Traffic Management Simulation Development

Final Report

Contract: BDK80 TWO #977-03

FIU Project: 212201540

Prepared for

The Florida Department of Transportation

by the

Florida International University

Lehman Center for Transportation Research

January 4, 2011

Disclaimer

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

Metric Conversion Chart

APPROXIMATE CONVERSIONS TO SI UNITS

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
LENGTH				
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
AREA				
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
VOLUME				
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
NOTE: volumes greater than 1000 L shall be shown in m^3				
MASS				
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
TEMPERATURE (exact degrees)				
^oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	^oC
ILLUMINATION				
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
FORCE and PRESSURE or STRESS				
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Traffic Management Simulation Development		5. Report Date January 03, 2011
		6. Performing Organization Code
7. Author(s) Mohammed Hadi, Chengjun Zhan, Patricio Alvarez		8. Performing Organization Report No.
9. Performing Organization Name and Address Lehman Center for Transportation Research Florida International University 10555 W. Flagler Street, EC 3680, Miami, FL 33174		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. BDK80 TWO #977-03
12. Sponsoring Agency Name and Address Office of Research and Development State of Florida Department of Transportation 605 Suwannee Street, MS 30, Tallahassee, FL 32399-0450		13. Type of Report and Period Covered Draft Final Report October 2008 – December 2010
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Microscopic simulation can provide significant support to traffic management center (TMC) operations. However, traffic simulation applications require data that are expensive and time-consuming to collect. Data collected by TMCs can be used as a primary source for the provision of the required data. The goal of this project is to explore the development of methods and tools for the use of microscopic traffic simulation models to support the TMC software implementation, operation, and testing on one hand, and the use of Intelligent Transportation System (ITS) data to support the development and calibration of simulation models on the other. The project products include software utilities that use the existing TMC databases and other available information for the preparation and calibration of microscopic simulation tools. In addition, the products include utilities to support testing of the TMC software modules and data archiving processes, as demonstrated by use cases of the tools developed in this study.

17. Key Word	18. Distribution Statement		
Traffic Simulation, Intelligent Transportation Systems, Traffic Detectors, TMC Software	Unrestricted		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 148	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Executive Summary

Traffic management center (TMC) software is an essential component of TMCs. The SunGuide software has been adopted as traffic management software for use at Florida Department of Transportation (FDOT) TMCs throughout the state. Microscopic simulation can provide significant support for TMC software operations. However, traffic simulation applications require data that are expensive and time-consuming to collect. Data collected by TMCs can be used as a primary source for the provision of this data.

The goal of this project is to explore the development of methods and tools for the use of microscopic traffic simulation models to support the TMC software implementation, operation, and testing on the one hand, and the use of Intelligent Transportation System (ITS) data to support the development and calibration of simulation models on the other. The specific objectives of the project are as follows:

- Development of software utilities that use the existing SunGuide databases and other available information for the preparation and calibration of microscopic simulation tools
- Development of software utilities to support the testing of ITS Data Warehouse processes by producing traffic sensor system (TSS) data, travel time data, and other measures, as needed, in the SunGuide archive format based on simulation outputs
- Development of software utilities to allow the exchange of data between the SunGuide software and virtual detectors in a simulation environment, for use in the SunGuide subsystem testing and operation evaluation
- Demonstration of the use and value of the developed SunGuide simulation support to project stakeholders, including FDOT Traffic Engineering Research Laboratory (TERL) and FDOT districts ITSs
- Documentation and technology transfer of project activities, results, and developed tools.

The review of literature in this study includes a review of the current state of traffic simulation modeling as it applies to the support of TMCs. In addition, an introduction of the components of the SunGuide software that are relevant to this study is provided. Additional discussion of the ITS data warehouse implementation in Florida is also included.

Based on the results of this study, it can be concluded that ITS data archives have the potential to provide cost-effective and detailed information for the development and calibration of simulation modeling applications. To achieve this goal, detectors in the field should be well-calibrated and maintained to reduce inconsistencies in the measurements. More importantly, advanced data maintenance, aggregation, filtering, and imputation capabilities should also be provided to these ITS data archives to ensure data quality and the optimal use of archived data.

This study has developed procedures and tools to use ITS data archives to populate microsimulation application datasets with the required volume and speed data. The developed procedures and tools allow the collection of data from the data archive, manipulation and aggregation of the data, and automatic modification of the input files for microscopic simulation tools. Although the CORSIM microscopic simulation tool was used for this study, the methods developed in this project can be easily extended for use with other simulation tools. In addition, these tools can be used in the estimation of traffic parameters based on ITS data archives for other planning, travel demand forecasting, and traffic analysis purposes.

This study also has developed procedures for the utilization of ITS data archives to support the use of simulation to calibrate simulation models for incident and no-incident conditions. The ITS data archives utilized include both incident management and point traffic detector data. It can be concluded from the results presented in this study that, when simulating incident conditions, the analyst should fine-tune simulation model parameters to produce the expected or measured drops in capacity during incidents and the expected impacts of incidents on various performance measures. Simply specifying lane blockages is not sufficient to produce the expected impacts of incidents in the simulation.

Two major software components have been developed in this project to satisfy these requirements. The two components are referred to collectively as SunSim. The first component is the SunSim core simulation support environment which supports the development of simulation models based on ITS data and user inputs. The second component is the SunSim TSS simulators, which are software utilities that allow the exchange of data between the SunGuide software and virtual detectors in a simulation environment for use in the SunGuide subsystem testing and operation evaluation.

A number of use cases are designed in this study to demonstrate the use of the developed simulation environment in evaluating the SunGuide software modules and algorithms. These use cases include software load test, travel time estimation based on point detectors, travel time estimation using Automatic Vehicle Identification (AVI) and/or License Plate Recognition (LPR) technologies, incident alarm threshold procedure testing, and ITS data warehousing process testing.

Table of Contents

List of Figures

List of Tables

List of Selected Acronyms and Abbreviations

Acronyms and abbreviations used in the report are listed below.

AIMSUN	An integrated transport modeling software
API	Application Programming Interface
ATIS	Advanced Traveler Information System
ATMS	Advanced Transportation Management System
AVI	Automatic Vehicle Identification
CA	Cellular Automata
CCTV	Closed-circuit Television
CORSIM	A microscopic traffic simulation software package
CSV	Comma-Separated Value
DLL	Dynamic Link Library
DMS	Dynamic Message Sign
DTA	Dynamic Traffic Assignment
EIS	Electronic Integrated Systems
ETC	Electronic Toll Collection
FDOT	Florida Department of Transportation
FIAS	Freeway Incident Analysis System
FHWA	Federal Highway Administration
FREEVAL	A software tool for Freeway Evaluation
GIS	Geographic Information System
GUI	Graphic User Interface
HAR	Highway Advisory Radio
HCM	Highway Capacity Manual
HIL	Hardware in the Loop
HOV	High Occupancy Vehicle
ICM	Integrated Corridor Management
ISM	Intermodal Strategy Manager
ITS	Intelligent Transportation System
LPR	License Plate Recognition

Matlab	A numerical computing environment developed by MathWorks
MOE	Measurement of Effectiveness
Paramics	A traffic micro-simulation software
PeMS	Performance Measurement System
RTMS	Remote Traffic Microwave Sensor
RTRDS	Real Time Route Diversion System
RWIS	Road Weather Information System
SIL	Software in the Loop
STEWARD	Statewide Transportation Engineering Warehouse for Archived Regional Data
SwRI	Southwest Research Institute
TERL	FDOT Traffic Engineering Research Laboratory
TMC	Traffic Management Center
TSS	Traffic Sensor System
TVT	Travel Time
VISSIM	A microscopic multi-modal traffic flow simulation software
XML	Extensible Markup Language

1. Introduction

1.1. Background

Traffic management centers (TMCs) are a critical component of advanced traffic management systems. At TMCs, crucial decisions are made in both real-time and off-line to optimize the performance of transportation systems. TMC software that supports an effective operation and management of transportation systems is essential to the function of TMCs. The SunGuide traffic management software has been adopted and installed for use at Florida Department of Transportation (FDOT) TMCs throughout the state. The SunGuide TMC software is a set of Intelligent Transportation System (ITS) software that allows the control of roadway devices, as well as information exchange, across transportation agencies.

Microscopic simulation modeling has been widely used in the transportation engineering field. These models can both benefit from and provide benefits to traffic management implementations and operations in two ways:

- First, simulation models can provide significant support to traffic management operations in both off-line and on-line applications. Off-line, simulation models can be used to test potential ITS deployments and alternative strategies by acting as a virtual environment that emulates real-world conditions. The SunGuide system has a strong capability to communicate with external hardware and software. This capability can be used to communicate with virtual ITS devices in the simulation environment in a similar manner to communicating with real-world ITS devices, allowing the testing of various modules, algorithms, and deployment strategies associated with the SunGuide within the simulation environment. This off-line application between traffic simulation software and the TMC software can potentially be extended for the on-line execution of simulation in conjunction with SunGuide operations to assess alternative operation strategies in real-time
- Second, the development and calibration of traffic simulation applications require data that are expensive and time-consuming to collect. Traffic, incident, and

other data collected by TMCs can be used as a primary source for the provision of this data. However, additional data filtering, cleaning, imputation, fusion, and aggregation capabilities would be required to use such data.

Recent developments of these simulation models include the ability to exchange data with external devices or external software applications. Several examples exist of hardware in the loop (HIL) and software in the loop (SIL). The most widely used of these applications involve the connection of the traffic signal controller hardware or software to simulation models. Other recent developments in simulation models also include the ability to code alternative traffic management algorithms and processes, using powerful Application Programming Interface (API) facilities. These facilities allow for the interfacing of advanced algorithms and methods, written in computer programming languages like C and C++, with virtual ITS devices that can be specified by the users. Examples include the use of APIs for traffic simulation software to model traffic responsive ramp metering, adaptive signal control, incident detection algorithms, and automatic dynamic message sign (DMS) messaging.

1.2. Project Objectives

The goal of this project is to explore the development of methods and tools for the use of microscopic traffic simulation models to support the SunGuide system implementation, operation, and testing on the one hand, and the use of ITS data to support the development and calibration of simulation models on the other. The specific objectives of the project are the following:

- Development of software utilities that use the existing SunGuide databases and other available information for the preparation and calibration of microscopic simulation tools
- Development of software utilities to support the testing of ITS Data Warehouse processes by producing traffic sensor system (TSS) data, travel time data, and other measures, as needed, in the SunGuide archive format based on simulation outputs

- Development of software utilities to allow the exchange of data between the SunGuide software and virtual detectors in a simulation environment, for use in the SunGuide subsystem testing and operation evaluation
- Demonstration of the use and value of the developed SunGuide simulation support to project stakeholders, including FDOT Traffic Engineering Research Laboratory (TERL) and FDOT districts ITS
- Documentation and technology transfer of project activities, results, and developed tools.

1.3. ITS Data

The SunGuide TMC software is a set of ITS software that allows for the control of roadway devices as well as the exchange of information across transportation agencies. The software represents a common software base that has been deployed by FDOT districts throughout the state of Florida.

The SunGuide system maintains operational data in several different places for use in report generation. Aggregated operational data are stored in Oracle database files, while the raw data are stored in comma-separated value (CSV) format files. Below is a description of the two archived SunGuide files that were used for the purposes of this study:

- Incident Archives: For each SunGuide incident record, the stored information includes incident timestamps (detection, notification, arrivals, and departures), incident ID, responding agencies, event details, chronicle of the event, and environmental information. The detection timestamp is the time when an incident is reported to the TMC and inputted in the SunGuide system. The notification timestamps are recorded per responding agency and refer to the time when such responding agencies are notified. The arrival and departure timestamps are also recorded per responding agency and refer to the time when responding agencies arrive and depart from the incident site

- Detector Data Archives: The traffic condition data are stored in TSS text flat files, with each file including data for all the detection stations for a 24-hour day. The TSS file contains one record for each lane of each detection station for every 20-second polling interval. Each TSS data record includes the timestamp, detection station name, lane number, speed, occupancy, and raw count information.

In addition, the Statewide Transportation Engineering Warehouse for Archived Regional Data (STEWARD) has been developed as a proof of concept prototype for the collection and use of ITS data in Florida (Courage and Lee, 2008). The current effort has concentrated on archiving point traffic detector data and travel time estimates. The STEWARD database contains summaries of traffic volumes, speeds, and occupancies collected from point traffic detectors. This database was also used in this study, as described when discussing the various products of this project.

Information from the above sources and other sources, possibly combined with data generated by simulation modeling, can be integrated to provide a rich data environment for advanced analysis to support the TMC decision making processes, and for the delivery of more accurate, reliable, and useful information.

1.4. Overview of the Project Activities

This section presents an overview of the project activities and associates these activities with the following chapters in this report:

- The literature review presented in Chapter 2 of this report presents a summary of the current state of traffic simulation modeling as it applies to the support of TMCs. Moreover, an introduction to the relevant components of the SunGuide software is provided herein. Additional discussion of the ITS data warehouse implementation in Florida is also included
- Chapter 3 includes a discussion of procedures and tools developed in this project to use ITS data archives for the population of microsimulation application datasets with the required volume and speed data. The developed procedures and tools allow for the collection of data from the data archive, the manipulation and

aggregation of the data, and the automatic modification of the input files to microscopic simulation tools. To this end, the CORSIM microscopic simulation tool was used in this study. However, the aforementioned tools and methods can be easily extended for use with other simulation tools. In addition, these tools can be used in the estimation of traffic parameters based on ITS data archives for other planning, travel demand forecasting, and traffic analysis purposes

- Chapter 4 presents procedures for the utilization of ITS data archives to support the use of simulation to calibrate simulation models for incident and no-incident conditions. The ITS data archives utilized include both incident management and point traffic detector data
- Chapter 5 outlines the two major software components that have been developed in this project to satisfy the aforementioned requirements. The two components are referred to collectively as SunSim. The first component is the SunSim core simulation support environment which supports the development of simulation models based on ITS data and user inputs. The second component is the set of SunSim TSS simulators, which are software utilities that allow the exchange of data between the SunGuide software and virtual detectors in a simulation environment for use in the SunGuide subsystem testing and operation evaluation
- Chapter 6 presents a number of use cases that are designed to demonstrate how the developed simulation environment is used to evaluate the SunGuide software modules and algorithms. These use cases include software load test, travel time estimation based on point detectors, travel time estimation using AVI/LPR, and incident alarm threshold procedure testing
- Appendices A through D present user and installation manuals and data schemas of the developed tools.

2. Literature Review

As discussed in the project objectives, one of the main tasks of this project is to develop an integrated simulation environment for SunGuide functionality evaluation. In this chapter, a summary is given about the current state of traffic simulation modeling as it applies to the support of TMCs. In addition, an introduction of the components of the SunGuide software that are relevant to this study is provided herein. Additional discussion of the ITS data warehouse implementation in Florida is also included.

2.1. Simulation Modeling

This section first provides a brief overview of the existing simulation tools. The subsequent subsections then describe the various levels of the implementation of simulation modeling for ITS.

2.1.1. Existing Simulation Tools

Simulation models are typically classified according to the level of detail, which represents the traffic stream as follows (Jeannotte et al., 2006):

- ***Macroscopic simulation models***: Macroscopic simulation models are based on the deterministic relationship of the flow, speed, and density of the traffic stream, as described above. The simulation in a macroscopic model takes place on a section-by-section basis rather than by tracking individual vehicles. Macroscopic models are less complicated and have considerably lower computer requirements than microscopic models. They do not, however, have the ability to analyze transportation improvements in as much detail as the microscopic models. The Freeway Evaluation (FREEVAL) tool, the computational engine of the HCM freeway facility, is an example of this procedure. Hadi et al. (2010) have recently used FREEVAL to assess the impacts of incident and incident management on system performance
- ***Mesoscopic simulation models***: Mesoscopic simulation models combine the properties of both microscopic (discussed below) and macroscopic simulation

models, as discussed earlier. As in microscopic models, the mesoscopic models' unit of traffic flow is the individual vehicle. Their movement, however, follows the macroscopic approach of traffic flow. Mesoscopic models provide less fidelity than the microsimulation tools, but are superior to the typical planning analysis and macroscopic modeling techniques. These models run much faster than microscopic models and require less effort to use and calibrate. They are particularly useful when combined with dynamic traffic assignment for simulating large networks that require the modeling of strategic decisions. For ITS applications, mesoscopic simulation models are useful to test applications that impact strategic driver decisions, such as dynamic message signs, traveler information systems, managed lanes, and congestion pricing. Examples of mesoscopic simulation models are Dynasmart-P, Cube Avenue, VISTA, DynaMIT, Transmodeler, and Dynameq

- ***Microscopic simulation models***: These models simulate the characteristics and interactions of individual vehicles. They produce trajectories of vehicles as they move through the network. Both vehicle movements and driver decisions are modeled, including car following, acceleration, deceleration, gap acceptance, and lane changing maneuvers. With these models, vehicles are tracked through the network over small time intervals (e.g., one-second or a fraction of a second). Modeling and computer requirements for microscopic models are large, usually limiting the network size and the number of simulation runs that can be completed. However, they are ideal to model strategies such as signal control, ramp metering, bus priority, and signal preemption. In addition, they are required if accurate simulated traffic detector measurements (volume, speed, and occupancy measurements) are the necessary outputs for the simulation. Examples of microscopic simulation models are VISSIM, CORSIM, AIMSUN, and PARAMICS. Another widely used tool is SimTraffic, which is the microscopic simulation tool that is integrated with the Synchro signal optimization program. However, the use of this tool is only appropriate for signalized arterials, and not for freeways.

A number of "hybrid" simulation modeling implementations have been proposed to combine the three modeling levels mentioned above, in order to utilize the individual strengths of each simulation modeling level.

As described later in this document, two important attributes of the simulation tools are needed for this project:

- The ability to simulate traffic detectors to test algorithms and modules that use data collected by these detectors, and
- The provision of an API extension of the microscopic simulation tool to allow for the coding of traffic management algorithms and modules.

These needs can only be achieved using microscopic simulation tools. However, the implementation could be expanded in the future to include a hybrid microscopic/mesoscopic implementation, if the modeling of routing decisions in a regional network is required. The use of macroscopic models such as FREEVAL could also be useful for specific ITS applications.

2.1.2. Simulation of ITS Using Traditional Data Collection

The simplest application of microscopic simulation models to ITS evaluation is to develop and calibrate a simulation model of a transportation system using data collected with the traditional data collection approaches. In such an application, the analyst uses the built-in features of the simulation model, in most cases combined with API modules, to simulate the operations of ITS devices. The four most commonly used simulation models (CORSIM, VISSIM, AIMSUN, and PARAMICS) allow, or can be extended using their APIs to allow, the coding of ITS devices and algorithms such as point detectors, ramp metering, incident detection, incidents, diversion plans, and signal control. These models (except CORSIM) permit the explicit modeling of dynamic message signs (DMS), vehicle re-routing in response to in-vehicle messages, and, in some cases, dynamic speed limit. Although CORSIM does not allow the modeling of DMS explicitly, the analyst can emulate DMS impacts by changing the simulation tool input of the percentages of traffic existing at off-ramps at different time periods of the simulation. All four simulation tools mentioned above also allow the modeling of managed lanes; however, in the case of CORSIM, the percentage of drivers using the managed lane has to be obtained using external means since there is no built-in dynamic traffic assignment module in CORSIM. In the case of the other three tools, dynamic traffic assignment can be used to split traffic between managed lanes, general use lanes, and other alternative routes.

Abdulhai et al. (1999) developed a general list of requirements for a microscopic traffic simulator to successfully model ITS. The study evaluated Paramics 1.5, which was a relatively new microscopic traffic simulation tool at the time of the evaluation. The performance of the tool was validated in two phases; first on a small freeway section and then on a large network of freeways and surface streets in the City of Irvine. The study concluded that the tool is an excellent framework with which to assess ITS strategies. The study also discussed a list of recommended improvements to the tool.

A scan of the literature indicates that there are numerous papers that applied simulation to assess ITS strategies. Examples of these applications include evaluating incident detection algorithms, incident management, travel time estimation, travel time prediction, probe surveillance ramp metering, signal control, DMS, ATIS, in-vehicle navigation systems, managed lanes, smart work zones, and more recently, IntelliDrive applications.

2.1.3. Simulation Modeling Based on ITS Data

This section reviews another category of studies that combine ITS and simulation. These studies discuss the use of ITS data to populate the input files in simulation tools and possibly to calibrate these simulation tools.

Few studies have investigated the use of archived ITS data for simulation modeling applications. Gomez et al. (2004) presented a procedure for constructing and calibrating a detailed model of a freeway, based on detector data using VISSIM. Field data used as input for the model was compiled from two separate sources: loop-detectors on the on-ramps and mainline, stored in a central database referred to as the Performance Measurement System (PeMS), and a manual survey of on-ramps and off-ramps. Gaps in both sources made it necessary to use both traffic detector data and manually collected data sets. A data processing algorithm was implemented to filter, aggregate, and correct the PeMS data. The developed procedure was applied to a 15-mile stretch of I-210 West in Pasadena, California. This site included a high occupancy vehicles' (HOV) lane with an intermittent barrier, a heavy freeway connector, twenty metered on-ramps with and without HOV bypass lanes, and three interacting bottlenecks. A data processing algorithm was implemented using the MATLAB software to filter, aggregate, and correct the PeMS data

Barceló et al. (2002, 2003) described the implementation of a microscopic simulation tool (AIMSUN) to support traffic management strategies in the Rhine-Main region, which includes the City of Frankfurt in Germany. The project integrated an ITS data warehouse with the AIMSUN modeling environment. This integration allowed the off-line analysis and fine-tuning of traffic management strategies. The results from this off-line analysis are integrated in an on-line process that allows the traffic management system to automatically implement the strategies predefined using the simulation environment. The developed integrated environment is illustrated in Figure 2-1. Figure 2-2 illustrates the architecture of the Intermodal Strategy Manager (ISM), which uses the predefined strategies in on-line applications. The pre-defined strategies include recommendations for regional and temporal diversion of traffic, as well as the diversion from road traffic to public transport. ISM supports the automatic selection of the corresponding strategies and the verification of the predefined conditions to activate these strategies.

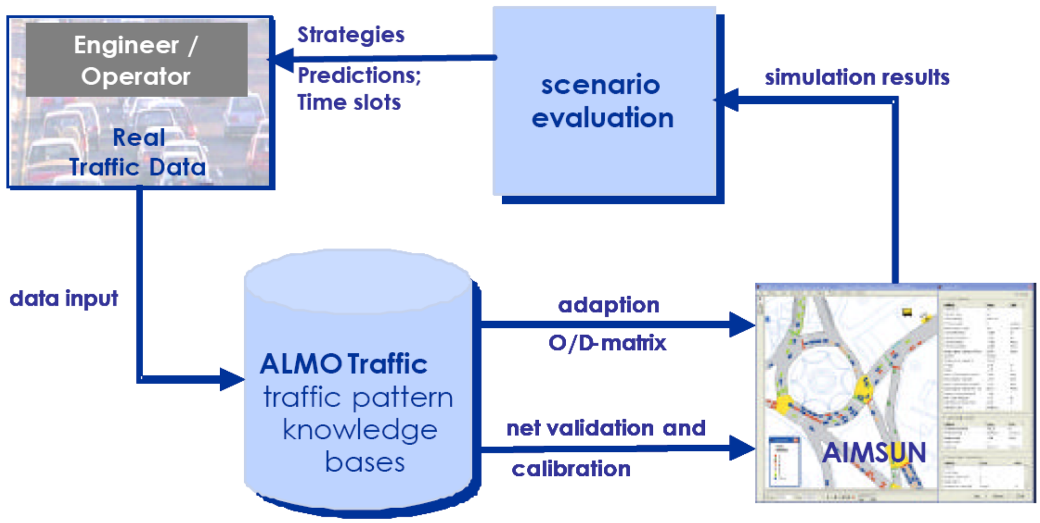


Figure 2-1 Integrated AIMSUN and ALMO Environment

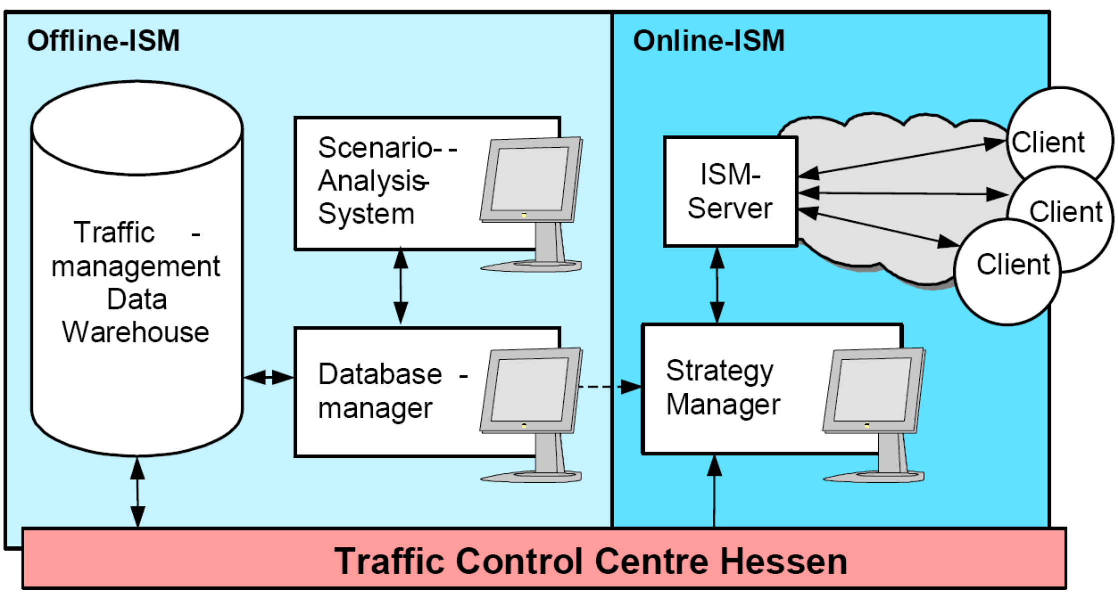


Figure 2-2 Conceptual Architecture of ISM

2.1.4. Real-Time Use of Simulation at TMCs

A number of studies have proposed the use of microscopic and mesoscopic simulation models in real-time applications at TMCs. The section includes a review of these studies.

2.1.4.1. Cellular Automata Models

A number of researchers have proposed the use of Cellular Automata (CA) models for TMC applications. In CA models, the simulated freeway is subdivided into homogeneous cells and the simulation time is divided into time-steps. The low computational cost of CA models as compared to microscopic simulation enable the large-scale modeling of urban traffic in a short time period, which is a requirement for real-time applications. In addition, CA models have fewer parameters than microscopic models, making them easier to calibrate.

Dailey and Taiyab (2002) presented a cellular automata model for traffic flow simulation and prediction (CATS). The inputs to the model are the real-time estimation of volume and density based on traffic sensor data. This model can be used to predict travel time and other measures, given the real-time estimation of traffic demands.

Chang et al. (2007) developed a decision support tool called the Freeway Incident Analysis System (FIAS). FIAS utilizes real-time data from toll collection and vehicle detection systems, in conjunction with historical data, to predict traffic flow measures and formulate recommended decision plans. With the developed model, the CA simulation models incident scenarios and produces estimates of the measures of effectiveness (MOEs), which are then displayed in an animated 2D or 3D environment. Chang et al. (2007) stated that the main difference between their model and other CA models is the ability to better simulate congested flows and breakdown conditions.

2.1.4.2. Mesoscopic Simulation/DTA Models

Dynamic Traffic Assignment models based on mesoscopic simulation have been proposed for real-time applications in previous studies. This section presents a review of some of these studies.

As part of an FDOT research project, Aved et al. (2007) proposed the use of a route diversion management system that utilizes a DTA/mesoscopic simulation tool, combined with real-time traffic information from field sensors and TMC operator information, in order to dynamically generate route diversion plans. The system, referred to as the Real-Time Route Diversion System (RTRDS), utilizes the Dynasmart-P mesoscopic simulation software developed for the Federal Highway Administration (FHWA). Aved et al. (2007) proposed to integrate the RTRDS with the Florida SunGuide software, as shown in Figure 2-3. They stated that, by utilizing RTRDS, TMC operators can choose either from route diversions generated dynamically or from a saved historical route diversion plan.

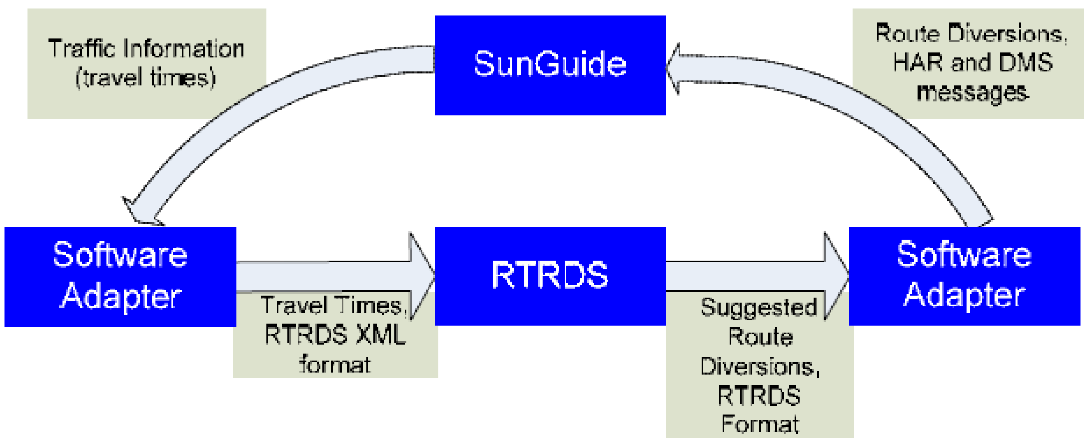


Figure 2-3 Proposed RTRDS Processes (Source: Aved et al., 2007)

The proposed RTRDS consists of the following four main components:

- A graphical user interface that permits the users to create/edit/modify route diversion plans, query for existing plans, review/edit DMS contents, etc.
- A set of software components (drivers) that "listen" for incoming traffic data and convert it into the RTRDS format
- The RTRDS server process, which determines the current traffic conditions and incidents, prepares input files for Dynasmart-P, executes Dynasmart-P, and processes the Dynasmart-P output; RTRDS writes updated route diversion plans back to the database where they can be accessed from the GUI
- Additional scripts to load information from a commercial map database.

Mahmassani et al. (2005) applied the Dynasmart-X (the real-time version of Dynasmart which differs from the Dynasmart-P software used in the FDOT project mentioned above) to the CHART network in Maryland. The goal of the CHART network application was to use the prediction and estimation capabilities of the simulation tool in conjunction with real time information to assess multiple traffic management strategies and scenarios in real-time. In this way, if an unplanned disturbance occurs, the operator at the TMC can make changes to the tool interface to reflect the disturbance (e.g., specifying an incident of a given severity as input to the model). This allows the simulator to adjust traffic pattern and diversion to account for changes to the physical network or control processes, and more accurately replicate field conditions.

Park et al. (2003) discussed a pilot study of conducting an online implementation of DynaMIT in Hampton Roads, Virginia. A test network was coded and DynaMIT input parameters were calibrated. DynaMIT was found to be able to perform relatively well in estimating and predicting sensor counts with a root mean square normalized (RMSN) error ranging between 0.15 and 0.25 for estimations, and between 0.25 and 0.4 for predictions. Although speed and travel times showed some discrepancies between the simulation and real-world conditions, further investigations indicated that the performance of DynaMIT can be significantly improved by adequately calibrating the simulation parameters.

2.1.4.3. Microscopic Simulation

The use of microscopic simulation for on-line applications at TMCs has been studied by a number of researchers. Fries et al. (2007) studied the effectiveness and timeliness of using microsimulation to support real-time decision making in a regional TMC. The study identified the time required to estimate the travel time with a desired combination of confidence level and margin of error for different incident scenarios. The researchers simulated incidents of 30-, 60-, and 120-minute duration. The study determined the number of simulation replications that produced a 95% confidence level with different margins of error. Two networks from South Carolina, each three to four miles long with two or three interchanges, were considered. The simulations were performed using

PARAMICS. Overall PARAMICS was found to meet the running time criteria. However, it should be noted that the simulated networks in the study were small with relatively light traffic. Thus, the running time of microscopic simulation models for real-time applications may need to be further investigated for a more realistic modeling of congested networks.

Zou et al. (2003) proposed an approach that combines a knowledge-based system and simulation for real-time incident management applications. The proposed system offers an immediate estimate of traffic conditions based on similar previous scenarios stored in the knowledge-based subsystem. The effectiveness of previously employed congestion management strategies is also available from the knowledge-based system for incident operations. The estimates of traffic conditions are presented initially to the operator as preliminary estimates. The estimates are also used to allow the microscopic simulation module to select a sub-network for modeling. The sub-network is selected by the knowledge-based system, with consideration to the most likely impacts given the characteristics of the incident. After the first simulation runs, the impacts initially estimated by the knowledge-based module are replaced by the estimates from the simulation. The simulation tool used to conduct this process was CORSIM. The spatial network segmentation was based on the constraint that the results from the simulation should be updated in a time period of less than five minutes for a one-hour simulation.

Another study by Barceló and García (2002) discussed the lessons learned and experience gained from four major European R&D projects. In one of these projects, the ENTERPRICE project, the decision support functions were based on the real-time simulation of alternative management strategies. The project developed a translator that automatically builds the AIMSUN simulation model with the scenario selected from a Geographic Information System (GIS).

In a more recent paper, Torday et al. (2008) discussed the development of an “on-line” version of the AIMSUN software for application at TMCs, as shown in Figure 2-4. The core of the model is the same as the off-line version of AIMSUN. However, the simulation is distributed over several computers to increase the computational efficiency. The simulation tool uses detector data to determine the current level of demand in the network. Based on the detector data, the tool selects an origin-destination (O-D) matrix from a stored matrix library using pattern-matching. First, all yearly traffic demands are grouped in 10 to 20 different patterns, each with a different O-D matrix. Each of these matrices is then further sliced into 96 15-minute O-D matrices. The tool models events such as accidents and highway construction. In addition, the tool simulates control and information dissemination strategies such as ramp metering, DMS, and dynamic speed limit systems. The AIMSUN on-line environment described above is said to be implemented at the Madrid TMC in Spain.

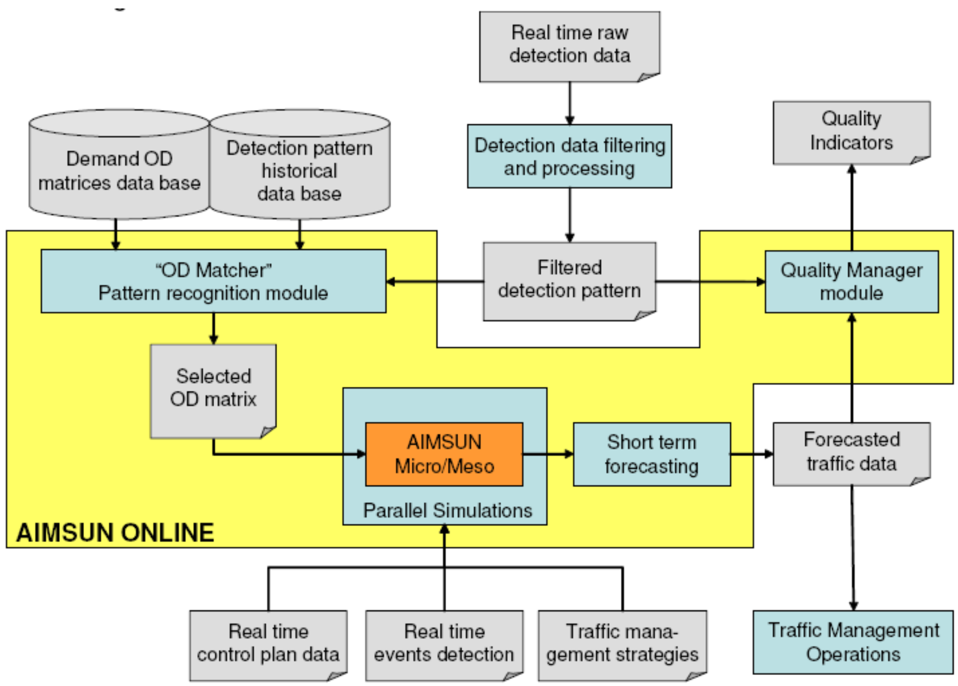


Figure 2-4 Architecture of Online Version of AIMSUN

2.2. SunGuide Software

The SunGuide TMC software is a set of ITS software that allows for the control of roadway devices as well as information exchange across transportation agencies. The software represents a common software base that has been deployed by FDOT districts throughout the state of Florida. The SunGuide development effort began in October 2003.

Since then, six major releases of the software and a number of versions of each release have been developed. The following SunGuide Software Project documents were reviewed at the beginning of this study:

- *Introduction to an Operational Concept for the Florida Statewide Library*, FDOT – OCD – 1.0, March 31, 2002
- *Software Requirements Specification, draft,* February 22, 2008
- *SunGuide: Concept of Operations*, January 3, 2005
- *SunGuide: Software Architecture Guidelines*, February 16, 2006
- *SunGuide: Software User's Manual*, December 4, 2008
- *Transportation Sensor System Device Driver Interface Control Document* November 21, 2003

Figure 2-5 provides a graphical view of Release 5.0 of the software. The following subsections include a discussion of some of the components of the SunGuide software as they relate to the subject of this study.

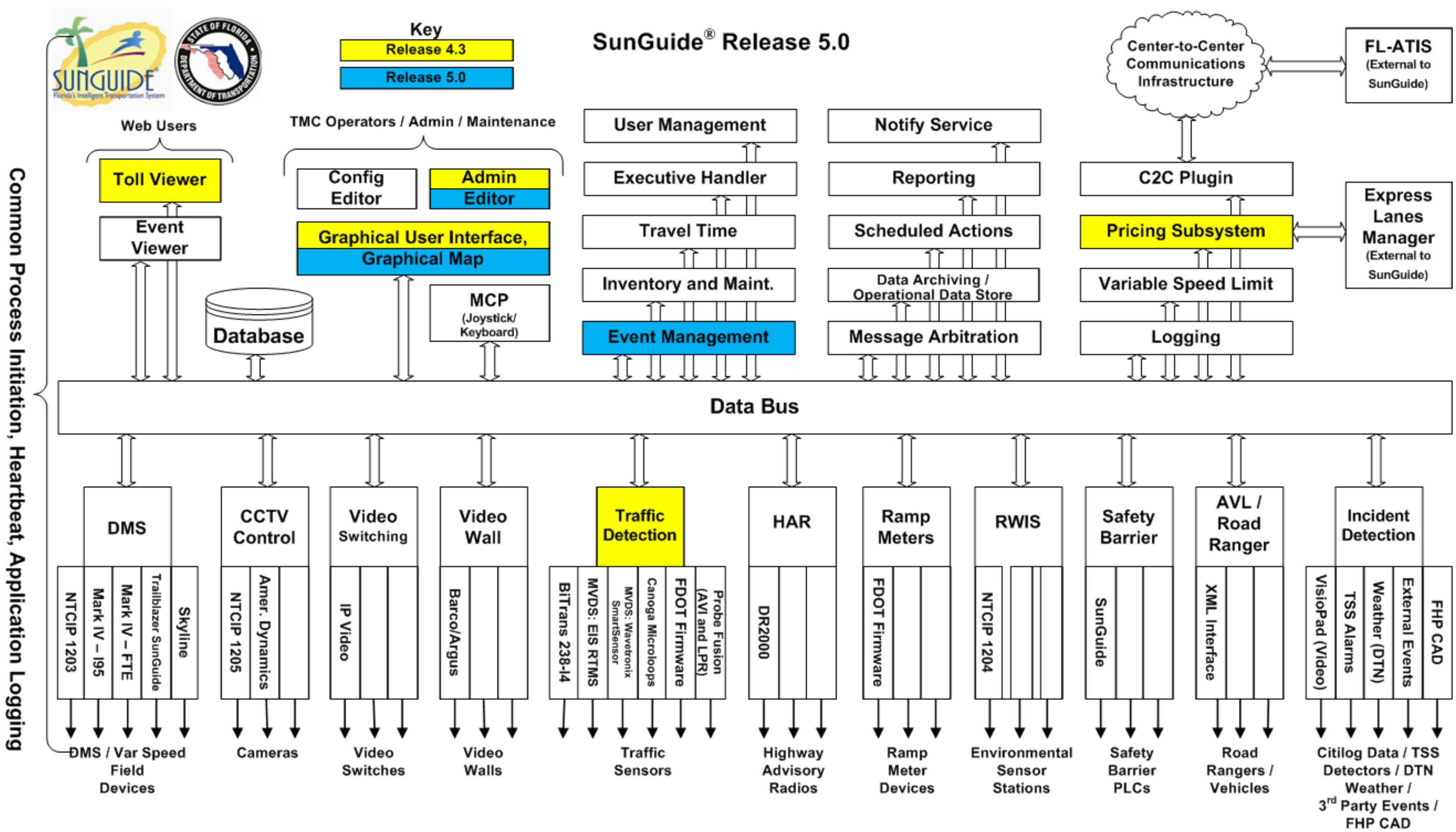


Figure 2-5 GraphicalView of Version 5.0 of the SunGuide Software

2.2.1. SunGuide Data Bus

It is important to note that the SunGuide architecture is not a “database-centric application" (i.e., an application that uses a database to exchange data between software processes). Rather, SunGuide utilizes an Extensible Markup Language Document Object Model (XML DOM), which is a tree-based application Data Bus, that stores the real-time data in the memory of the server running it. As shown in Figure 2-5, the SunGuide Data Bus subsystem provides both a real-time exchange of data between the subsystems and clients of the SunGuide, and a framework to which processes attach and exchange data with other subsystems. Depositing data to the bus is done in a structured, common format, and extracting data from the bus requires an appropriate privilege level. The primary advantage of this architecture is performance because Data Bus access is extremely fast, as opposed to having all the real-time data moving through a long-term store such as relational databases.

2.2.2. SunGuide Subsystems and Drivers

A subsystem in the SunGuide environment is a software process that implements a set of closely related functional requirements. A subsystem provides data to the Data Bus to make the data available to other SunGuide subsystems and interfaces. As shown in Figure 2-5, examples of these subsystems are DMS, CCTV Control, Video Switching, Video Wall, Traffic Detection, Highway Advisory Radio (HAR), Roadway Weather Information System (RWIS), Safety Barriers, AVL/Road Rangers, Incident Detection, Travel Time Estimation, Congestion Pricing, 511, Variable Speed Limit, Response Plan Generation, and the Center-to-Center plug-in. When processes log into subsystems (e.g., the Graphical User Interface logs into the DMS subsystem), they become “clients” of the subsystem to which they log on. As a client, they can subscribe for status data, which is typically retrieved from the Data Bus. They can also transmit command information if the subsystem supports this capability. These subsystems can be “clients” of other subsystems as well. The Data Bus process is the only process that connects directly to subsystems, as all other clients connect to subsystems through a connection to the Data Bus.

The SunGuide requires the development of drivers associated with the aforementioned subsystems. A driver is utilized to implement a vendor-specific protocol or an ITS standard protocol. Drivers do not communicate directly to the Data Bus; rather, each driver communicates to a subsystem. The details of the particular protocol being implemented are only important for the communication between the device and the driver. The subsystems do not communicate using these protocols, but communicate using standard XML messages. This approach allows a subsystem to be able to communicate with the devices of various vendors by developing a driver that communicates using the protocol of the vendor's device. The subsystem treats all devices in the same manner, independent of protocol. The SunGuide architecture utilizes XML Interface Control Documents (ICDs) to provide the subsystems with a user-friendly interface.

A generic framework referred to as the SunGuide Generic Subsystem was developed to allow base functionality to be provided. This generic subsystem allows subsystems and drivers to utilize the same module for the common functionality required of all processes. This includes the logging of messages to the software Status Logger, communicatng with the Executive Handler (EH), connecting to an Oracle database, and other common tasks.

As explained later in this document, the subsystem and driver concept is particularly important to this study. Emphasis is placed on this concept because the simulated traffic detectors utilize drivers already developed and associated with the traffic detection subsystem in the current version of the SunGuide.

2.2.3. Map, Interface, and Databases

The SunGuide system includes a mapping subsystem that displays primary and secondary roads along with the status information available from the Data Bus. The map connects to the Data Bus and is capable of displaying traffic conditions, incidents, lane closures, and device status in real-time. The displayed components are color-coded for the quick identification of troublesome spots within a graphical representation of the network. This representation also allows the operator to pan and zoom in to regions of interest that can be either expanded or shrunk depending on the operator's needs. Icons are displayed for field equipment locations, incidents, and lane closures, and an operator can access the field equipment information by selecting these icons. Finally, the GUI provides access to various subsystems (e.g., DMS, CCTV, incident management, etc.) using industry standard browser techniques.

2.2.4. SunGuide Archive

The SunGuide system maintains data in several different places for use in report generation. Aggregated operational data are stored in Oracle database files, while the raw data are stored in comma separated (CSV format). Three of the SunGuide archive files are of particular interest to microscopic traffic simulation: incident archive, detector data archive, and travel time archive files. These archives are produced in the CSV file format and contain 24 hours of data (midnight to midnight). A description of the three archived files is given below.

- ***Incident Archives**:* For each SunGuide incident record, the stored data includes timestamp, incident ID, operator, event details, and history of the event
- ***Detector Data Archives**:* The traffic conditions data are stored in TSS text flat files, with each file including data for a 24-hour day. The TSS file contains one record for each lane of detection stations, for each 20-second polling interval. Each TSS detector record includes timestamp (HH:MM:SS 24-hour format), detector identifier, speed, occupancy, and raw count data
- ***Travel Time Archives**:* These archives include records produced for each travel time link, with one record per time slice. Each record includes the timestamp, travel time link identifier, travel time (in minutes), and link status ("in service" or "failed") data.

2.2.5. Transportation Sensor System

The tool developed in this study will make use of results from virtual traffic detectors in the simulation. These will be interfaced to the SunGuide TSS Subsystem using the existing SunGuide device drivers. The SunGuide TSS acquires data from the detection field devices (speed, volume, and occupancy) and communicates information between these detector device drivers and the SunGuide Data Bus. The TSS subsystem currently supports a number of device drivers, including BiTrans B238-I4, Remote Traffic Microwave Sensor (RTMS) by Electronic Integrated Systems (EIS), Smart Sensors by Wavetronix, 3M Canoga Microloops, FDOT Firmware, and Probe Fusion (AVL and LPR).

For the TSS subsystem, detector mapping information and roadway geometry are retrieved from the back-end Oracle database at startup. The user can add and map additional detectors, update or remove existing detectors, modify the polling cycles, or update the roadway geometry information, etc. The TSS subsystem can then send an addDetectorReq message to each driver containing the specified detectors. After receiving a response from the driver, the mapDetectorReq message is sent to map the zone numbers of the detector to links and lanes. Once a detector has been mapped to links and lanes, threshold values (e.g., for speed and/or occupancy) may be set for the links/lanes. After receiving the request from the TSS subsystem, the driver begins polling the detectors with the polling cycle specified in the addDetectorReq message. For each poll, the driver will send a linkUpdateMsg to the TSS subsystem, which contains detector data updates on speed, volume, and occupancy.

2.3. STEWARD Data Warehouse

The Statewide Transportation Engineering Warehouse for Archived Regional Data (STEWARD) has been developed as a proof of concept prototype for the collection and use of ITS data (Courage and Lee, 2008). STEWARD archives data in a database that supports the generation of reports and queries. The development of this prototype has demonstrated that data from TMCs throughout Florida can be centrally archived in a practical manner, and that a variety of useful reports and other products can be created. The current effort has concentrated on archiving the SunGuide TSS and the Travel Time (TVT) information. These archives are produced based on the TSS and TVT archives described in Section 2.2.4.

Some of the useful functions of STEWARD as reported in the final report of the central data warehouse project (Courage and Lee, 2008) are as follows:

- Identifying detector malfunctions
- Providing calibration guidance for detectors
- Performing quality assessment data reliability tests
- Providing daily performance measures
- Facilitating periodic reporting requirements
- Providing data for research and special studies

The STEWARD database contains summaries of traffic volumes, speeds, occupancies, and travel times aggregated by 5-, 15-, and 60-minute periods, as specified by the user. Using a web-based interface, the user can specify date and time ranges and the detector locations for which the data is needed. The user may then download all generated reports in comma-delimited formats that can be easily imported into database management tools.

As discussed in the previous section, the SunGuide raw data archive contains one record for each lane at each station, for each 20-second polling interval. To streamline the data processing and retrieval, the STEWARD database contains one record per station representing all of the lanes at that station, accumulated over periods of 5, 15, and 60 minutes. Reports are generated from the TSS data at the detector, station, and system levels. With regard to the travel time database, the one-minute travel times for each link produced by SunGuide are aggregated and grouped into 5-, 15-, and 60-minute intervals. STEWARD also includes a quality assessment procedure to identify bad or suspicious data. Figure 2-6 shows a flow chart of the TSS data archiving process.

The following information is maintained in STEWARD for the TSS archive data:

- Station ID and Lane Number - Unique identifiers assigned by STEWARD
- Description - A physical description of the location
- Road - The name given to the facility
- Coordinates - Latitude and longitude
- Milepost - Required for sequential ordering of stations
- Lane Type - Freeway, entrance, or exit ramp
- Direction
- Detector Type - Loop, RTMS, etc.
- Maximum Speed - Normally the speed limit
- Count Station - The number assigned by the FDOT Statistics Office or District Planning Office for generating traffic count data files from the SunGuide detectors

The following information is maintained in a database for each travel time link in the travel time archive data:

- A description of the origin and destination

- Direction
- Distance weighted average speed limit (required to compute the travel time variability measures)
- Link Length

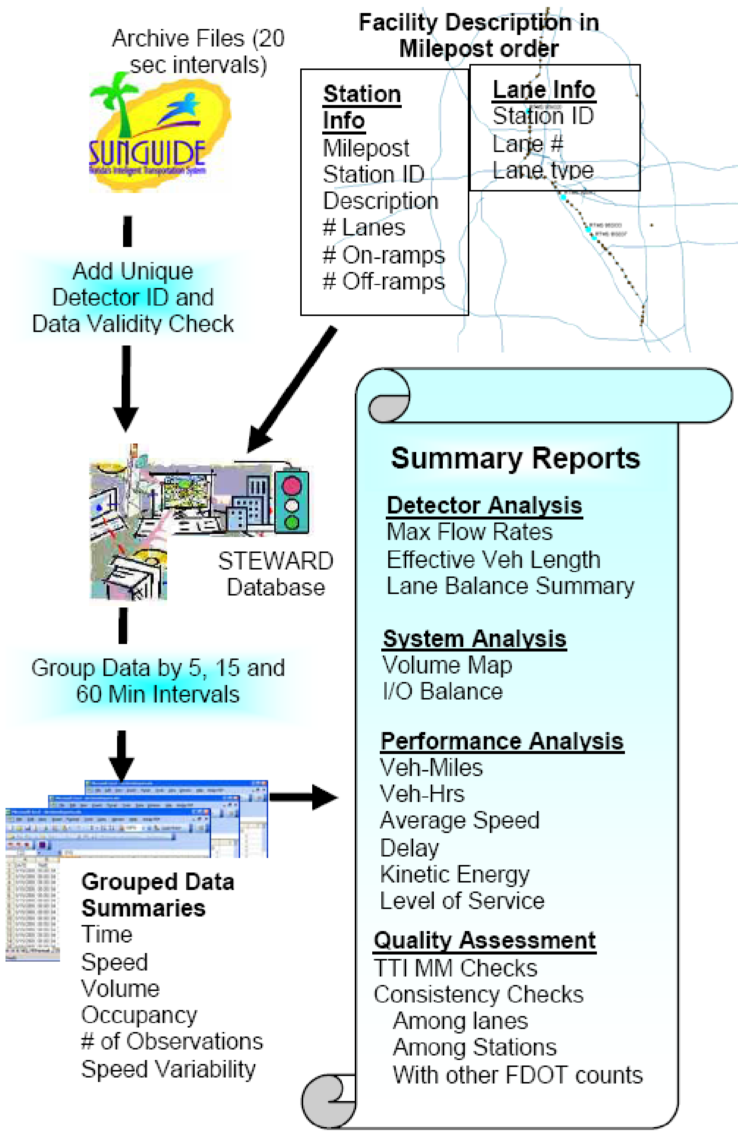


Figure 2-6 STEWARD Archives TSS Flow Chart

2.4. Current Simulator Use in SunGuide Testing

The testing of many SunGuide software requirements has been conducted at Southwest Research Institute (SwRI) during the Factory Acceptance Test (FAT) and at the FDOT Traffic Engineering Research Lab (TERL) utilizing software simulators of field devices. Below is a list of simulation programs that are (or have been) used in conjunction with SunGuide operations at TERL:

- Skyline DMS simulator
- RTMS simulator
- SwRI RWIS simulator
- SwRI Safety Barrier Simulator
- RTMS SimStarter (Auto-launcher for SwRI RTMS simulator)
- SwRI AVL/RR simulator

These simulators, however, are limited to generating a dummy stream of data and can simulate only one station at a time.

2.5. Summary and Analysis

Based on the discussion presented in Section 2.1, it appears that many applications exist in which simulation has been applied to model ITS, including incident management, smart work zones, traffic detectors, ramp metering, probe surveillance, travel time estimation and prediction, DMS messaging, and traveler information systems. Most of these applications, however, utilize simulation models that were developed and calibrated based on data collected using traditional data collection methods. A limited number of studies have used ITS data warehouse and/or operational databases for off-line simulation development and calibration. These models were then used in an off-line environment to generate ITS strategies to support TMC planning and operation activities. In recent years, a few locations in Europe and the United States have started exploring the use of simulation models for real-time applications at TMCs.

The SunGuide software has a flexible architecture that allows for the adding of devices, and emulation of these devices, using the interfaces between the drivers, subsystems, and the Data

Bus, as described in Section 2.2. This flexible architecture, combined with the powerful API capabilities of existing microscopic simulation tools, will make it possible to exchange data between the SunGuide software and virtual detectors in selected microsimulation tools for use in the SunGuide subsystem testing and operation evaluation, which is one of the objectives of this study.

As discussed in Section 2.3, the STEWARD data warehouse project has proved it is possible to centrally archive data from TMCs throughout Florida for use in a variety of applications. The data archive stored in STEWARD can form a foundation for the development and calibration of future simulation tools. In addition, simulation models can be used to emulate archived data for the testing of various archiving and data processing tasks.

In this study, two of the existing simulators listed in Section 2.4 were used as a basis for the development of the interface between microscopic simulation and SunGuide: the RTMS simulator and SwRI AVI/LPR simulator. Whereas the RTMS simulator is owned by FDOT, the SwRI AVI/LPR is owned by SwRI; thus, an agreement was signed between SwRI and the research team regarding the use of this tool.

3. Use of ITS Data to Support Simulation Model Development

3.1. Introduction

Traditionally, traffic simulation applications have been developed using volume data collected by tube and/or manual turning volume counts. These applications are calibrated using data from travel time studies combined with volume data and field observations of queues and other traffic conditions. The collection of the required data, however, is expensive, particularly for large simulated systems. In addition, the data collected using traditional methods are typically for only one or a few days, which may not represent the traffic demands and conditions throughout the year.

ITS agencies have used devices such as traffic detectors, closed circuit television cameras (CCTV), electronic toll readers, and license plate readers to collect traffic parameter measurements for operational purposes. In recent years, these agencies have started archiving the data collected by these devices (FHWA 2004). Because ITS detectors and communicators are already in place to collect data for operational purposes, the extra cost to archive and manage the data is relatively low. As ITS data archives become more widely available, the utilization of such archives for the development and calibration of simulation applications will in turn become an increasingly attractive option. This utilization will provide a significantly lower cost and a more efficient data collection method in comparison to traditional methods, and will increase safety by reducing the need for personnel to go out into the field for data collection purposes.

The additional details provided by the ITS data, both in time and space resolutions, will allow better representations of real-world environments in simulation applications. For example, the use of archived ITS data will allow the simulation of seasonal variations in traffic, special events, accidents, work zones, weather events, other types of incidents, and incident management strategies. This chapter discusses the development of procedures and tools for the utilization of data from the ITS data archives to support the development of simulation models. Chapter 4 discusses the use of ITS data for calibrating simulation models.

3.2. Functionalities Provided

This study has developed procedures and tools that use the STEWARD data to populate microsimulation application datasets with the required volume and speed data. The procedures and tools developed allow for the collection of data from the data archive, the manipulation and aggregation of the data, and the automatic modification of the input files to microscopic simulation tools. To achieve this end, the CORSIM microscopic simulation tool was used in this study. However, the tools and methods developed in this project can be easily extended for use with other simulation tools. In addition, the tool can be used in the estimation of traffic parameters based on ITS data archives for other planning, travel demand forecasting, and traffic analysis purposes.

The use of ITS data that covers a long period of time provides the opportunity to classify the days throughout the year into different patterns. For example, on certain corridors, it may be important to differentiate between different seasons or to simulate days with special events. In addition, it is necessary to exclude days with unusual demands or congestion when simulating typical day patterns. Thus, a procedure was developed to categorize the demand data for different days into patterns based on the similarity of travel demands as measured by the traffic detectors.

ITS data can include inconsistent, non-balanced, and missing measurements. A procedure was therefore developed to produce consistent and balanced traffic demands and to estimate missing traffic demands based on the demands measured. Another functionality provided was the automatic segmentation of the time period for each identified pattern into sub-periods of similar demands. Free-flow speeds, a required input to simulation tools, were also estimated based on the archived data. The details of the procedures developed in this study to implement the required functionalities are discussed in the following section.

3.3. Developed Tool

This section presents a discussion of the modules developed to deliver the functionalities identified in the previous section.

3.3.1. Data Quality Check

The STEWARD data warehouse has advanced data filtering and imputation procedures that ensure a relatively high data quality. However, it was determined that there are conditions in which fully automated corrections by STEWARD cannot be performed and thus require manual inputs. In this study, a module was included in the developed tool to identify whether these specific conditions exist and, if so, alert the analyst via a warning message. If the analyst agrees that there are problems with the data, these problems are then corrected. The specific conditions are as follows:

- When unreasonable free flow speeds may be estimated based on detector measurements; for example, a situation in which most locations on a corridor are estimated to have free flow speeds around 55 mph, except at one or two locations where the free flow speeds are above 70 mph.
- When the volume for some of the lanes is substantially lower than other lanes at the same detection station. This may or may not be correct depending on the actual conditions at the site. In some cases, it is due to detector malfunctions or a lack of appropriate calibrations.

3.3.2. Identification and Selection of Simulated Patterns

A module was developed to categorize the demand data for different days into patterns, based on the similarity of the time series of volume counts on different days. The *k*-means clustering algorithm (Alpaydin, 2004) was used for the categorization. The analyst can specify all or a subset of the detector measurement to be used in the categorization. This is an iterative partitioning algorithm that minimizes the sum of time series distances to cluster centroids, summed overall clusters (Alpaydin, 2004). In this study, the times series distance is measured by the Euclidian distance, defined as follows:

$$dist((v_j, c_k) = \sum_i (v_j(t_i) - c_k(t_i))^2 \qquad \forall j \in k \tag{3-1}$$

$$c_k(t_i) = \frac{1}{n_k} \sum_j v_j(t_i)^2, \forall j \in k \tag{3-2}$$

where

$v_j(t_i)$ = time series measurement *j* at time interval *i* from STEWARD,

$c_k(t_i)$ = centroid of cluster *k* at time interval *i*, and

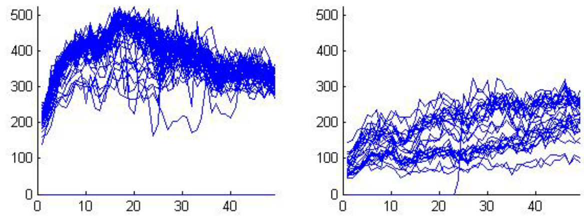
n_k = total number of time series in cluster *k.*

The optimization routine used in the clustering algorithm achieves a local optimal that can be varied each time the algorithm is run, depending on the starting point of the optimization. Thus, the analyst should run the algorithm for a number of replications to associate the measured daily demands with the clusters. The results presented in this paper are based on ten replications of the algorithm.

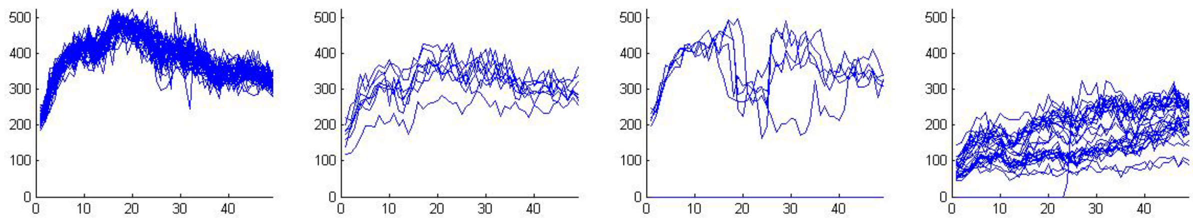
The procedure was applied using the built-in *k*-means function in MATLAB. However, it was desired to have the final product be a shared function that can be called by other applications; as such, this study utilized MATLAB's capability to convert the procedure to a .Dynamic Link Library (DLL) which can then be called from the .NET programming environment.

With the developed module, the analyst has the option of specifying the number of clusters that result from the analysis. Figure 3-1 shows the results of applying the data selection procedure to a set of 40 days using different numbers of clusters. The initial dataset contains weekdays; weekends; and days with incidents, bad weather, special events, and detector malfunctions. Of course, the more clusters that are used, the more homogeneous each cluster will be. However, too many clusters will not be useful, since the analyst's aim in most cases is to identify major differences in the patterns in order to be able to simulate a limited number of them. Figure 3-1 shows the results of the clustering when specifying two, four, and ten as the number of patterns to be obtained from the clustering procedure. As can be seen from Figure 3-1(a), specifying two patterns is not sufficient, since the algorithm basically classifies the days into a weekday and a weekend pattern. Figure 3-1(b) shows the results of requesting four patterns to be produced. The procedure was able to classify the patterns into two different weekday clusters. The first cluster from the left in Figure 3-1(b) represents higher demand weekdays compared to those days represented by the second pattern in the figure. The third pattern from the left represents incident days, and the fourth pattern represents weekends. Figure 3-1(c) shows the results of the analysis when ten patterns are specified. A visualization routine was also included in the developed tool to allow the analyst to associate each pattern with specific days. This allowed the determination of the reasons for the difference in the patterns, such as different seasons, different weather, special events, incidents with varying attributes, and so on. By examining the resulting patterns and the associated information, the analyst can determine which cluster to use in the analyses, what days should be excluded as outliers, and which clusters should be divided further into sub-clusters. For example, based on the data included in Figure 3-1, the analyst may decide to simulate two weekday patterns and one heavy weekend day pattern. In addition, the analyst may want to classify incident days further into different incident categories and use these days in calibrating simulation models for incident conditions. It is interesting to note that the second pattern from the left in Figure 3-1(c) does not have any detector measurements, as this pattern represents days with malfunction of the detection station at the given location.

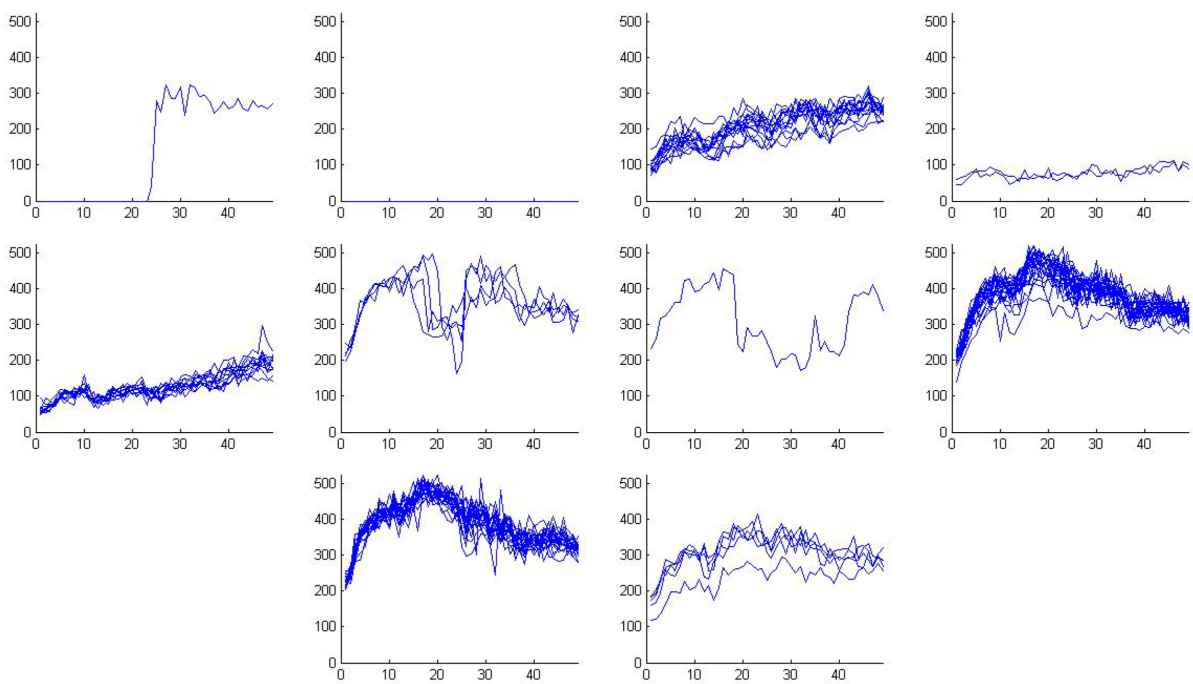
Another way of classifying days with different traffic patterns based on STEWARD data is to allow the analyst to specify thresholds for low, medium, and heavy demand days based on volumes. Alexis (2008), for example, classified measured traffic patterns into three categories based on demand thresholds, as part of the integrated corridor management (ICM) modeling effort.



(a) Two clusters. Vertical axis is traffic volume per five minutes and horizontal axis is time in minutes.



(b) Four clusters. Vertical axis is traffic volume per five minutes and horizontal axis is time in minutes.



(c) Ten clusters. Vertical axis is traffic volume per five minutes and horizontal axis is time in minutes.

Figure 3-1 Results of Clustering Using Different Number of Clusters

3.3.3. Time Period Segmentation

Microscopic simulation requires segmenting the day into discrete time intervals. Traditionally, analysts have divided the day into intervals that represent different peak periods during the day (e.g., AM, PM, and midday). These periods are then simulated separately. The analysts can also subdivide the peak period into subintervals to account for the variation in demands within the peak period. CORSIM and most other microscopic simulation tools allow coding sub-intervals to be ran in the same run. With more detailed data available from the ITS archives, it is useful to automate this segmentation of the time periods.

A procedure was developed in this study to segment the 24-hour or peak period volumes based on the measurements from all or a subset of the detection stations. The segmentation was done using an algorithm referred to as the Bottom-Up algorithm that has been used in data mining for linear piece-wise segmentation (Keogh et al., 2001). First, the Bottom-Up algorithm creates the finest possible approximation of the time series, so *n* segments are used to approximate the *n*-length time series. Next, the cost of merging each pair of adjacent segments is calculated, and the algorithm begins to iteratively merge the lowest cost pair until a stopping criteria is met.

The number of segments to represent the time series can be selected by the user. There is a trade-off between the number of segments and the complexity of the developed simulation application, thus it is desirable to select the lowest number of segments which capture the main temporal variations in demands.

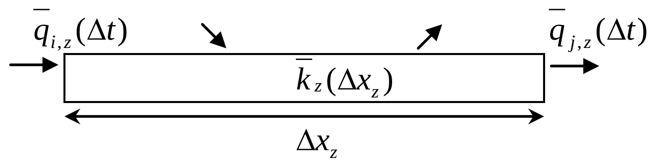
3.3.4. Spatial Conciliation and Estimation of Missing Demands

Although STEWARD implements a data filtering and imputation methods, it was found that inconsistencies between adjacent detector measurements still exist. In addition, in many cases, detectors are not placed on the ramps. A procedure was therefore developed to resolve inconsistencies and non-balanced traffic between upstream and downstream detectors, and to estimate missing link measurements based on other link measurements.

To illustrate this point, consider the following segment:

$$\overline{I}_{k,z}(\Delta t) \qquad \overline{O}_{l,z}(\Delta t)$$

where



Δt	= period of time,
Δx_z	= length of the section z,
$\bar{q}_{i,z}$	= average flow at location *i* in section *z* during Δt,
$\bar{q}_{j,z}$	= average flow at *j* in section *z* during Δt,
$\bar{k}_z$	= average density in section *z* during Δt,
$\bar{I}_{k,z}$	= average in-flow at ramp *k* in section *z* during Δt, and
$\bar{O}_{l,z}$	= average off-flow at ramp *l* in section *z* during Δt.

The conservation principal results in the following equations:

$$\bar{k}_z \Delta x_z = \left(\bar{q}_{i,z} - \bar{q}_{j,z} + \bar{I}_{k,z} - \bar{O}_{l,z}\right)\Delta t \tag{3-3}$$

$$\left(\bar{k}_z(t+\Delta t) - \bar{k}_z(t)\right)\Delta x_z = \left(\bar{q}_{i,z} - \bar{q}_{j,z} + \bar{I}_{k,z} - \bar{O}_{l,z}\right)\Delta t \tag{3-4}$$

With all variables as defined above, this equation is applied between every two consecutive detection stations. Further, we introduce in the formulation error terms to account for errors in detector measurements of volume and occupancy, resulting in the following formulation:

$$\left(\bar{k}_z(t+\Delta t) + \bar{\delta}_z(t+\Delta t) - \bar{k}_z(t) - \bar{\delta}_z(t)\right)\Delta x_z = \left(\left(\bar{q}_{i,z} + \bar{\varepsilon}_{i,z}\right) - \left(\bar{q}_{j,z} + \bar{\varepsilon}_{j,z}\right) + \left(\bar{I}_{k,z} + \bar{\varepsilon}_{k,z}\right) - \left(O_{l,z} + \bar{\varepsilon}_{l,z}\right)\right)\Delta t \tag{3-5}$$

where

$\bar{\varepsilon}_{x,z}$	= flow correction at location *x* in section *z*, and
$\bar{\delta}_z(X)$	= density (occupancy) in section *z* during period *X*.

$$\left(\bar{\varepsilon}_{i,z} - \bar{\varepsilon}_{j,z} + \bar{\varepsilon}_{k,z} - \bar{\varepsilon}_{l,z}\right)\Delta t - \left(\bar{\delta}_z(t+\Delta t) - \bar{\delta}_z(t)\right)\Delta x_z =$$

$$= -\left(\bar{q}_{i,z} - \bar{q}_{j,z} + \bar{I}_{k,z} - \bar{O}_{l,z}\right)\Delta t + \left(\bar{k}_z(t+\Delta t) - \bar{k}_z(t)\right)\Delta x_z \qquad (3\text{-}6)$$

$$\bar{k}_z(t+\Delta t) = \frac{\alpha \times Occ_z(t+\Delta t)}{L+C},\ \bar{k}_z(t) = \frac{\alpha \times Occ_z(t)}{L+C} \qquad (3\text{-}7)$$

where

$Occ_z(X)$ = average occupancy in section z during period X,

L = average vehicle length,

C = detector length, and

α = factor that converts $(L+C)$ into the number of vehicles per mile or kilometer (5280 or 1000 respectively).

However, the occupancy on a segment can be assumed to be the average of the occupancy at upstream and downstream detector locations, resulting in the equation:

$$Occ_z(t+\Delta t) = \frac{1}{2}\left(Occ_i(t+\Delta t) + Occ_j(t+\Delta t)\right),\ Occ_z(t) = \frac{1}{2}\left(Occ_i(t) + Occ_j(t)\right) \qquad (3\text{-}8)$$

where

$Occ_i(t)$ = average occupancy in detector i from section z at the beginning of Δt,

$Occ_j(t+\Delta t)$ = average occupancy in detector j from section z at the end of Δt,

$Occ_j(t)$ = average occupancy in detector i from section z at the beginning of Δt, and

$Occ_i(t+\Delta t)$ = average occupancy in detector j from section z at the end of Δt.

Finally, by substituting from Equation 8 into Equation 6, the following is derived:

$$\left(\bar{\varepsilon}_{i,z} - \bar{\varepsilon}_{j,z} + \bar{\varepsilon}_{k,z} - \bar{\varepsilon}_{l,z}\right)\Delta t - \left(\bar{\delta}_z(t+\Delta t) - \bar{\delta}_z(t)\right)\Delta x_z =$$

$$= -\left(\bar{q}_{i,z} - \bar{q}_{j,z} + \bar{I}_{k,z} - \bar{O}_{l,z}\right)\Delta t + \frac{\alpha}{2(L+C)}\left(\left(Occ_i(t+\Delta t) + Occ_j(t+\Delta t)\right) - \left(Occ_i(t) + Occ_j(t)\right)\right)\Delta x_z$$

(3-9)

where $\overline{q}_{i,z}, \overline{q}_{j,z}, \overline{I}_{k,z}, \overline{O}_{l,z}, Occ_i(t+\Delta t), Occ_j(t+\Delta t), Occ_i(t)$ and $Occ_j(t)$ can be obtained from the detector measurements in the ITS data warehouse.

For steady state conditions (where no queue occurs), the problem can be simplified assuming that the density in section *z* does not vary significantly during *Δt*, resulting in the following:

$$\left(\overline{\delta}_z(t+\Delta t) - \overline{\delta}_z(t)\right) = \frac{\alpha}{2(L+C)}\left(\left(Occ_i(t+\Delta t) + Occ_j(t+\Delta t)\right) - \left(Occ_i(t) + Occ_j(t)\right)\right) = 0 \qquad (3\text{-}10)$$

Thus, the conservation equation below is obtained:

$$\overline{\varepsilon}_{i,z} - \overline{\varepsilon}_{j,z} + \overline{\varepsilon}_{k,z} - \overline{\varepsilon}_{l,z} = -\overline{q}_{i,z} + \overline{q}_{j,z} - \overline{I}_{k,z} + \overline{O}_{l,z}\,, \forall z \qquad (3\text{-}11)$$

It is possible to formulate several optimization criteria to minimize the error values $\overline{\varepsilon}_{i,z}, \overline{\varepsilon}_{j,z}, \overline{\varepsilon}_{k,z}, \overline{\varepsilon}_{l,z}, \overline{\delta}_z(t+\Delta t)$ and $\overline{\delta}_z(t)$ (10). Three different formulations of quadratic error summation minimization and linear programming optimization were investigated. The first quadratic errors summation minimization minimizes the summation of the squares of all error corrections, subject to complying with all conservation equations of the system, and constraining all corrections to reasonable maximum and minimum pre-defined values. The second formulation is similar to the first formulation, but differs in that the error corrections are weighted by the original volumes. The third is a linear programming problem that minimizes the maximum correction. Testing revealed that the results from the first formulation were as good or better than results from the other two formulations, and were thus used for the rest of this study. This formulation is given below:

Minimization

$$\sum_{i,j,k,l}\varepsilon^2 + \sum_{z,t}\delta^2(t) \qquad \forall z \in \text{System} \quad (\Delta t \text{ given}) \qquad (3\text{-}12)$$

subject to

$$\left(\bar{\varepsilon}_{i,z} - \bar{\varepsilon}_{j,z} + \bar{\varepsilon}_{k,z} - \bar{\varepsilon}_{l,z}\right)\Delta t - \left(\bar{\delta}_z(t+\Delta t) - \bar{\delta}_z(t)\right)\Delta x_z =$$

$$= -\left(\bar{q}_{i,z} - \bar{q}_{j,z} + \bar{I}_{k,z} - \bar{O}_{l,z}\right)\Delta t + \frac{\alpha}{2(L+C)}\left(\left(Occ_i(t+\Delta t) + Occ_j(t+\Delta t)\right) - \left(Occ_i(t) + Occ_j(t)\right)\right)\Delta x_z \qquad \forall z$$

$$\varepsilon_{i,lower} \le \bar{\varepsilon}_{i,z} \le \varepsilon_{i,upper} \qquad \forall i, z$$

$$\varepsilon_{j,lower} \le \bar{\varepsilon}_{j,z} \le \varepsilon_{j,upper} \qquad \forall j, z$$

$$\varepsilon_{k,lower} \le \bar{\varepsilon}_{k,z} \le \varepsilon_{k,upper} \qquad \forall k, z$$

$$\varepsilon_{l,lower} \le \bar{\varepsilon}_{l,z} \le \varepsilon_{l,upper} \qquad \forall l, z$$

$$\delta_{z,lower} \le \bar{\delta}_z(t) \le \delta_{z,upper} \qquad \forall z, t$$

The above formulation was also extended to cases where additional information is available from other sources which may have levels of reliability differing from that of the ITS data. Such sources may include short term counts, previous corridor studies, or old tube counts. In these cases, the analysts will have the option of assigning weights to different information to account for their different levels of reliability.

During testing of the above model, it was determined that the results from the optimization should be examined to determine if there are large volume corrections due to measurements at one or two locations which are clearly not consistent with other measurements in the system. For these instances, it is advised to take the measurements at these locations out of the optimization model. This will be illustrated using the case study presented later in this document.

This study uses MATLAB to code and solve the optimization formulations of the problem. A built-in quadratic optimization function referred to as *quadprog* was used. The MATLAB code was compiled as a DLL application that can be called from the .NET modeling environment.

The correction of the inconsistencies and non-balanced volumes must account for recurrent bottleneck locations that prevent a portion of the demand from being served during a given time period. In these cases, the counts from the data archives may not actually represent the actual demands, but volumes constrained by downstream bottleneck throughputs. Thus, a procedure was developed to approximate the demand for the ramps and mainline locations affected by the bottlenecks during the constrained demand periods. The procedure first detects the presence of bottlenecks and the affected locations. In addition, it identifies the time period during which the demand is constrained (traffic queuing) and the time period during which the queue is dissipating. The sum of the volumes during these periods represents the total demands. The challenge is to distribute these demands among the sub-periods (e.g., 15-minute intervals) during the queuing and queue dissipation periods. One of three options is therefore given to the analyst: the assumption of linear increase and decrease in traffic flow (triangular pattern) during the period, as shown in Figure 3-2; inputting the distribution of demands as the proportions of demand for each sub-period during the queuing and queue dissipation periods; or allowing the model to automatically specify these proportions based on measurements at detector locations specified by the user.

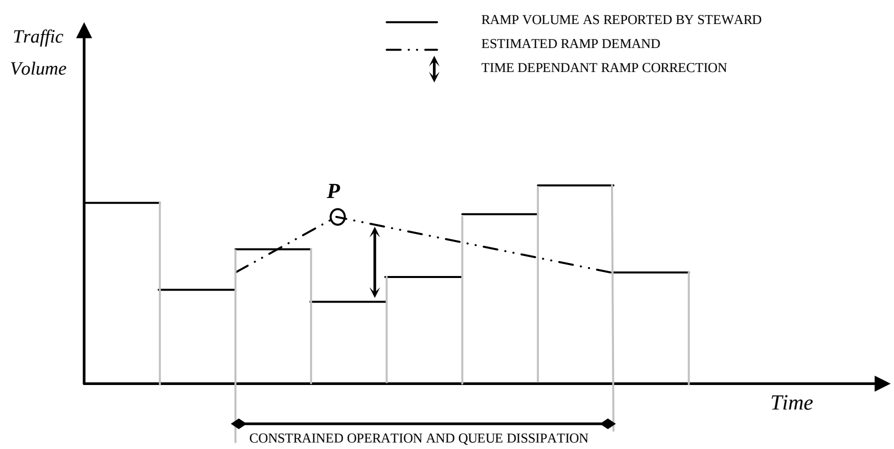


Figure 3-2 Ramp Volume Correction

3.3.5. Free Flow Speed Estimation

One of the important inputs to simulation tools is the free flow speed. The Highway Capacity Manual (HCM) 2000 (TRB, 2000) defines free-flow speed as the mean speed of passenger cars, measured during periods of low to moderate flow (up to 1,200 pc/h/ln). Thus, a module was developed to estimate the free flow speeds based on STEWARD data as the average speeds during low traffic conditions of less than 1,200 pc/hr/lane.

3.4. Case Study

The feasibility of using ITS data to develop microscopic simulation applications was tested using a freeway corridor equipped with ITS devices, including traffic detectors and CCTV cameras. The traffic corridor is the eastbound section of State Road (SR) 826, also known as the Palmetto Expressway, located in Miami, Florida. This corridor includes six interchanges and begins a quarter-mile west of the NW 67th Avenue interchange and ends a quarter-mile east of the NW 12th Avenue interchange, for a total length of 6.5 miles. This study focuses on the AM peak period between 5:00 and 10:00 AM and the PM peak period between 4:00 and 6:00 PM.

3.4.1. Data Collection

Volume, speed, and occupancy data were collected from the STEWARD ITS data warehouse. These parameters were measured by true presence microwave detectors located at 0.3- to 0.5-mile intervals on the test section. In total, there are 21 detectors on the eastbound direction of SR-826. The data were downloaded at the five-minute aggregation level.

3.4.2. Pattern Selection

This section presents a comparison between four options to demonstrate the use of the pattern selection procedure. Figure 3-3 shows a comparison between four cases for two consecutive hours during the PM peaks (4:00 PM to 6:00 PM). The first case uses the average of three consecutive day volume measurements collected at random from STEWARD. In the second case, the average of only two of these three days was used to exclude one day which, based on manual inspection of the data, seemed to involve incident conditions. The other two options utilize the pattern selection procedure with 22 days and 44 days, respectively. For these two options, the pattern selection procedure identified 16 days and 33 days, respectively, to belong to a typical recurrent traffic pattern cluster. This indicates that for this corridor, the volumes for 30% of the days are non-typical and should not be included when estimating the average typical demands on the corridor. It can thus be seen that using a bigger sample provides a more stable estimation of the average volumes. This stability helps in acquiring better data for the other procedures of this study.

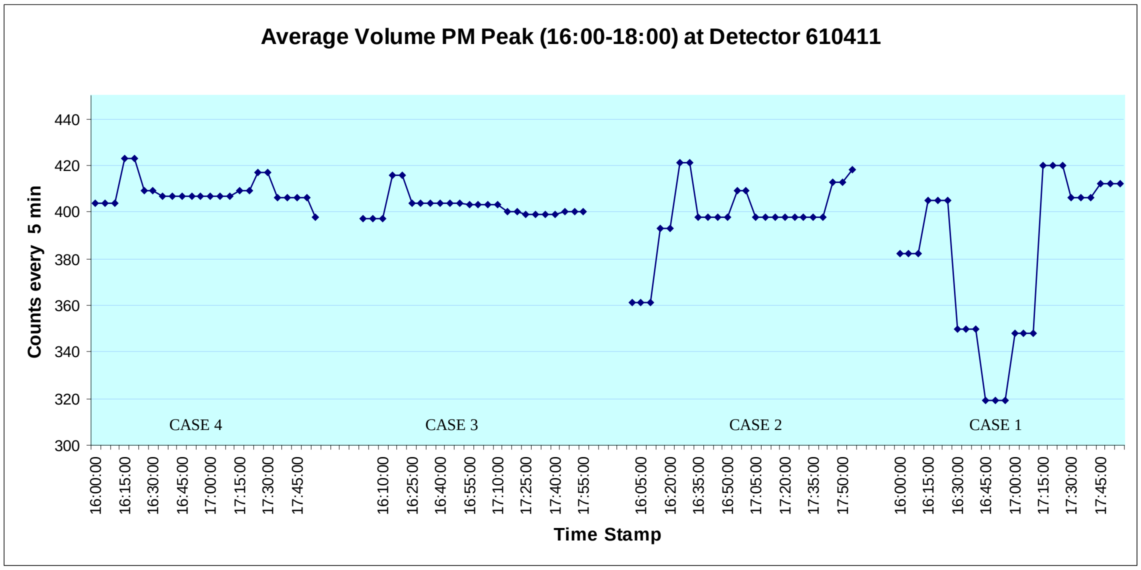


Figure 3-3 Average Volumes based on Different Cases

3.4.3. Period Segmentation

The five consecutive hours during the AM peaks (5:00 AM to 10:00 AM) were segmented using the period segmentation procedure mentioned earlier in this paper. Figure 3-4 shows a comparison using four, six, eight, twelve, and twenty segments in the segmentation procedure. Based on the results, the analyst can select the appropriate period segmentation for the scope of the analysis desired.

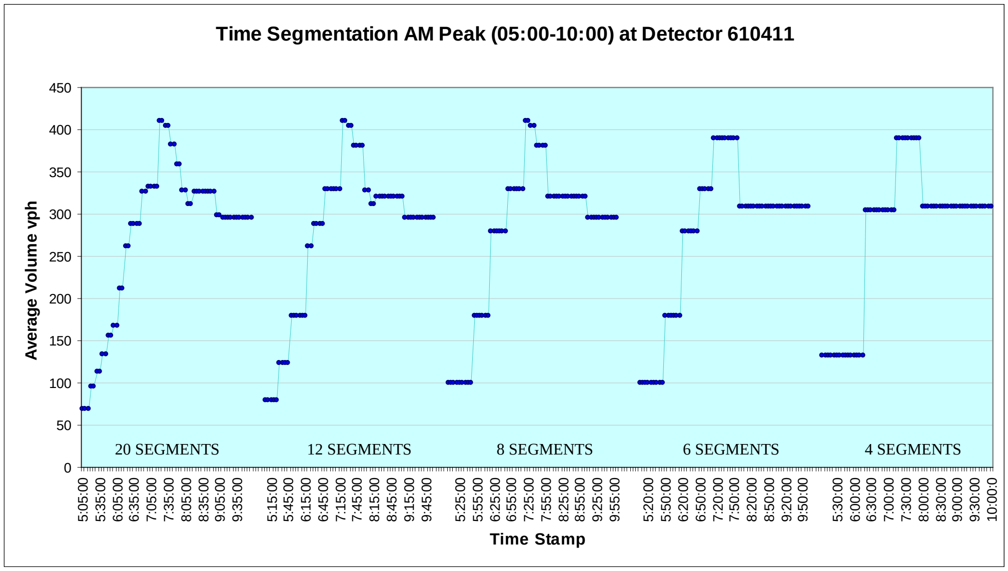


Figure 3-4 Time Segmentation based on Volumes

It is interesting to quantify the difference in performance between the segmentation of a time series using the segmentation algorithm developed in this study versus a baseline segmentation, in which the length of each subinterval is fixed at 15 minutes and the developed procedure is not used. This performance was assessed using data for the two PM peak hours from one detector station on SR-826. Eight, six, and four segments were requested for this data using the segmentation algorithm. The algorithm produced variable periods ranging from 10 to 40 minutes.

As a measure of the quality of the segmentation, the sum square error between the segmented series and the original STEWARD data is calculated as follows:

$$error(v(t_i), r(t_i)) = \sum_i (v(t_i) - r(t_i))^2 \tag{3-13}$$

where

$v(t_i)$ = time series volume value at time interval *i* from STEWARD, and

$r(t_i)$ = average volume value for the resulting time segment which represents the volume of the segment that covers interval *i*.

Figure 3-5 shows that using a higher number of periods improves the quality of the segmentation. Requesting eight segments in the segmentation algorithm produced a significantly lower error as compared to the other numbers of segments. It is interesting to see that using the four segments produced by the developed segmentation procedure was able to achieve the same error as that obtained by using eight consecutive 15-minute intervals without the segmentation procedure. The algorithm with eight segments produced a 34% reduction in the error compared to using eight segments with a consecutive 15-minute interval.

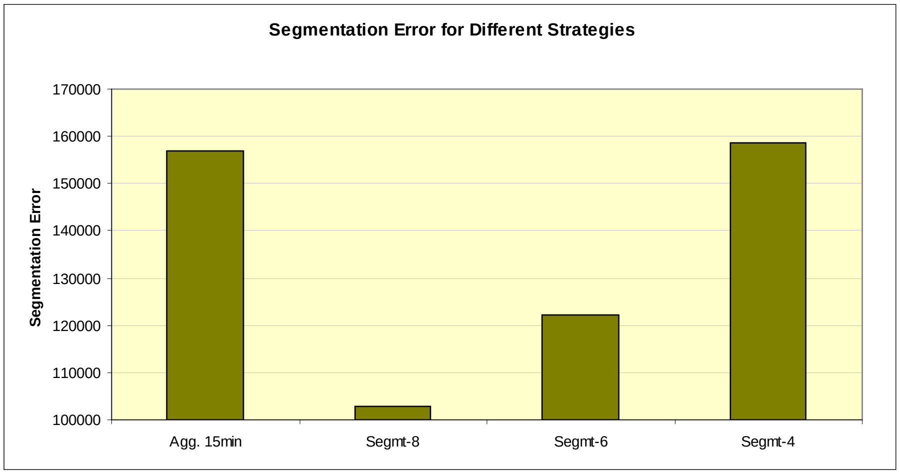


Figure 3-5 Square Error Comparisons for Different Approximation Strategies

3.4.4. Spatial Conciliation and Missing Demand Estimation

The spatial conciliation procedure was used to correct inconsistencies between traffic detector measurements and to estimate missing volumes. On SR-826, traffic detectors are available at 0.3- to 0.5-mile intervals on the mainline. However, there are a number of ramps that do not have detectors. At other ramps, the locations of the detectors do not allow accurate measurements of the volumes. Fortunately, the detection stations on the mainline were located such that each ramp volume can be calculated as the difference between the volumes of the upstream and downstream links. The following cases were thus compared:

- In Case 1, volume measurements from both mainline and ramp locations (where available) were used
- In Case 2, ramp count measurements were not used. Rather, these measurements were calculated based on upstream and downstream locations
- Case 3 is an extension of Case 2, wherein mainline detector station 610011 was removed from the optimization model, as discussed later in this section.

Sensitivity analysis showed that data collected for only a few days (e.g., three days) exhibits inconsistencies between detector locations and required substantial corrections. Using longer periods of time (data from 30 and 60 days) reduced the inconstancies significantly and produced better results. As such, 60-day data was used in this study.

Figure 3-6 shows the results of using only mainline detectors results as in Case 2. It appears that the correction of the volumes on the mainline is less than 12% in most instances, and that the majority of the corrections occurred in the second half of the corridor. Most of the corrections for the ramps occur for the last four on-ramps. The volumes for these ramps were reduced significantly during the correction process. Further examination indicates that this is due to the lower than expected volume measurements at the last mainline detector location. For this reason, the spatial conciliation algorithm decreased the volumes on the on-ramps to reduce the total upstream arrivals at this location. It also increased the volume significantly at this last detector.

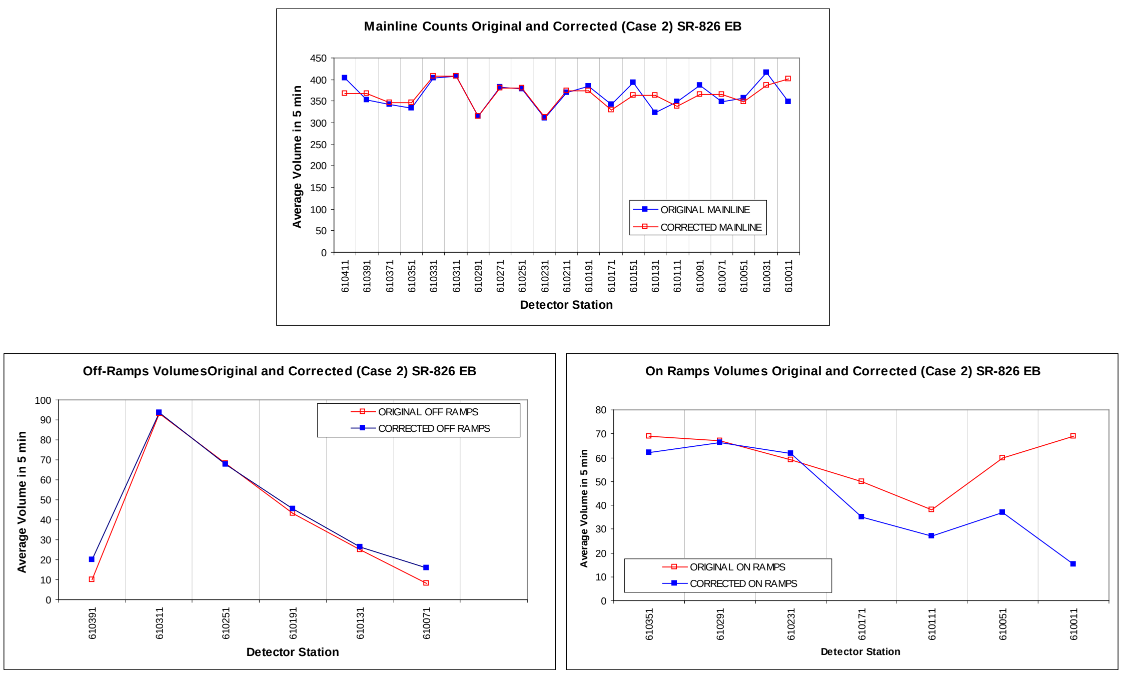


Figure 3-6 Mainline and Ramp Volume Corrections for Case 2

Figure 3-7 shows the results of Case 3, in which this last detector station was removed from the optimization due to significant inconsistencies with the rest of the system. It is noteworthy to observe how removing this detector station reduces the correction needed in adjacent stations. Manual counts for a short period of time confirmed that the last detector had data quality problems. The above results indicate that it is useful for the analyst to examine the results and revise the inputs to the optimization process, if the results from the optimization show significant corrections due to one or two suspicious detection station measurements.

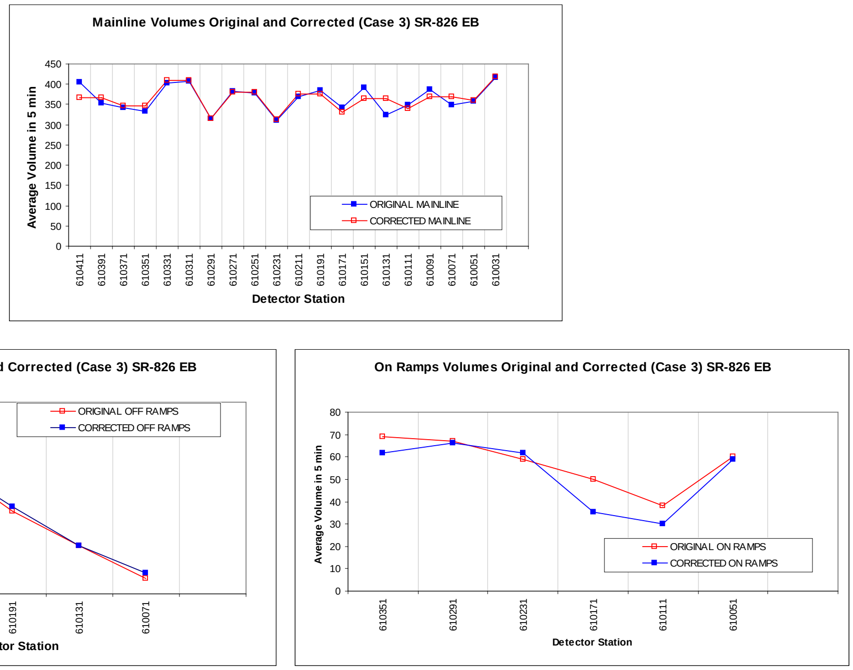


Figure 3-7 Mainline and Ramp Volume Corrections for Case 3

The results presented above also provide additional insights regarding the impacts of data quality reported by the data warehouse. If significant errors remain in the data, the quality of the results will be affected.

3.5. Conclusions

ITS data archives have the potential to provide cost-effective and detailed information for the development and calibration of simulation modeling applications. To achieve this goal, detectors in the field should be well-calibrated and maintained to reduce inconsistencies in the measurements. More importantly, advanced data maintenance, aggregation, filtering, and imputation capabilities should be provided to these ITS data archives to ensure data quality and the optimal use of archived data.

This chapter has illustrated the development and application of a series of data manipulation procedures for the utilization of ITS data archives to support simulation modeling. The procedures allow for the extraction of volume data collected from ITS data archives, automatic identification of temporal patterns in the data, automatic segmentation of daily demands into dynamically captured sub-periods to best fit the variations in the demands, resolution of possible spatial inconsistencies in the data, and the estimation of missing volumes. The developed procedures have been implemented as an automated tool for simulation model generation. The tool provides a graphic interface for users to download data from the STEWARD data warehouse, identify and select ideal traffic patterns, perform segmentation on traffic demands, conduct spatial conciliation to reconcile data inconsistency and estimate missing volumes, and generate new simulation model files based on the purified data. Although the main objective of the developed procedures and the tool is to produce data for microscopic simulation applications, they can also be used to support other applications, such as macroscopic, mesoscopic, and demand forecasting modeling applications. The details of the developed tool and associated documentations are discussed in Chapter 5 and Appendices A through C.

4. Use of ITS Data to Support Simulation Model Calibration

4.1. Introduction

A large number of studies have investigated methods to calibrate simulation models for recurrent conditions. For example, Volume 3 of the Traffic Analysis Toolbox series produced by the Federal Highway Administration (FHWA 2004) includes guidelines for calibrating traffic microsimulation modeling tools, while Volume 4 of that series (FHWA, 2007) presents guidelines that are specific to the calibration of the microscopic simulation tools for CORSIM. However, few studies discussed the use of ITS data to calibrate these simulation models. In addition, very limited research has been done to identify methods to validate and calibrate simulation models for incident conditions.

Traffic incidents have significant impacts on the performance of transportation systems. These incidents can reduce roadway capacity and result in excessive queues and delays. Advanced Transportation Management Systems (ATMS) have been applied successfully to minimize the negative impacts of incidents on traffic operations. However, to improve the effectiveness of ATMS, there is a need to assess how different types of incidents affect traffic operations, and the impacts of various incident management strategies and technologies on system performance. Many of the existing studies used for such assessments have utilized analytical queuing and shock wave analysis methods. Hadi et al. (2010) have also utilized the freeway facility analysis procedures in the Highway Capacity Manual (HCM) (TRB, 2000) for this purpose. This procedure uses shock wave analysis combined with other HCM procedures to assess the impacts of queuing, such as for cases in which there is a drop in capacity due to incidents.

Microscopic traffic simulation modeling has been proposed as a more detailed, flexible, and potentially more accurate, approach for assessing the impacts of incidents and management strategies. A microscopic simulation tool can model individual vehicles on a roadway network, typically on a second-by-second, or fraction of a second, basis. Microscopic simulation models have the benefit of being able to model complex roadway geometries, traffic control devices, integrated multi-facility operations, variations in vehicle characteristics, and variations in driver behaviors. These simulation model capabilities are beyond those of the existing analytical methods. However, modeling microscopic driver behaviors is difficult under normal traffic conditions and even more difficult under incident conditions. Limited research has been conducted to understand how incidents impact microscopic driver behaviors, such as lane changing and vehicle following, which are critical to the accuracy of microscopic traffic simulation modeling. These microscopic behaviors directly impact the macroscopic traffic performance measures, such as throughput, density, speed, and queue length, as assessed by the simulation models.

As described earlier, FDOT has used traffic detectors to collect measurements of traffic flow parameters for operational purposes. The utilization of these archives for the development and calibration of simulation models for both incident and no-incident conditions is therefore a logical option. This utilization can significantly lower cost and facilitate a more effective development and calibration, as compared to the utilization of data collected using traditional methods. The additional details provided by the ITS data will also allow for better representations of real-world environments in simulation applications. This chapter discusses the utilization of ITS data archives to support the use of simulation to calibrate simulation models for incident and no-incident conditions. The ITS data archives utilized include both incident management and point traffic detector data.

4.2. Previous Studies on Incident Impacts

Traffic incidents can introduce two types of changes that can impact traffic operations; changes in the roadway environment and changes in driver behaviors. Examples of changes in the roadway environment include capacity reductions due to lane and/or shoulder closures and changes to traffic control strategies, such as ramp metering, incident site control, and dynamic message sign (DMS) activations. On the other hand, changes in driver behaviors include changes in the microscopic (tactical and operational) driving behaviors, such as lane-changing, car-following, speed, gap acceptance, and accelerating behaviors. In addition, incidents can modify strategic driving behaviors such as changes to trip route, mode, and time choices.

There are a limited number of studies that have investigated the impacts of traffic incidents. These studies focused mostly on the impacts to macroscopic traffic measures, such as the reduction in capacity, rather than on microscopic traffic parameters such as lane-changing and car-following behaviors. The reason for this focus is that the impacts on macroscopic measures can be more easily assessed by utilizing the current data collection technologies. This section includes a review of the studies available on the subject.

4.2.1. Incident Impacts on Roadway Environment

The studies of incident impacts on the roadway environment have focused mostly on the reductions in capacity due to incidents. These capacity reductions were studied by Goolsby (1971), after which he concluded that an incident blocking one lane out of three lanes reduces capacity by about 50%. He also concluded that an incident blocking two lanes out of three lanes reduces the capacity by about 79%. The HCM 2000 (TRB, 2000) provides estimates of the remaining capacity during incident conditions as a function of the number of the blocked lanes (or shoulder) and the number of lanes of the highway section under consideration. The HCM estimates have been widely used in studies that investigated the effects of incident management strategies on system performance.

A study by Qi and Smith (2001) found, based on data collected in Hampton Roads, Virginia, that the capacity reduction with one lane blocked out of three lanes can be modeled as a Beta distribution with an average of 63% and a standard deviation of 14%. The study also found that the capacity reduction due to two blocked out of three lanes can be modeled as a Beta distribution with an average of 77% and a standard deviation of 12%.

Knopp et al. (2009) found that in the case of a blocked driving lane, the queue discharge rate for each available lane is reduced by 50%. They also found that the queue discharge rate is reduced by 30% when the driving lanes are open but there is an incident on the shoulder or on the opposite direction of travel.

Hadi et al. (2007) adjusted the parameters of three widely used microscopic simulation tools to determine their abilities to replicate the reported reductions in capacities due to traffic incidents. From the results, they concluded that it was possible to fine tune the parameters of the three simulation tools to simulate the drops in capacities due to incident lane blockages.

4.2.2. Incident Impacts on Microscopic Driver Behavior

One existing study investigated the impacts of incidents on the microscopic behaviors of drivers. The study collected empirical trajectory data using a digital camera mounted under a helicopter (Knopp, 2009). The analysis showed considerably different behaviors between driving under normal conditions and driving at the incident sites. It was found that at incident sites, drivers choose a longer headway, have a higher reaction time, and reduce speed. These behavioral changes lead to queue discharge rates during incident conditions, which are from 60% to 75% of the normal queue discharge rates per lane, according to the study.

4.2.3. Incident Impacts on Strategic Driver Behavior

The study of incident impacts on driver behaviors has focused largely on changes at the strategic behavior level, and particularly on changes in route choice behavior. Several researchers have used the stated preference approach to determine the percentage of travelers changing trip decisions in response to information disseminated by Advanced Traffic Information Systems (ATIS). The studies concluded that, based on these types of surveys, the disseminated information can result in up to 60% to 70% of the freeway traffic exiting the freeway ahead of an incident location (Barfield et al., 1989; Benson, 1995; Madanat et al., 1995; and Chatterjee et al., 2002). However, limited information is available about the actual diversion due to traveler information, as reflected by revealed preference or field measurements. Several European field studies have found that DMS compliance rates range between 27% to 44% (Tarry and Graham, 1995). Furthermore, Knopp et al. (2009) found that for major incidents up to 50% of travelers take another route.

Luk and Yang (2003) developed a simulation modeling framework to assess the performance of ATIS under different conditions. They assumed the average diversion rate to be 15% and the highest diversion rate to be 30%. In another study, Cragg and Demetsky (1995) used the CORSIM microscopic simulation tool to analyze route diversion strategies from freeways to arterial roads. The study concluded that there was often an optimal diversion percentage beyond which the system delays increased. This diversion percentage is expected to be different for different systems, depending on both traffic and incident conditions and the original and alternative routes.

4.3. Methodology

The availability of detailed incident and traffic detector data from ITS data archives allows the identification of important parameters required to model incidents in simulation models. These parameters include traffic demands during incident conditions, diversion rates, capacity reductions due to incidents, and the resulting impacts on various performance measures. This section describes the estimation and utilization of these parameters for simulating incident conditions using the CORSIM microscopic simulation tool. Although CORSIM is used here, similar procedures can be used for other simulation tools.

Simulating incident impacts on traffic requires the following steps:

1. Estimation of traffic demands during no-incident and incident days;
2. Estimation of capacity during incident and no-incident conditions;
3. Fine-tuning of simulation model parameters with the objective of producing, in the simulation, the capacities estimated for incident and no-incident conditions; and
4. Fine-tuning of simulation model parameters to produce the observed performance measures, based on traffic detector data.

These steps are discussed in the next section.

4.3.1. Estimation of Traffic Demand

The first challenge in setting a simulation model for incident conditions is to estimate the traffic demands during these conditions. In Chapter 3, this research developed procedures to estimate the demands in a system based on archived ITS detector data. However, during incident conditions, the volumes measured by the ITS system are severely constrained by the capacity remaining at the incident location. Thus, in many cases, the queues extend well beyond the simulated system boundary, preventing the direct estimation of the true demands for each interval based on the detector measurements. A procedure was therefore developed in this study to estimate the traffic demands based on volume measurements from different days, combined with the incident day volume measurements.

First, the *k*-means clustering algorithm, discussed in Chapter 3 was used to identify "typical" days that are expected to have traffic demand patterns similar to the incident day if there would have been no incident in the system. The algorithm utilizes the time-variant detector measurements at each detection station to classify the days into groups with similar traffic patterns. By examining these patterns, the analyst can clearly identify the typical day pattern.

The average demands for the typical days were then calculated and used as the initial estimates of the demands during the incident day. These average demands were subsequently adjusted to account for the expected difference between the average demands of the typical days and the demands during the incident day. The demands during the incident day are expected to be different from the average demands due to two reasons: the stochastic variations in demands between days and the traffic diversions due to incident conditions. As stated above, due to the constrained traffic conditions, the true demands during incident conditions are difficult to estimate for short time intervals based on traffic detector measurements. However, it is possible to calculate the differences in the cumulative traffic volumes between the average day and the incident day. These differences in volumes can be used to estimate the differences in demands between the average and the incident day, as long as the accumulations of volumes compared include all of the demands before the queue formation, during the queue, and after the queue discharge, as shown in Figure 4-1. The accumulated difference in demand can be distributed uniformly among the time intervals impacted by the incident.

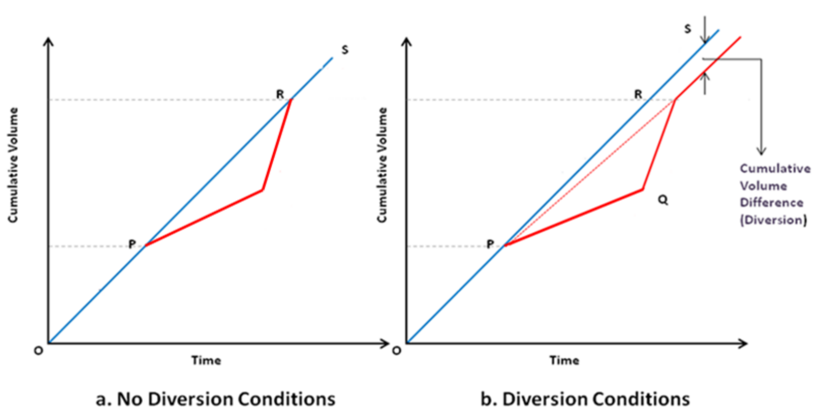


Figure 4-1 Estimation of the Difference between the Demands during Average and Incident Days

The above analysis should be repeated at all detector locations since the diversion rates at each location are expected to differ depending on the locations of the diversion points for the particular incident under consideration. This will allow the determination of the mainline, on-ramp, and off-ramp demands for the incident day at different corridor locations, based on the differences calculated from the average demands of the typical days.

4.3.2. Estimation of Capacity

The next step is to estimate the capacity during incident and no-incident conditions for each highway segment. For no-incident conditions, this could be achieved at bottleneck locations by plotting the fundamental diagrams, based on ITS data, at various detector stations and determining the maximum traffic flow from these diagrams. However, if the simulated segments do not exhibit recurrent congestion, the measurements on the congested side of the fundamental diagram will be insufficient to allow for the plotting of the full diagram. In these cases, the capacity for no-incident conditions can be estimated using the HCM procedures (TRB, 2000).

The drop in capacity due to the incident can be estimated using values reported in the HCM (2000) or in the literature, as reviewed earlier in this chapter. As previously stated, the HCM estimates the remaining capacity during incident conditions as a function of the number of blocked lanes (or shoulder) and the number of lanes of the highway segment under consideration. However, if detector measurements are available, the remaining capacity during incident conditions can also be estimated based on volume measurements at a detection station downstream of the incident location. This will be discussed in more detail later in this chapter.

4.3.3. Adjusting Simulation Parameters for Capacity Assessment

The next step is to fine-tune a number of calibration parameters for the selected simulation tool to achieve the measured or expected capacities during incident and no-incident conditions. Based on past experiences with simulation studies, Volume 4 of the FHWA Traffic Analysis Toolbox (2007) identified a few CORSIM input parameters for adjustment when calibrating freeway capacity for no-incident conditions. These parameters include the car-following sensitivity factor and multiplier, lag acceleration and deceleration time, and Pitt car-following constant.

CORSIM has a set of incident-specific parameters that can be used for modeling incidents. However, unlike the capacity for no-incident conditions, there are limited examples with which to guide the analyst when selecting appropriate values for these parameters. The incident-specific parameters in CORSIM include lane blockage details (number, location, and duration), rubberneck factor, incident length, and incident warning sign location. To model the capacity drops during incidents, CORSIM allows the analyst to specify what lane(s) are blocked, in addition to a rubberneck factor for each adjacent lane. Incidents in CORSIM can be specified at any longitudinal position on a freeway link, can extend over a user-specified length of the roadway, and have a start and end time.

CORSIM uses the rubberneck factor (as a percentage, the default value of which is zero) to increase the time headway between vehicles and reflect driver distraction while observing the incident. This, in turn, will result in a capacity reduction for those lanes adjacent to the blocked lanes or shoulder. CORSIM also allows the specification of the incident length, which is the length of the roadway affected by the incident. The CORSIM User’s Manual recommends estimating the incident length as the number of vehicles involved in the incident plus one vehicle length. For example, in a two-vehicle collision with an assumed vehicle length of 20 feet, the manual recommends coding a 60-foot incident length (FHWA, 2006). As discussed below, it was found that coding this length is not adequate to produce the expected drops in capacity due to incidents.

The impacts of varying the rubberneck factor and incident length parameters on the drop in capacity due to incidents was investigated by modeling a simple hypothetical system consisting of a single lane link. The incident length was varied from 60 ft. to 2,500 ft., with the rubberneck factor varied between 35% and 90%. Figure 4-2 shows that the effect of the rubberneck factor depends on the length coded for the incident. For short lengths, the effect of the rubberneck factor on reducing capacity is very small. In fact, for the 60-ft incident length recommended by the CORSIM User’s Manual discussed above, the resulting drop in capacity is zero for all rubberneck factors investigated. As stated earlier, CORSIM uses the specified rubberneck factor to increase the time headways between vehicles. If the specified incident length in the simulation is too short, this change cannot be fully achieved and thus the capacity will not drop to the expected levels. As can be seen in Figure 4-2, the full impact of the rubberneck factor on capacity drop is achieved at an incident length of 1,000 ft. The recommendation of the user manual that leads short incident length to be specified in the inputs should therefore be revised based on the above results.

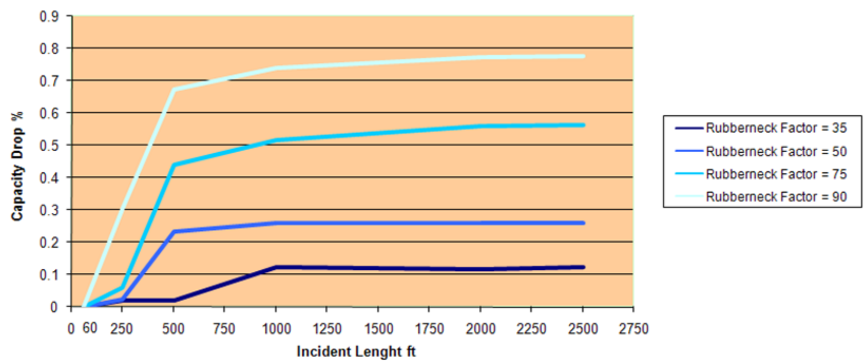


Figure 4-2 Effects of Incident Length and Rubberneck Factor on Capacity Drop

Another incident parameter in CORSIM is the warning sign upstream of the incident. This warning sign represents a location at which vehicles start changing lanes in response to the specified lane blockage. The default value for the warning sign location is 1,500 ft. ahead of the incident. In the investigation undertaken by this study, varying the warning sign location did not have a significant effect on the capacity drop. However, as discussed in the next section, the sign location was found to have significant effects on the *other* performance measures assessed when simulating incidents.

4.3.4. Parameter Adjustment based on Other Measures

In addition to the drops in capacity calibrated as described above, simulation models should be able to replicate the changes in performance measures such as throughputs, speed, occupancies, and queue measures at different locations of the system due to incidents. These measures can be estimated based on the data collected by detection stations. From the detailed investigation conducted in this study of a variety of input parameters to CORSIM, it was found that the parameters that affect the performance measures during incident conditions are the deceleration for non-emergency conditions parameter and the location of the incident warning sign parameter.

4.4. Case Study

A case study is presented in this section to explore the use of the processes discussed in the previous section to support the simulation of incident conditions based on ITS data. The case study includes the simulation of three incidents located on the eastbound section of the SR-826 limited access highway in Miami, FL. The simulated corridor includes six interchanges beginning west of the NW 67th Avenue interchange and ending east of the NW 12th Avenue interchange, for a total length of 6.5 miles. The simulation was conducted using the CORSIM microscopic simulation tool for weekday traffic conditions between 4:00 AM and 11:00 AM. Each scenario was run ten times using ten different random seed numbers, with the results reported in this study based on the averages from these runs.

First, the corridor was simulated for an average no-incident weekday. The simulation was then conducted for three different incident days. All selected incidents were one-lane blockage incidents located on the SR-826 eastbound (EB) at locations that are in close proximity to each other. The three incidents occurred in the morning peak period and were detected by the TMC between 7:39 AM and 7:50 AM. Figure 4-3 depicts the incident locations and nearby microwave detections stations on SR-826.

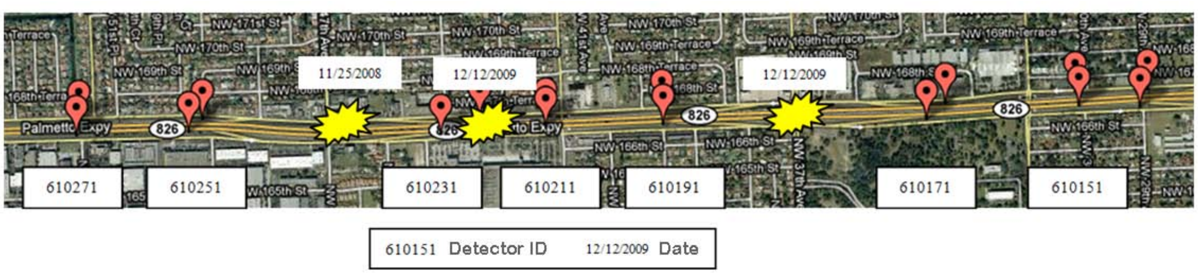


Figure 4-3 Locations of the Selected Incidents and Detector Stations

4.4.1. Estimation of Traffic Demand

As described in the Methodology section, the demands for the incident days were estimated based on the difference between the cumulative volumes during the incident days and the average demands of the no-incident days. It was found that the differences between the incident days and the average no-incident days at the incident locations were 4%, 22%, and 4% for

Incidents 1, 2, and 3, respectively. The lane blockage duration of Incident 2 was very long compared to Incidents 1 and 3. Thus, the estimated demands for the incident days show the expected trend of higher diversion rates with longer incident durations. The difference between incident and typical day demands was obtained for each detection station to allow the calculation of the mainline and ramp demands.

4.4.2. Estimation of No-Incident Capacity

During no-incident conditions, SR-826 EB does not have mainline bottlenecks that result in recurrent congested operations. Thus, it was not possible to estimate the segment capacity based on detector measurements. This study estimated the no-incident highway capacities of the corridor segments using the HCM procedures (TRB, 2000). The capacity for the three-lane segment on which the three incidents occurred was estimated to be about 2,200 vphpl (vehicle per hour per lane).

4.4.3. Estimation of Capacity Drop

Figure 4-4 shows the time series for throughput and speed measurements at the three incident locations based on data extracted from the STEWARD data warehouse. Region I in these figures is the region of lane blockage while region II is the congested period after lane clearance (the queue discharge period).

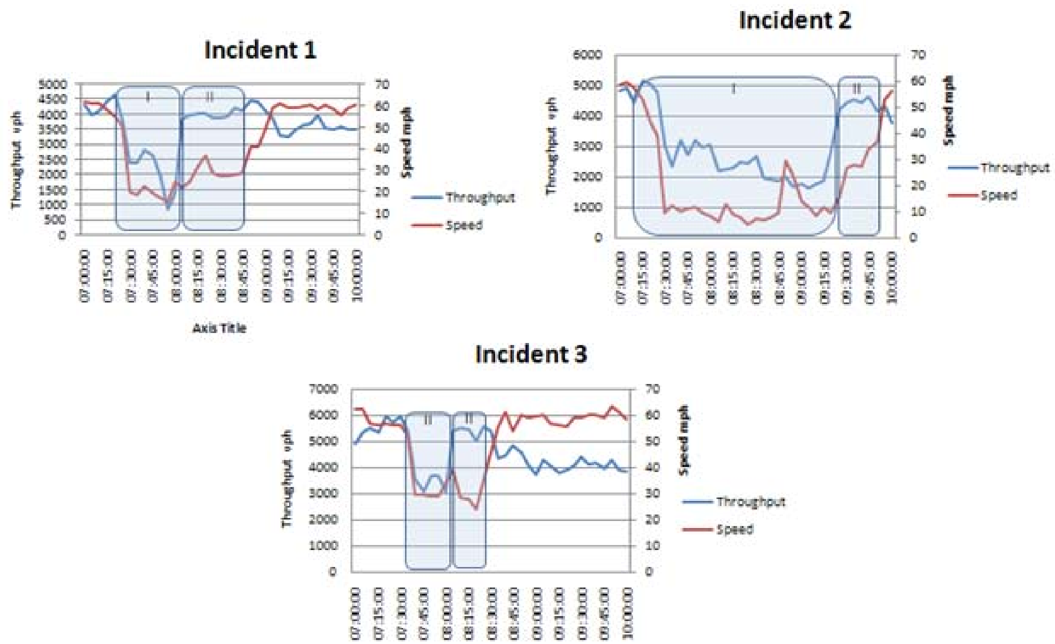


Figure 4-4 Throughput and Speed Measurements for the Three Investigated Incidents

It is interesting to note the following for the three investigated incidents, based on Figure 4-4:

- Just after the incidents occurred, the capacity dropped to between 2,600 and 3,000 vph, depending on the incident considered. This constituted a 54% to 60% drop in capacity, assuming a 6,600 vph no-incident capacity. The drop in capacity due to one-lane blockage incidents on three-lane segments is 51% according to HCM (TRB, 2000). Qi and Smith (2001) estimated the average drop in capacity to be 63%, as described earlier.
- During the lane blockage period for Incidents 1 and 2, additional significant drops in capacity occurred (down to 1,000 vph in the case of Incident 1 and 1,800 vph for Incident 2). Further examination of the incident management database indicated that these additional drops are due to the arrival of fire trucks that blocked additional lanes during incident management activities. The magnitudes of the drops are expected to be functions of incident attributes and the intensity of incident management activities at the incident sites. This dynamic nature of lane blockages during the incident management process may need to be modeled when simulating incident conditions.
- After lane clearance, the queue starts dissipating. However, the queue does not dissipate at 6,600 vph, which is the estimated capacity of the section, but at lower rates. The

queue discharge rates were 4,000, 4,500, and 5,400 vph, for Incidents 1, 2, and 3, respectively. These rates correspond to 39%, 32%, and 19% reductions in segment capacity. There are two reasons for not achieving the estimated capacity of the section during queue discharge. First, when the incident is moved to the shoulder, the presence of the involved vehicles and emergency vehicles on the shoulders still results in a drop in capacity due to rubberneck effects. The HCM (TRB, 2001) estimated that crashes with emergency vehicles on the shoulder experience a 17% drop in capacity on three-lane freeways. Furthermore, Knopp et al. (2009) found that the queue discharge rate is reduced by 30% due to shoulder incidents. Another factor that affects the maximum throughput during the queue discharge period is that the rate at which a queue is discharged is expected to be lower than the capacity, as defined by the HCM. The results of this study show that the queue discharge rate varies between 4,000 vph and 5,400 vph, indicating that this rate is a function of the intensity of emergency vehicle operations and the nature of the incident on the shoulder.

4.4.4. Adjusting Simulation Parameters for Assessment of Capacity

The next step is to adjust the simulation model parameters to ensure that the simulation model is able to produce the desired capacity during incident and no-incident conditions. For no-incident conditions, the lane capacity was reproduced in the simulation by adjusting the values of two parameters: the car-following sensitivity factors and the lag acceleration and deceleration times. As stated earlier, these parameters have been used in previous studies to adjust for the capacity assessed in the CORSIM applications (FHWA, 2007).

The next step was to fine-tune other parameters in the simulation model to allow the simulation to replicate the drops in capacity due to incident lane blockages, as observed in the real-world. To achieve the drops in the CORSIM simulation model, a search was performed to find the values of the rubberneck factor, incident length, and lane blockage that produce these capacity drop levels during the incident period.

Figure 4-5 shows the drop in throughputs in the real-world and the simulation model with the fine-tuned incident calibration parameters for the three incidents. It was found that CORSIM was able to reproduce the capacity reduction fairly accurately with a moderate level of effort. As stated earlier, there were three different throughput levels during Incidents 1 and 2, and two different throughput levels during Incident 3. It was possible to replicate these levels in the simulation, as shown in Figure 4-5, by coding time-variant rubberneck factors and an incident length of 1,000 ft. The values of the rubberneck factor that produced the observed drops in capacity due to one-lane blockage (a 54% to 60% drop in capacity, as discussed in the previous section) ranged from 70% to 80% for those lanes adjacent to the blocked lanes. For the queue discharge phase, the rubberneck factors required were 70%, 65%, and 50% to produce the observed drops in capacity for Incidents 1, 2, and 3, respectively.

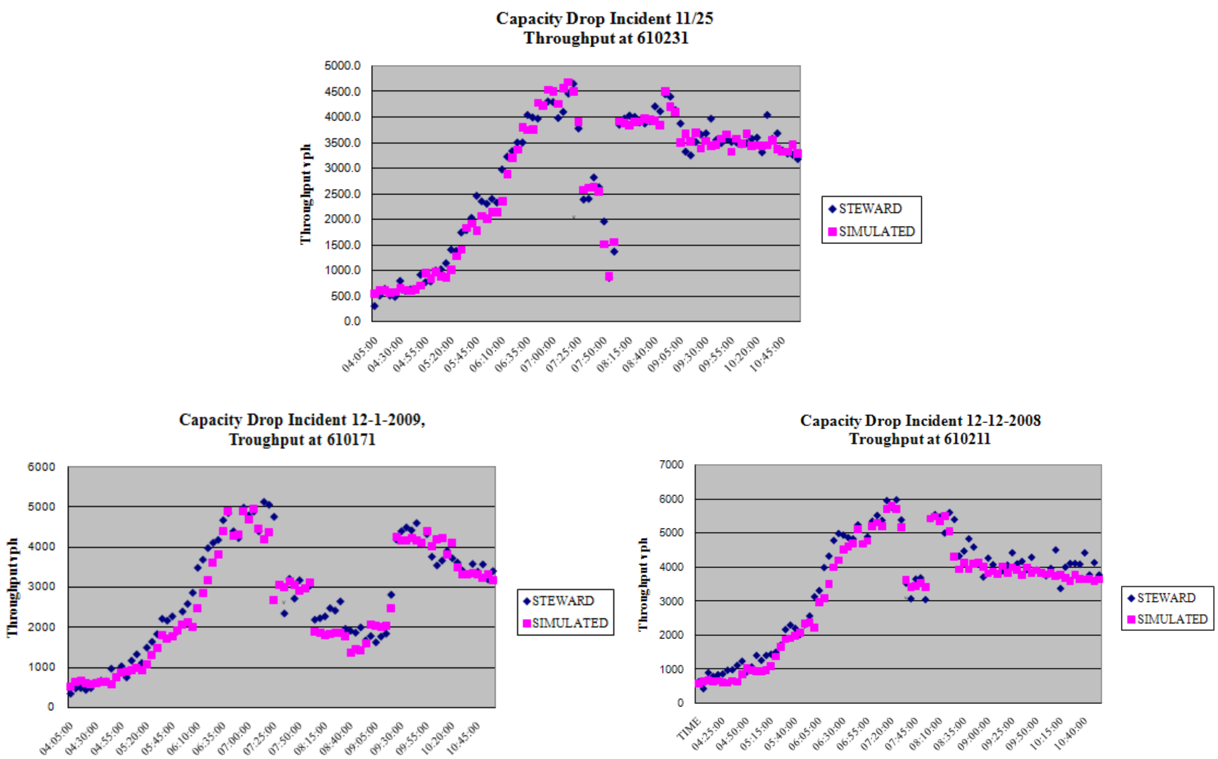


Figure 4-5 Comparison of Capacity Drop based on Detector Measurements and Simulation

4.4.5. Parameter Adjustment based on Other Measures

In addition to the drops in capacity calibrated as described above, it is important that the simulation models be able to replicate the changes in performance measures, such as throughputs, speeds, and occupancies. In order to compare the simulation results with traffic detector field measurements, the traffic detector readings in the simulation model were extracted at 20-second intervals, and aggregated at 5-minute and 15-minute intervals, for comparison purposes.

As stated in the methodology, two parameters were found to impact the performance measures when fixing the assessed segment capacity in CORSIM: the deceleration for non-emergency conditions parameter and the location of incident warning sign parameter. These parameters were varied to obtain the best goodness-of-fit between the observed and simulated measurements. Three measures were used to evaluate the goodness-of-fit for the throughput and speed measurements under incident and normal conditions. These measures were the absolute mean percent error (AMPE), the normalized root mean square error (RMSN), and the root mean square percent error (RMSPE). Table 4-1 shows the aggregated values of these measures for all detection stations during the period of analysis. Table 4-1 indicates that, for all simulated conditions (i.e., the no-incident and three incident conditions), the deviations of the simulated throughputs from the measured throughputs, as measured by the RMSN and RMSPE, were between 12% and 15%. In addition, the AMPE for the throughput ranged from 1% to 10%. It thus appears that the goodness-of-fit of throughput measurement for incident conditions was as good as those for no-incident conditions. However, Table 4-1 shows that the RMSN, RMSPE, and AMPE for speed measurements were higher for incident conditions (7% to 14%) compared to normal conditions (1% to 5%). This is to be expected, since unlike incident conditions, the normal conditions are free-flow conditions, which are easier to simulate.

Figure 4-6 shows the simulated throughput and speed measurements as compared to real-world detector measurements from the STEWARD data warehouse for selected detector locations during Incident 1. As shown in this figure, the calibrated CORSIM model was able to reproduce reasonably well the spatial and temporal variations of the observed volume and speed measurements.

Table 4-1 Goodness-of-Fit of Throughput and Speed

Goodness-of-Fit Measure	Non-incident Conditions	Incident 1	Incident 2	Incident 3
Throughput Goodness-of-Fit				
RMSN	0.12	0.13	0.15	0.12
RMSPE	0.13	0.13	0.15	0.13
AMPE	0.06	0.01	0.06	0.10
Speed Goodness-of-Fit				
RMSN	0.05	0.14	0.13	0.10
RMSPE	0.05	0.11	0.08	0.11
AMPE	0.01	0.11	0.13	0.07

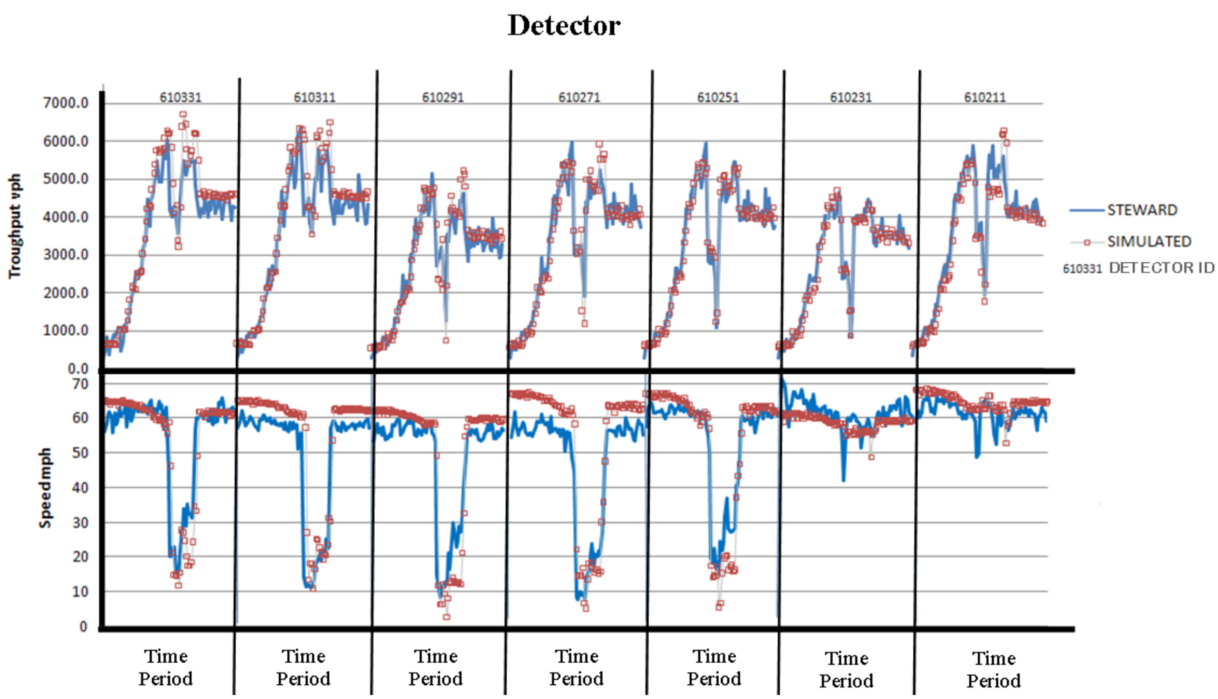


Figure 4-6 Simulation and Real-World Speed and Throughput Measurements for Incident 1

4.4.6. Investigation of the Significance of the Modeling Efforts

This section includes a discussion of the simulation model performance when different approaches were used to simulate the incident. These approaches varied in the level of effort required to collect and use ITS data, and to fine-tune the model parameters. The comparison was made based on the simulation of Incident 1, as introduced earlier in this paper. The following approaches were investigated and are sorted below based on the increasing level of effort required:

- Case IC1: In this case, a one-lane blockage was coded with no rubberneck factor. This is the least effort required to code a one-lane blockage incident. This case used only the lane blockage duration in the incident database
- Case IC2: This case involves the use of the rubberneck factor (RF), in addition to lane blockage specifications, to produce the capacity reductions suggested by the HCM. This case does not require ITS data, but requires the fine-tuning of the incident simulation parameters to achieve the predetermined capacity drops due to incidents
- Case IC3: This case is similar to case IC22, but is based on the capacity reductions reported by Knopp et al. (2009) during lane and shoulder blockage incidents, rather than those suggested by the HCM (*2*)
- Case IC4: In this case, the rubberneck factors were adjusted to produce the observed capacity reductions, based on ITS detector data. As discussed earlier, there were three different capacity levels during the period impacted by Incident 1. In Case IC42, all three levels of capacity drops were replicated
- Case IC5: This case is the same as Case IC42, but with the adjustment of the incident sign location in an attempt to reproduce the observed detector measurements
- Case IC6: This case is the same as Case IC52, but with the adjustment of the non-emergency deceleration in an attempt to reproduce the observed detector measurements.

The results for Input Case 1 (IC1) in Table 4-2 show that the capacity drop created by specifying one-lane blockage in the simulation was not sufficient to create a queue. In Input Case 2 (IC2), the input parameters were fine-tuned to produce the capacity drops in the HCM (TRB, 2001). The results indicate that a queue was generated in the simulation, but that it was significantly shorter than what is estimated from the real-world data. Increasing the drops in capacity during lane and shoulder blockages to the values suggested by Knopp et al. (2009), as in Input Case 3 (IC3), resulted in longer queues in the simulation, producing closer results to real-world conditions, as shown in Table 4-2. Results from Input Case 4 (IC4) indicate that even better queue estimates could be obtained when using the capacity drop information based on traffic detector data. Results from Input Cases 5 and 6 (IC5 and IC6) indicate that additional improvements could be obtained by the fine-tuning of the sign location and non-emergency deceleration to produce the detector speeds and occupancies observed.

Table 4-2 Queue Measures Based on Simulation Outputs and Real-World Measurements

Change to Input Parameters	Time to Dissipate the Queue (min.)	Average Time in Queue (min.)	Average Queue Length (miles)	Maximum Queue Length (miles)
Observed Values	125.0	89.5	3.6	4.9
IC1[2]: Code one-lane blockage only	0.0	0.0	0.0	0.0
IC2[2]: Adjust RF[1] to produce capacity reductions based on HCM	50.0	41.8	0.7	0.8
IC3[2]: Adjust RF[1] as recommended by Knopp et al. (*8*)	95.0	74.3	1.7	2.0
IC4[2]: Adjust RF[1] to produce observed capacity reductions	105.0	79.8	2.3	2.7
IC5[2]: Adjust RF[1] and sign location	110.0	81.2	2.2	2.7
IC6[2]: Adjust RF[1], sign location, and non-emergency deceleration	120.0	94.3	2.9	4.1

Notes: 1 RF: Rubberneck Factor. 2: IC1 to IC6 are Input Case 1 to Input Case 6.

4.5. Conclusions

It can be concluded from the results presented in this study that, when simulating incident conditions, the analyst should fine-tune simulation model parameters to produce the expected or measured drops in capacity during incidents, and the expected impacts of incidents on various performance measures. Thus, simply specifying lane blockages is not sufficient to produce the expected impacts of incidents in the simulation.

ITS data can provide valuable information to support the simulation of incidents. Data from existing ITS deployments allow the validation and calibration of simulation models based on macroscopic measures. However, IntelliDrive technologies will provide a significantly richer data source that will allow the validation and calibration of simulation models based on microscopic traffic parameters.

Traffic detector data combined with incident management data provide information that is critical to the successful use of simulation for the modeling of incidents. The analysis of ITS data indicates that there are different levels of capacities during the periods impacted by lane-blockage incidents. The first level is associated with the drop in capacity due to the initial blockage of lanes by the incident and the associated rubbernecking effects. The second drop in capacity can occur due to emergency vehicle operations that may block additional lanes. This drop in capacity is expected to be a function of the intensity of emergency vehicle operations and the nature of the scene of the incident. The lanes originally blocked may be opened gradually during the incident period; for example, a two-lane blockage incident may become a one-lane blockage incident, if one of the two blocked lanes is opened. Emergency vehicle operations on the shoulder after opening the blocked lanes may also impact the segment capacity due to the rubbernecking effects. Again, this is expected to be a function of the intensity of emergency vehicle operations and the nature of the scene of the incident. Thus, in this study, modeling the dynamic nature of capacity during the incident is found to produce better simulation results.

At least in cases where there are significant emergency vehicle operations at the incident sites, reproducing capacity drops using HCM estimates, based on lane-blockage information at the beginning of the incident, is not sufficient to reproduce the full impacts of the incidents. Therefore, the utilization of archived traffic detector and detailed incident management data is important to allow for the fine-tuning of simulation model parameters to reproduce various incident impacts on capacity, volume, speed, queue, and other measures.

5. Development of Tools to Support SunGuide Testing

5.1. Introduction

As stated in Chapter 1, the objectives of this study include the support of the use of ITS data for simulation model development, and the use of simulation to support the SunGuide software and the associated data archive testing. As shown in Figure 5-1, two major software components have been developed in this project to satisfy these requirements. The two components are referred to collectively as SunSim. The first component is the SunSim core simulation support environment that supports the development of simulation models based on ITS data and user inputs. The second component is the SunSim TSS simulators, which are software utilities in a simulation environment that allows for the exchange of data between the SunGuide software and virtual detectors in the SunGuide subsystem testing and operation evaluation.

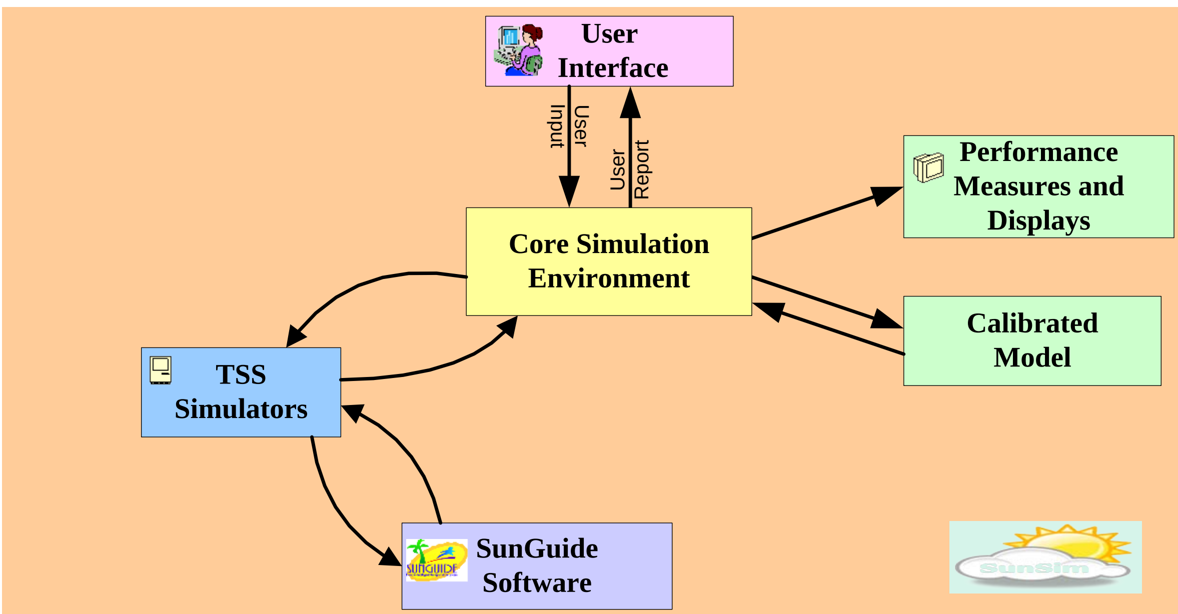


Figure 5-1 Simulation Environment Developed for SunGuide Software Evaluation

5.2. SunSim Core Simulation Development Environment

The SunSim core simulation environment is a software implementation of the methods described in Section 3 to manipulate ITS data for use in the development of simulation models. The inputs to this tool are ITS data archives and a simulation model input file with no input demands. The outputs are a simulation model input file with all input parameters, and after running the simulation, a simulation output and SunGuide TSS file with traffic measurements from the simulation model detectors. In addition, errors of different types could be introduced to this TSS file to emulate real-world detector errors.

The system architecture of the SunSim core simulation development tool is shown in Figure 5-2. The application adopts the typical three-tier architecture. The three-tier approach is a client–server architecture in which the user interface (presentation tier), application tier (business logic), and data tier are separated from each other, resulting in modular software which allows any of the three tiers to be upgraded or replaced independently as requirements or technology change. Appendix A includes the installation guide for the tool and Appendix B includes the user's guide.

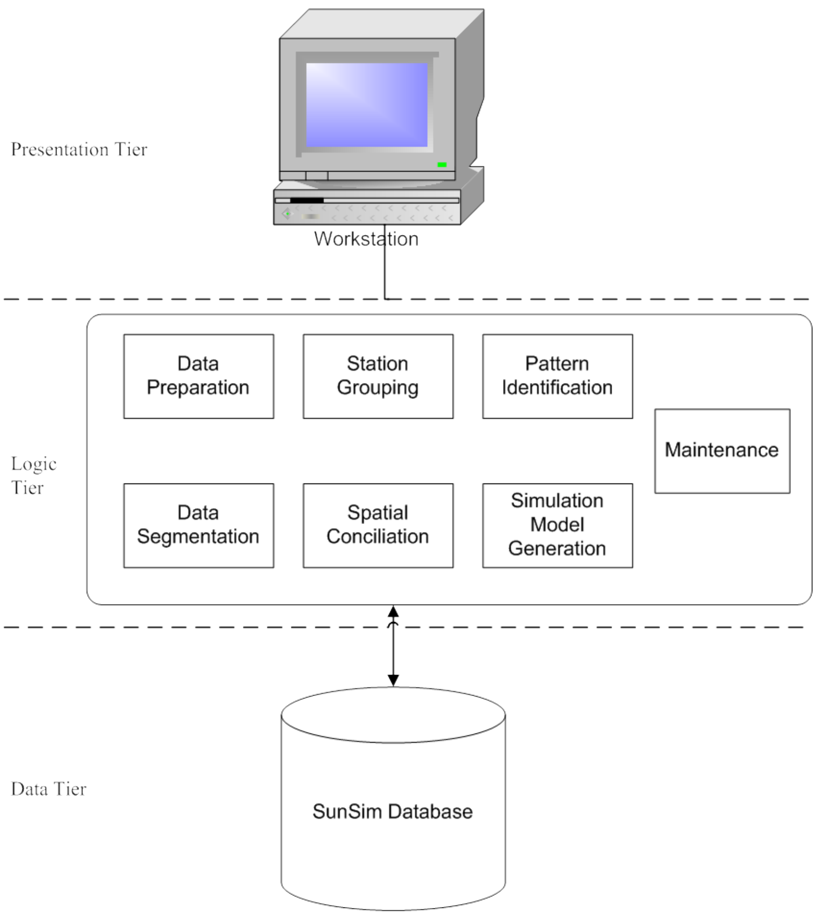


Figure 5-2 SunSim Core Simulation Development System Architecture

The data tier is an Oracle rational database created for storing traffic data, traffic patterns (based on clustering algorithm results), grouped stations (based on user input as described later in this section), period segmentation results, and spatial conciliation results (see Chapter 3 for a description of these processes). All intermediate results of the manipulation processes are stored in the database. The data schema of the database is presented in Appendix C. The user can either start from scratch to generate a CORSIM model based on selected criteria, or continue previous efforts and skip some of the finished steps.

The presentation tier is the interface shown on the user's computer screen. It provides the users with drop-down lists, buttons, and dialog boxes for selecting the desired functions. Figure 5-3 shows a screenshot of the user interface.

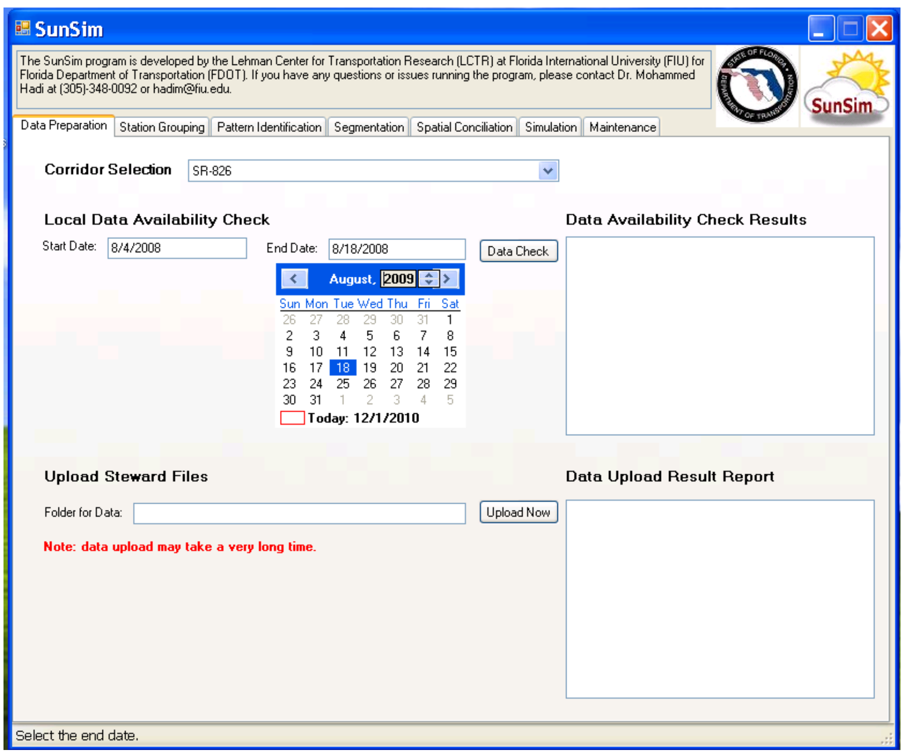


Figure 5-3 User Interface for Data Availability Check

The business logic (application) tier performs the required functionality of the application. Seven major components are included: data preparation, station grouping, pattern identification, data segmentation, spatial conciliation, simulation model generation, and an independent component for intermediate result maintenance.

Figure 5-4 shows the flow chart for the major components of the business logic in the SunSim core simulation development application. First, the user should use the “Data Preparation” component to check for data availability. If traffic data for all or part of the analysis time range is not available in the SunSim database, the user should download the missing data from the STEWARD website and use the “Data Preparation” component for uploading this data to the local SunSim database.

The next step in the process is to categorize the traffic conditions measured in different days into different patterns and use these inputs in simulation modeling (see Chapter 3). To accomplish this pattern identification, the user should first group the available data stations by using the “Station Grouping” component of the interface (see Appendix B for more details). Next, the user should select the “Pattern Identification” component for identifying the desired patterns to be simulated. The pattern identification process is then implemented by calling a clustering function written in MATLAB. The clustering is performed by minimizing the distances between the time series of traffic measurements and the cluster centroid (the average time series in the cluster). The results from this process are clusters of traffic patterns. The number of patterns resulting from the clustering may be specified by the user within the SunSim interface. The members of a particular cluster “look alike” but at the same time they “look different” when compared to members from other clusters; thus, the pattern identification process classifies the days into different clusters based on the similarity of the traffic patterns within each day. Engineering judgment is required to decide on the number of clusters required to produce meaningful traffic patterns in the analysis. The user should vary the number of clusters, examine the results, and make a decision regarding what cluster to use in the analysis. Once a user selects the desired traffic patterns, the related data station and the dates of the days that are grouped in each pattern will be stored in the backend local database for future use.

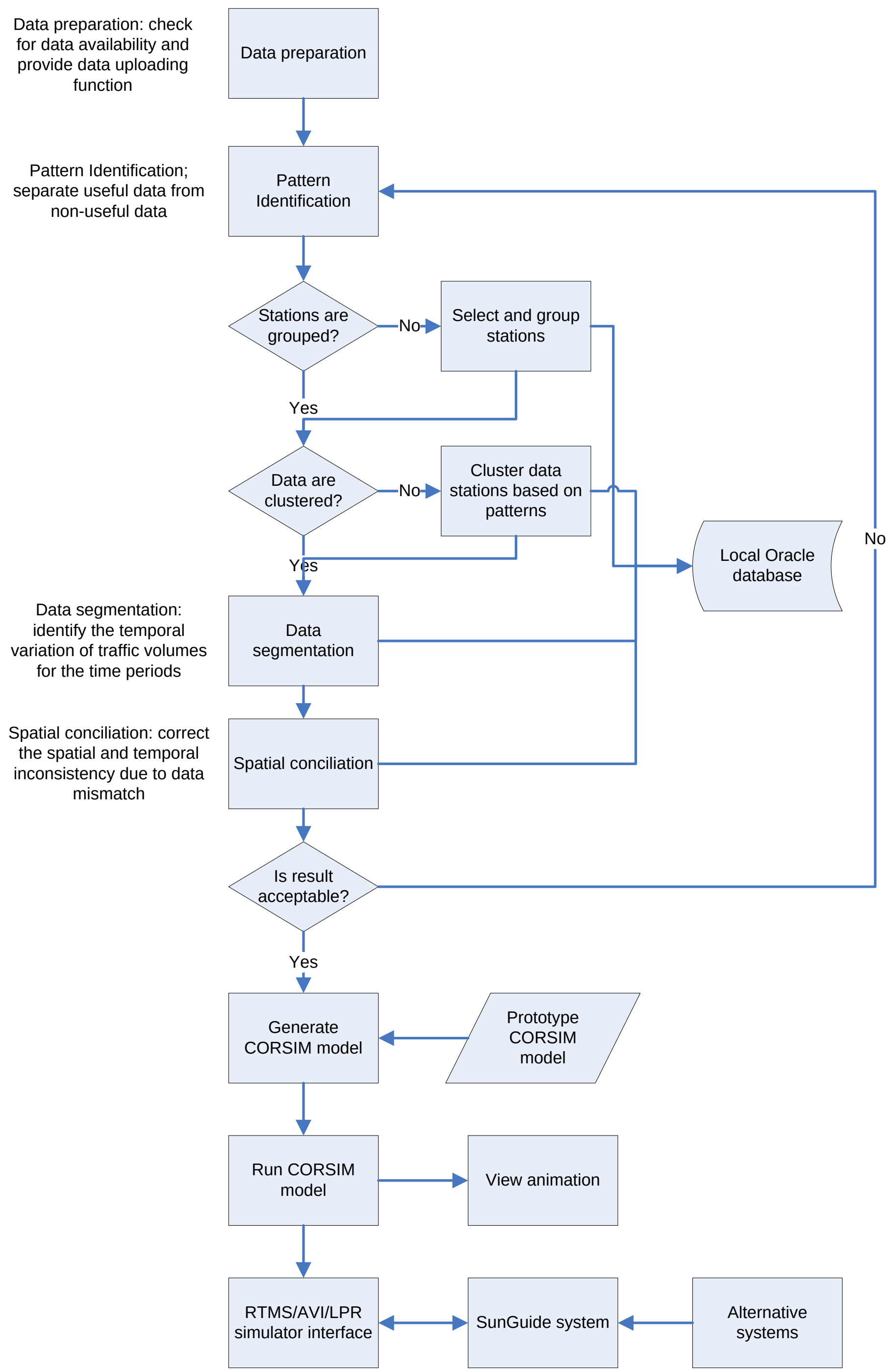


Figure 5-4 SunSim System Flow Chart

The next step to generate a simulation model is data segmentation (period segmentation as described in Chapter 3). In this process, the simulated traffic pattern is segmented into time periods for the analysis (e.g., 5:00 - 5:45 PM, 5:46 - 6:45 PM, etc.). This process aims to describe with as much accuracy as possible the temporal variation of the volume as a piece-wise collection of periods, wherein the volume can be considered constant in each period. This is required because simulation models represent the demand variations by requiring the user to input traffic demands for a series of periods with a constant demand in each period. The demands of the intervals within each period should be similar to each other, but different from the demands of intervals in other adjacent periods. The data segmentation process automatically finds the length of each period as well as the value of the demand for these periods. The data segmentation component allows the user to specify the desired number of time periods to be used in the segmentation.

However, because the data comes from detector readings, it is likely to contain errors. Thus, a process referred to as “Spatial Conciliation” is needed to ensure the spatial and temporal consistency of the measurements (see Chapter 3). This process is conducted for all time periods selected for analysis in the segmentation step, and is repeated automatically for these periods by the developed tool, with little user interaction.

Once the above tasks are finished, the “Simulation Model Generation” component can be used to automatically generate a CORSIM model input file, run the simulation, perform result analysis, and generate TSS raw data files as inputs for SunGuide system testing (again, errors could be introduced to this file to reflect real-world errors). It should be noted that to generate a CORSIM model input file, the user needs to input a CORSIM input file (without the specifications of demands) that includes all the inputs except the demands. These missing inputs are automatically added to the input file by the developed tool.

The SunSim program provides an independent “Maintenance” component for system maintenance tasks such as deleting unwanted data records and eliminating intermediate results. Details of how to use the SunSim core simulation development are described in the User’s Manual included in Appendix B of this report.

5.3. SunSim TSS Simulators

The second component of SunSim is the TSS simulators developed to support the testing of the modules of the SunGuide software. This component constitutes a bridge between a simulation tool (CORSIM in this case) and the SunGuide software, resulting in what can be described as a software-in-the-loop simulation for SunGuide testing purposes. A detailed description of the SunGuide software is presented earlier in Chapter 2. However, there are two SunGuide concepts that are again mentioned here in order to understand the approach used to build the interface between the SunGuide TMC software and simulation. These two concepts are subsystems and drivers. In the SunGuide, a subsystem is a software process that implements a set of closely-related functional requirements; it thus provides data to the Data Bus, making the data available to other SunGuide subsystems and interfaces. As shown in Figure 1, examples of these subsystems are: DMS, CCTV Control, Video Switching, Video Wall, Traffic Detection, HAR, RWIS, Safety Barriers, AVL/Road Rangers, Incident Detection, Travel Time Estimation, Congestion Pricing, 511, Variable Speed Limit, Response Plan Generation, and the Center-to-Center plug-in.

The SunGuide software requires the use of drivers associated with the subsystems. A driver is utilized to implement a vendor-specific protocol or an ITS standard protocol for an ITS device. Drivers do not communicate directly with the Data Bus; rather, each driver communicates with a subsystem.

The simulation environment developed in this project to support the testing of the SunGuide software will simulate and make use of the virtual traffic detectors and readers in the simulation model. In this project, these virtual detectors were interfaced with the SunGuide TSS using the existing SunGuide device drivers. The SunGuide TSS subsystem acquires data from the detection field devices (including speed, volume, and occupancy data) and communicates information between the drivers of these detectors and the SunGuide Data Bus, for use with other SunGuide subsystems and processes. For example, the Travel Time Estimation subsystem uses the speed data collected by the TSS subsystem to later disseminate the travel time information through the DMS subsystem. The TSS subsystem currently supports the protocol of a number of device drivers, including point detectors, electronic toll collection (ETC) readers, and license plate readers (LPR).

This study includes connecting the virtual point detection stations, ETC readers, and LPR readers in the simulation to the SunGuide software, thus allowing the emulation of traffic sensors that are connected through drivers to the SunGuide TSS subsystem. For this purpose, the study developed bridge programs (called simulators) between the virtual detectors in the utilized simulation tool (CORSIM) and the TSS subsystem (see Figure 5-1). The connection is established between the simulation environment and the SunGuide TSS subsystem using existing device protocols with the implementation of the simulator programs. The protocols used are those associated with a true presence microwave detector product (the RTMS detector) and with a number of ETC readers and LPR readers. The simulators read results from running simulation models and communicate with the SunGuide TSS subsystem using existing product drivers. From the point-of-view of the SunGuide system, there is no difference in communicating and using the data of virtual and real-world field devices. The Southwest Research Institute (SwRI), the developer of the SunGuide software for FDOT, has developed an AVI/LPR simulator that can emulate the communication between the AVI/LPR field devices and the SunGuide device drivers. EIS Inc., the vendor of the RTMS detector, has also developed a primitive RTMS simulator that can simulate one RTMS detector at a time. Thus, to reduce the requisite developmental effort, these two simulators were used for developing the simulators in this study. Both simulators were enhanced in this study to allow the use of results from CORSIM and/or other simulation tools as “input” to the SunGuide TSS subsystem, and to allow support for the simultaneous simulation of multiple virtual detectors on one computer, using the multi-thread mechanism. With these enhancements, both simulators support simulating multiple detectors on the same computer and transmitting data to the SunGuide system.

The enhanced RTMS simulator is designed to accept input data from several data sources: the TSS output files from the SunSim core simulation development application described earlier in this section, raw TSS data files from historical SunGuide outputs, or inputs randomly generated by the simulator itself. To improve performance and minimize data sharing or dead lock issues, the selected data files are pre-uploaded into a local Access database and an index created for quick data retrieval by the simulator program, as needed. The details of how to use the RTMS simulator program are described in the Simulator User's Manual included in Appendix D of this document.

The AVI/LPR simulator program developed by SWRI supports one type of detector driver, which is the "ProbeFusionDriver". This program supports four types of protocols, which are "InexZamirZap", "SercoPips", "SiritFlex", and "TranscoreAllegro". The SunGuide documents discuss the use of these protocols and can be reviewed by the user to ensure that this program works as intended. The Simulator User's Manual in Appendix D describes the steps for using the AVI/LPR simulator program to communicate with the SunGuide system.

6. Use Cases for System Evaluation

A number of use cases are designed in this study to demonstrate the use of the developed simulation environment in evaluating the SunGuide software modules and algorithms. These use cases include software load test, travel time estimation based on point detectors, travel time estimation using AVI/LPR, incident alarm threshold procedure testing, and ITS data warehousing process testing. These use cases and evaluation results are discussed in detail below.

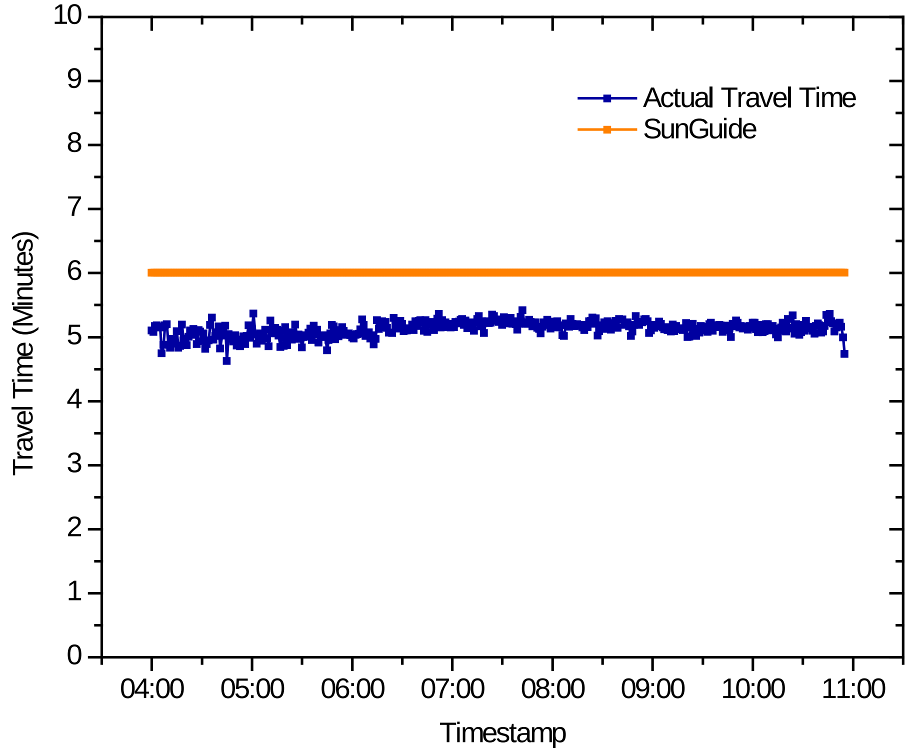
6.1. Load Test

One of the primary concerns of the FDOT was uncertainty in the maximum number of detectors that the SunGuide software can communicate with and process data from for various applications. Although the SunGuide software has been implemented for relatively large networks, the limits on the number of detectors remains unknown. In this study, a hypothetical freeway network with a large number of detectors was simulated. Data were fed from the simulated detectors to the SunGuide system using the model developed herein. It was found that the SunGuide software was able to perform as expected with the maximum number of detectors investigated (450 detection stations), which was deemed to be a sufficient maximum number according to FDOT personnel.

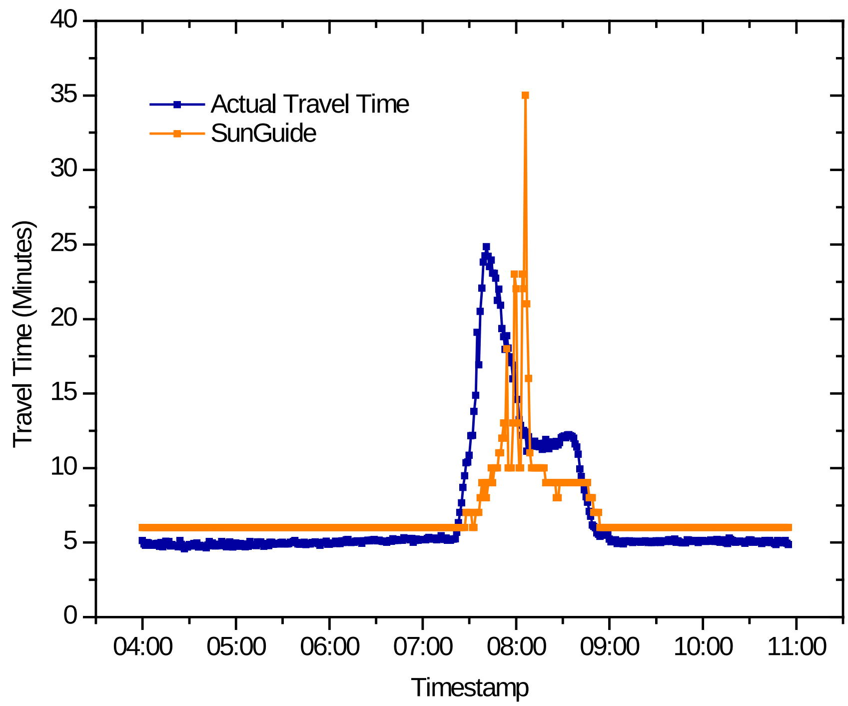
6.2. Travel Time Estimation based on Point Detectors

This use case includes the evaluation of the accuracy of the SunGuide estimation of travel time based on point detectors under incident and no-incident conditions. In addition, the impact of various influential factors could be tested, including the various types of detector errors, missing data, detector spacing, and SunGuide input parameters. The SunGuide software uses a point-to-point (PP) speed-based method to estimate the travel time based on point detector measurements. In this method, the TMC operations staff must associate highway segments with the speed measurement from each detection station.

Figure 6-1 presents the evaluation results of the use case of travel time accuracy under different conditions (i.e., incident and non-incident conditions). In this use case, the eastbound section of SR-826, located in Miami-Dade County, Florida, is chosen as the study corridor. The simulated section is 5.3 miles long and extends from NW 67th Avenue to the Golden Glades Interchange. In the simulation model, 17 point detection stations are modeled with the same configuration as the real-world detection stations. The simulation time period is chosen as a typical workday morning period from 4:00 AM to 11:00 AM. The simulation model is then developed and calibrated based on ITS data, as described in Chapters 3 and 4 of this document. Using the developed environment, the accuracy of the SunGuide travel time estimation is examined under incident and no-incident conditions. The reported speed, volume, and occupancy data from the virtual detectors in the simulation models are sent to the SunGuide software using the communication between the developed simulators and the SunGuide TSS subsystem through the related device drivers. The travel time is set to be computed every minute by the SunGuide software, and the results are then compared with the actual travel time outputs from the CORSIM simulation runs. Figures 6-1(a) and 6-1(b) present the travel time comparisons for the no-incident and incident conditions, respectively.



(a)



(b)

Figure 6-1 Travel Time Accuracy for Incident and No-incident Conditions

As mentioned before, the SunGuide software uses the PP algorithm for travel time estimation. In addition, the measured speeds are capped by the speed limit (for SR-826, the speed limit is 55 mph) before the calculation of the travel times to prevent reporting short travel times, which may encourage speeding. Figure 6-1(a) shows that the SunGuide software produces a reasonably good travel time estimates during no-incident conditions. However, due to the use of the speed cap, the estimates of travel time in SunGuide are slightly higher than the simulated travel time reported in the CORSIM output, which is expected.

Figure 6-1(b) presents the travel time estimation results for incident conditions. For this simulation case, the attributes of a real-world incident with one-lane blockage was retrieved from the FDOT District 6 SunGuide incident management database. This one-lane blockage incident occurred at 7:23 AM. After 35 minutes, the incident was moved to the shoulder and the blocked lane was open. The incident was cleared completely at 8:45 AM, 82 minutes after it started. Figure 6-1(b) shows that that the travel time estimates produced by the SunGuide software are not satisfactory under these incident conditions. One reason for this appears to be that the travel time estimation conducted by the SunGuide software is based on the current traffic conditions, which cannot capture the dynamic changes in the queue length during incident conditions. Based on this finding, the research team has investigated new algorithms for travel time estimation to be implemented in the SunGuide software, as described in the final report of the "Decision Support Tools to Support the Operations of Traffic Management Centers" project sponsored by the FDOT central office. In addition, the project evaluated the accuracy and reliability of travel time estimation under different conditions, set-ups, input parameters, and detector errors. In that project, the simulation models developed, calibrated, and described in Chapters 3 and 4 of this report were used to test different travel time estimation methods under different conditions (for more details, see the aforementioned final report of the project).

6.3. Travel Time Estimation Using AVI/LPR

This use case evaluates the SunGuide estimation of travel time under incident and no-incident conditions for the AVI/LPR detector type. However, one important distinction between the evaluation presented in this section for travel time estimation based on AVI/LPR and that presented in Section 6.2 for travel time estimation based on the RTMS detector, is that the accuracy of disseminated travel time in this case study assumes that the travel time is delivered through in-vehicle or hand-held equipment, and that the information is updated every two minutes. The evaluation in Section 6.2 assumes that the delivery is made at the DMS locations, thus the provided time latency is significantly longer, particularly during incident conditions when traffic is moving slowly and takes a longer time to cross the links for which travel times are estimated. It is therefore important that the results of Sections 6.2 and 6.3 are not compared in terms of the accuracy of travel time estimation based on AVI/LPR versus point detectors. Another use case, however, is currently being studied by the research team that will allow this comparison.

Figures 6-2 through 6-11 present the evaluation results of travel time accuracy under non-incident and incident conditions for different AVI/LPR matching percentages. The eastbound section of SR-826 located in Miami-Dade County, Florida is chosen as the study corridor. The simulated section is 6.6 miles long and extends from before the NW 67th Avenue interchange to the Golden Glades Interchange. Two paired AVI/LPR detectors are modeled in the simulation, with one located just after the starting point and the other located at the end of the simulated corridor. The simulation time period is chosen as a typical workday morning period from 5:00 AM to 11:00 AM. The accuracy of the SunGuide travel time estimation is examined under incident and no-incident conditions using the developed environment. The reported speed, volume, and occupancy data from the virtual detectors in the simulation models are subsequently sent to the SunGuide software, using the communication between the developed simulators and the SunGuide TSS subsystem through the related device drivers. The travel time is set to be computed every two minutes by the SunGuide software and is assumed to be instantly disseminated to travelers using hand-held or in-vehicle equipment, as discussed above. The results are then compared with the manually calculated travel times from the vehicle trajectory data retrieved from the CORSIM simulation runs.

Scenarios with 100%, 50%, 20%, 10%, 5%, and 1% matching rates were designed and evaluated for the non-incident situation. For the 1% matching rate scenario, there were few matched cases and the travel time was reported only two times by the SunGuide software during the entire six-hour period. Therefore, the 1% matching rate scenario was excluded from the comparison.

Figures 6-2 through 6-6 show that the SunGuide system estimates and the manually calculated values are very similar, independent of the matching rate. At this level of congestion, vehicles are traveling close to free flow speed, and measuring the speeds of a small proportion of the vehicles is sufficient to produce accurate speed estimation. However, as the matching rate drops below 20%, the number of two-minute intervals without travel time estimates starts to increase with the decrease in the matching rate. At a 10% matching rate, there are noticeable two-minute intervals with no reported travel times (see Figure 6-5). When the matching rate drops even further to 5% or 1%, the travel times are only reported at scattered two-minute intervals, as shown in Figures 6-6 and 6-7. Based on these results, a matching rate of 10% or higher is recommended if the AVI/LPR is the sole detector type for travel time calculation.

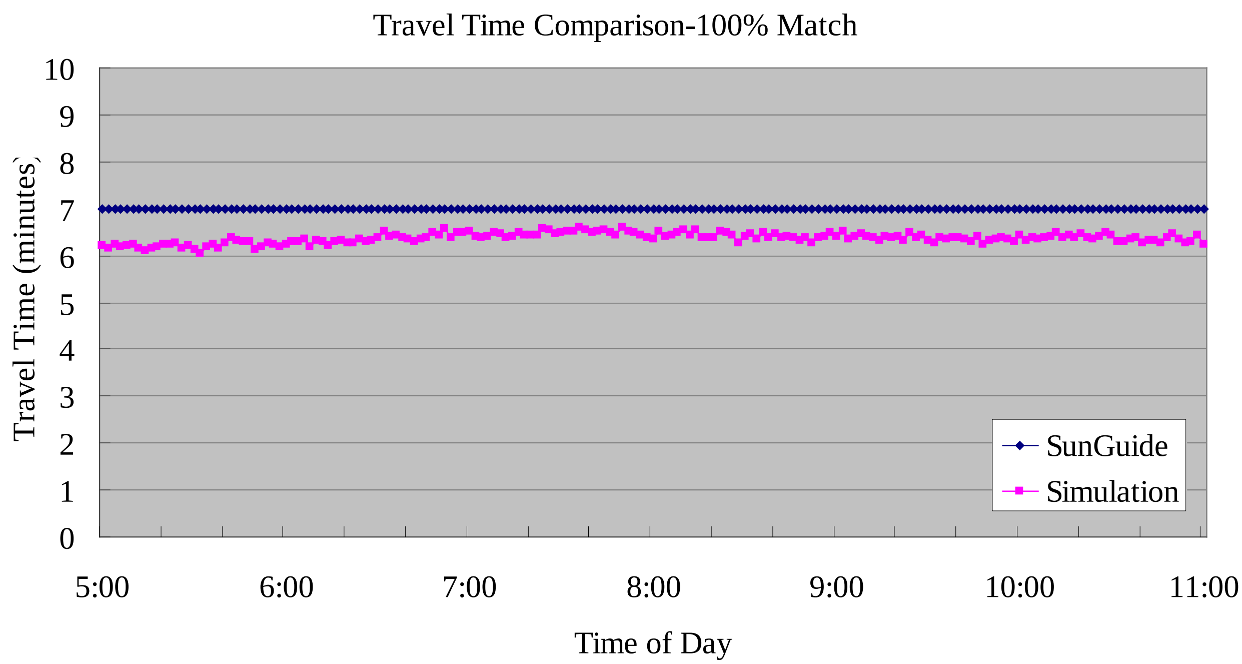


Figure 6-2 Travel Time Comparison for No-incident Conditions, 100% Match

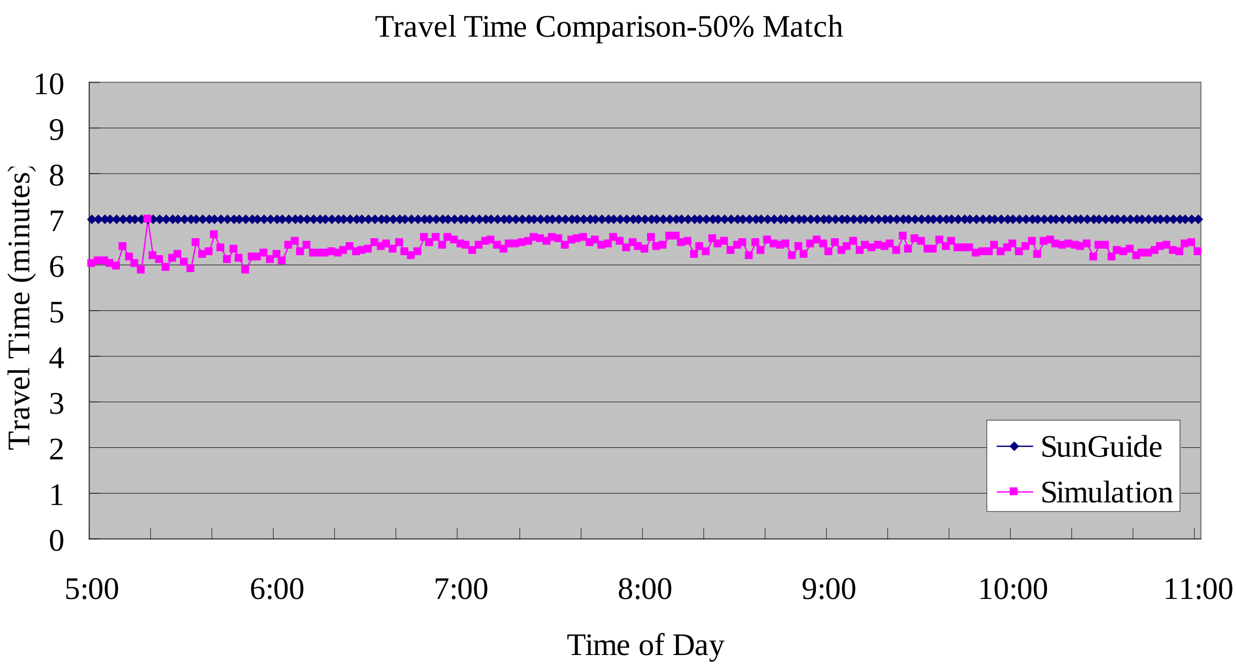


Figure 6-3 Travel Time Comparison for No-incident Conditions, 50% Match

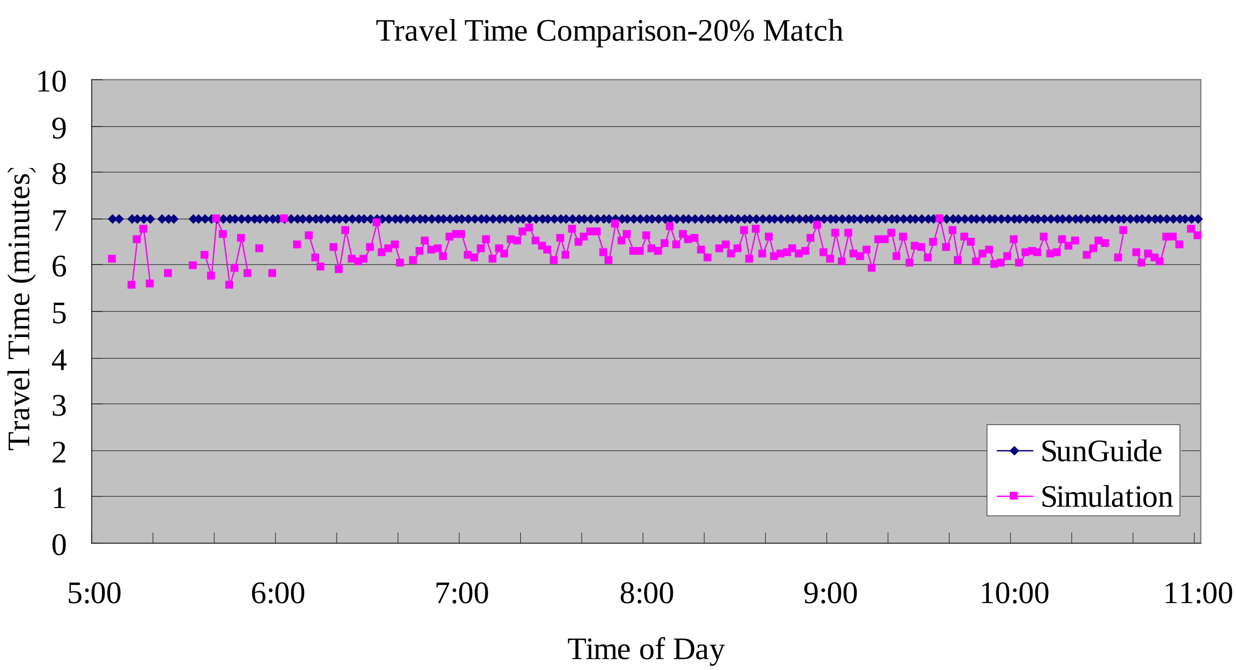


Figure 6-4 Travel Time Comparison for No-incident Conditions, 20% Match

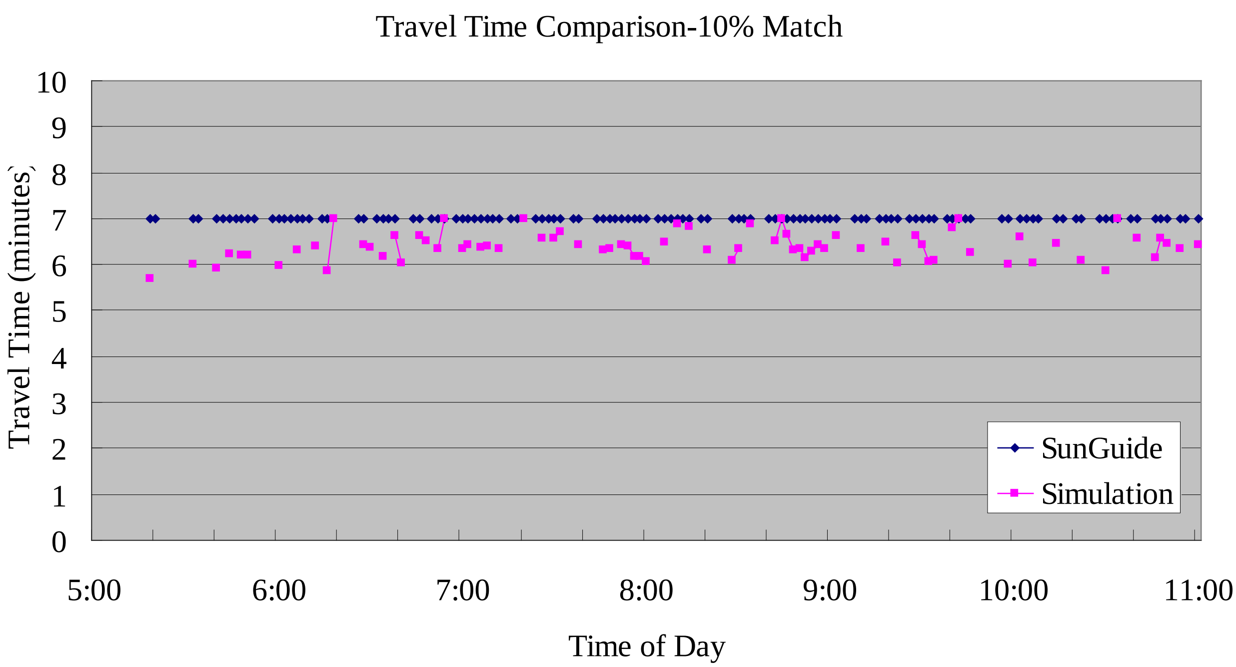


Figure 6-5 Travel Time Comparison for No-incident Conditions, 10% Match

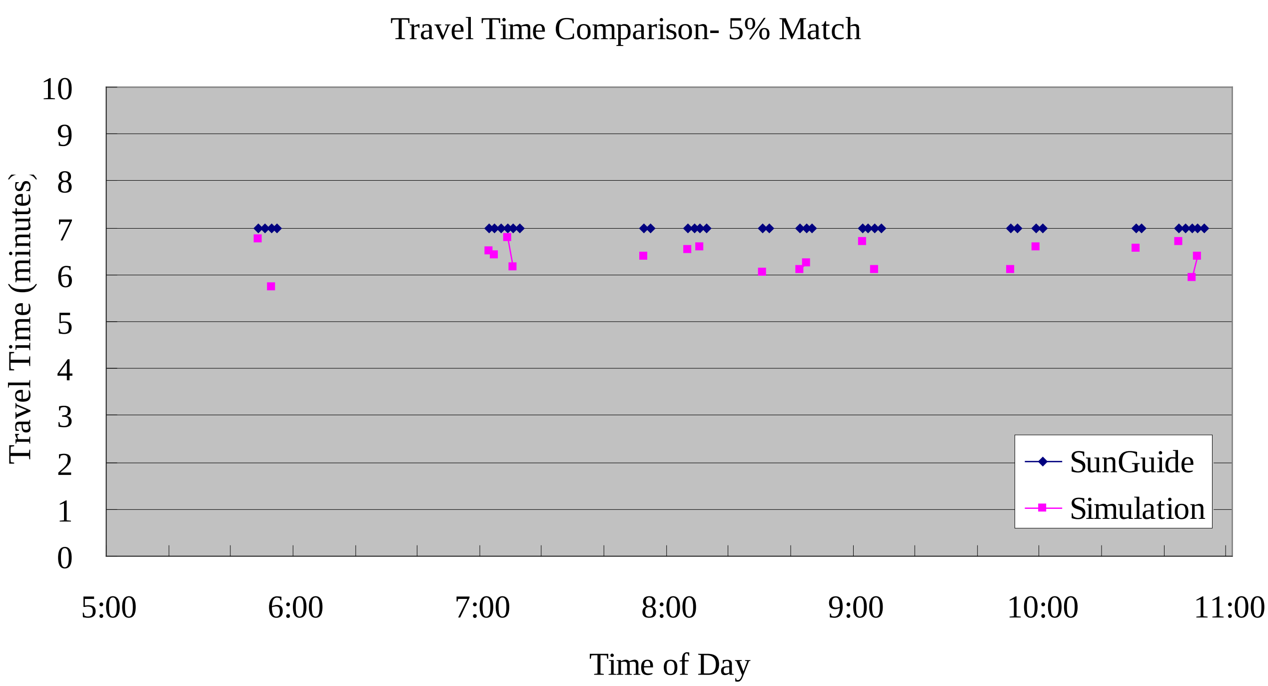


Figure 6-6 Travel Time Comparison for No-incident Conditions, 5% Match

Scenarios with 100%, 50%, 20%, 10%, 5%, and 1% matching rates were also evaluated for the incident situation. Similarly to the use case for the RTMS detectors, for this AVI/LPR simulation case, the attributes of a real-world incident with one-lane blockage was retrieved from the FDOT District 6 SunGuide incident management database. This one-lane blockage incident occurred at 7:35 AM. After 25 minutes, the incident was moved to the shoulder and the blocked lane was open. The queue was completely dissipated at 8:20 AM, 45 minutes after the incident started.

Figures 6-7 through 6-11 show the comparison of the accuracy of travel time estimation for the scenarios with 100%, 50%, 20%, 10%, and 5% matching rates. The results for the scenario with a 1% matching rate are not reported because of the unacceptable number of travel times reported by SunGuide for this case. Figure 6-7 shows that with a 100% matching rate, the travel times reported by the SunGuide system correspond well with the manual calculations based on simulation results. The accuracy of travel time estimation is much higher that the results obtained based on point detectors, as reported in Figure 6-1(b). However, it is interesting to note that even at a 100% matching rate, there is still a discrepancy between the estimation of the incident clearance time and the lane blockage clearance time due to the fast moving recovery shockwave. Figures 6-8 and 6-9 show that, with the 50% and 20% matching rates, the SunGuide travel estimation based on AVI/LPR can generate reasonable travel time and matching results for the travel times reported. As the matching rate drops to 10%, however, the reported travel times from both methods cannot predict the travel time changes caused by the incident, as shown in Figure 6-10. Depending on the arrival time at the incident location, vehicles may experience different delays. Figure 6-11 shows that the results are almost unacceptable for both methods when the matching rate drops to 5%. In these instances, the travel times can only be reported during some of the two-minute periods, and most of them are scattered. In summary, the figures show that a matching rate of higher than 10% is recommended in order to have acceptable travel time estimation results for the incident scenario.

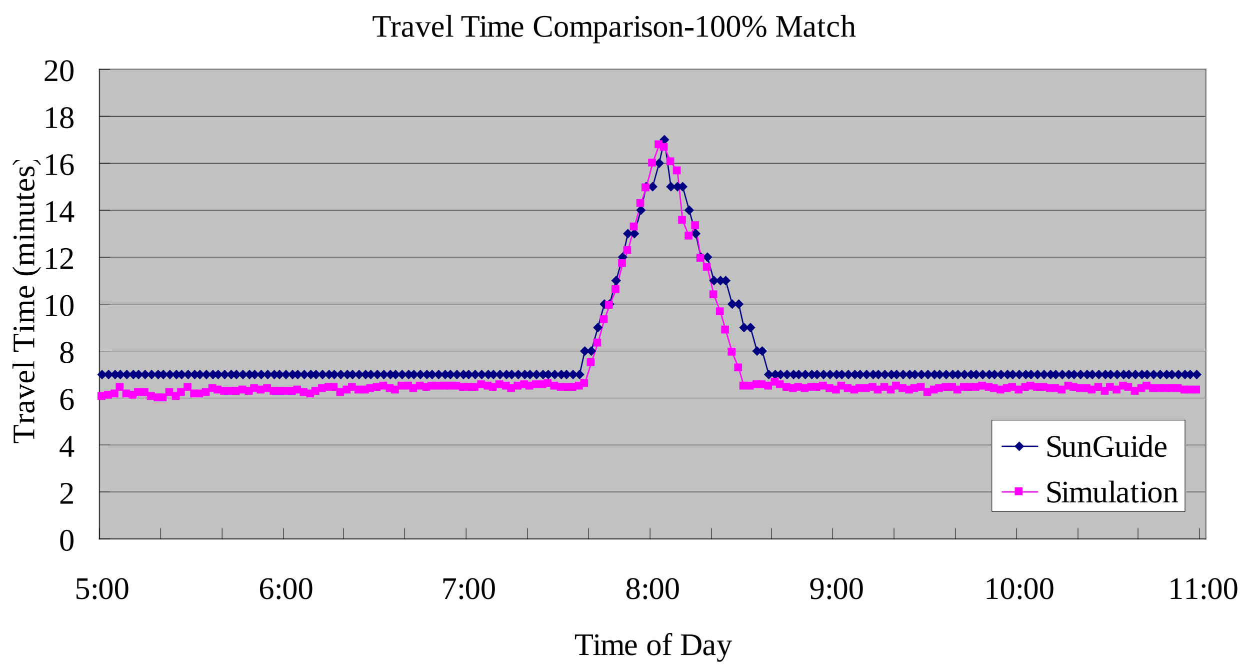


Figure 6-7 Travel Time Comparison for Incident Conditions, 100% Match

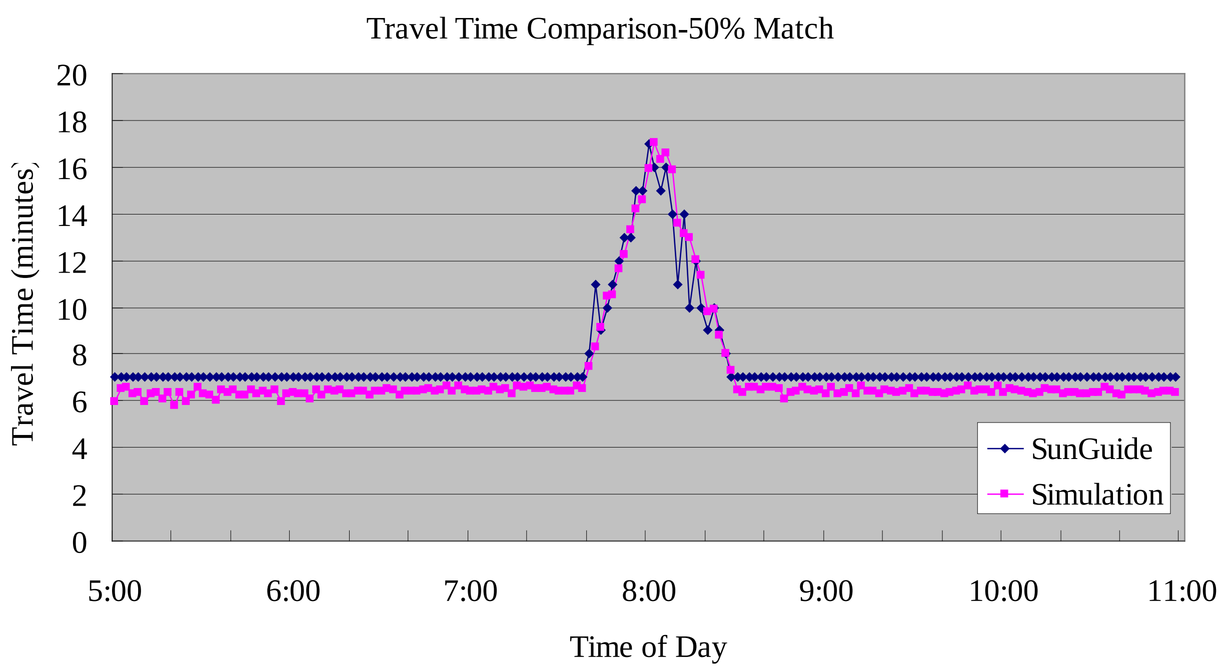


Figure 6-8 Travel Time Comparison for Incident Conditions, 50% Match

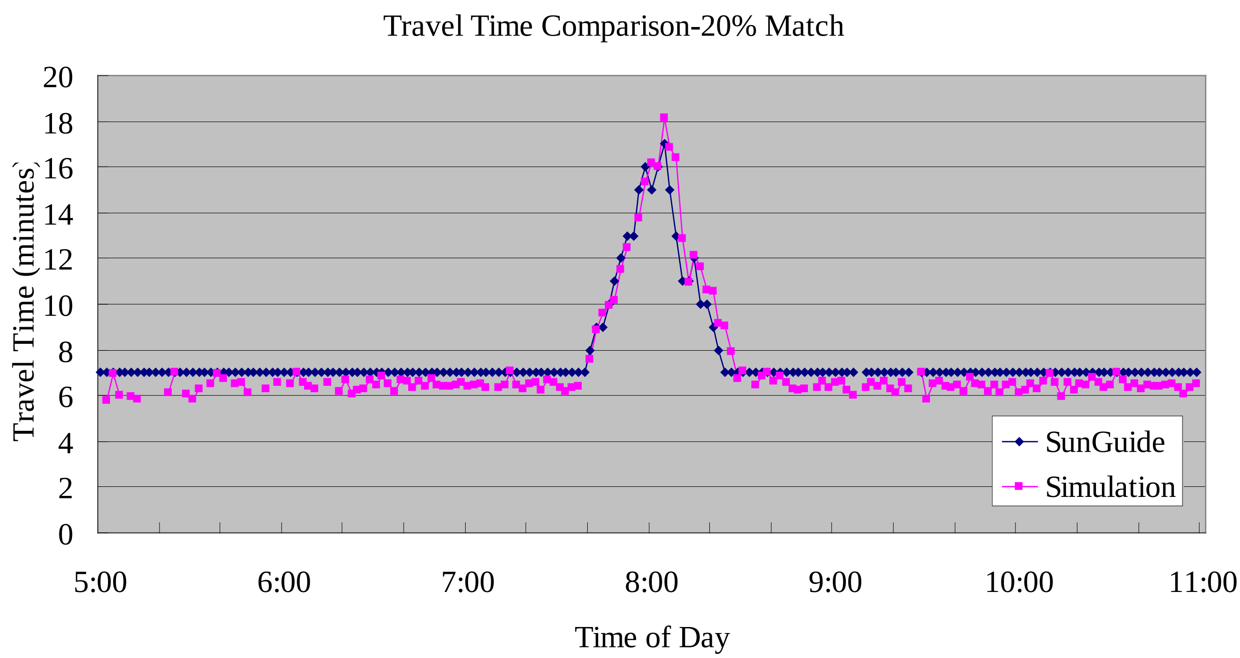


Figure 6-9 Travel Time Comparison for Incident Conditions, 20% Match

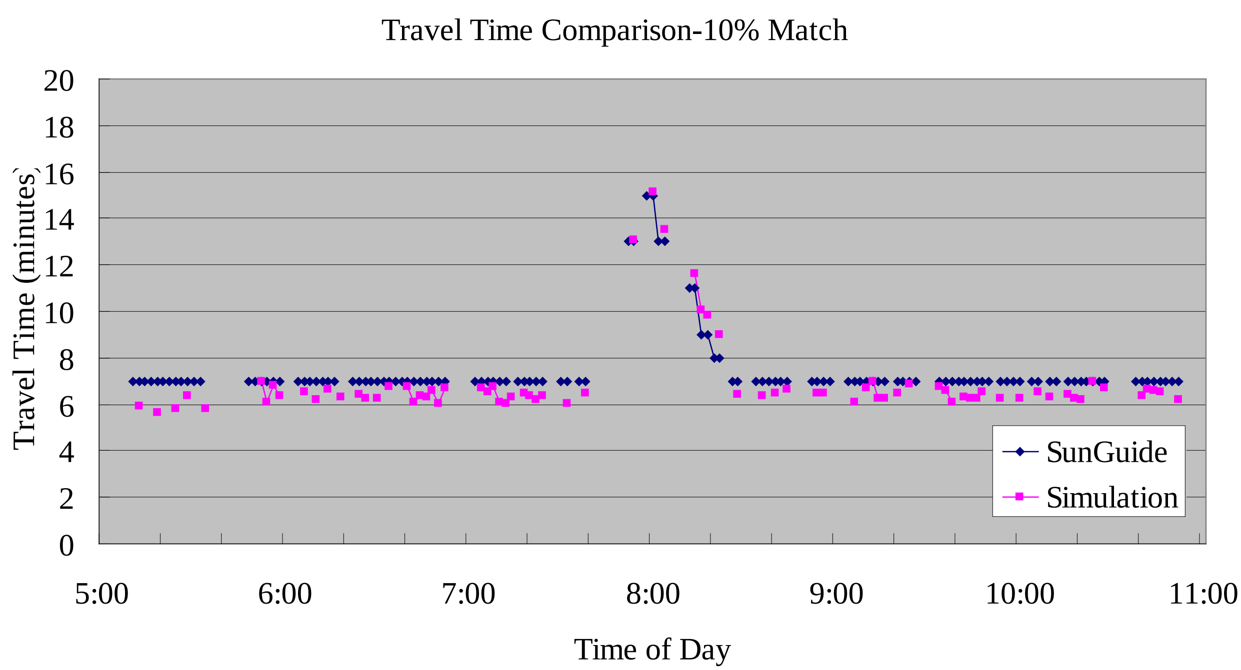


Figure 6-10 Travel Time Comparison for Incident Conditions, 10% Match

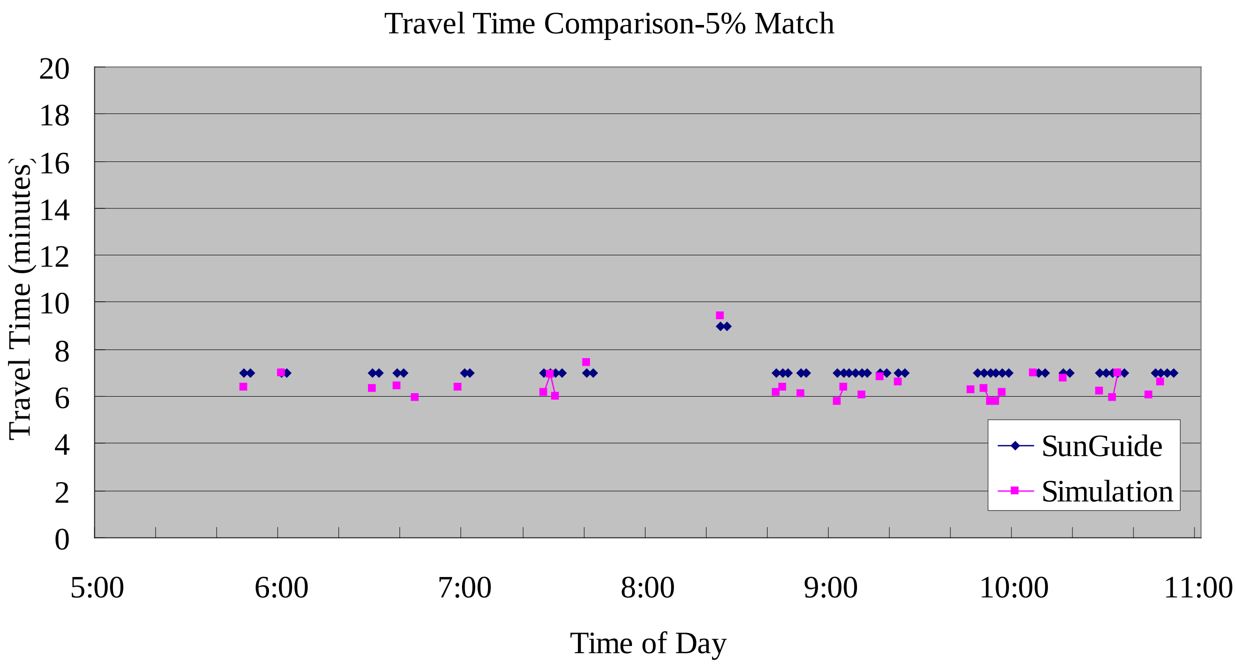


Figure 6-11 Travel Time Comparison for Incident Conditions, 5% Match

6.4. SunGuide Incident Alarm Procedure

The SunGuide software produces an alarm referred to as the TSS alarm based on TSS traffic detector data. The objective of the alarm is to draw the operator's attention to conditions which indicate that an incident may have occurred. However, it was found that the implementation of the alarm in previous versions of the SunGuide was not useful. In this prior implementation, the software generates an alarm every time the speed or occupancy of a link crosses a specified threshold. Thus the traffic data will often oscillate rapidly around this "Alarm Threshold", causing a large number of alarms to be generated and flooding the operators with alerts.

The latest version of the SunGuide software (Version R5.0.3) includes an enhancement to address this problem. It proposes adding an additional "Recovery Threshold" to the TSS alarms, which is meant to prevent the Sunguide from generating a large number of alerts when the traffic data is oscillating around the Alarm Threshold. Once an alarm is generated for a given Alarm Threshold, a second alarm will be generated only if the traffic data has improved past the configured Recovery Threshold since the previous alarm. Likewise, alarms which have not been addressed by an operator will only be removed from the alert box when the traffic condition has improved past the Recovery Threshold. In this way, new alarms are not triggered unless the traffic dips back below the Alarm Threshold. Figure 6.12 presents a visual description of this enhancement.

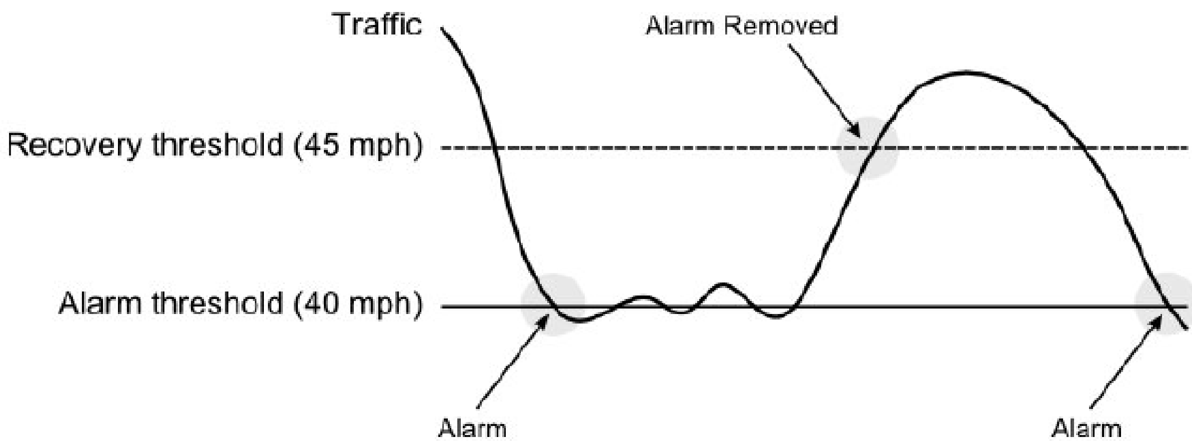


Figure 6-12 TSS Alarm Enhancement in SunGuide R5.0.3

Simulation is ideal to test the new enhancement and to set the thresholds to maximize the performance of incident detection without generating excessive false alarms. In this study, the RTMS simulator was used to generate input traffic data to the SunGuide system and evaluate the enhancements to the TSS alarms. Nodes and links were added for the real-world detectors on State Road 826 (SR-826) and it was possible to display the traffic parameters (speed, volume, and occupancy) on the operator map. The display of multiple detectors was examined to ensure the link coloring in the display and the alarms were working correctly. However, a single detector “DS-1501N” was chosen for the detailed evaluation of the proposed enhancement.

The requirements and the overall evaluation results are listed in Table 6-1. The final evaluation results show all the requirements are implemented successfully in SunGuide R5.0.3. Multiple scenarios using different traffic data combinations were designed for detector “DS-1501N” to cover all the possible cases for requirements a, b, c, d, f, and g. The details are discussed later in this section. Requirement e (TSS Alarm Configuration) was evaluated using the 49 detectors on SR-826 and it showed that the Recovery Threshold was configurable for all detectors.

Table 6-1 TSS Threshold Requirements Proposed by SwRI

Name	Specification	Test Result
a. TSS Alarm Triggering	A TSS alarm shall be triggered when the speed or occupancy values cross the Alarm Thresholds configured for that link	Successful
b. TSS Alarm Triggering	A TSS alarm shall not be triggered when TSS reports no traffic (i.e., speed, volume and occupancy = 0).	Successful
c. TSS Alarm Retriggering	Once a TSS alarm has been triggered it shall not be retriggered until the traffic data for that link subsequently crosses the Recovery Threshold	Successful
d. TSS Alarm Removal	TSS alarms that have not been addressed by an operator shall be removed when the traffic data that triggered the alarm subsequently crosses the Recovery Threshold.	Successful
e. TSS Alarm Configuration	The Recovery Threshold shall be configurable per link in the Admin Editor.	Successful
f. TSS Link Data Screen	The Operator Map shall represent the data region between the threshold and recovery threshold with yellow in the "TSS Link Data" screen.	Successful
g. TSS Alarm Operator Map View	A link where the traffic data currently falls between the Recovery Threshold and the Alarm Threshold shall be displayed in yellow on the Operator Map.	Successful

In SunGuide v5.0.3, the added enhancement allows the end users to add thresholds for either speed, occupancy, or a combination of both. Therefore, for requirements a and c (TSS Alarm Triggering and Retriggering), scenarios were designed to cover the three possible threshold settings (speed, occupancy, and speed and occupancy). For each of the three threshold settings, five runs with different parameter settings were conducted to evaluate:

- if a TSS alarm would be triggered when the speed or occupancy values fell below the configured Alarm Thresholds (requirement a),

- if a TSS alarm would be triggered again when the speed or occupancy values increased above the Alarm Thresholds but below the Recovery Thresholds and then fell again below the Alarm Thresholds (requirement c - scenario 1, requirements f and g),
- if a TSS alarm would be triggered again if the speed or occupancy values increase above the Recovery Thresholds and then fell again below the Alarm Thresholds (requirement c - scenario 2, requirement d).

Table 6-2 lists the testing results for the three scenarios, each with five runs. Table 6-2 (a) is for using only speed as the parameter, Table 6-2 (b) is for using occupancy only, and Table 6-2 (c) is for using both speed and occupancy. Results for five runs are listed in Tables 6-2 (a) and 6-2 (b). Results for only one run are listed in Table 6-2 (c) since it contains a lot of boundary conditions and presenting the results from one run will make it clearer. For the use of speed and occupancy case, results of the five runs are all similar and presenting the results of one run in Table 6-2 (c) is enough to show the effects. Results for evaluating the enhanced threshold design requirements a and c are listed in Table 6-2. The design requirements d, f, and g cannot be quantified and were checked visually when conducting the tests for the three scenarios. It was found that the design requirements d, f, and g are all met with the three test scenarios (speed only, occupancy only, speed and occupancy), each with five runs. It was concluded that based on the conducted testing, the enhanced threshold design requirements d, f, and g were met and performed as expected. Results in Table 6-2 (a) to (c) also show that the design requirements a and c perform as expected.

Table 6-2 (a) TSS Threshold Evaluation Results for Speed Only

Run No.	Parameter: Speed (mph)		Detector Value	Result	Event ID
	Alarm Threshold	Recovery Threshold			
1	10	20	9	Alarm triggered	634299246001406000
			15	Alarm not retriggered	N/A
			22->9	Alarm retriggered	634299269401406000
2	12	30	10	Alarm triggered	634299257801406000
			20	Alarm not retriggered	N/A
			35->10	Alarm retriggered	634299264401406000
3	15	35	12	Alarm triggered	634299272401406000
			25	Alarm not retriggered	N/A
			38->12	Alarm retriggered	
4	20	40	18	Alarm triggered	634299282401406000
			30	Alarm not retriggered	N/A
			45->18	Alarm retriggered	634299278201406000
5	23	43	20	Alarm triggered	634299291001406000
			33	Alarm not retriggered	N/A
			45->20	Alarm retriggered	634299278201406000

Table 6-2 (b) TSS Threshold Evaluation Results for Occupancy Only

Run No.	Parameter: Occupancy (percent)		Detector Value	Result	Event ID
	Alarm Threshold	Recovery Threshold			
1	70	10	80	Alarm triggered	634299313805312000
			60	Alarm not retriggered	N/A
			9->80	Alarm retriggered	634299322808437000
2	75	15	80	Alarm triggered	634299331611875000
			60	Alarm not retriggered	N/A
			13->80	Alarm retriggered	634299334813281000
3	80	20	85	Alarm triggered	634299338214531000
			60	Alarm not retriggered	N/A
			15->85	Alarm retriggered	634300149640194000
4	65	10	70	Alarm triggered	634300153041131000
			60	Alarm not retriggered	N/A
			7->70	Alarm retriggered	634300155042069000
5	60	10	70	Alarm triggered	634300157642850000
			50	Alarm not retriggered	N/A
			7	Alarm retriggered	634300159644725000

Table 6-2 (c) TSS Threshold Evaluation Results for Speed plus Occupancy

Run No.	Parameter: Speed (mph) and Occupancy (percent)				Detector Value		Result	Event ID
	Alarm Threshold		Recovery Threshold		Speed	Occ.		
	Speed	Occ.	Speed	Occ.				
1	10	60	30	10	8	70	Alarm triggered	634300195260975000
					15	70	Alarm not retriggered	N/A
					15	20	Alarm not triggered	N/A
					15->35->8	70	Alarm retriggered	634311127624922500
					15->35->8	70	Alarm not retriggered	N/A
					15	20	Alarm not retriggered	N/A
					15	20->8->70	Alarm not retriggered	N/A
					8	20->8->70	Alarm retriggered	634300199463006000

The enhanced threshold design requirement b is for scenarios with missing traffic data for all the traffic parameters (i.e., speed, volume, and occupancy = 0). Table 6-3 shows the evaluation results for the designed test scenarios. There is an “or” relationship between the speed and occupancy thresholds. It means that an alarm should be triggered when either of the speed or occupancy threshold condition is met. As desired, when there is no traffic input data (speed, volume, occupancy = 0), no alarm is triggered. When any of the three parameters has input and the alarm threshold value is met, an alarm is triggered and an event is generated by the system.

Table 6-3 TSS Threshold Evaluation Results for No Traffic Data Scenario

			Speed Threshold		Occupancy Threshold			
Speed	**Volume**	**Occupancy**	**alarm**	**recovery**	**alarm**	**recovery**	**triggering alarm**	**event id**
0	0	0	10	30	60	10	X	X
8	0	0	10	30	60	10	√	634300172251131000
8	10	0	10	30	60	10	√	634300169649413000
8	10	70	10	30	60	10	√	634300174451756000
8	0	70	10	30	60	10	√	634300176651600000
0	10	0	10	30	60	10	X	X
0	10	70	10	30	60	10	√	634300182550381000
0	0	70	10	30	60	10	√	634300185856756000

6.5. Assessment of Data Warehousing Processes

One of the useful applications of the developed tools is the generation of TSS files based on measurements from virtual detectors in traffic simulation software. In this way, different types of errors and missing data can be introduced to emulate real-world traffic detector data. These TSS files can be used to generate data archive records. Once the data warehouse records are generated based on simulation detector data, they can be examined in comparison to the actual measures of traffic experience in the simulation. This examination allows for the testing of the effects of data warehousing processes such as filtering, imputation, smoothing, and aggregation on the quality of the saved data.

Initially, this application of simulation models was planned as a use case in this project. However, since the STEWARD proof-of-concept research project ended before this use case was ready for testing, it was not possible to use the assessment of data warehouses as a use case in this project. However, the developed tool can be utilized as described in this section to support the activities of the upcoming FDOT ITS data warehouse implementation project.

Appendix A: Installation Manual of the SunSim Core Simulation Support Tool

Before the installation of the SunGuide Core Simulation Support Tool, the user needs to install the database backup file, the MATLAB DLL package, and the SunSim interface program.

Prerequisites

First, the user must make sure that **Microsoft Office** is installed on the same computer as the support tool will be, since the program utilizes Excel software as part of its operations. The user also needs to make sure that the **Oracle client runtime components** are installed on this computer. If CORSIM will be run, CORSIM should be installed on this computer as well; however, if the end user does not need to run CORSIM simulation, the CORSIM installation can be skipped.

MATLAB DLL Installation

The following are the steps to install the MATLAB DLL:

Step 1: Unzip the download MatlabDLL.zip to “c:\MatlabDLL” directory, which should contain four sub-folders: “close_figures”, “cluster”, “newseg”, and “spatial_conc_deliver”. There is another file “MCRInstaller.msi” that will be stored in the“c:\MatlabDLL” directory, which is the Matlab runtime library.

Step 2: Click on “MCRInstaller.msi” to install **Matlab runtime library**.

Step 3: Go to “close_figures\distrib”, right-click on “_install.bat”, and **check to make sure that the “path” environment variable is correctly set** (you will need to include the folder for the Matlab runtime library, i.e., C:\Program Files\MathWorks\MATLAB Component Runtime\v70\runtime\win32); if not, an error message will be displayed stating that

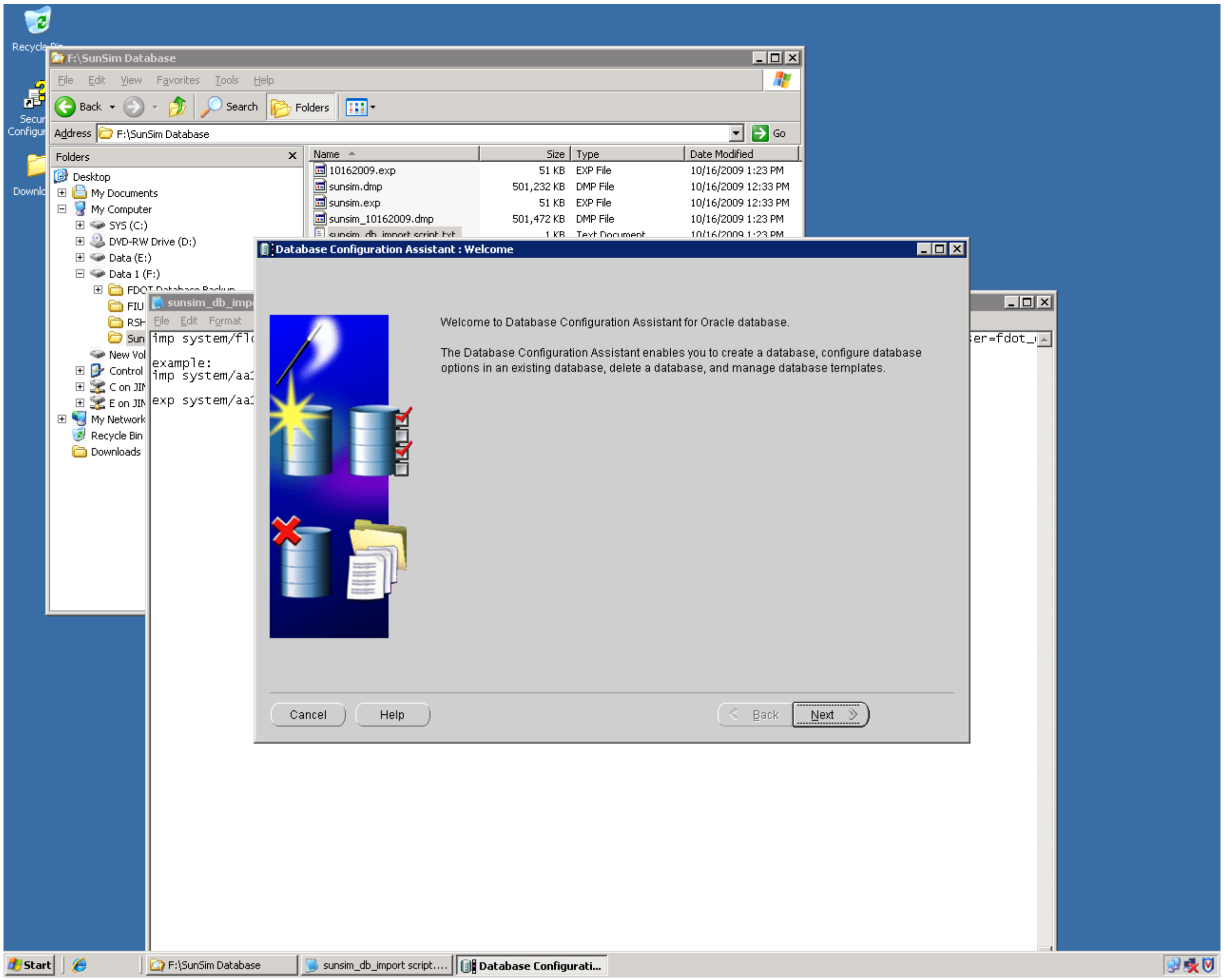
“LoadLibrary("<libraryname>.dll") failed - The specific module could not be found” when attempting to execute Step 4.

Step 4: Click on “_install.bat” again to register the DLL file (close_1_0.dll), and then do the same for “cluster_deliver”, “newseg”, and “spatial_conc_deliver”.

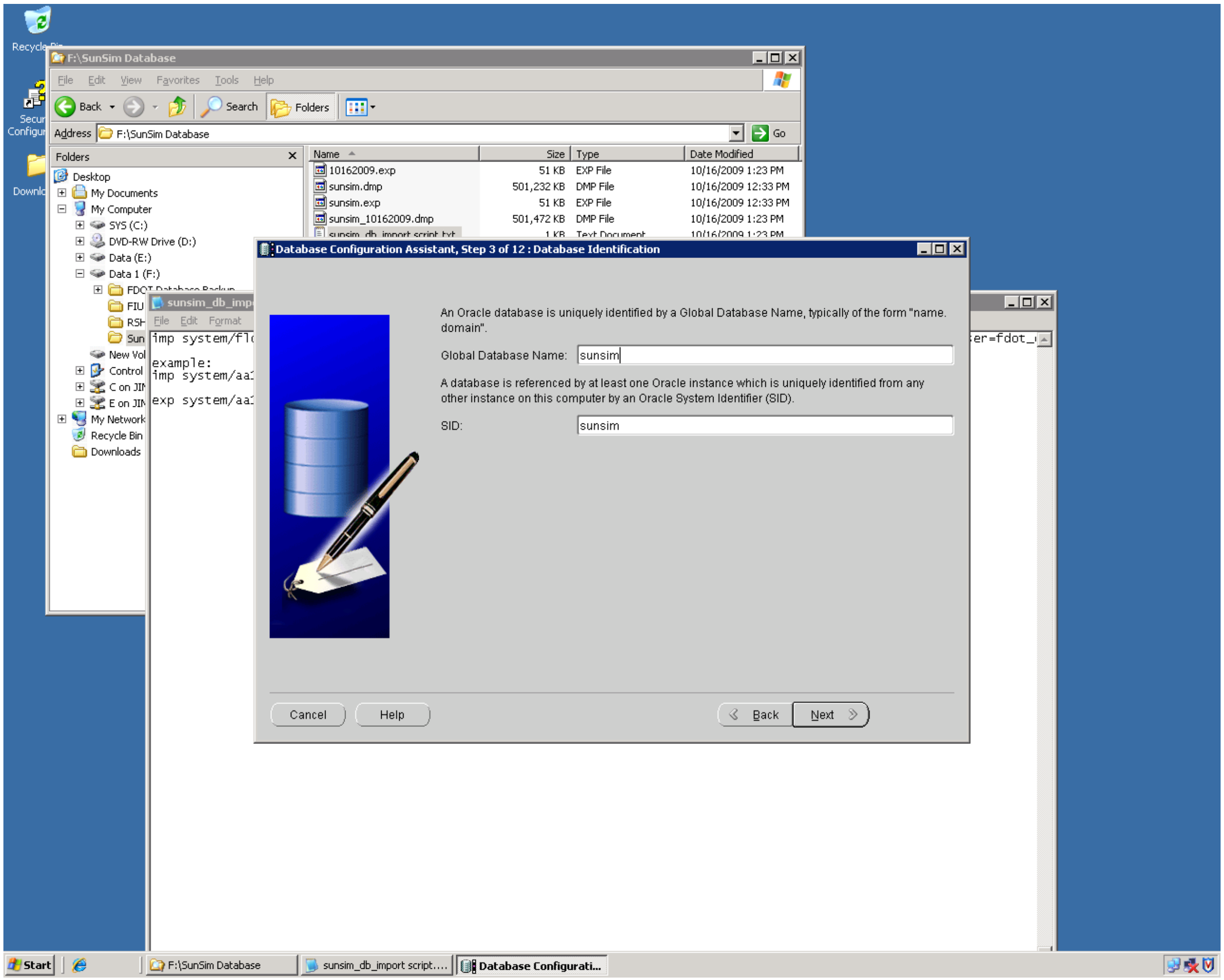
Oracle Database Preparation

The steps to prepare the Oracle database are given below.

Step 1: Make sure that the Oracle database program is installed on your server. Login to your server, choose “Start”->”All Programs”->”Oracle-OraDb10g_home1”->”Configuration and Migration Tools”->”Database Configuration Assistant” to generate the interface shown.



Step 2: Choose “Next”, choose “Create a Database”, choose “Next”, choose “General Purpose”, choose “Next”, choose a name for the database (e.g., sunsim), then “Next”, choose a password for “system” database account, then choose “Next” to create the database.

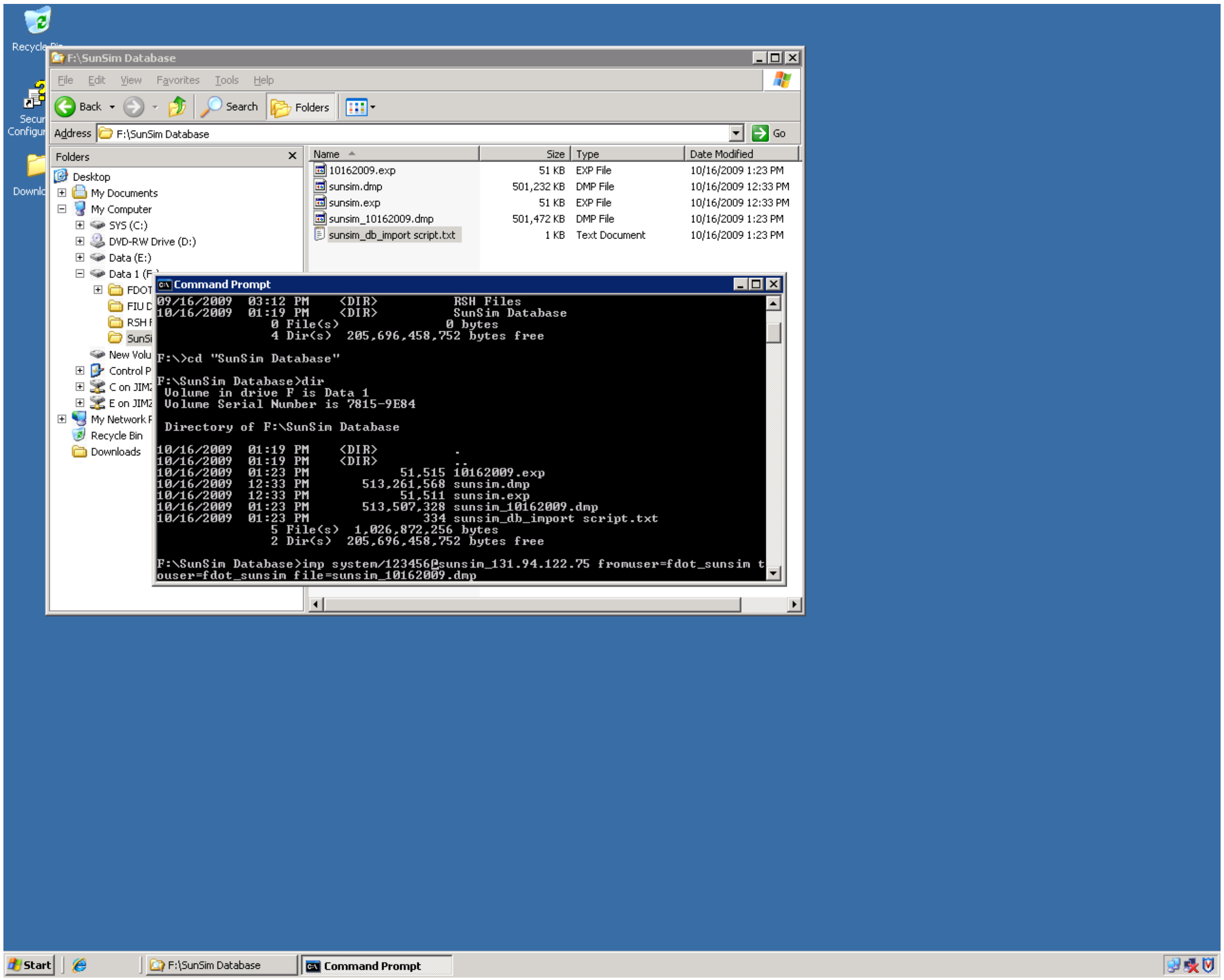


Step 3: Create a user “fdot_sunsim” for the newly created database and assign DBA privileges to the account (either use the web interface or the enterprise management console and other tools).

Step 4: Click “Start”->”Accessories”->”Command Prompt” to open a command prompt window. Enter “imp system/<password>@sunsim_<db server IP> fromuser=<desired username> touser=fdot_sunsim file=<oracle dump file you downloaded>” to import the database schema and data records

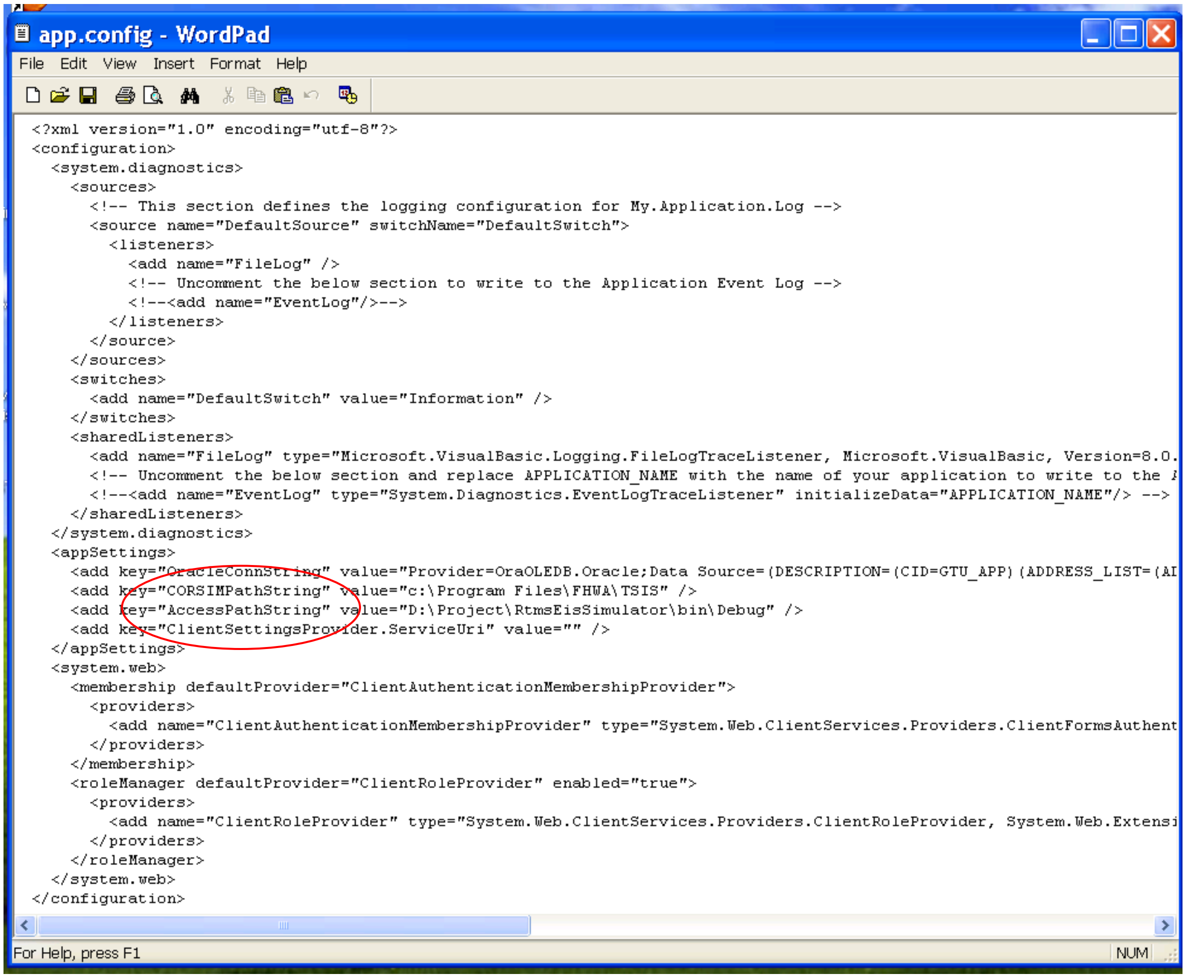
For example: imp system/123456@sunsim_131.94.122.75 fromuser=fdot_sunsim touser=fdot_sunsim file=sunsim.dmp

Make sure there is no error in the importing process. This completes the database schema generation and data record importing. An example database is included with the tool that can be loaded. This database contains several months of detector data (Oct. 2008 to Jan. 2009), which can be used for program testing and evaluation.



Interface Program Installation

Step 1: Modify the SunSim.exe.config file. Three values should be modified: "OracleConnString", "CORSIMPathString", and "AccessPathString". The first value is for the Oracle database connection and the second is for the CORSIM program location (for cases in which CORSIM is to be installed). The third value is for the folder that will store the generated TSS files (optional, if CORSIM simulation is configured).



app.config - WordPad

File Edit View Insert Format Help

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.diagnostics>
    <sources>
      <!-- This section defines the logging configuration for My.Application.Log -->
      <source name="DefaultSource" switchName="DefaultSwitch">
        <listeners>
          <add name="FileLog" />
          <!-- Uncomment the below section to write to the Application Event Log -->
          <!--<add name="EventLog"/>-->
        </listeners>
      </source>
    </sources>
    <switches>
      <add name="DefaultSwitch" value="Information" />
    </switches>
    <sharedListeners>
      <add name="FileLog" type="Microsoft.VisualBasic.Logging.FileLogTraceListener, Microsoft.VisualBasic, Version=8.0.
      <!-- Uncomment the below section and replace APPLICATION_NAME with the name of your application to write to the A
      <!--<add name="EventLog" type="System.Diagnostics.EventLogTraceListener" initializeData="APPLICATION_NAME"/> -->
    </sharedListeners>
  </system.diagnostics>
  <appSettings>
    <add key="OracleConnString" value="Provider=OraOLEDB.Oracle;Data Source=(DESCRIPTION=(CID=GTU_APP)(ADDRESS_LIST=(AI
    <add key="CORSIMPathString" value="c:\Program Files\FHWA\TSIS" />
    <add key="AccessPathString" value="D:\Project\RtmsEisSimulator\bin\Debug" />
    <add key="ClientSettingsProvider.ServiceUri" value="" />
  </appSettings>
  <system.web>
    <membership defaultProvider="ClientAuthenticationMembershipProvider">
      <providers>
        <add name="ClientAuthenticationMembershipProvider" type="System.Web.ClientServices.Providers.ClientFormsAuthent
      </providers>
    </membership>
    <roleManager defaultProvider="ClientRoleProvider" enabled="true">
      <providers>
        <add name="ClientRoleProvider" type="System.Web.ClientServices.Providers.ClientRoleProvider, System.Web.Extensi
      </providers>
    </roleManager>
  </system.web>
</configuration>
```

For Help, press F1 NUM

Step 2: Conclude the procedure by running the program.

Appendix B: User's Manual for the SunSim Core Simulation Support

This appendix includes the User Manual's for the SunSim program developed to allow the generation of inputs to simulation models based on ITS data. The application includes the following major modules that will be described in detail later in this document:

1) ***Data Preparation:*** Allows the user to check traffic data availability and whether the needed data are currently stored in the local Oracle database. If it is determined that the needed data are not available, the module provides an easy way to upload traffic data from the STEWARD data archive to the local database, allowing the performance of operations on different aggregation levels of the data.
2) ***Station Grouping:*** Allows the user to group available data stations together for pattern identification and other uses.
3) ***Pattern Identification:*** Clusters the traffic patterns from different days into different categories based on the similarities in traffic demands, and then saves the desired patterns for future use.
4) ***Data Segmentation:*** Automates the segmentation of time intervals into groups of similar intervals (traditionally this process is accomplished manually).
5) ***Spatial Conciliation:*** Estimates missing demands and resolves inconsistencies between upstream/downstream traffic counts.
6) ***Simulation:*** Automatically generates the desired CORSIM simulation files, runs the simulation, and then examines the simulation results.
7) ***Maintenance:*** Used for historical data clean up and database re-initialization; the user is expected to utilize this module on an as-needed basis.

B.1 Data Preparation

The different tasks performed by SunSim are conducted utilizing a local database. The database details are explained in the Local Database Schema in Appendix C of this document. The Data Preparation tab provides an interface for the user to check the data availability in the local Oracle database. If the required traffic data for a specific day are not available, this tab allows the user to upload STEWARD data files to the local database. Figure B.1 shows the interface for the data check function. To check if traffic data are available in the local database system, the user should select the desired corridor, specify the start and end dates, and then click on the "Data Check" button. If traffic data are missing for some of the days within the specified date range, a message indicating that traffic data are missing will be displayed in the right-hand box, as shown in Figure B.2. The user can then go to the STEWARD data archive website (http://cdwserver.ce.ufl.edu/steward/index.html) and download data for the missing days at five-minute aggregation intervals. These downloaded data files should be put in a specific folder for data uploading, as described below.

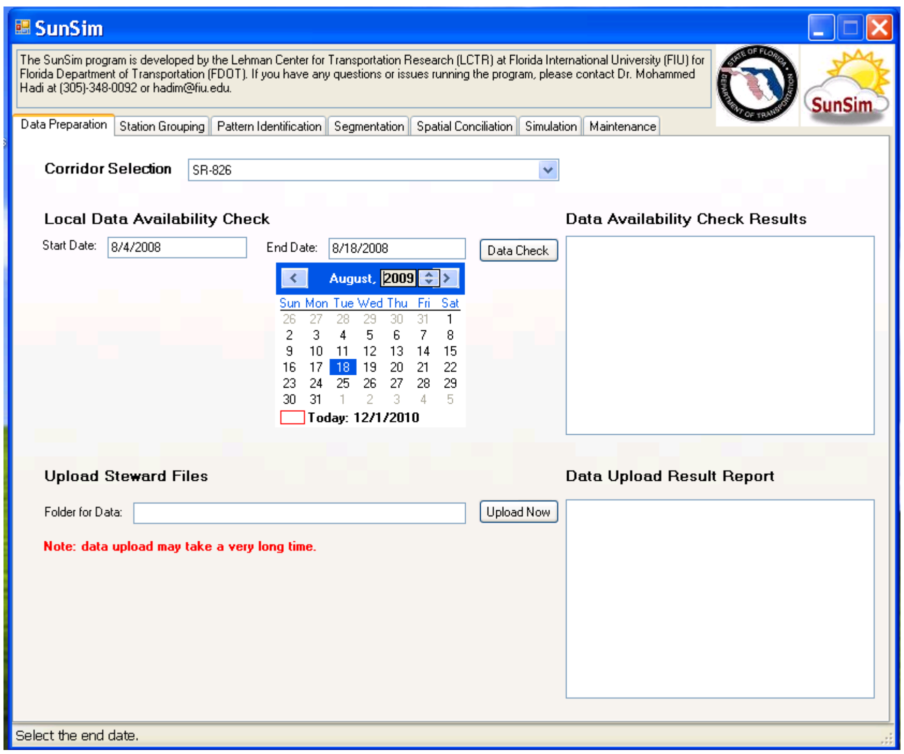


Figure B-1: User Interface for Data Availability Check

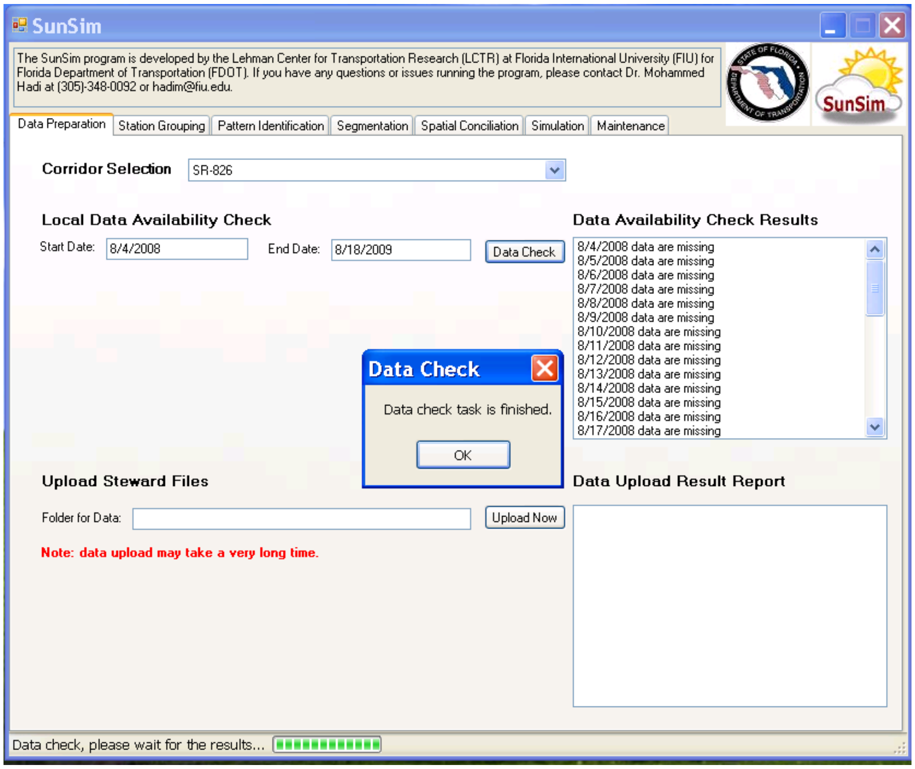


Figure B-2: User Interface for Data Availability Check Results

Figure B.3 shows the interface that allows the user to click on the data upload textbox to upload the STEWARD traffic data files saved on the hard drive to the Oracle database. A dialog box for folder selection permits the end user to upload multiple files in one click. The user can simply put all the downloaded data files under a specific folder, select the folder name, and click the "OK" button. The program will automatically identify which data files are new and which files have already been uploaded, and will only upload the new data files. Data upload results will be shown in the lower right listbox.

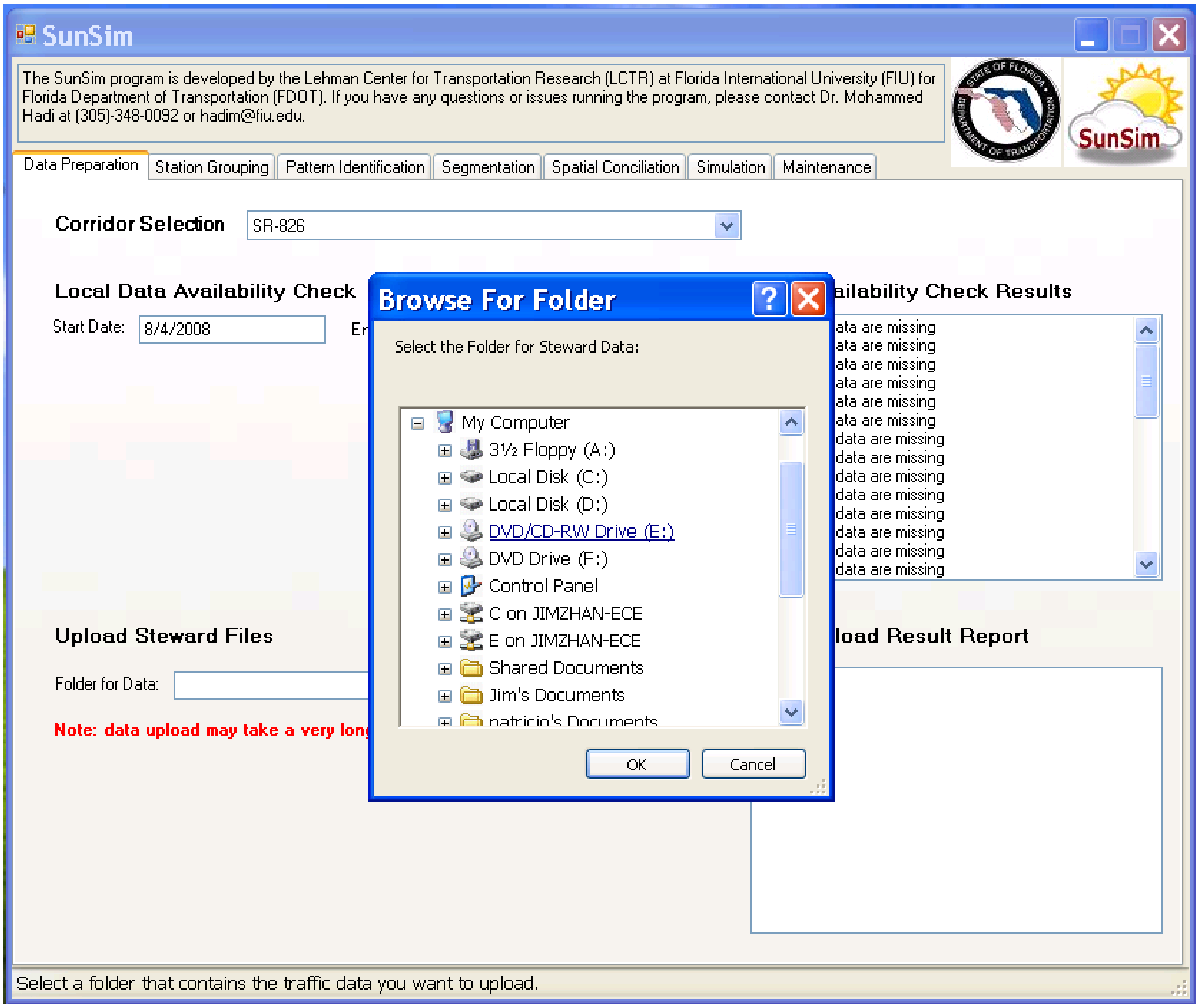


Figure B-3: User Interface for Data Upload

B.2. Station Grouping

The Station Grouping tab provides an interface for the user to organize detector data stations into groups that can then be used to identify the desired traffic patterns, utilizing a clustering algorithm. The clustering algorithm classifies traffic days from different days into clusters that represent different patterns occurring throughout the year. The data station grouping selected by the user must be saved in a local database as a base for further data processing.

To use this module, the user should first select a corridor to display all available detector data stations in the “Available Stations” list. The end user can then use the “>>” and “<<” buttons to select or deselect available data stations from the “Selected Stations” list. After providing a name for the group, the user can click on the “Save Station Group As” button to save the selected station list into a defined detector data station group. After performing the save operation, the group will be available from the “Select Group” dropdown menu for future use. The interface of the detection station selection (Data Selection tab) is shown in Figure B.4.

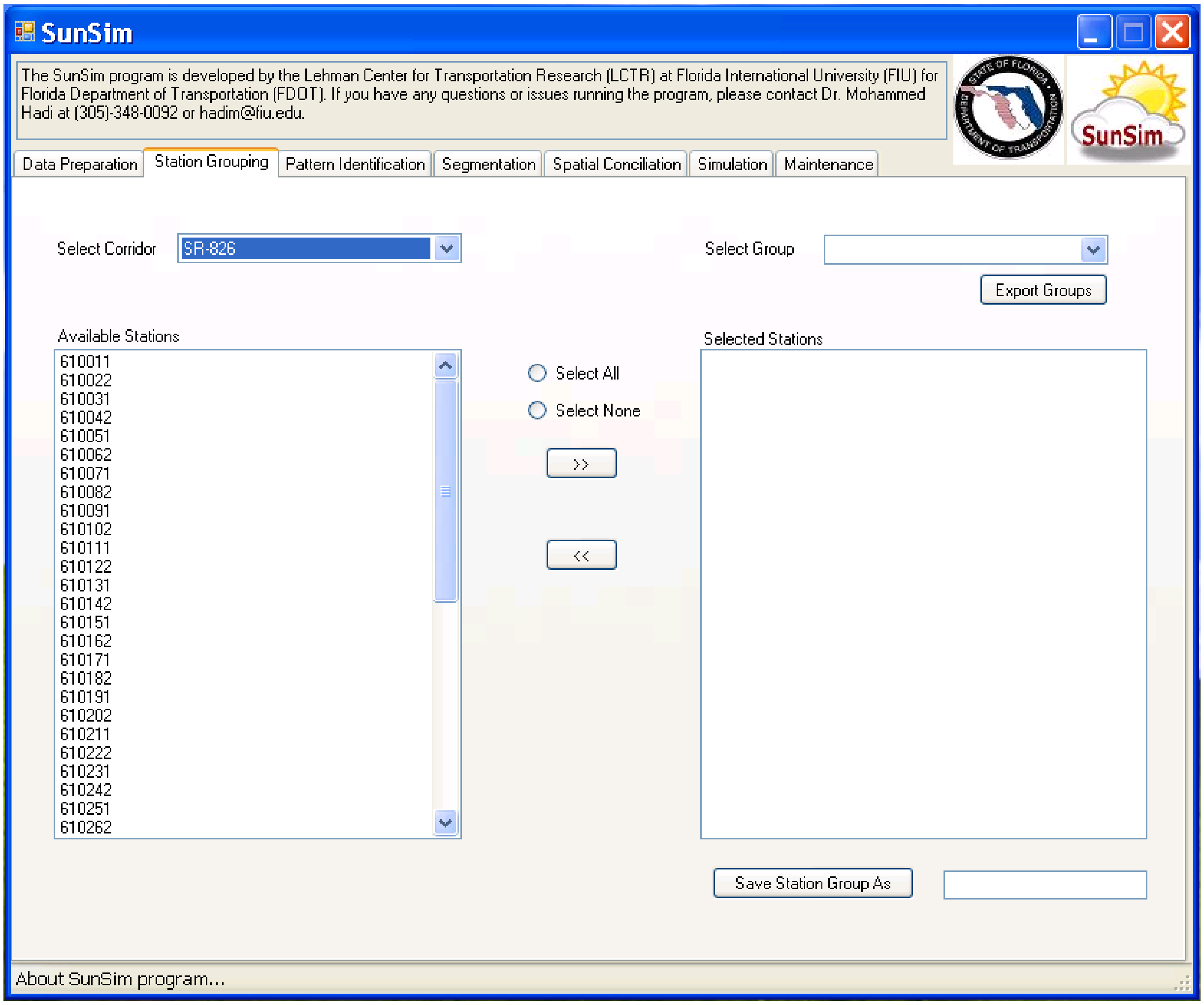


Figure B.4: User Interface for Detection Station Selection (Data Selection Tab)

Pattern Identification

The Pattern Identification tab is used to identify the desired traffic patterns using a data clustering algorithm. Figure B-5 shows the interface for selecting start and end dates for traffic data. After selecting the dates, the user should also choose the start and end times of day, the aggregation interval, and the predefined group name of the analysis in order to conduct the clustering task for pattern identification. The aggregation levels allowed are 5-minute, 10-minute, 15-minute, 30-minute, or one-hour intervals. The user should make sure that the traffic data for the selected dates are available in the local database using the Data Preparation module, as previously introduced.

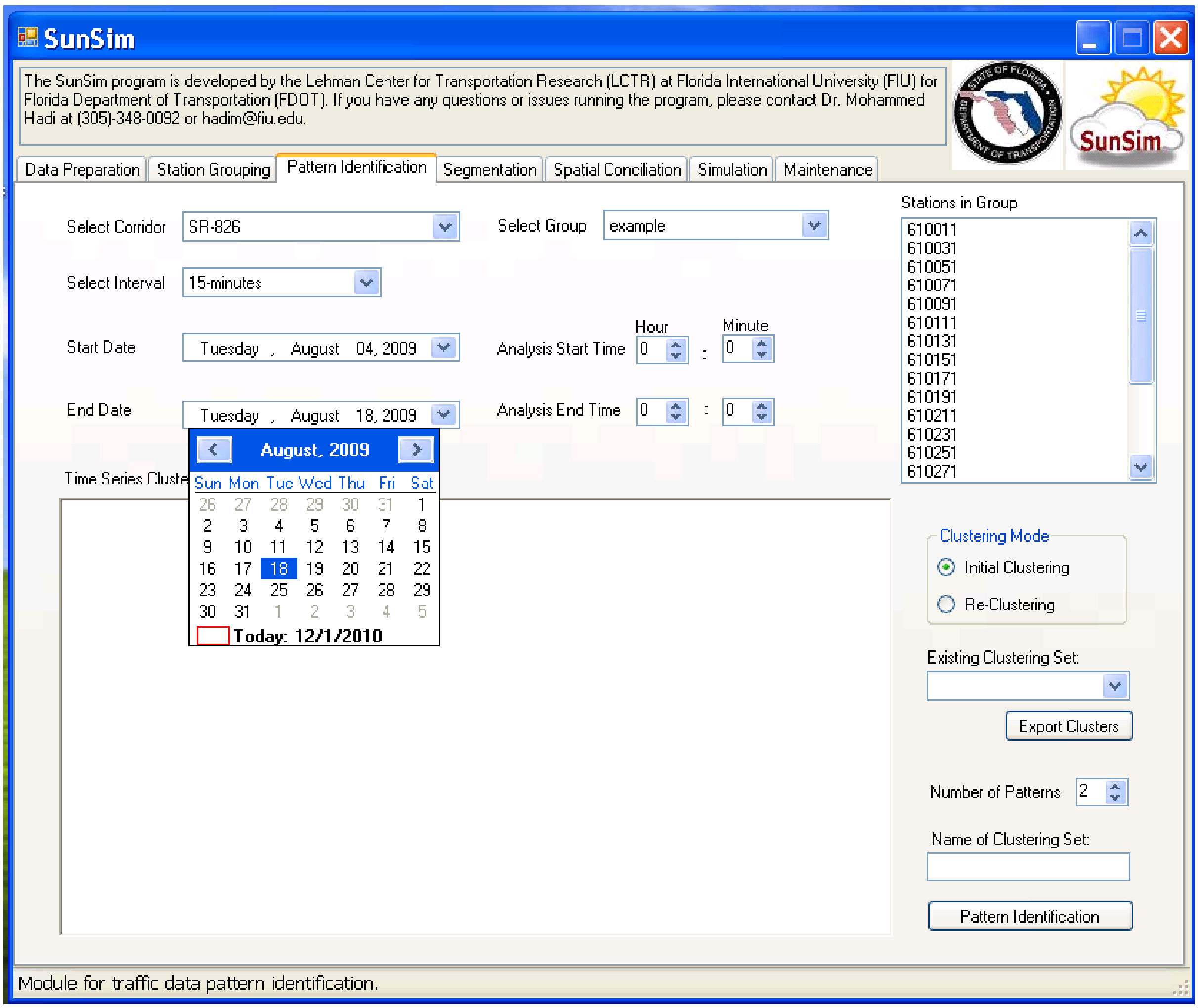


Figure B-5: User Interface for Date Selection of Traffic Data

After the user selects the start/end date and time, the aggregation interval, and the predefined detector data station group, the user can click the "Number of Patterns" control to assign the number of patterns desired, as shown in Figure B-6. The user can experiment with between two and ten different numbers of desired patterns and examine the results. In addition, the end user should select the clustering mode, as explained further on, and assign a name for the clustering result set which will be saved in the database. There are two modes for clustering, an initial clustering or a re-clustering for a previously saved cluster. For an initial clustering, a new clustering set name to associate with the clustering results should be provided. For re-clustering, the previously saved clustering name will be used unless the user wants to provide another name for the result set. The next step is for the user to click on the "Pattern Identification" button to run the clustering process using the uploaded traffic data.

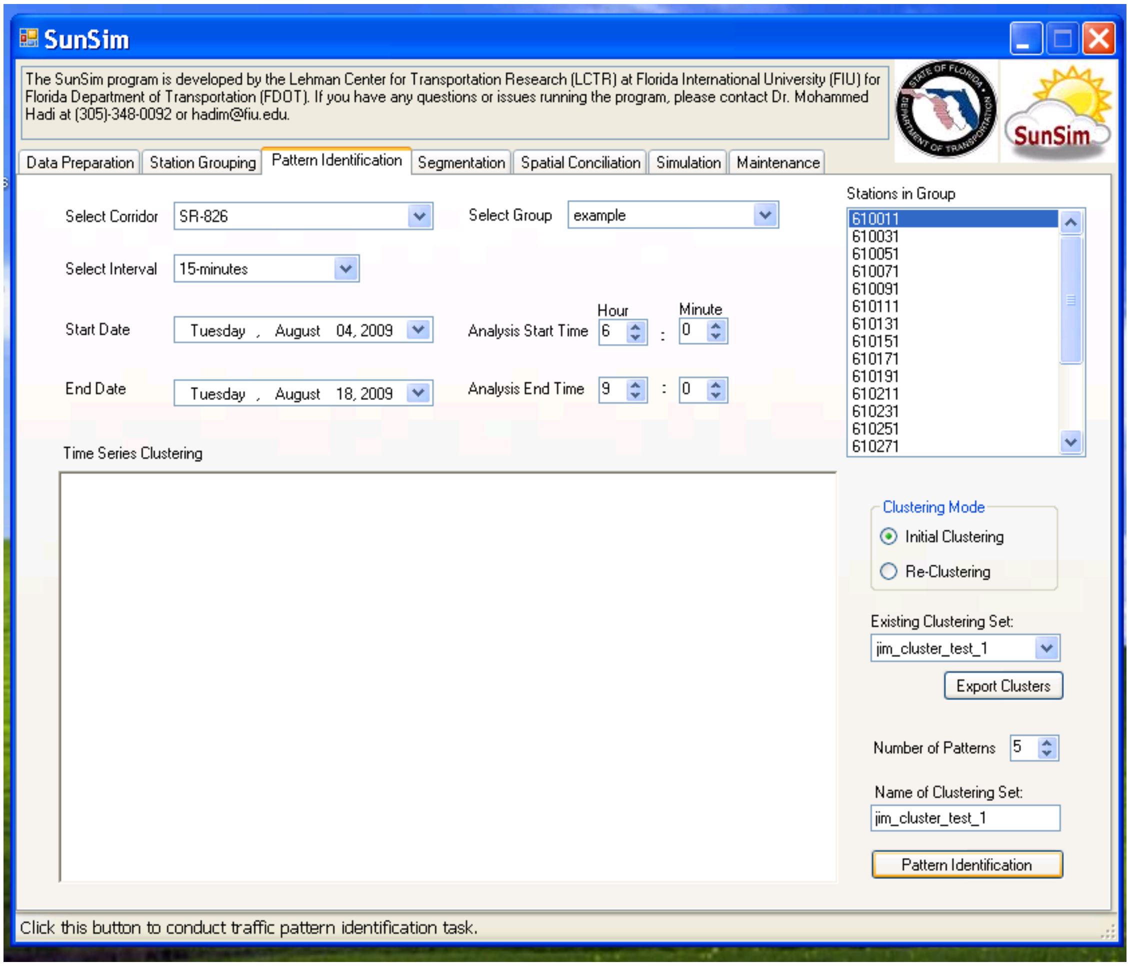


Figure B-6: User Interface for Pattern Identification

The identified patterns of the clustering results are now displayed. Figures B.7 through B.9 show examples of data clustering results; in these examples, there are five identified patterns displayed. The end user can visually compare these patterns and choose the pattern(s) desired to meet their criteria for further analysis (for example, the user can select the pattern that represents a typical day in the analysis). These patterns will be saved in the database along with the related date/time information. Further modules, such as data segmentation and spatial conciliation, will use the data clustering results to conduct time interval identification and demand estimation analyses, as described in the next sections.

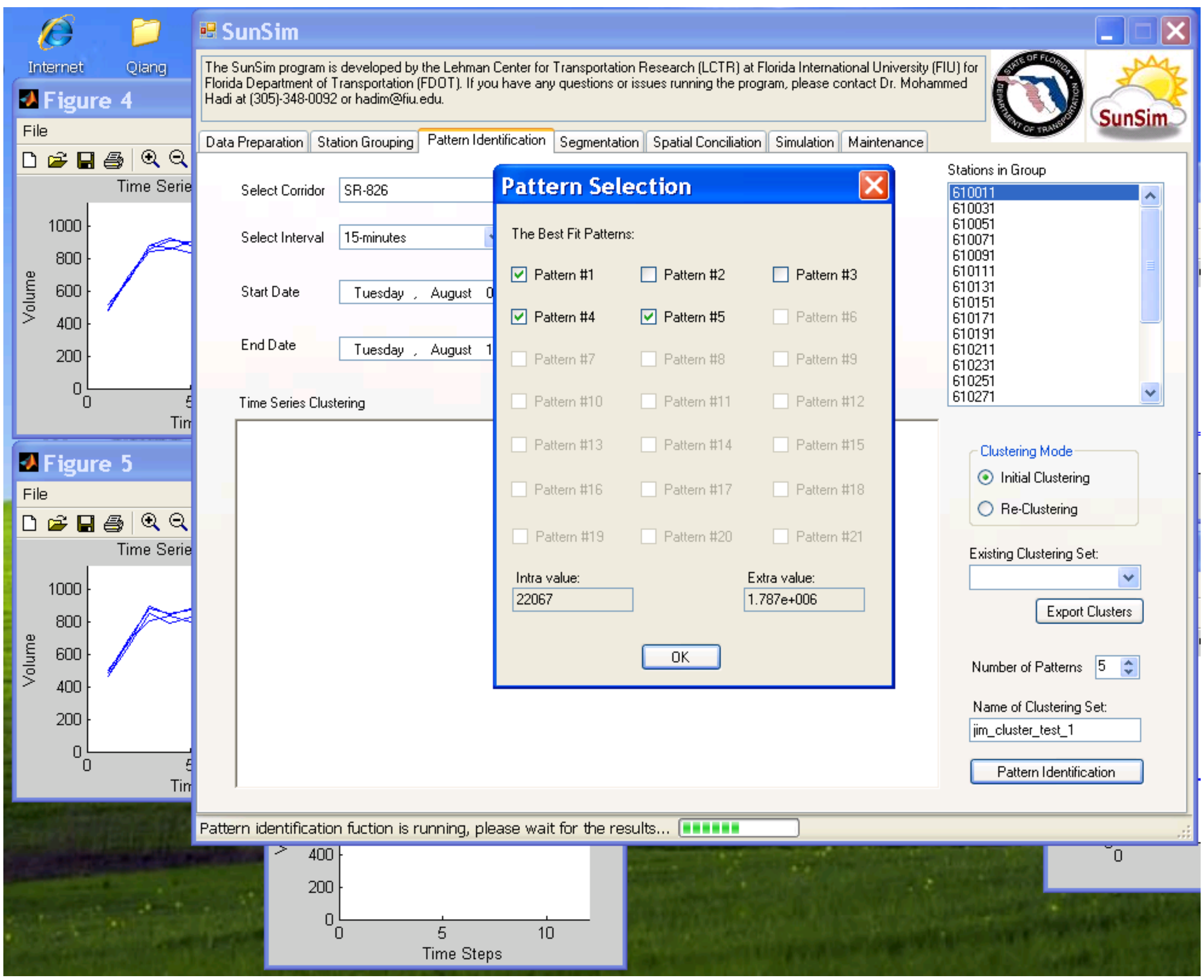


Figure B-7: User Interface for Data Clustering Results

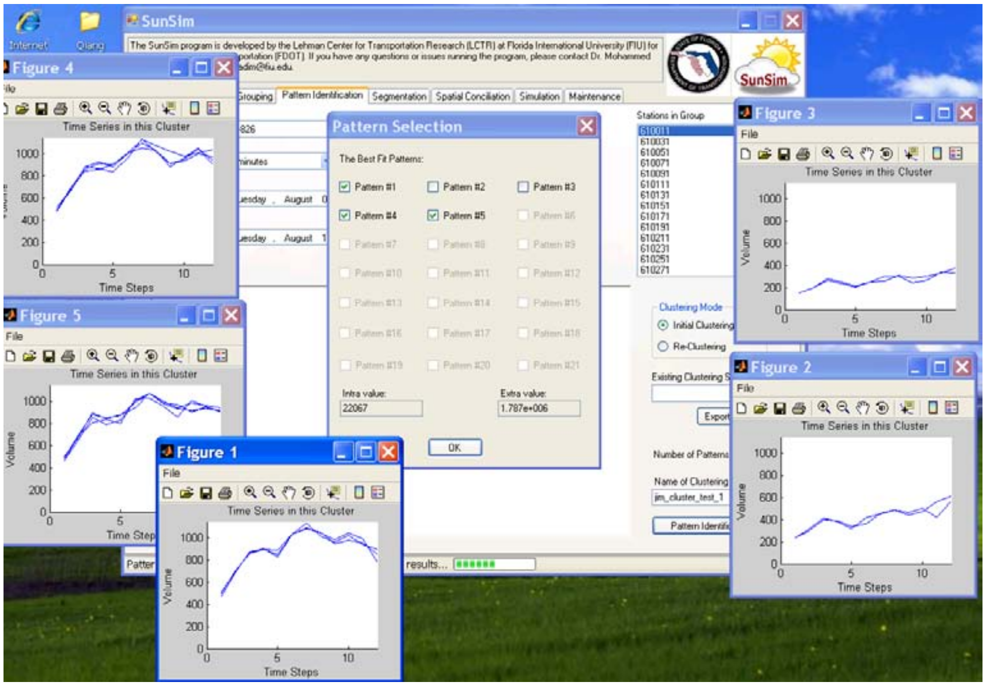


Figure B-8: User Interface for Data Clustering Results

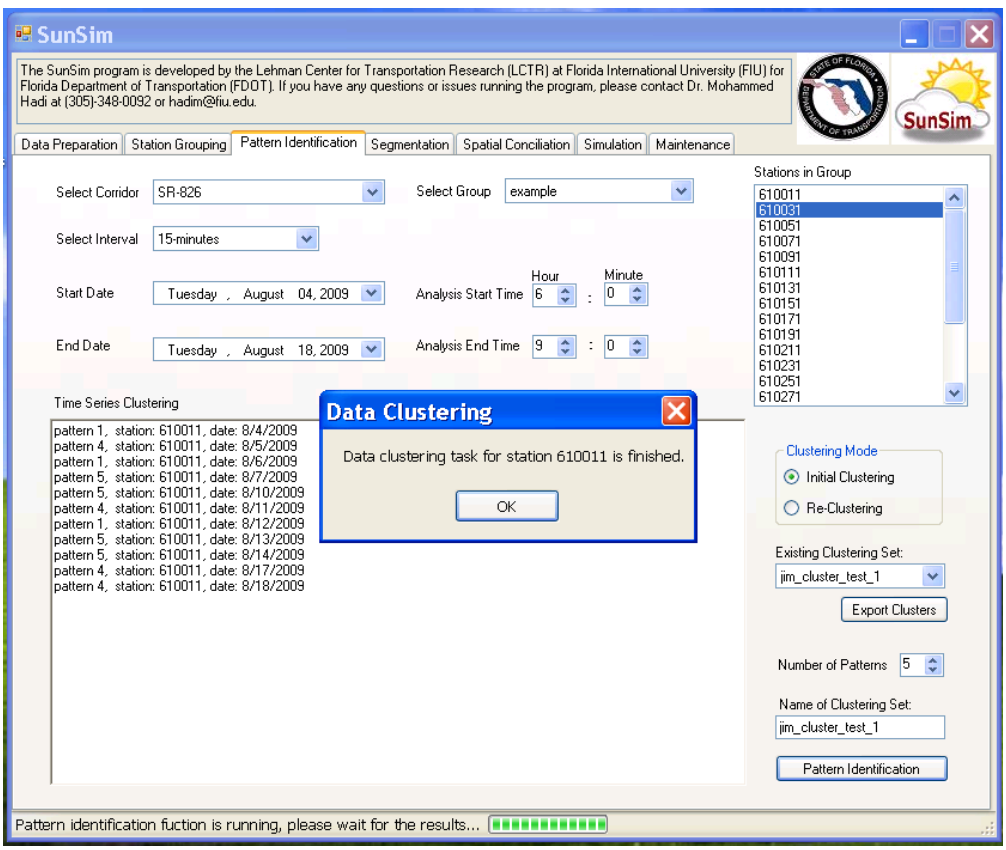


Figure B-9: User Interface for Data Clustering Results (cont.)

B.3 Data Segmentation

Microscopic simulation generally requires segmenting the simulation period into discrete time intervals since traffic characteristics may be time-variant. Traditionally, analysts make this decision manually (e.g., 15-minute, 30-minute, or 1-hour intervals during the peak hours). The data segmentation module was developed to automate the segmentation of the study time period into multiple time intervals.

Figures B.10 and B.11 show the data segmentation module interface. This module uses the data clustering module results, as described in the previous section. After a user selects a desired clustering result set, the detector data stations will be shown in the “Stations for Segmentation” list, and the user can select one or multiple stations from the list for inclusion in the segmentation task. The segmentation is performed using data obtained from only the selected stations. The data segmentation results are displayed as a figure on the end user’s screen, as shown in Figure B-11. The user should save the segmentation results for usage in subsequent steps (the spatial conciliation and the CORSIM simulation file generation tasks). By saving the segmentation results, the number of time intervals, the user-specified length of each time interval, and the associated start/end time of each interval will be saved for spatial conciliation and CORSIM simulation file generation.

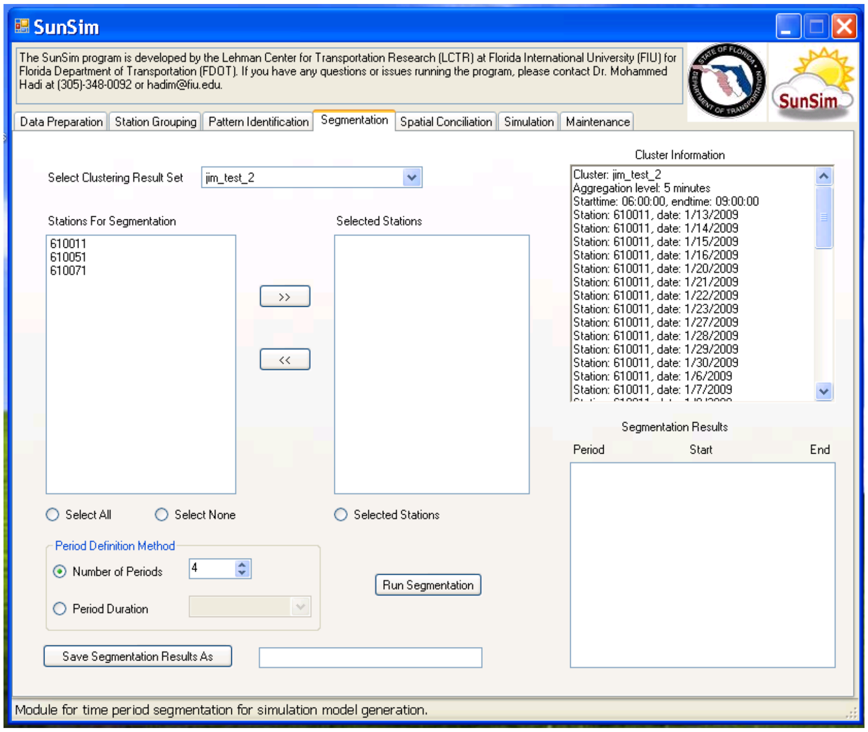


Figure B-10: User Interface for Data Segmentation

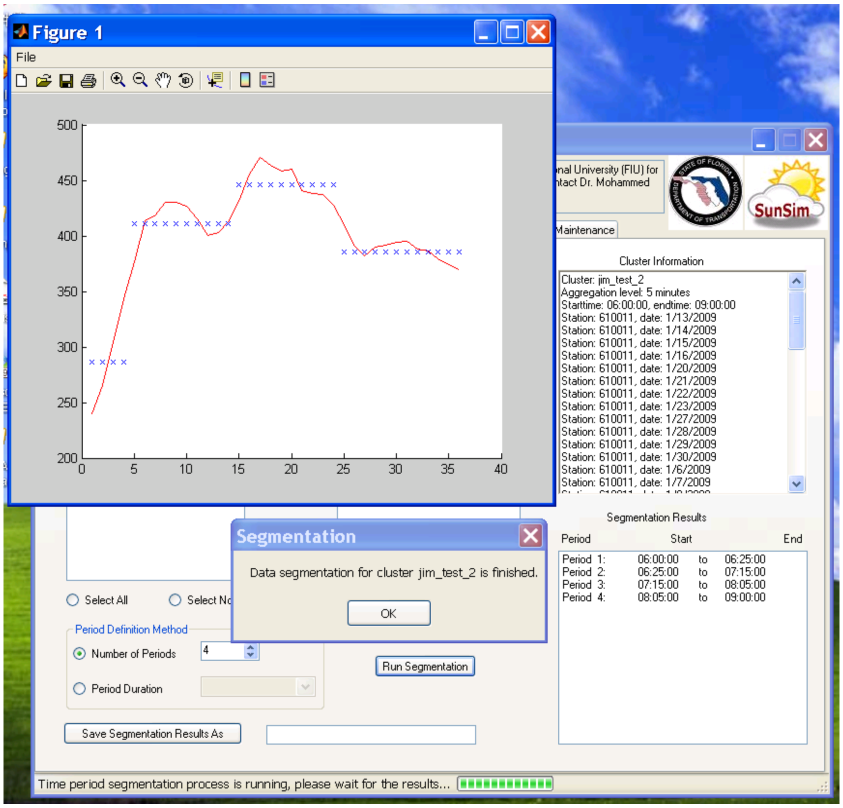


Figure B-11: User Interface for Data Segmentation (cont.)

B.4 Spatial Conciliation

The Spatial Conciliation module is used to estimate missing demands and correct the inconsistencies and unbalanced volume data between adjacent network nodes. The user interface of the module is shown in Figure B-12, with a sample result screen shown in Figure B-13. The reconciled traffic data can be used for CORSIM simulation file generation.

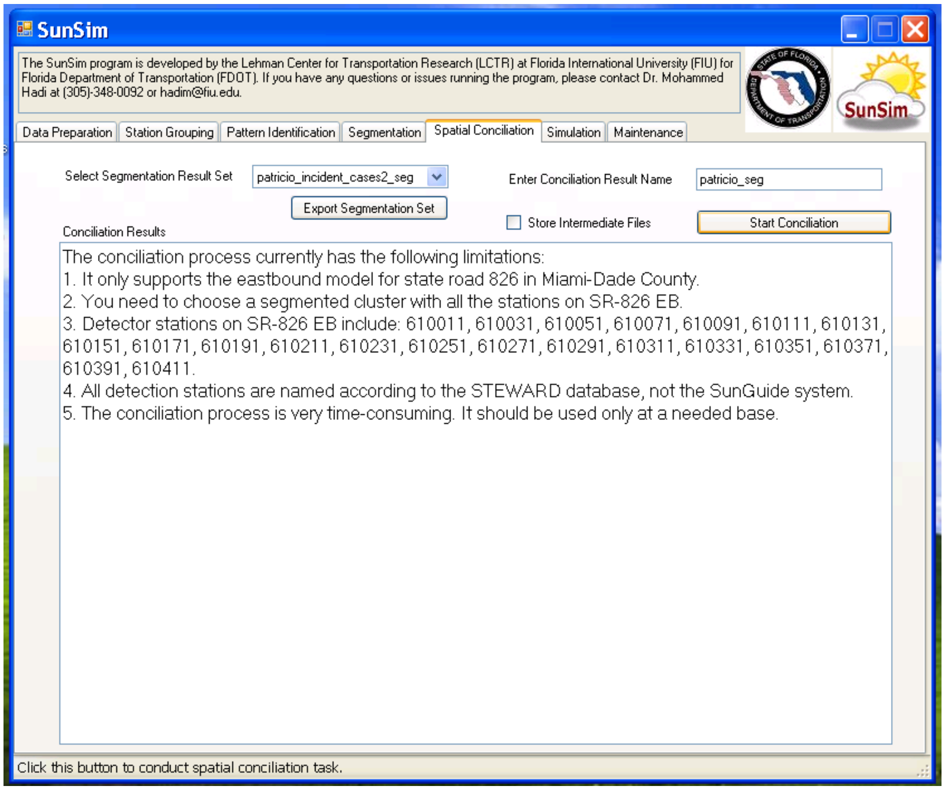


Figure B-12: User Interface for Spatial Conciliation

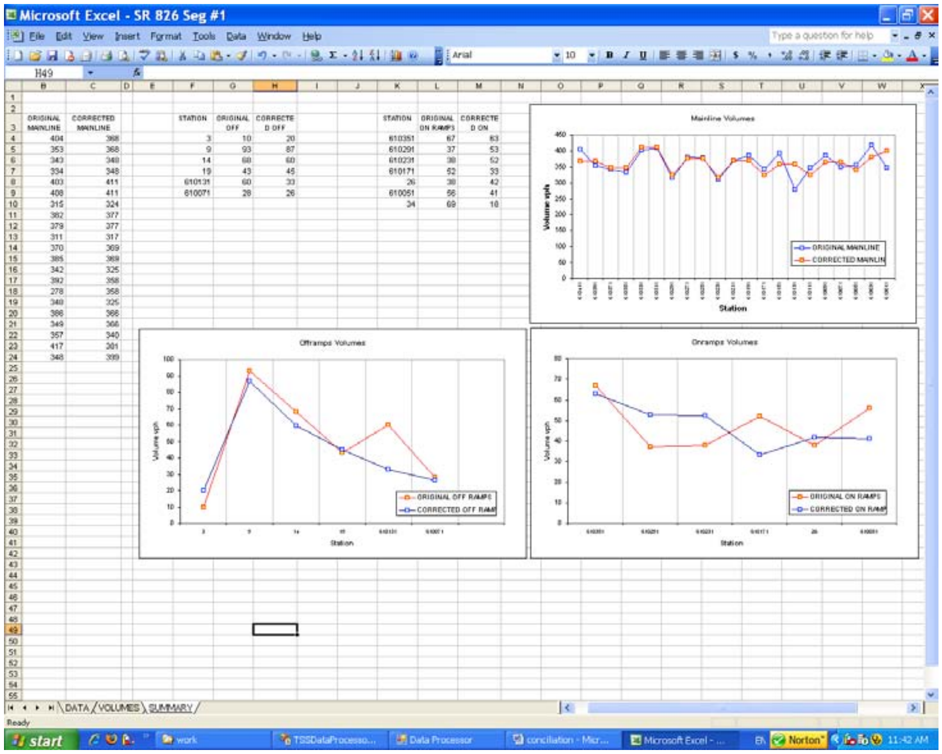


Figure B-13: An Example Result Set for Spatial Conciliation

B.5 Simulation

This module is used to generate the desired CORSIM simulation input file and run the simulation. The module will use a pre-coded network for geometric settings and other parameters (except demands). It then applies the results from the data segmentation step for the time interval setting and the spatial conciliation results for traffic demand inputs to generate a new CORSIM simulation file. After a CORSIM simulation file is generated, the system provides the capability to run CORSIM simulation, extract simulation results, and view the animation, all under one interface. In addition, the module is able to generate TSS-like data files from CORSIM outputs; these files can then be directly fed into the SunGuide system, using the simulator program developed in this project and described previously. The interface is shown in Figure B-14.

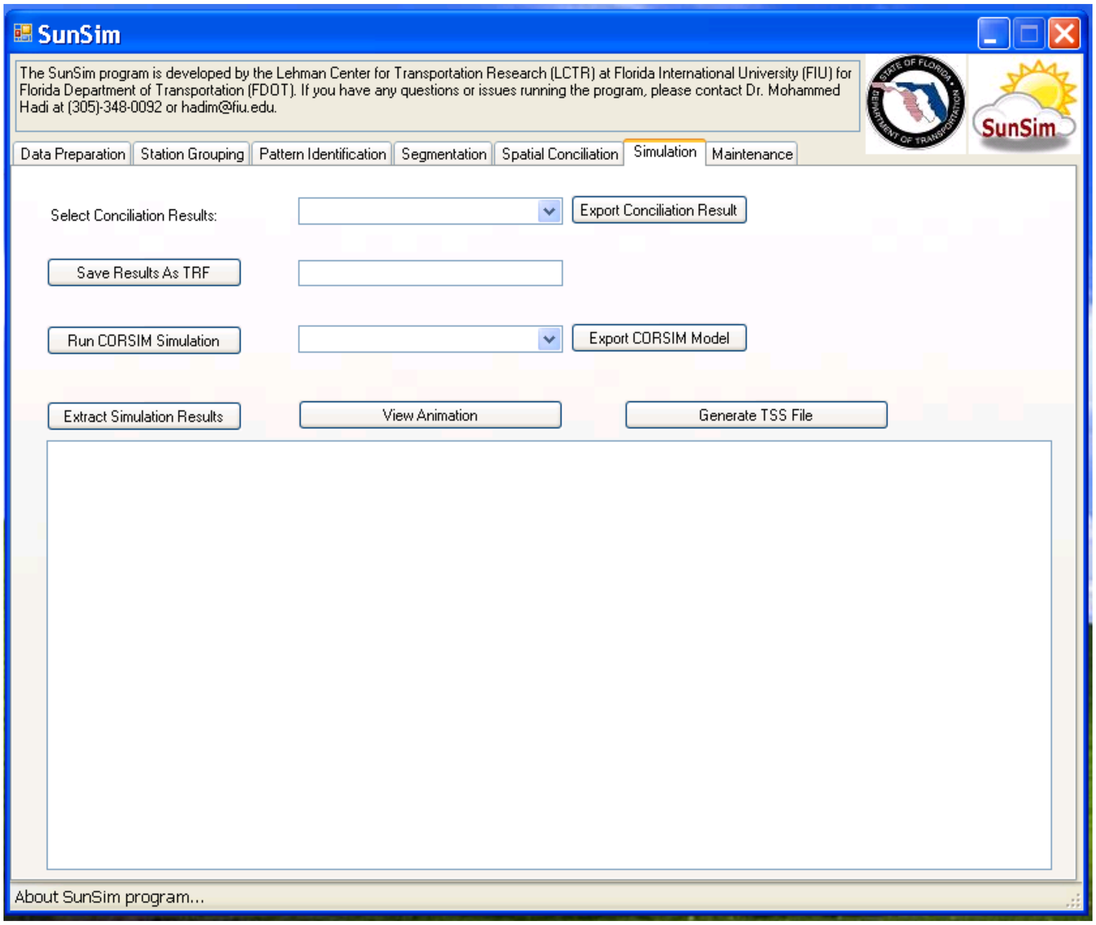


Figure B-14: User Interface for the Simulation Module

B.6 Maintenance

The Maintenance module is used to delete historical and temporary data in order to "clean up" the system. The interface is shown in Figure B-15. The users can view the historical data records from the dropdown menu and select the ones desired for removal. In addition, the module has the ability to restore previously deleted records and permanently delete data records that are no longer needed.

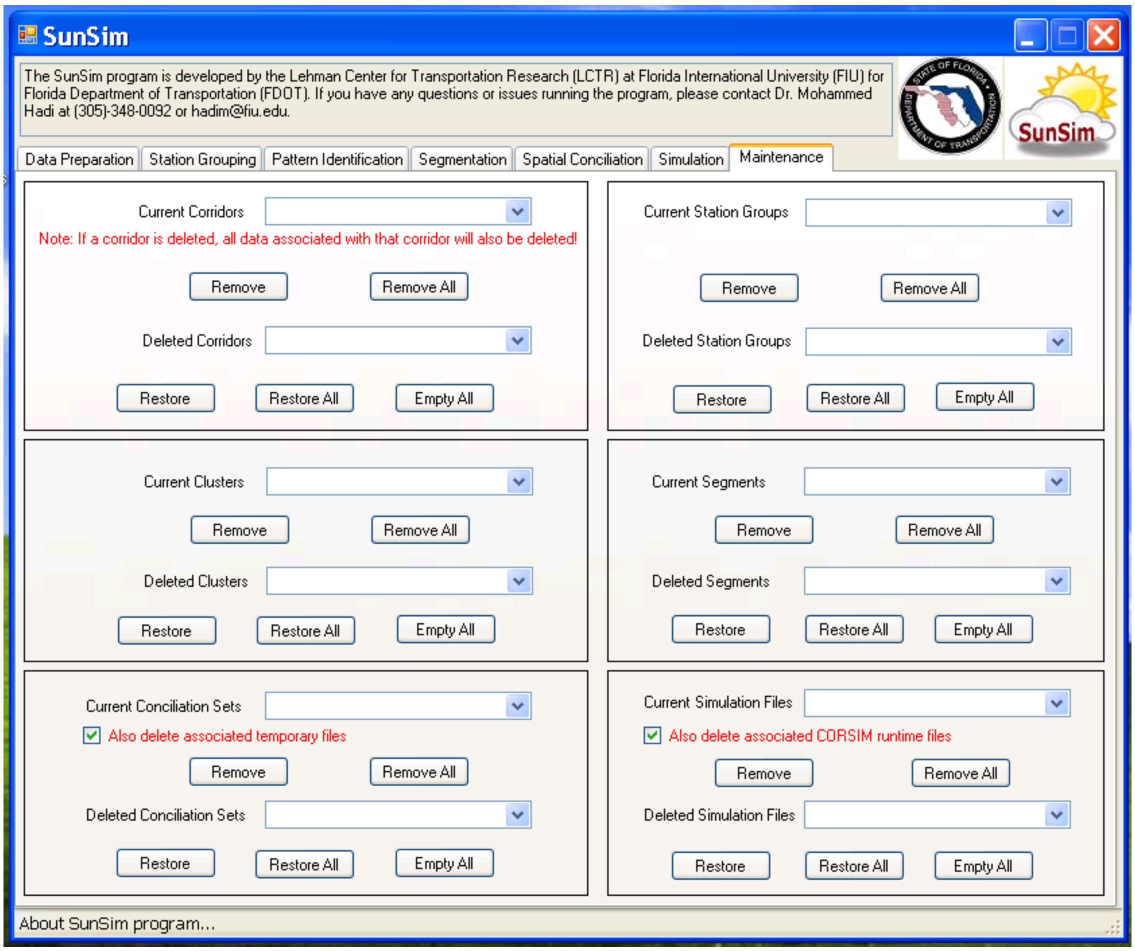


Figure B-15: User Interface for the Maintenance Module

Appendix C: Data Schema of the Core Simulation Support Tool

As described in Appendix B, the TSS detector data are downloaded from the STEWARD data archive website and stored in a back-end local Oracle database. Although the STEWARD data archive provides data at different aggregation levels, it does not provide a mechanism for the application developed in this study to directly connect it for the manipulation of data; a local Oracle database must therefore be created for this application.

To ensure data integrity, TSS detector data are downloaded from the STEWARD data archive at the 5-minute level for an entire 24-hour day. Data that are not in 5-minute intervals or for 24 hours are automatically rejected by the application. Valid data are then stored in a table called “STEWARDTable”. The schema of table “STEWARDTable” is shown in Table C-1. Since all the data records are in 5-minute intervals by default, only the start time information of a record is stored in the “Time” field. Each data record can store the volume/speed/occupancy information for up to six mainline lanes, three on-ramp lanes, and three off-ramp lanes, which should be able to cover all cases for Florida’s freeway facilities. In order to facilitate the quick retrieval of data records, indexes are also generated for date fields “DateString”, “Time”, and “Station_ID”, since they are most likely to be in the searching rules for traffic data.

Table C-1: Data Fields for Table "STEWARDTable"

Field Name	Date Type	Size	Explanation
DateString	Varchar2	10	Original date information for traffic record
Time	Varchar2	8	Original time information for traffic record
Station_ID	Varchar2	6	Steward data station number
Station_Desc	Varchar2	50	Text description of data station location
Station_MP	Number	8(3)	Station milepost information
Lane1_vol	Number	10(0)	Volume for mainline lane 1 for a 5-min period
Lane1_spd	Number	10(0)	Average speed for mainline lane 1
Lane1_occ	Number	10(3)	Average occupancy for mainline lane 1
…	…	…	Volume/speed/occupancy for lane 2-5
Lane6_vol	Number	10(0)	Volume for lane 6 (if exists, otherwise, null)
Lane6_spd	Number	10(0)	Speed for lane 6
Lane6_occ	Number	10(3)	Occupancy for lane 6
Onramp1_vol	Number	10(0)	Volume for on-ramp lane 1 (if any)
Onramp1_spd	Number	10(0)	Speed for on-ramp lane 1
Onramp1_occ	Number	10(3)	Occupancy for on-ramp lane 1
…	…	…	Volume/speed/occupancy for on-ramp lane 2
…	…	…	Volume/speed/occupancy for on-ramp lane 3
Offramp1_vol	Number	10(0)	Volume for off-ramp lane 1 (if any)
Offramp1_spd	Number	10(0)	Speed for off-ramp lane 1
Offramp1_occ	Number	10(3)	Occupancy for off-ramp lane 1
…	…	…	Volume/speed/occupancy for off-ramp lane 2
…	…	…	Volume/speed/occupancy for off-ramp lane 3
Facility	Varchar2	15	Corridor name for traffic record

Since the "STEWARDTable" will contain the actual daily traffic records and will have an extremely large size, a separate table "UploadRecord" is used to keep a quick record of the days for which data are uploaded. By using this table, the application can quickly identify if data for a selected date is available and relay this information to the end users. The schema of table "UploadRecord" is shown in Table C-2.

Table C-2: Data Fields for Table "UploadRecord"

Field Name	Date Type	Size	Explanation
District	Varchar2	2	District information for an upload record
Facility	Varchar2	15	Corridor name for an upload record
Startdate	Varchar2	10	Start date of an upload record
Enddate	Varchar2	10	End date of an upload record
Dayofweek	Varchar2	15	Day of week selection style for an upload record (e.g., "all days", "Mondays", "weekdays", "weekends", etc)
Direction	Varchar2	2	Directional information for an upload record

The table "CorridorTable" is used to keep track of the corridors that have data in the system. This table will be automatically updated when an end user uploads data from the STEWARD data archive. The application will compare the corridor information with that stored in the table "CorridorTable". If a new corridor is found, it will be automatically added to "CorridorTable". The schema of "CorridorTable" is shown in Table C-3. It should be noted that the "Status" field is added for the record deletion and restoration feature. A value of zero (status = 0) indicates a normal record and a value of 1 (status = 1) indicates a previously deleted record. Although the deleted record is still stored in the database, a user will not see it during standard operations. A user can use the maintenance function of the program to restore a previously deleted record, or choose to permanently delete it from the system later.

Table C-3: Data Fields for Table "CorridorTable"

Field Name	Date Type	Size	Explanation
CorridorID	Number	10	Auto increment ID number for a corridor
Name	Varchar2	50	Full name for a corridor
Status	Number	10	Status of the record

Tables "GroupTable" and "GroupStationTable" are created to facilitate the data clustering process (traffic pattern identification). A user may create multiple detector data station groups and associate those groups with appropriate data stations (detectors). Later on, traffic patterns can be classified and identified using data from those data stations. User selected cluster information, which is the date and time information of the desired traffic data, is stored in the "ClusterTable" table. Tables C.4 through C.6 show the data schema for tables "GroupTable", "GroupStationTable", and "ClusterTable". Again, the "Status" field is added to indicate the status of a record. A value of zero (status = 0) indicates a normal record and a value of 1 (status = 1) indicates a previously deleted record.

Table C-4: Data Fields for Table "GroupTable"

Field Name	**Date Type**	**Size**	**Explanation**
GroupID	Number	10	Auto increment ID number for a user defined group
GroupName	Varchar2	50	User defined name for a group
CorridorID	Number	10	ID of the corridor this group belongs to (foreign key, should exist in "CorridorTable")
Status	Number	10	Status of the record

Table C-5: Data Fields for Table "GroupStationTable"

Field Name	**Date Type**	**Size**	**Explanation**
GroupID	Number	10	ID of the group (foreign key, should exist in "GroupTable")
StationName	Number	6	STEWARD data station ID
Status	Number	10	Status of the record

Table C-6: Data Fields for Table "ClusterTable"

Field Name	Date Type	Size	Explanation
ClusterName	Varchar2	50	User defined cluster name
StationID	Varchar2	6	STEWARD data station ID
Pattern	Number	10(0)	Traffic pattern number
DateString	Varchar2	10	Date of the data record
Interval	Number	10(0)	Aggregation level, minimum: 5-minute
StartTime	Varchar2	8	Start time of selected traffic data
EndTime	Varchar2	8	End time of selected traffic data
Status	Number	10	Status of the record

Appendix D: User Manual for RTMS and AVI/LPR Simulators

D.1 Pre-Requisites to Run the Simulators

The pre-requisites to run the simulators are as follows:

1. The user must make sure that the Oracle client run-time libraries are installed on the computer that will run the simulators, as the program will need to directly retrieve detector configuration data from the SunGuide database.

2. The user must then check that the virtual RTMS and/or AVI/LPR detectors which are coded in the simulation are also coded in the SunGuide system, with the IP address pointing to the computer that will run the simulators, and check that the assigned port numbers are not conflicting with any other programs as well (as a general rule, port numbers below 255 are reserved for system usage).

D.2 CORSIM and RTMS/AVI/LPR Detector Mapping:

If CORSIM output will be used as inputs to the detector simulators, the user needs to make sure that the output files are stored on the computer and that there is a Microsoft Access database file "DetectorMapping.mdb" under the same folder of the program. This Access database should contain three tables: "CORSIMOutput", "TSSOutput", and "Detector". The "CORSIMOutput" table is updated by the RTMS or AVI/LPR simulator automatically and the end users should not modify it. The "TSSOutput" table is used to support the import of historical TSS outputs and feed the data back to the SunGuide system for testing purposes. The end user is not expected to directly modify the contents of the "TSSOutput" table. If historical TSS output data will be used to feed into the simulator, no CORSIM and SunGuide mapping information is needed and the simulator will directly match the detector name in the "TSSOutput" table to the SunGuide detector names. On the other hand, before an end user can feed CORSIM outputs to the SunGuide system, the system administrator needs to check that the "Detector" table contains the mapping information between SunGuide detector ID and CORSIM simulation network virtual detector ID. The data fields for the table "Detector" are listed in Table D-1. As can be seen, there are three IDs: "StewardID", "CORSIMID", and "SunGuideID". The "StewardID" is optional and corresponds to the detector name in the STEWARD database; the “SunGuideID” corresponds to the name specified by FDOT personnel in the SunGuide system; and the “CORSIMID” is the ID coded in the CORSIM simulation network. In addition, there are two other fields: “NumOfLanes” specifies the number of lanes at the detector site and “WithRamp” specifies if there is an on/off-ramp at the detector site.

Table D-1: Data Fields for Table “Detector”

Field Name	**Date Type**	**Size**	**Explanation**
StewardID	Number	10	Detector ID in Steward database (optional)
CORSIMID	Number	10	Detector ID in CORSIM simulation network
SunGuideID	Text	50	Detector ID in SunGuide System
NumOfLanes	Number	10	Number of freeway lanes at the site
WithRamp	Boolean	Yes/No	If the detector site has an on/off ramp

D.3 Running the RTMS Simulator

The major interface of the RTMS simulator program is shown in Figure D-1. There are two menu items that an end user can select from, “File” or “Help”. The “File” menu provides the items that are available for the simulator program and the “Help” item provides license and copyright information.

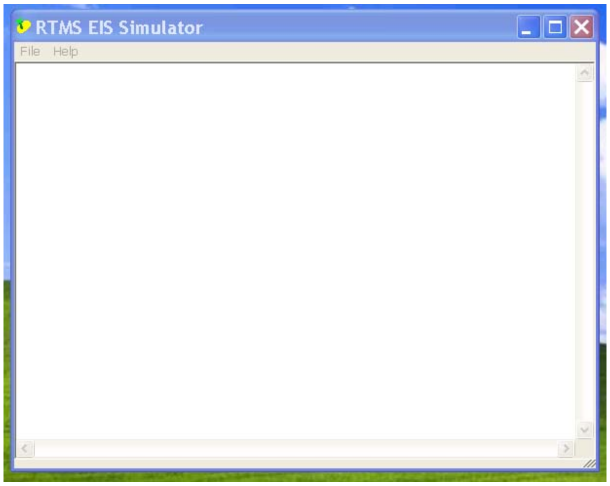


Figure D-1: Startup Interface for RTMS Simulator

When an end user clicks on the "File" menu, six possible menu items are displayed, as shown in Figure D-2. A user can choose to "Start Server", "Stop Server", "Configure Data", "Debug Logging", "Clear Display", or "Exit".

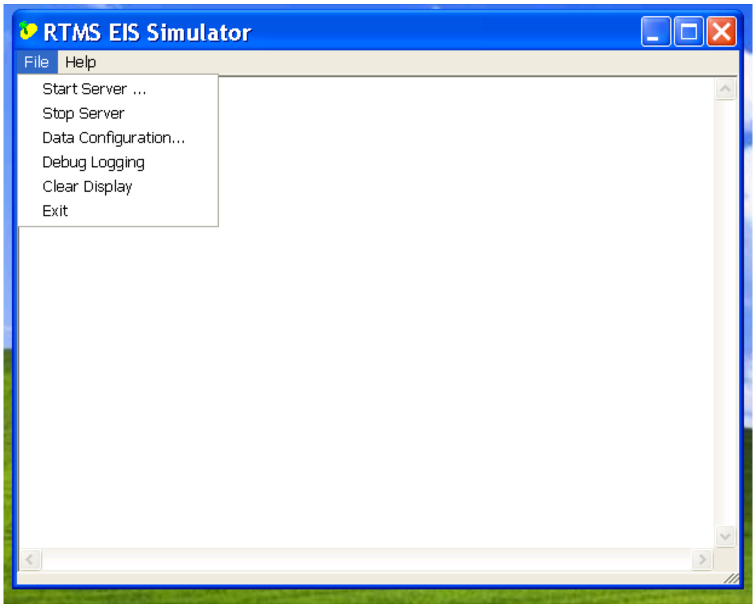


Figure D-2: RTMS Simulator Interface

When clicking on "Start Server", the pop-up window shown in Figure D-3 is displayed. This display includes a list of the available SunGuide RTMS detectors, which are specified in the SunGuide system using the Administrators interface, as shown in Figure D-4. The operators should specify "EIS" as the protocol type, and enter the IP address of the computer running the RTMS simulator program as the "Port Server IP". The RTMS simulator program will automatically retrieve the information for all RTMS detectors that have a "Port Server IP" address pointing to the computer running the program. As shown in Figure D-5, in addition to displaying Sunguide Detector ID, the pop-up window also displays the detector port number, CORSIM ID, and secondary CORSIM ID (for ramps), if available. The reason for having two detector IDs for detectors covering on/off-ramp lanes in CORSIM is that CORSIM requires coding the detectors for freeway mainline and ramps separately with different IDs.

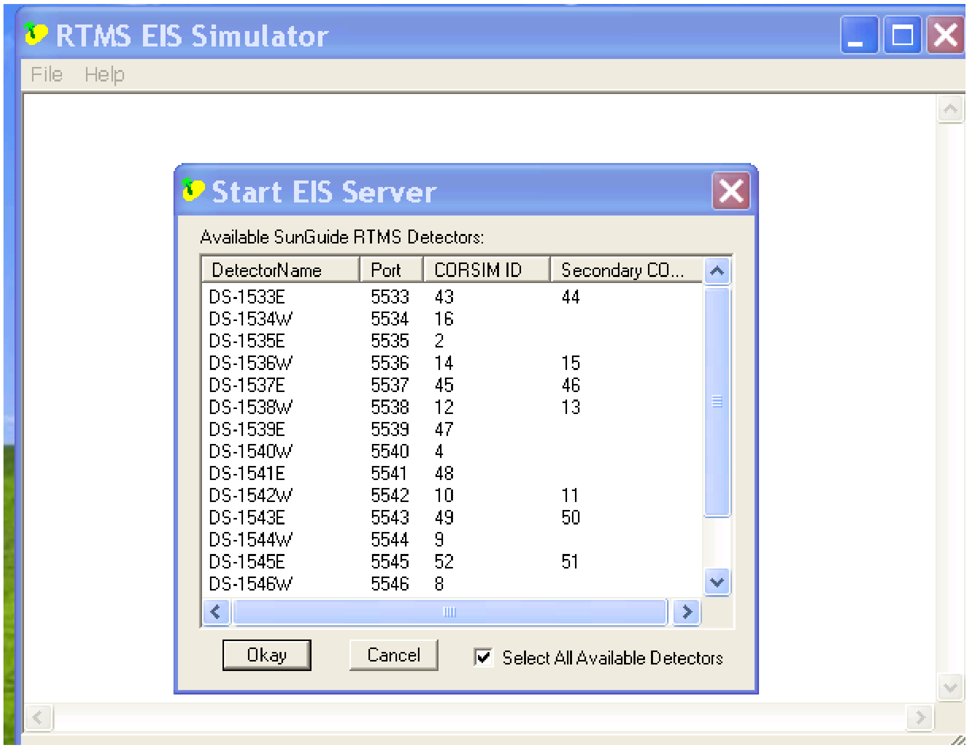


Figure D-3: RTMS Simulator Interface for "Start Server"

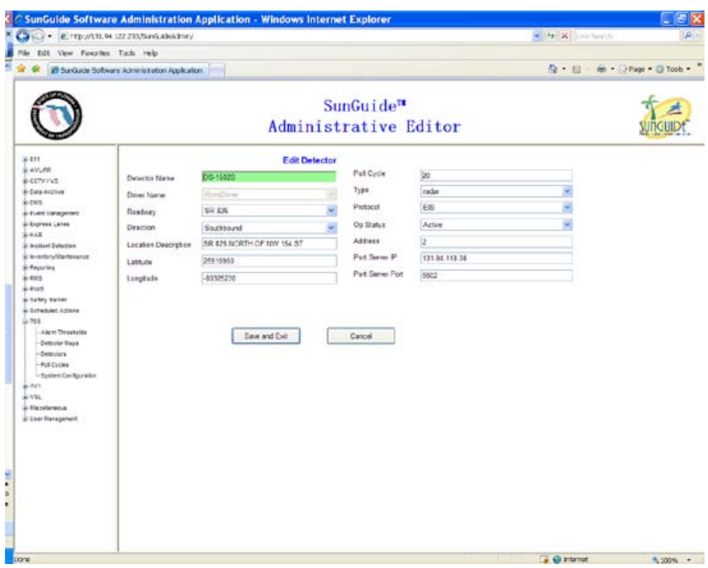


Figure D-4: SunGuide System Interface for RTMS Detector Coding

By default, all available detectors are chosen to run and wait on the designated ports for connection request. If an end user wants to select only a few of the detectors, the user can hold the SHIFT or CTRL key and use the mouse pointer to select the detectors he/she wants to start, as shown in Figure D-5.

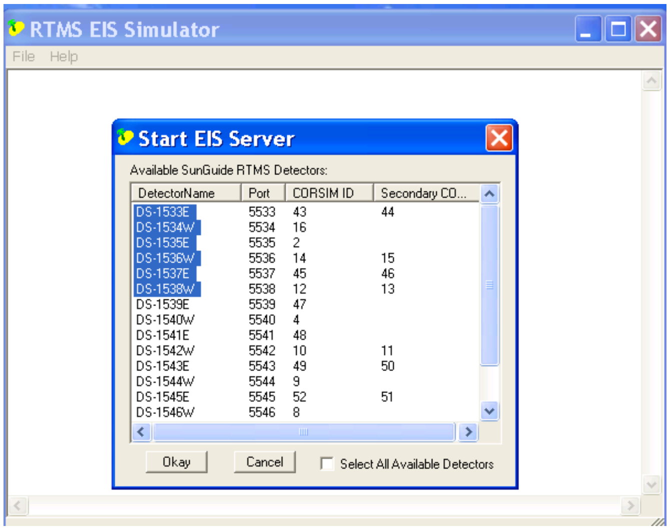


Figure D-5: RTMS Simulator Interface for Detector Selection

After selecting the list of detectors to simulate, the end user can click on the “Okay” button to start the simulators and listen on the designated ports for SunGuide connection requests. A set of waiting/connecting information is shown on the user interface, as illustrated in Figure D-6.

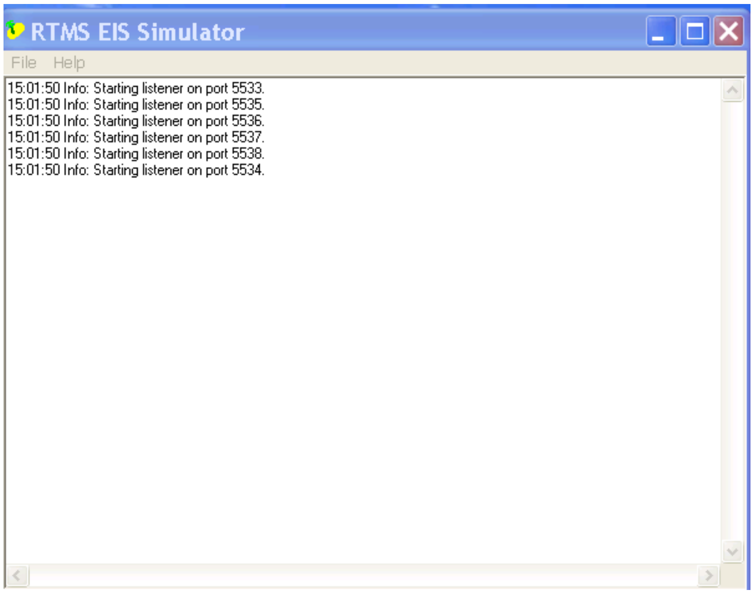


Figure D-6: RTMS Simulator Interface for Detectors Waiting for Connection

At anytime, if the end user wants to stop the simulators, he/she can choose "Stop Server" and a dialog box will be displayed to ask confirmation, as shown in Figure D-7. If the user chooses "Okay", all the simulators (one for each detector) will stop listening on the designated ports and the simulator program will stop sending data for established connections to the SunGuide system.

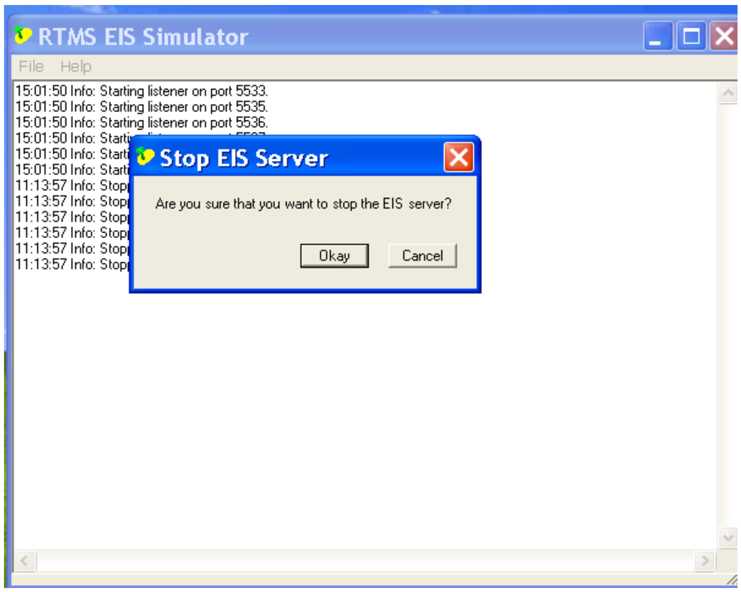


Figure D-7: RTMS Simulator Interface for "Stop Server"

At anytime, the end user can choose "Configure Data" to specify how and what data will be populated to the SunGuide system. As shown in Figure D-8, there are four possible ways to feed data into the SunGuide system: fixed values, fluctuating values, changing values, or using CORSIM outputs. The default option is to use fixed values. Once selected, the desired option is indicated in bold. The meanings of the four data feeding methods are as follows:

- Fixed values: The simulator program will feed the SunGuide detectors with fixed speed, volume, and occupancy data values specified by the user. The range for speed values is from 0 to 100 mph. The range for volume is from 0 to 100 vehicles per reporting period (every 20 seconds). The range for occupancy is from 0% to 100%

- Fluctuating values: The simulator program will feed the SunGuide detectors with fluctuating speed, volume, and occupancy data, calculated using average and variance values specified by the end user
- Changing values: The simulator program will feed the SunGuide detectors with user specified non-fixed values. The end user needs to specify the minimum, maximum, start, and pace (i.e., the value to change for each time interval) values of the speed, volume, and occupancy for detectors. The end user can also specify if the values will go up or down
- CORSIM or raw TSS output: the simulator program will feed the SunGuide detectors with the matched detector outputs from CORSIM or the historical TSS output data. When this option is selected and the “Direct TSS Match” checkbox is checked, the simulator knows that the end user wants to directly input historical TSS output to the SunGuide system. If the “Direct TSS Match” checkbox is not checked, CORSIM simulation output will be used. In any event, the end user should specify a text file (either from CORSIM output or historical TSS outout) as the input to the simulator program. In addition to specifying the CORSIM/TSS output file, the end user can also specify when to start the use of CORSIM simulation or TSS outputs to allow for the use of a combination of CORSIM/TSS outputs and one of the other three methods described above. The simulator program can automatically determine the length of the period covered by CORSIM/TSS output, based on the data in the CORSIM/TSS output file. Outside this time period, the simulator program will feed the SunGuide detectors with one of the previous three methods as chosen by the end user. Figure D-9 shows an example of this scenario. Suppose in this case that the CORSIM simulation is from 16:00 to 18:00; the RTMS simulator program will then feed the SunGuide system with “Changing Values” for the time period from 0:00 to 16:00 and from 18:00 to 24:00, and feed the SunGuide system with CORSIM outputs from 16:00 to 18:00.

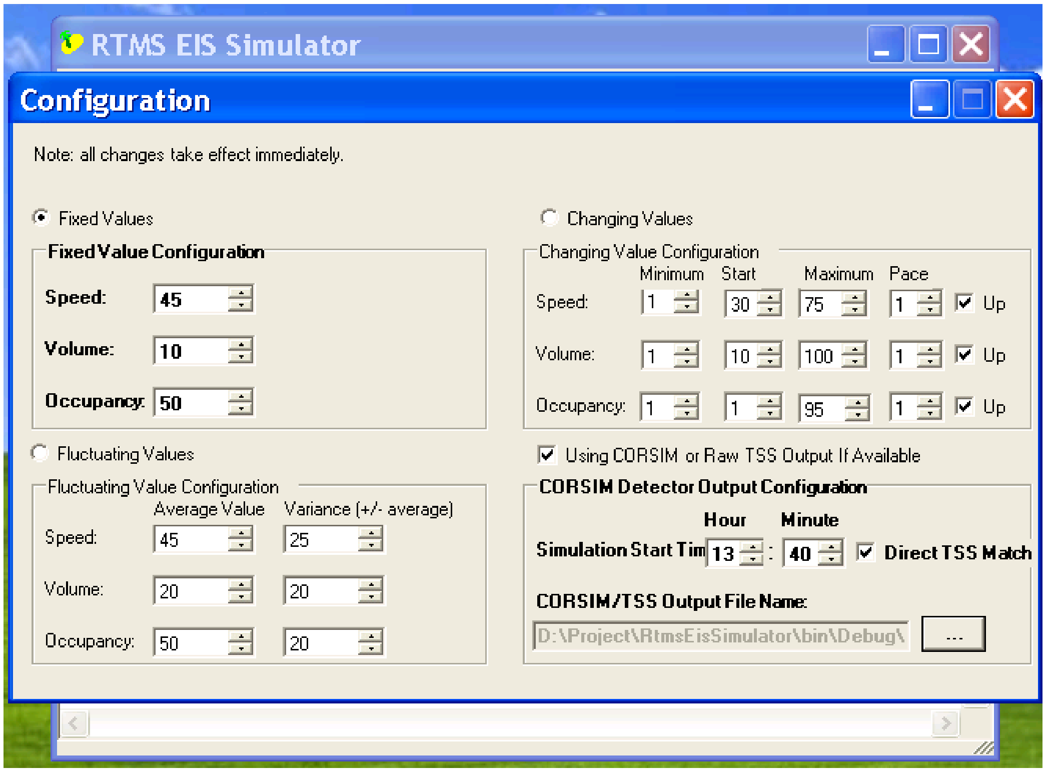


Figure D-8: RTMS Simulator Interface for “Data Configuration”

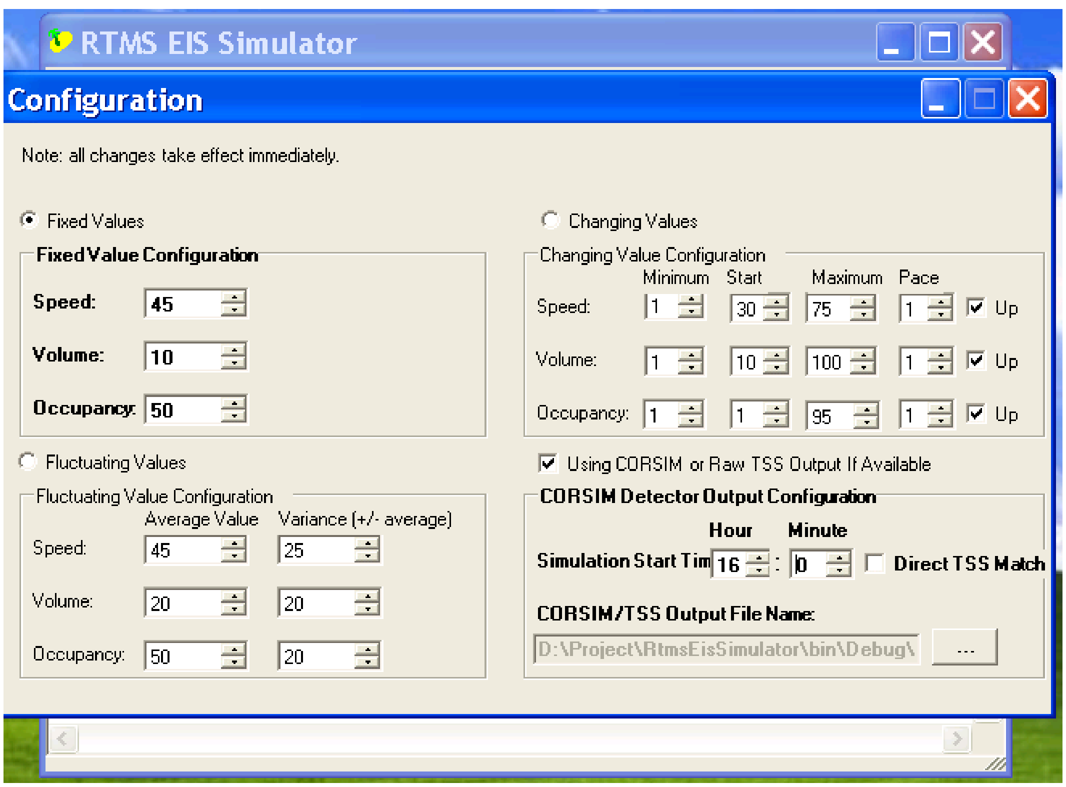


Figure D-9: RTMS Simulator Interface for “CORSIM Output”

If the end user chooses the “Debug Logging” menu, a text file is generated at the back-end to log the program console outputs. This option is mainly for debugging issues and problems that may not be encountered by general users. When an end user selects the “Clear Display” menu, the interactive message for “virtual” detector waiting/connection will be eliminated. This option can be used to clean messages shown on the screen when clutter occurs. When an end user selects the “Exit” menu, the program will terminate.

Running the AVI/LPR Simulator

The basic interface for the AVI/LPR simulator is shown in Figure D-10. The program was originally developed by SwRI and modified by FIU to support CORSIM output to the SunGuide TSS subsystem. The program supports one type of detector driver, which is the “ProbeFusionDriver”. The program supports four types of protocols: “InexZamirZap”, “SercoPips”, “SiritFlex”, and “TranscoreAllegro”. The end users should read the related SunGuide documents for the specific meanings of these protocols and code them in the SunGuide system to ensure that this program works correctly.

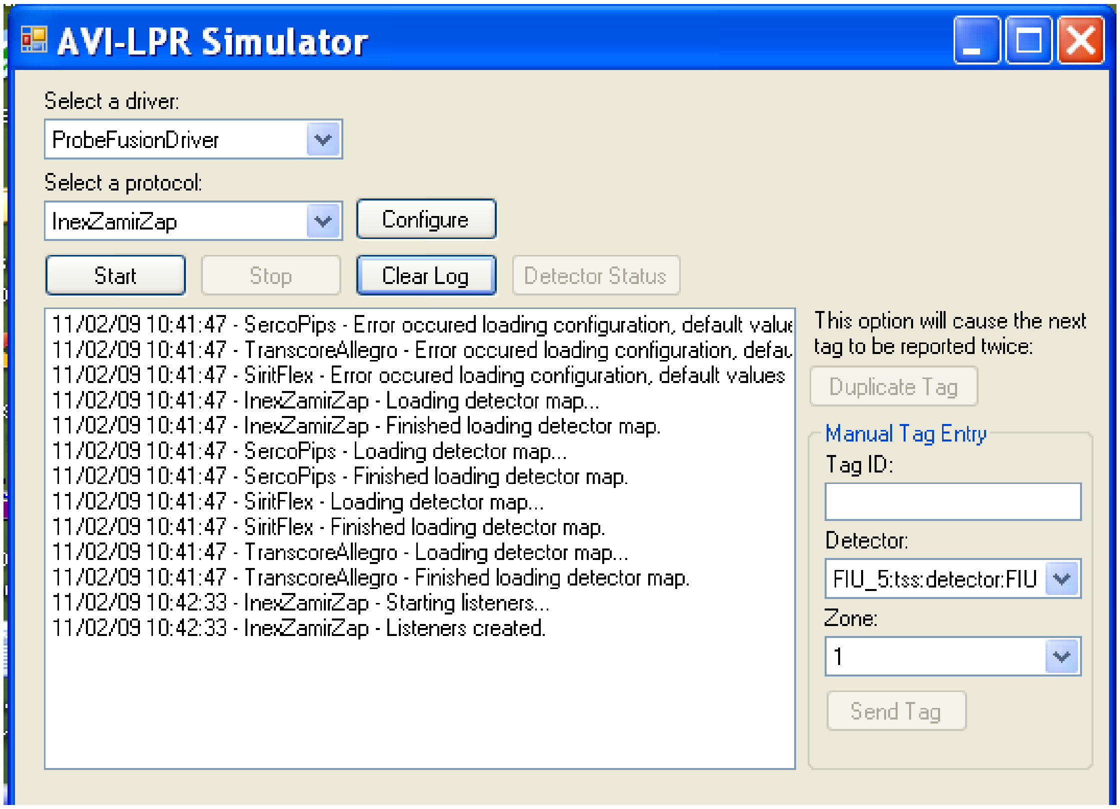


Figure D-10: AVI/LPR Simulator Interface

To be able to use the program, detectors should be coded in the SunGuide system using the "Administrative Editor" application. The SunGuide interface for coding an AVI/LPR detector is shown in Figure D-11. In Figure D-11, an LPR detector named "FIU_5" is coded. The only supported protocol for LPR by SunGuide is "InexZamirZap". The supported protocols for AVI detectors are "SiritFlex" and "TranscoreAllegro". The user should read the related SunGuide documents to correctly configure the AVI/LPR detectors in the SunGuide system. The user also needs to specify the longitude and latitude of the detector, and most importantly, the "Port Server IP" field should be filled with the IP address of the computer running the AVI/LPR simulator, with an appropriate port number specified to ensure that there is no conflict with other applications.

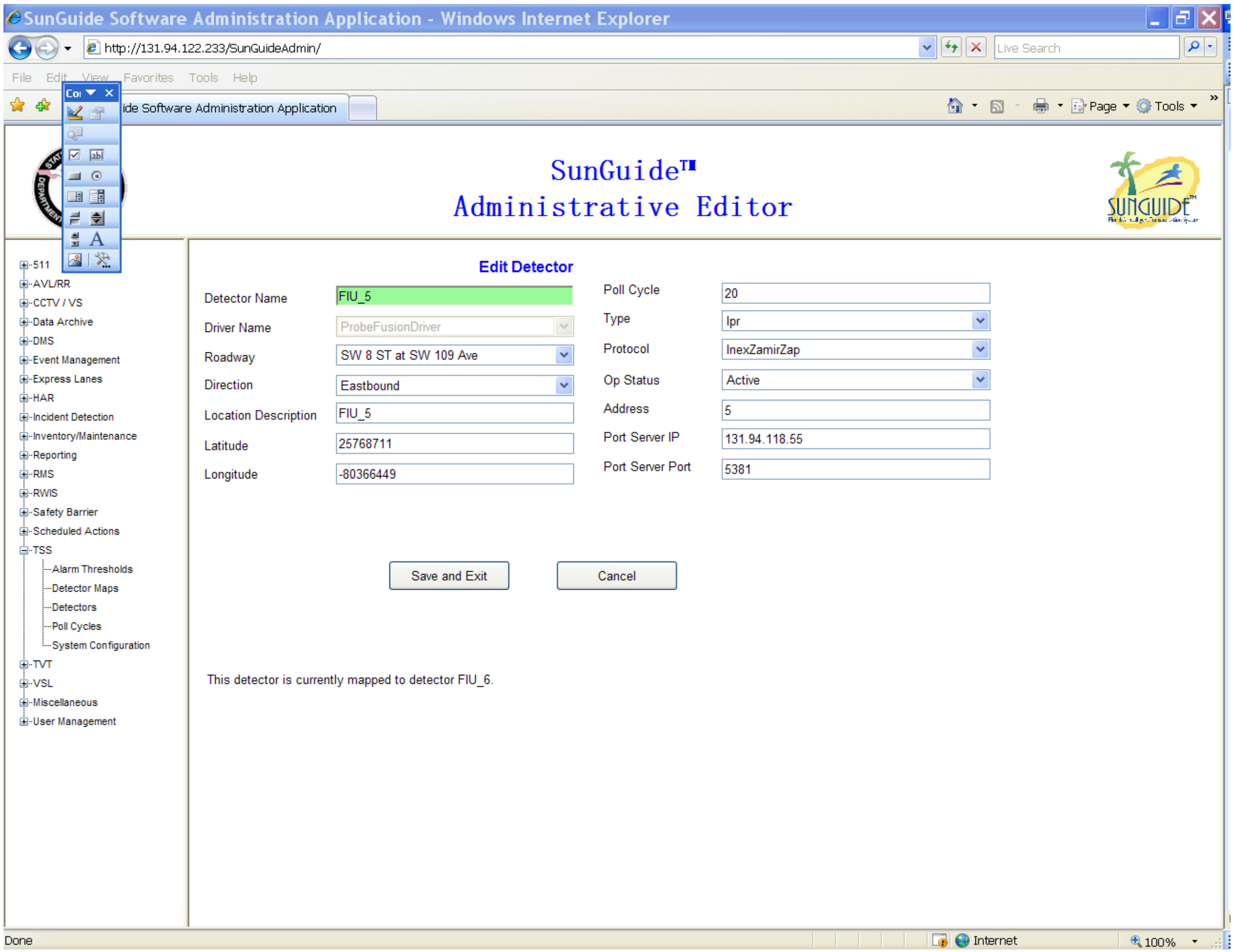


Figure D-11: SunGuide Interface for Coding AVI/LPR Detectors

After the detectors are coded, the system administrator or the end user needs to conduct the initial editing and coding of two XML files within the program folder. These two files are "detectorData.xml" and "mapDetectorData.xml". In addition, if CORSIM output is to be used, the table "Detector" in the "DetectorResult.mdb" access database should also be modified to match the SunGuide detectors with CORSIM detector IDs.

After all the coding and editing tasks are finished, the end user can start the AVI/LPR simulator. Figure D-12 is an example of running the LPR detector simulator. Detector information specified in the XML files is retrieved by the program when it runs, as shown in the right hand detector information dropdown menu ("FIU_5" and "FIU_6" in this example). The program also shows detector zone information, which is usually corresponds to lane information on the detector site. The "Tag ID" box allows end users to manually enter new vehicle tag numbers and use the "Send Tag" button to deliver the information to the SunGuide system. In the main display of this simulator, there are five options that can be selected by pressing buttons: "Start", "Stop", "Clear Log", "Detector Status", and "Configure". The "Start" and "Stop" buttons are used to start and stop the AVI/LPR listeners; the appropriate messages will be shown in the textbox when the end user clicks on a respective button. The "Clear Log" button is used to simply eliminate all the displayed log messages. The "Detector Status" button is used to show the status of the simulated detectors, which can be "Activate" or "Inactivate". Figure D-13 shows the interface for displaying such statuses. In this example, one pair of detectors, "FIU_5" and "FIU_6", is coded. Figure D-13 shows information about the detector names, port numbers, operation status, and the current passing vehicle tags. The "Configure" button is a little bit more complex, and is used to configure the run-time properties of the simulated detectors. When an end user clicks on the "Configure" button, a dialog box window will be displayed, as shown in Figure D-14.

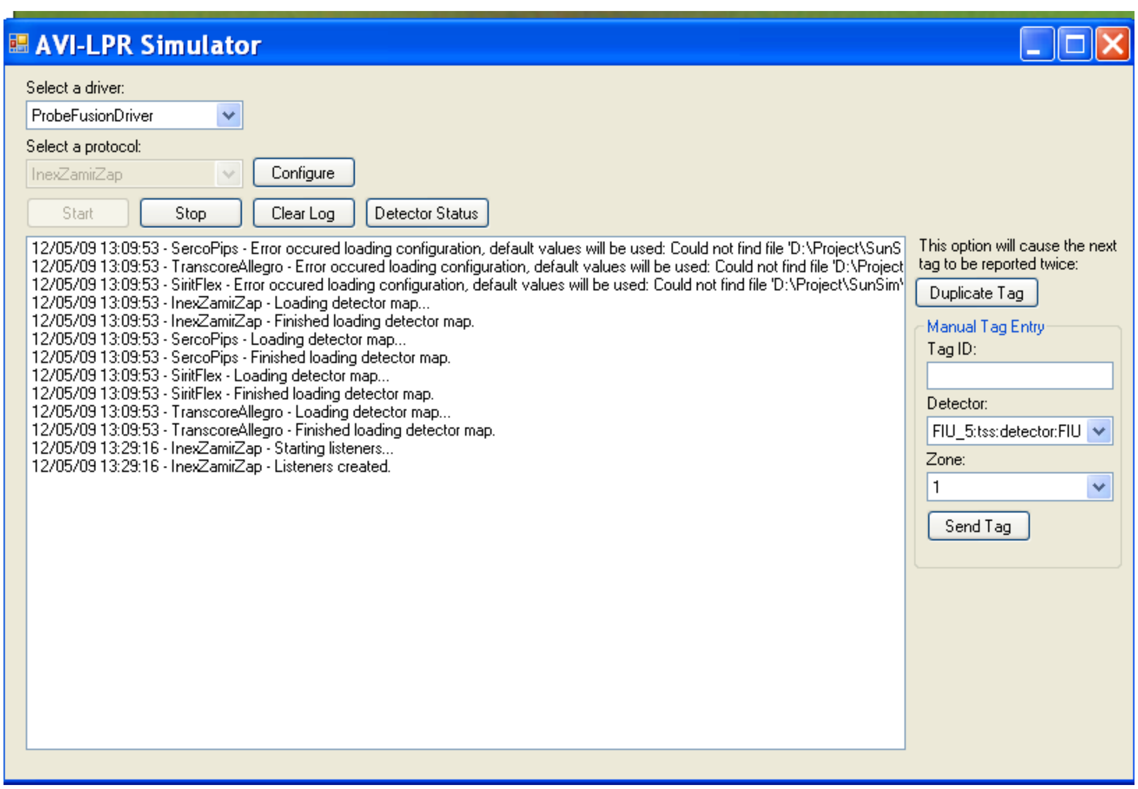


Figure D-12: AVI/LPR Simulator Interface for Starting the LPR Detector Simulators

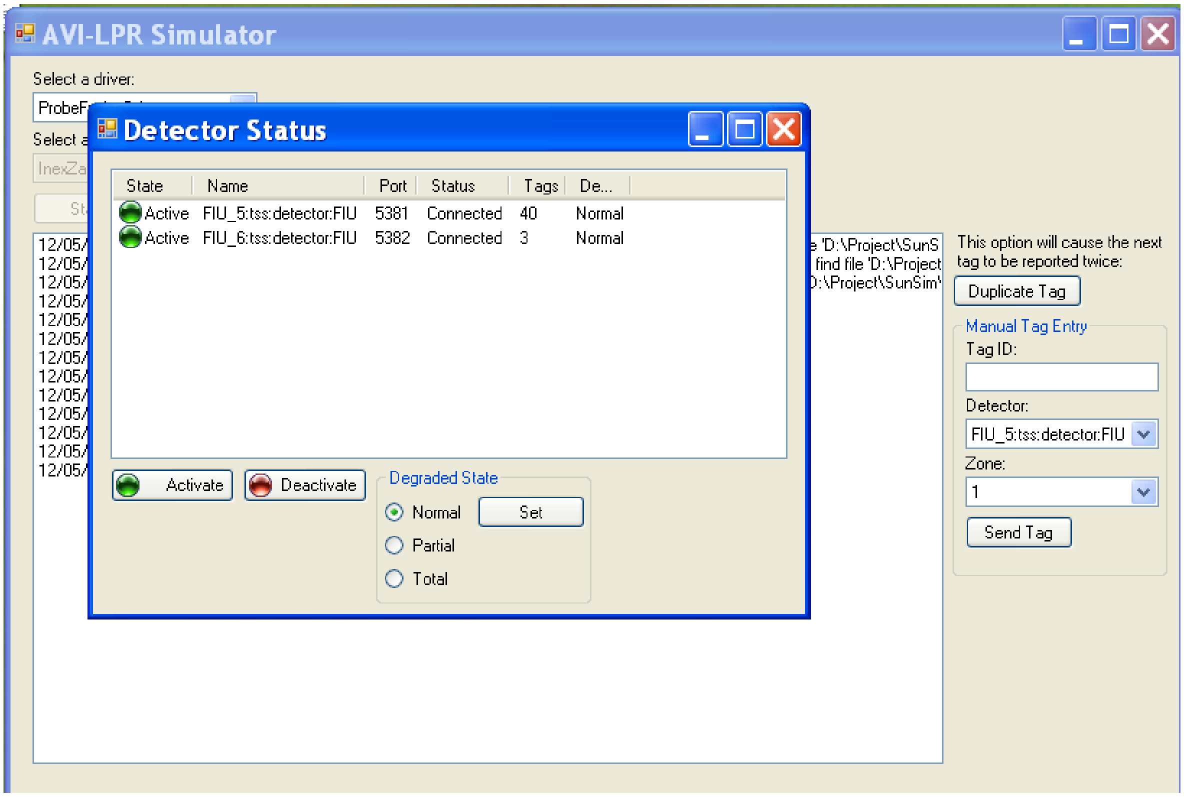


Figure D-13: AVI/LPR Simulator Interface for Showing Detector Status

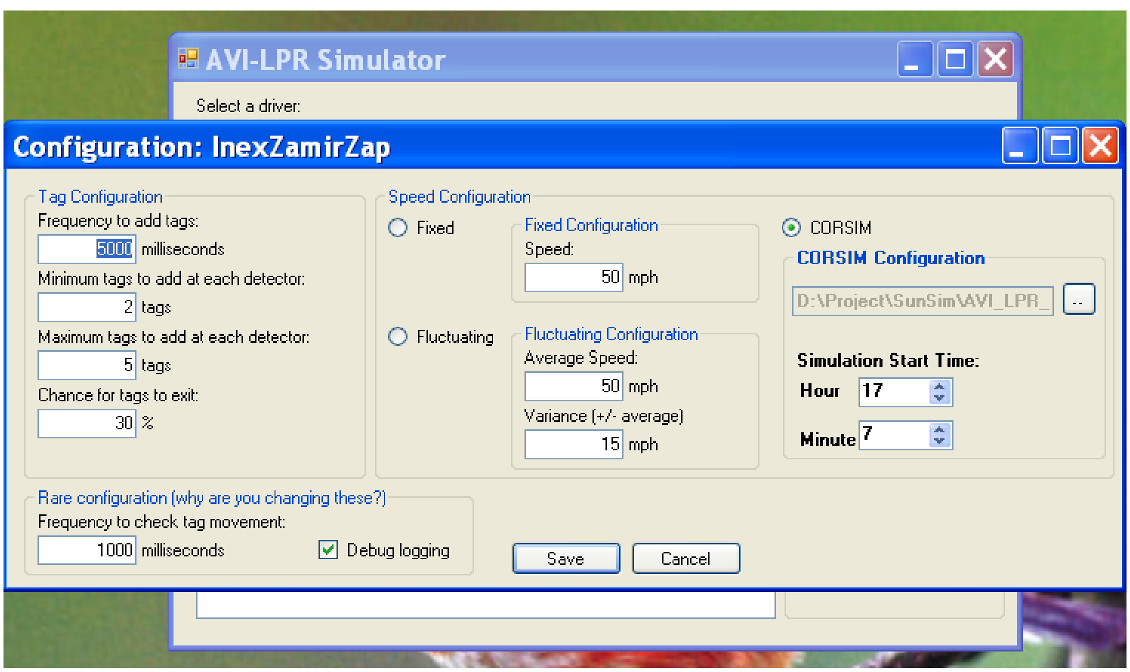


Figure D-14: AVI/LPR Simulator Interface for Configuration Settings

As shown in Figure D-14 above, there are three major configuration sections: tag, speed, and rare configurations. Tag configuration is used to specify the frequency with which new vehicle plates (tags) are added, minimum and maximum number of vehicles added per every specified interval, and the chance of vehicles to exit the freeway. The rare configurations are used to specify the time interval for vehicle movements and the debug-logging settings. The speed configuration section is used to specify the speed variable of added vehicles (tags). There are three options for fixed speed, fluctuating speed, or CORSIM speed. If the fixed speed option is used, all of the vehicles will have the same speed, while the fluctuating speed option calculates speed based on a specified average speed and a variance. The CORSIM speed option allows the program to extract CORSIM outputs and populate the speed data to the SunGuide system. Before one can use the “CORSIM” option, an end user should make sure that the “DetectorResult.mdb” Access database file exists under the executable program folder, and that the mapping of SunGuide and CORSIM detector IDs has been setup in the “Detector” table, similar to the RTMS simulator program. In addition, the end user must use the “…” button to select the CORSIM output file (i.e., the user should have a CORSIM model for the corridor/arterial streets that he/she wants to simulate). The user should then run the CORSIM model to generate the output file, which will be used as simulator inputs to the SunGuide system.

References

Abdulhai, B., J. Sheu, and W. Recker. "Simulation of ITS on the Irvine FOT Area Using Paramics 1.5 Scalable Microscopic Traffic Simulator: Phase I: Model Calibration and Validation," California PATH Research Report, Report for MOU 359, April 1999.

Alexis, V. *Integrated Corridor Management Analysis, Modeling, and Simulation Experimental Plan for the Test Corridor*. Publication FHWA-JPO-08-035. FHWA, U.S. Department of Transportation, 2008.

Alpaydin, E. *Introduction to Machine Learning.* The MIT Press, Massachusetts, 2004.

Aved, A., T. Do, G. Hamza-Lup, A. Ho, L. Hoang, L. Hsia, K. Hua, F. Liu, and R. Peng. "A Real-Time Route Diversion Management System," Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Seattle, WA, September, 2007 pp.1131-1136.

Barceló, J., D. García, and H. Kirschfink. "Scenario Analysis a Simulation Based Tool for Regional Strategic Traffic Management," Proceedings of the 9th World Conference on Intelligent Transport Systems, Chicago, 2002.

Barceló, J., and D. García. "Some Issues Concerning the use of Simulation in Advanced Traffic Management Systems," International Transactions in Operational Research, 1999, Vol. 6, pp. 123-143.

Barceló, J., H. Kirschfink, and A. Torday. "An Integrated Software Platform to Assist Advanced Traffic Management Decisions," Proceedings of the 10th World Conference on ITS, Madrid, 2003.

Barfield, W., L. Conquest, J. Spyridakis, and M. Haselkorn. "Information Requirements for Real-Time Motorist Information Systems." Proceedings of the Vehicle Navigation and Information Systems Conference (VNIS), New York, IEEE, 1989, pp. 101-112.

Benson, B. G. Motorist Attitudes about Content of Variable-Message Signs. Transportation Research Record, No. 1550, 1996, pp. 48-57.

Chang, H., S. Baek, H. Kim, A. Shah, J. Lee, and N. Mahalik. “Development of Distributed Real-Time Decision Support System for Traffic Management Centers Using Microscopic CA Model,” Iranian Journal of Science & Technology, Transaction B, Engineering, 2007, Vol. 31, No. B2, pp. 155-166.

Chatterjee, K., N. B. Hounsell, P. E. Firmin, and P. W. Bonsall. Driver Response to Variable Message Sign Information in London. Transportation Research Part C, Vol. 10, No. 2, 2002, pp. 149-169.

Cragg, C., and M. Demetsky. Simulation Analysis of Route Diversion Strategies for Freeway Incident Management. Virginia Transportation Research Council, Charlottesville, Virginia, 1995.

Courage, K. G., and S. Lee. ”Development of a Central Data Warehouse for Statewide ITS and Transportation Data in Florida: Phase II Proof of Concept,” Final Report Prepared for the Florida Department of Transportation by the University of Florida Transportation Research Center, Tallahasse, FL, 2008.

Dailey, D., and N. Taiyab. “A Cellular Automata Model for Use with Real Freeway Data,” Produced by ITS Research Program, University of Washington Seattle, Washington for the Washington State Department of Transportation, January 2002.

Federal Highway Administration (FHWA). Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software. Publication NO. FHWA-HRT-04-040, Washington, D.C., 2004.

Federal Highway Transportation (FHWA 2004). *Archived Data Management Systems - A Cross-Cutting Study*. Publication FHWA-JPO-05-044. FHWA, U.S. Department of Transportation, 2004.

FHWA. CORSIM User’s Guide. U.S. Department of Transportation, 2006.

FHWA. Traffic Analysis Toolbox Volume IV: Guidelines for Applying CORSIM Microsimulation Modeling Software. Publication NO. FHWA-HOP-07-079, Washington, D.C., 2007.

Fries, R., I. Inamdar, M. Chowdhury, K. Taaffe, and K. Ozbay. "Feasibility of Traffic Simulation for Decision Support in Real-Time Regional Traffic Management," Transportation Research Record, No. 2035, Washington, D.C., 2007, pp. 169-176.

Gomez, G., A. May, and R. Horowitz. "Calibration of VISSIM for a Congested Freeway," Institute of Transportation Studies California Partners for Advanced Transit and Highways (PATH) Report UCB-ITS-PRR-2004-4, University of California, Berkeley, March 2004.

Goolsby, M. E. "Influence of Incidents on Freeway Quality of Service." Proceedings of the 50th TRB Annual Meeting, Transportation Research Board, Washington, D.C., 1971.

Hadi, M., P. Sinha, and A. Wang. "Modeling Reductions in Freeway Capacity due to Incidents in Microscopic Simulation Models." Journal of the Transportation Research Board, Transportation Research Record 1999, Transportation Research Board, Washington, D.C., 2007, pp. 62-68.

Hadi, M., Y. Xiao, C. Zhan, and A. Gan. "Use of the HCM Freeway Facility Procedures Combined with ITS Data Archives to Assess the Benefits of Incident Management Systems." Journal of the Transportation Research Board (in press). Transportation Research Board, Washington, D.C, 2010.

Hillier, F., S. Frederick, J. Gerald, and K. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, 2004.

Jayakrishnan, R., O. Jun-Seok, and S. Abd-El-Kader. "Calibration and Path Dynamics Issues in Microscopic Simulation for Advanced Traffic Management and Information Systems," Presented at the 80th Annual Meeting of the Transportation Research Board, Washington, D.C., 2001.

Jayakrishnan, R., H. Mahmassanim, and T-Y Hu. "An Evaluation Tool for Advanced Traffic Information and Management Systems in Urban Networks," Transportation Research 2C, 1994, pp.129-147.

Jayakrishnan, R., and C. Rindt. "Distributed Computing and Simulation in a Traffic Research Test Bed," Computer–Aided Civil and Infrastructure Engineering, Vol. 14, No. 6, November 1999, pp. 429-443.

Jeannotte, K., A. Chandra, V. Alexiadis, and A. Skabardonis. “Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis Tools,” Prepared for the Office of Operations Federal Highway Administration, Washington, D.C., 2006.

Jun-Seok, O., C. Cortes, R. Jayakrishnan, and L. Der-Horn. “Microscopic Simulation with Large-Network Path Dynamics for Advanced Traffic Management and Information Systems,” Institute of Transportation Studies, University of California, Irvine, 1999.

Keogh, E., S. Chu, D. Hart, and M. Pazzani. “An Online Algorithm for Segmenting Time Series,” International Conference on Data Mining, 2001.

Knoop, V.L., S. P. Hoogendoorn, and K. Adams. “Capacity Reductions at Incidents Sites on Motorways.” European Journal of Transport and Infrastructure Research., Vol. 9 Issue 4, Delft, Netherlands, 2009, pp. 363-379.

Knoop, V.L., S. P. Hoogendoorn, and H. J. Van Zuylen. “Route Choice Under Exceptional Traffic Conditions.” International Conference on Evacuation Management, 23-25 September 2009, The Hague, Netherlands.

Luk, J., and C. Yang. “Comparing Driver Information Systems in a Dynamic Modeling Framework.” Journal of Transportation Engineering, Vol. 120, Issue 1, Jan/Feb 2003 p.p. 42-50.

Madanat, S., C. Yang, and Y. M. Yen. “Analysis of Stated Route Diversion Intentions under Advanced Traveler Information Systems Using Latent Variable Modeling.” Transportation Research Record, No. 1485, 1995, pp. 10-17.

Mahmassani, M., S. Eisenman, X. Fei, and X. Zhou. “Application of DYNASMART-X to the Maryland CHART Network for Real-Time Traffic Management Center Decision Support,” Paper # 4969, Maryland Transportation Initiative, University of Maryland, College Park, MD, USA, Proceedings of the 12th World Congress On Intelligent Transportation Systems, San Francisco, CA, November 2005.

Narupiti, S., and M. Mustafa. “Quality of Travel Time Estimation from Probe Vehicles: A Simulation Study,” Journal of the Eastern Asia Society for Transportation Studies, Vol. 7, 2007, p. 2517-2532.

Park, B., and I. Yun. “Development of ITS Evaluation Test-Bed Using Microscopic Simulation – City of Hampton Case Study,” A Research Project Report for the Center for ITS Implementation Research (ITS), A U.S. DOT University Transportation Center, August 2003.

Tarry, S., and A. Graham. “The Role of Evaluation in ATT Development.”, Traffic Engineering and Control. Vol. 36, No. 12, London, England, 1995, pp. 688- 693.

Transportation Research Board (TRB). *Highway Capacity Manual 2000.* Transportation Research Board of the National Academies, Washington, D.C., 2000.

Qi, L., and B. Smith, “Characterization of Accident Capacity Reduction.” Research Report No. UVACTS-15-0-48, Center for Transportation Studies, University of Virginia, Charlottesville, September 2001.

Zou, N., P. Lin, G. Chang, and S. Rahwanji. “An Integrated Knowledge-Base/Simulation System for Realtime Incident Management,” Presented at the 82nd TRB Annual Meeting, Washington, D.C., 2003.