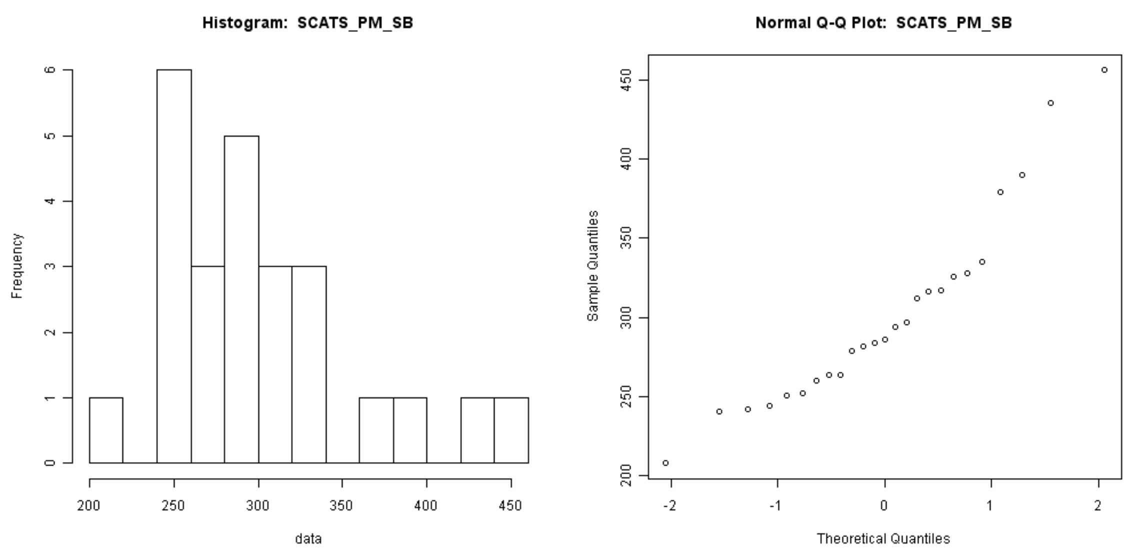

Figure 79. SCATS PM Southbound Field Travel Time Distribution

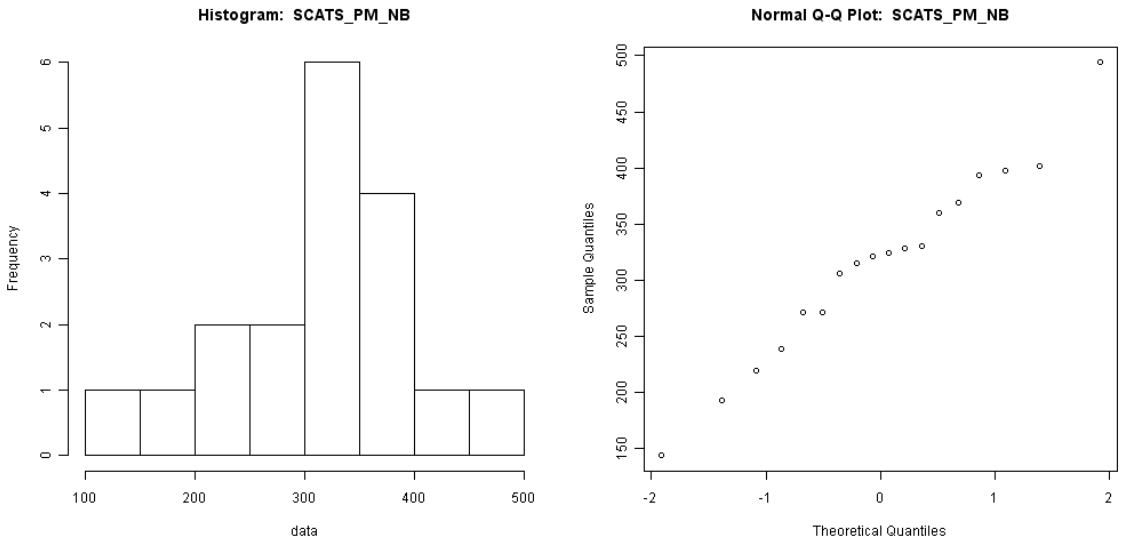

Figure 80. SCATS PM Northbound Field Travel Time Distribution

Simulated Travel Time Distributions

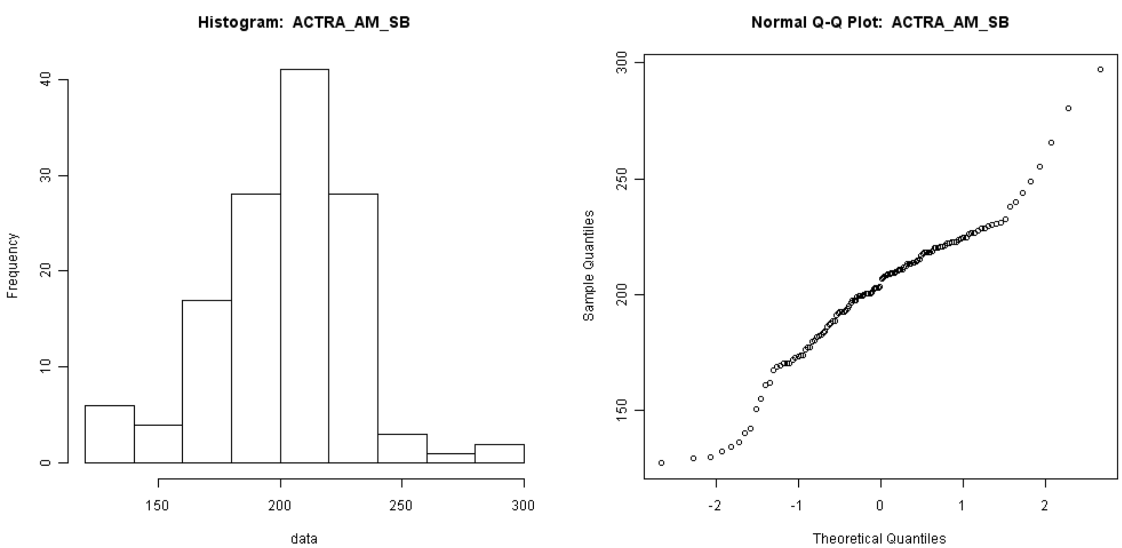

Figure 81. ACTRA AM Southbound Simulated Travel Time Distribution

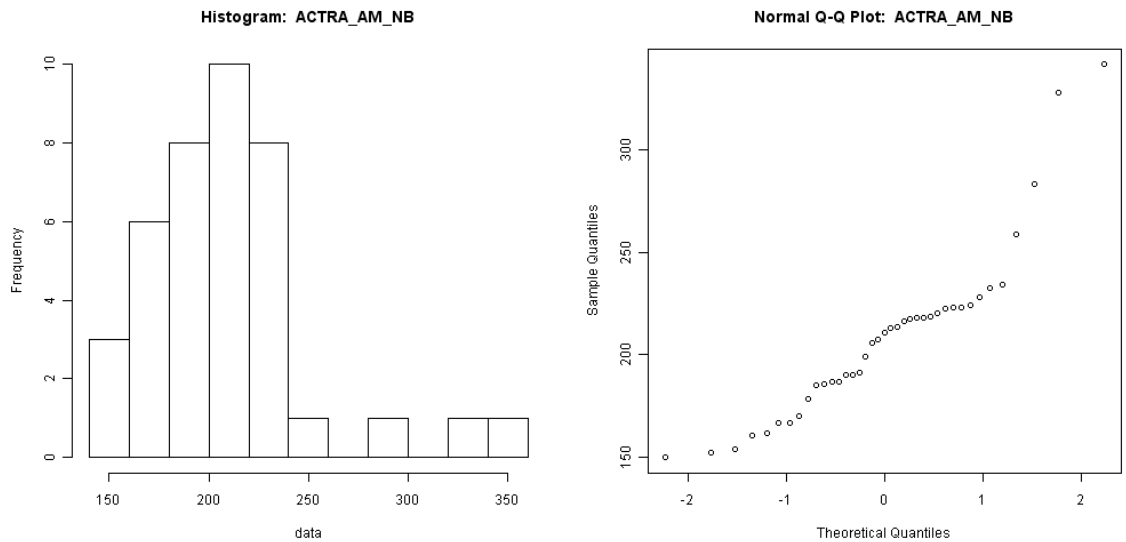

Figure 82. ACTRA AM Northbound Simulated Travel Time Distribution

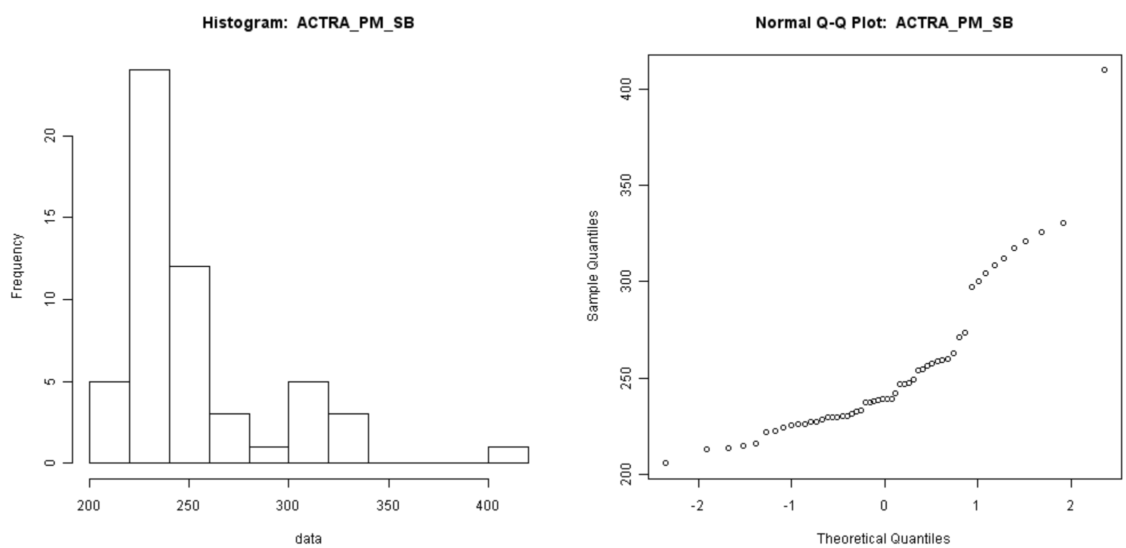

Figure 83. ACTRA PM Southbound Simulated Travel Time Distribution

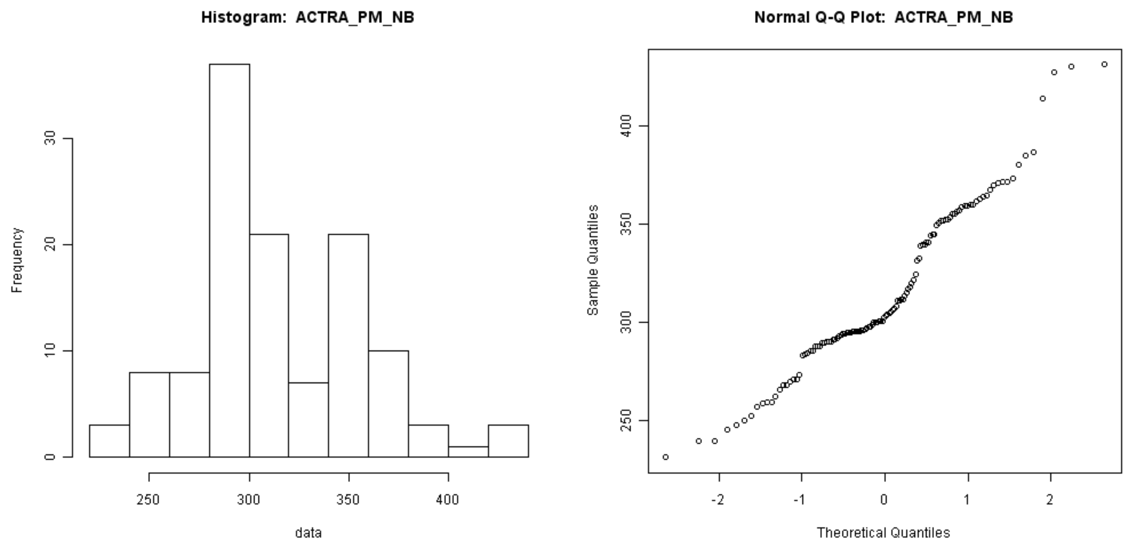

Figure 84. ACTRA PM Northbound Simulated Travel Time Distribution

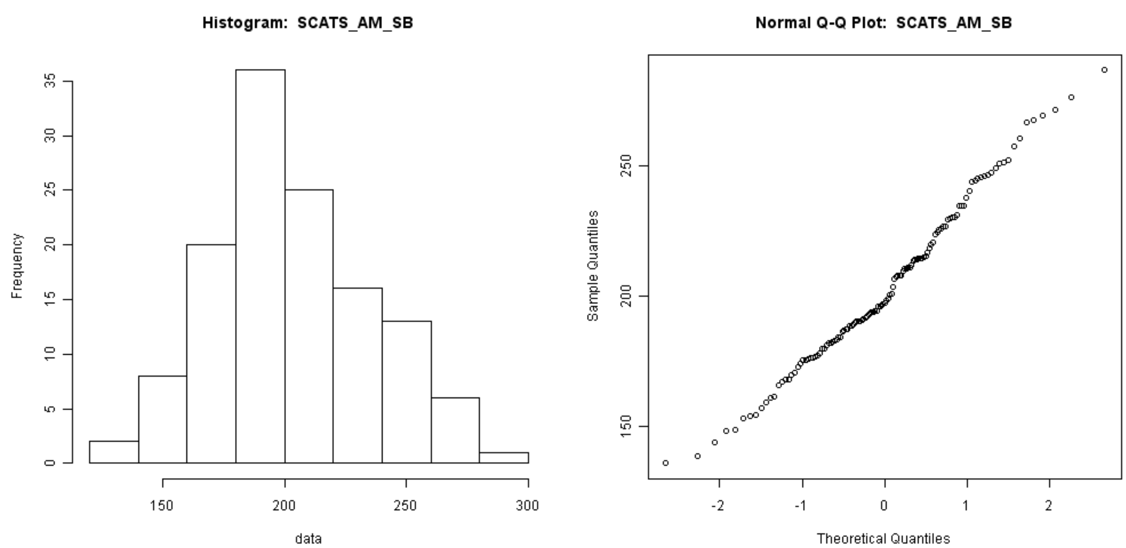

Figure 85. SCATS AM Southbound Simulated Travel Time Distribution

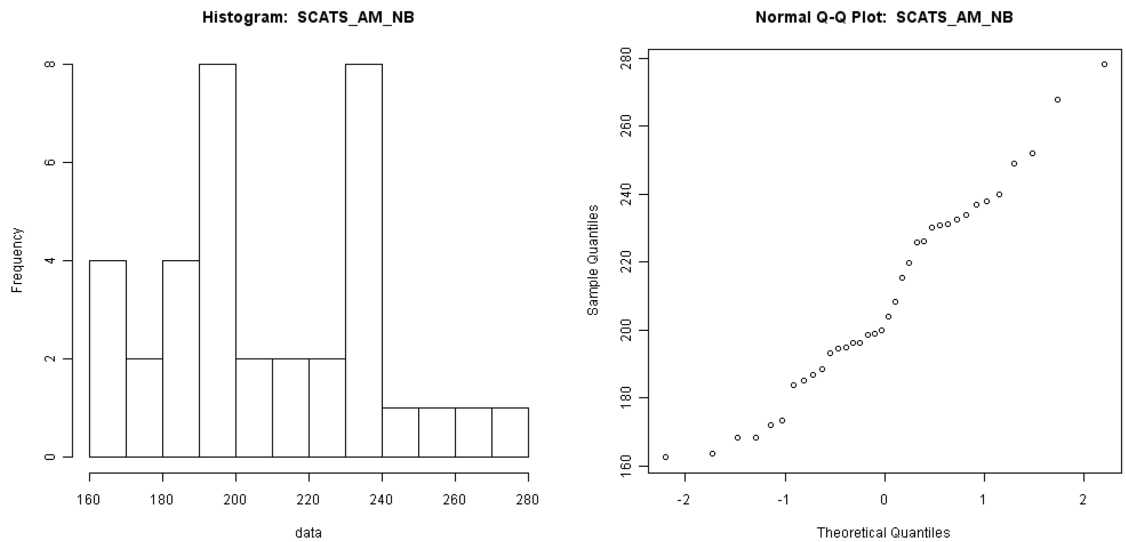

Figure 86. SCATS AM Northbound Simulated Travel Time Distribution

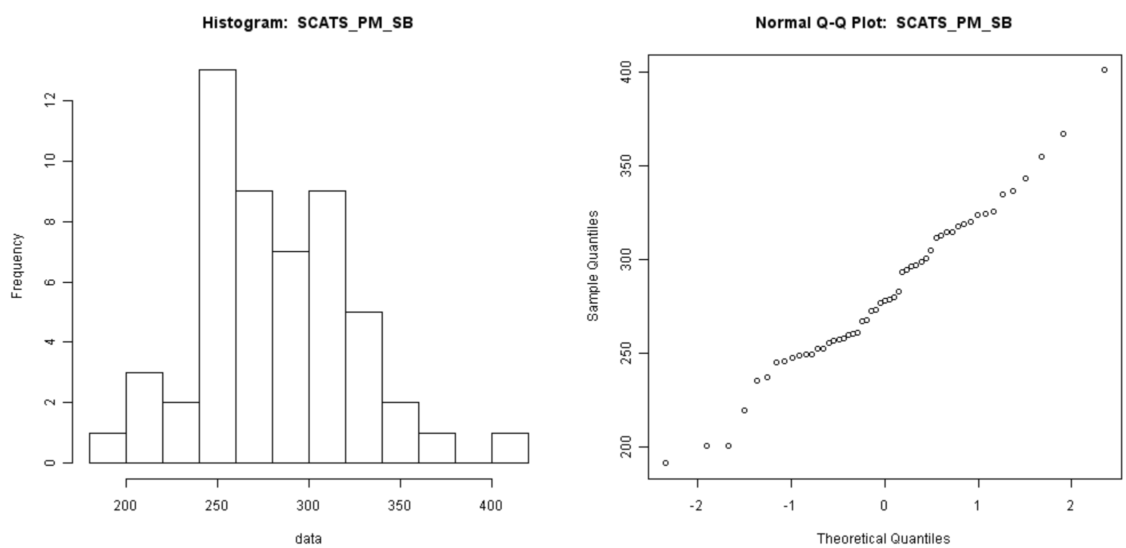

Figure 87. SCATS PM Southbound Simulated Travel Time Distribution

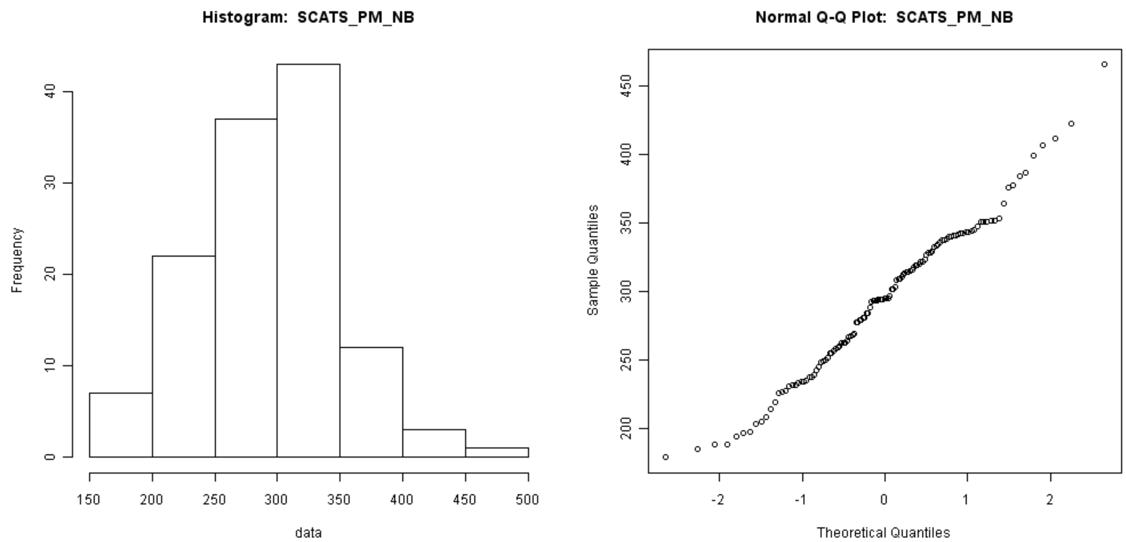

Figure 88. SCATS PM Northbound Simulated Travel Time Distribution

APPENDIX B

FIGURES

The following figures represent the turning movements for each of the intersections in the study network. Each scenario has its own set of figures. Additionally, all of the figures demonstrate the laneage at each intersection based on observation.

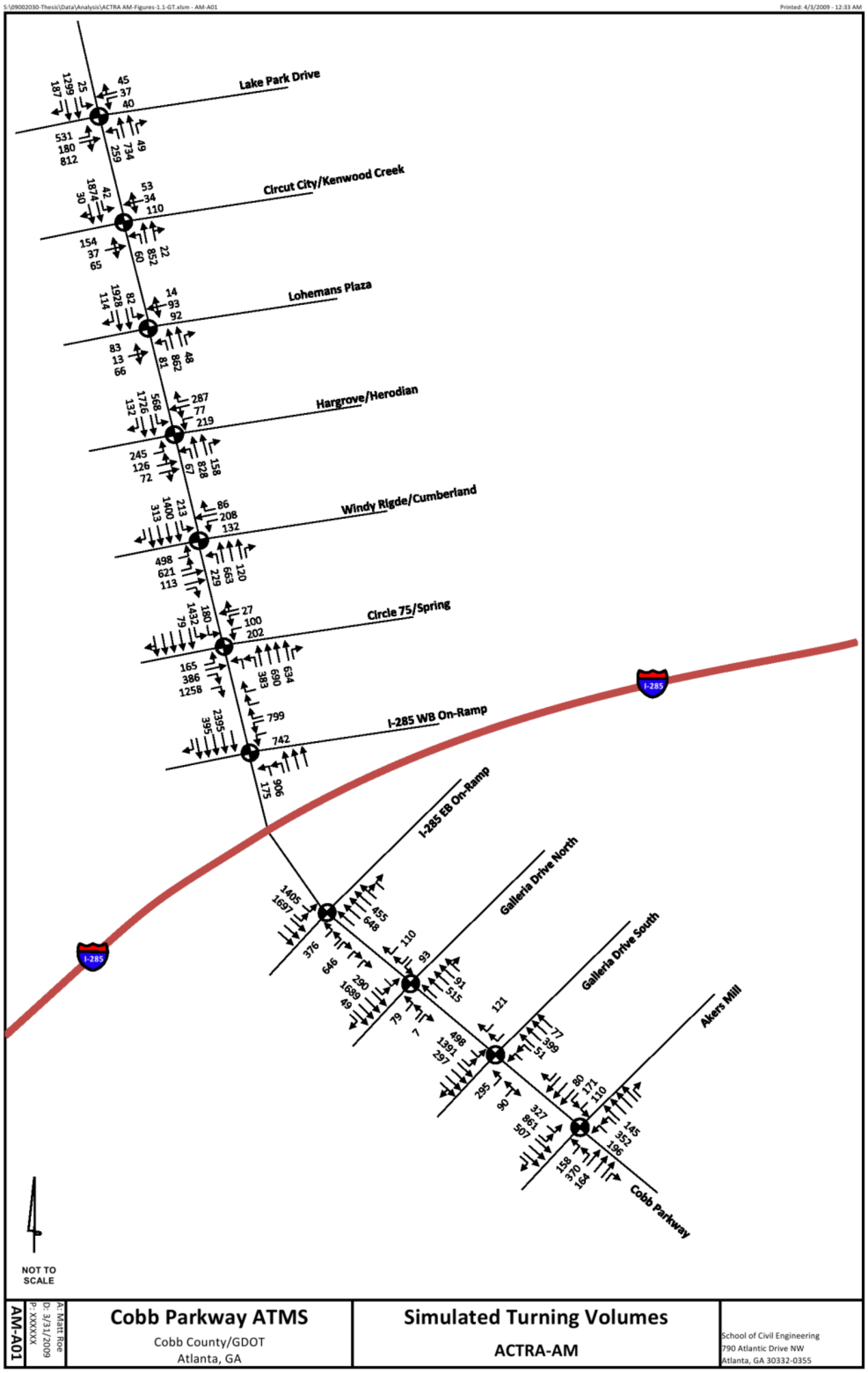

Figure 89. ACTRA AM Simulated Turning Volumes

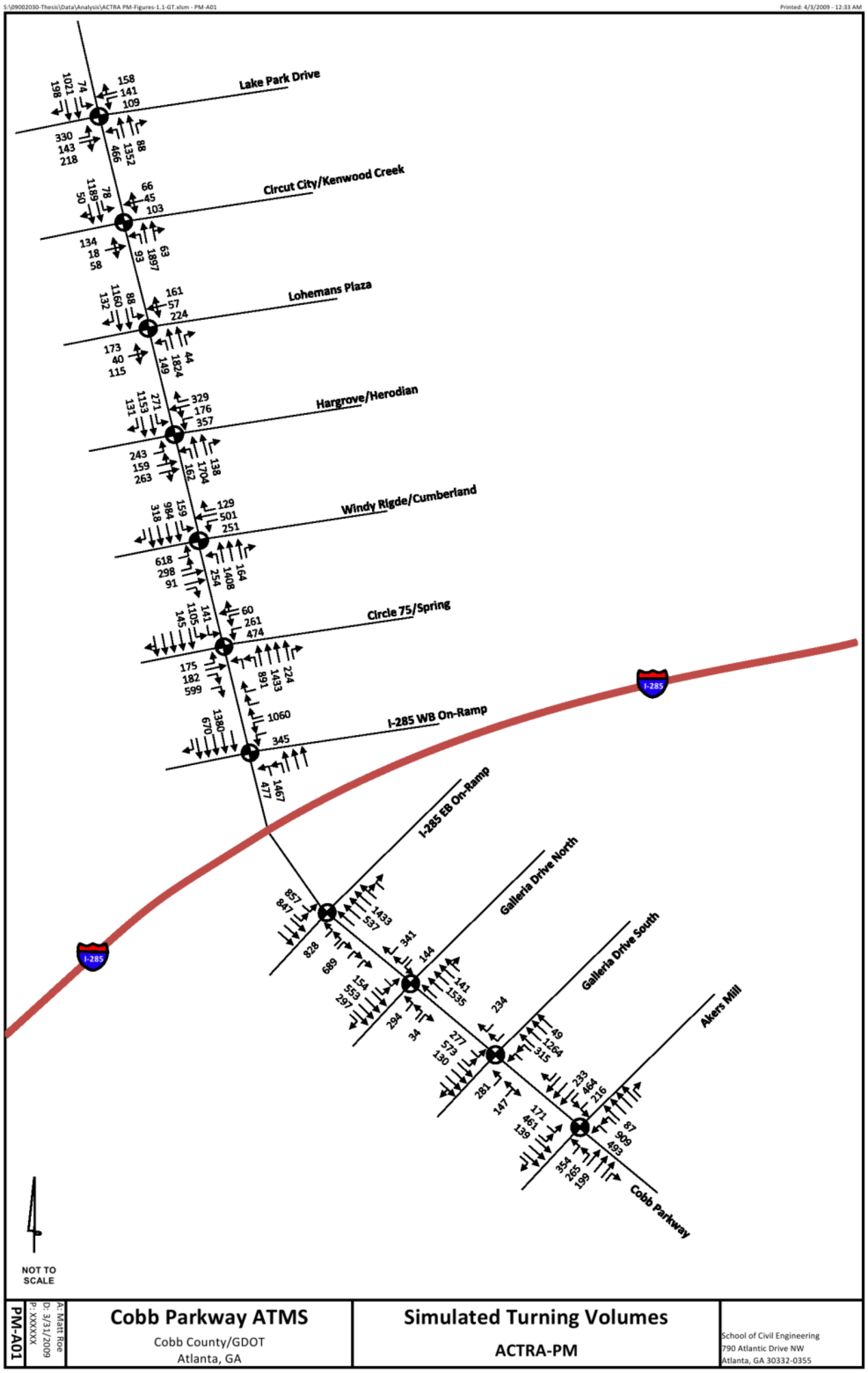

Figure 90. ACTRA PM Simulated Turning Volumes

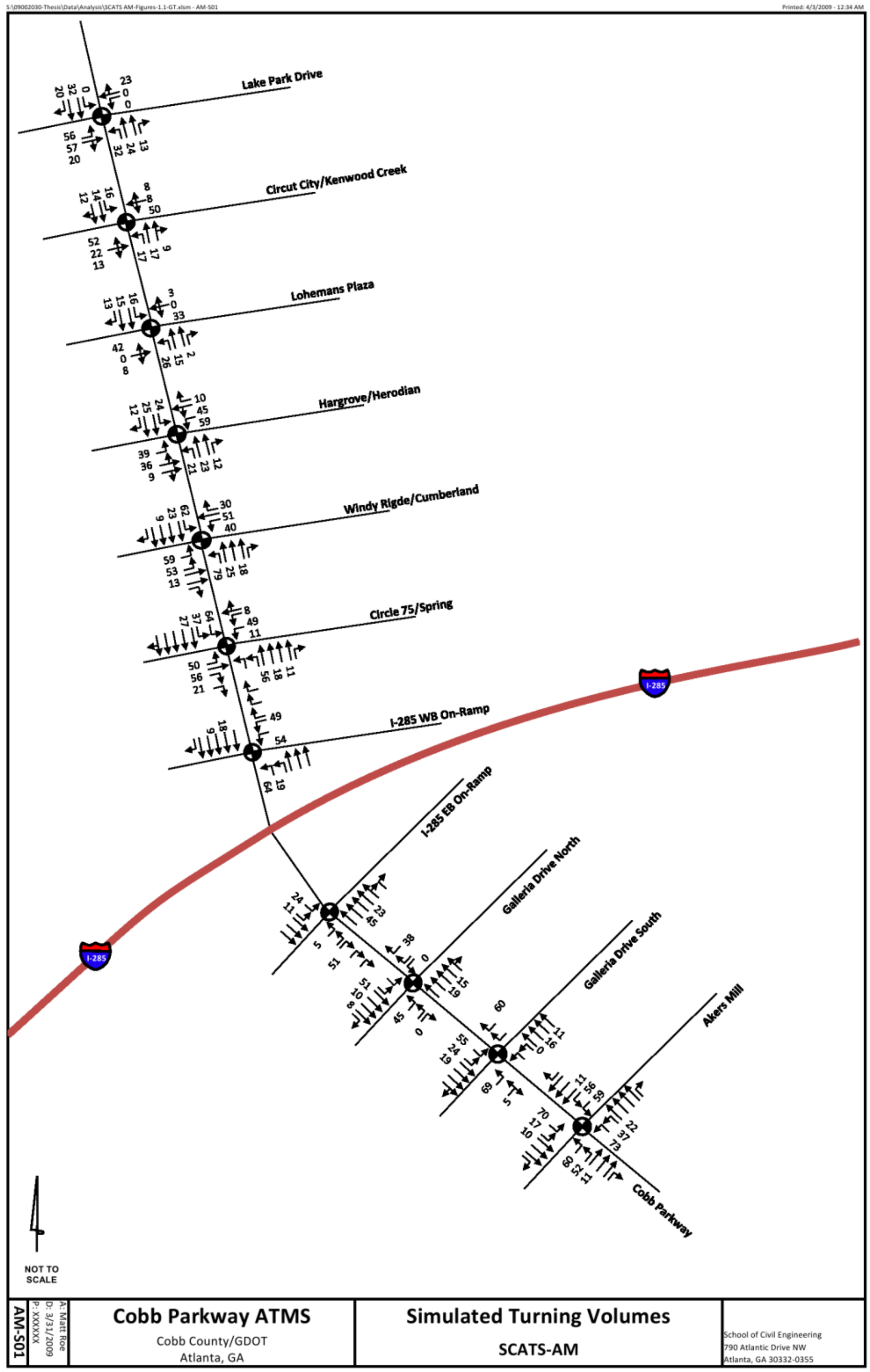

Figure 91. SCATS AM Simulated Turning Volumes

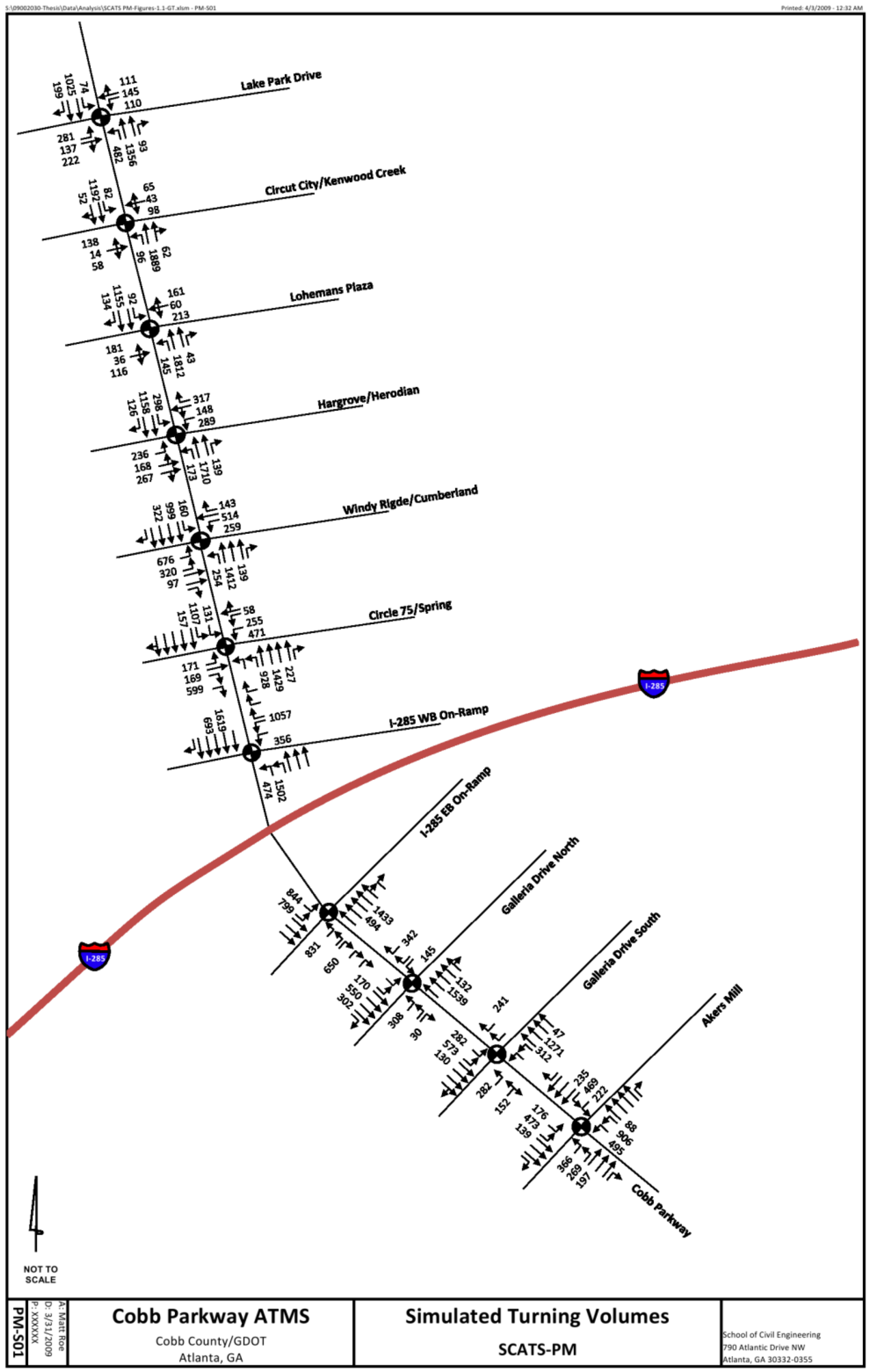

Figure 92. SCATS PM Simulated Turning Volumes

REFERENCES

1. AG, P.T.V., *VISSIM 4.30 User Manual*. 2007, Karlsruhe, Germany: Planung Transport Verhehr AG.

2. AG, P.T.V., *VISSIM 4.30-00 COM Interface Manual*. 2007, Karlsruhe, Germany: Planung Transport Verkehr AG.

3. AG, P.T.V., *VAP 2.16 Interface User Manual*. 2007, Karlsruhe, Germany.

4. Miller, D., *Developing a Procedure to Identify Parameters for Calibration of a VISSIM Model*, in *Civil Engineering*. 2009, Georgia Institute of Technology: Atlanta, GA.

5. Park, B. and H. Qi, *Development and Evaluation of a Calibration and Validation Procedure for Microscopic Simulation Models*. 2004, Virginia Transportation Research Council.

6. Park, B. and J. Won, *Microscopic Simulation Model Calibration and Validation Handbook.* 2006, Virginia Transportation Research Council.

7. America, P., *NEMA Editor Manual*. 2006: PTV America.

8. Goldsman, D., *Introduction to Simulation*, in *2007 Winter Simulation Conference*. 2007, IEEE, Piscataway, NJ, USA: Washington, DC, USA. p. 26-37.

9. Goldsman, D., *Simulation Output Analysis*, in *1992 Winter Simulation Conference*. 1992, IEEE, New York, NY, USA: Arlington, VA, USA. p. 97-103.

10. Kelton, D., *Analysis of Output Data*, in *1994 Winter Simulation Conference*. 1994, IEEE, New York, NY, USA: Lake Buena Vista, FL, USA. p. 62-68.

11. Smadi, A. and S. Birst, *Use of Hardware-in-the-Loop Traffic Simulation in a Virtual Environment* in *Applications of Advanced Technologies in Transtportation Engineering*. 2006. p. 635-640.

12. Chatila, H. and Z. Li, *US-95 ACS-Lite System Evaluation*. 2007, Idaho Transportation Department. p. 21.

13. Toledo, T. and H.N. Koutsopoulos, *Statistical Validation of Traffic Simulation Models.* Transportation Research Record, 2004. **1876**: p. 142-150.

14. Ciuffo, B.F., V. Punzo, and V. Torrieri, *A Framework for the Calibration of Microscopic Traffic Flow Models*, in *Transportation Research Board Annual Meeting 2007*. 2007.

15. Park, B. and J. Won, *Simulation Model Calibration and Validation: Phase II: Development of Implementation Handbook and Short Course*. 2006, Virginia Transportation Research Council.

16. Yu, L., X. Li, and W. Zhuo, *A Genetic Algorithm-Based Appraoch to the Calibration of VISSIM Using GPS Data*, in *Applications of Advanced Technologies in Transtportation Engineering*. 2004. p. 406-411.

17. Dowling, R., A. Skabardonis, and V. Alexiadis, *Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Software*, in *Traffic Analysis Toolbox*. 2004, Federal Highway Administration.

18. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1989: Addison Wesley Longman, Inc.

19. Fu, M.C., C.-H. Chen, and L. Shi, *Some Topics for Simulation Optimization*, in *2008 Winter Simulation Conference*. 2008: Austin, TX, USA. p. 27-38.

20. Liu, H.X., et al., *Large-Scale Traffic Simulation Through Distributed Computing of Paramics* in *PATH Research Report*. 2004, California Department of Transportation. p. 26.

21. Ziliaskopoulos, A., J. Zhang, and H. Shi, *Hybrid Mesoscopic-Microscopic Traffic Simulation Model: Design, Implementation, and Computational Analysis* in *Transportation Research Board 85th Annual Meeting*. 2006.

22. Mohamed, Y. and S.M. AbouRizk, *Framework for Building Intelligent Simulation Models of Construction Operations* Journal of Computing in Civil Engineering, 2005. **19**(3): p. 277-291.

23. Fang, F.C. and L. Elefteriadou, *Some Guidelines for Selecting Microsimulation Models for Interchange Traffic Operational Analysis* Journal of Transportation Engineering, 2005. **131**(7): p. 535-543.

24. Chabrol, M. and D. Sarramia, *Object Oriented Methodology Based on UML for Urban Traffic System Modeling*, in *UML 2000, LINCS 1939*, A. Evans, S. Kent, and B. Selic, Editors. 2000. p. 425-439.

25. Prevedouros, P., D. Brauer, and R.J. Sykes, *Development of Interactive Visualization Toool for Effective Presentation of Traffic Impacts to Nonexperts.* Transportation Research Record, 1994. **1463**: p. 35-44.

26. Easa, S.M., et al., *Three-Dimensional Transportation Analysis: Planning and Design.* Journal of Transportation Engineering, 2002. **128**(3): p. 250-258.

27. Goldsman, D. and G. Tokol, *Output Analysis Procedures for Computer Simulations*, in *2000 Winter Simulation Conference*. 2000, IEEE, Piscataway, NJ, USA: Orlando, FL, USA. p. 39-45.

28. Seila, A.F., *Advanced Output Analysis for Simulation*, in *1992 Winter Simulation Conference*. 1992, IEEE, New York, NY, USA: Arlington, VA, USA. p. 190-7.

29. Ycuesan, E., *Nonparametric Techniques in Simulation Analysis: A Tutorial*, in *1994 Winter Simulation Conference*. 1994, IEEE, New York, NY, USA: Lake Buena Vista, FL, USA. p. 99-105.

30. Alexopoulos, C. and D. Goldsman, *To Batch or not to Batch*, in *2003 Winter Simulation Conference*. 2003, IEEE, Piscataway, NJ, USA: New Orleans, LA, USA. p. 481-9.

31. Keenan, D., *M62-M1 Interchange - A Combined TTRANSYT/VISSIM Micro-simulation Assessment*, in *Traffic Engineering and Control*. 2005, Hemming Group Ltd. p. 178.

32. Hunter, M., et al., *Cobb Parkway ATMS Evaluation*. 2009, Georgia Institute of Technology: Atlanta, GA.

33. Hayter, A.J., *Probability and Statistics for Engineers and Scientists*. 2nd ed. 2002, Pacific Grove, CA, USA: Duxbury Thompson Learning.