Joint Calibration of a Microscopic Traffic Simulator and Estimation of Origin-Destination Flows

by

Deepak Darda
B. Tech in Civil Engineering (2000)
Indian Institute of Technology, Bombay, India

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author.. ....
Department of Civil and Environmental Engineering
May 24, 2002

Certified by...
Moshe E. Ben-Akiva
Edmund K. Turner Professor of Civil and Environmental Engineering
Thesis Supervisor

Certified by...
Mithilesh K. Jha
Principal of Transportation Planning, Jacobs Civil Inc.
Thesis Supervisor

Certified by.. ....
Haris N. Koutsopoulos
Operations Research Analyst, Volpe National Transportation Center
Thesis Supervisor

Accepted by ....
Oral Buyukozturk
Chairman, Department Committee on Graduate Students

Joint Calibration of a Microscopic Traffic Simulator and

Estimation of Origin-Destination Flows

by

Deepak Darda

Submitted to the Department of Civil and Environmental Engineering
on May 24th, 2002, in partial fulfillment of the
requirement for the degree of

Master of Science in Transportation

Abstract

Microscopic Traffic simulators can be effectively used to analyze various transportation strategies (especially ITS related). However, microscopic traffic simulators need to be calibrated in order to yield meaningful results. Calibration is the process of adjusting the model parameters to closely replicate the observed behavior. In this thesis, a module for joint calibration of a microscopic traffic simulator along with estimation of Origin-Destination (OD) flows has been developed. The developed framework takes into account the interactions between the various model parameters and the OD flows. An optimization-based framework has been proposed for the joint calibration of model parameters and estimation of dynamic OD flows. A systematic search approach based on the Box algorithm is adopted for calibration of the parameters. Depending on the problem size, a sequential or a simultaneous OD estimation algorithm is employed. Since the calibration of the parameters depends on the estimated OD flows and vice versa, the proposed framework is iterative. The developed framework has been implemented in MITSIMLab, a microscopic traffic simulation laboratory. Case studies on three networks with varying levels of complexity illustrate the potential of the calibration approach.

Thesis Supervisor: Moshe E. Ben-Akiva
Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Thesis Supervisor: Mithilesh K. Jha
Title: Principal of Transportation Planning, Jacobs Civil Inc.

Thesis Supervisor: Haris N. Koutsopoulos
Title: Operations Research Analyst, Volpe National Transportation Systems Center

Acknowledgements

I would foremost like to thank my advisors Professor Moshe Ben-Akiva, Dr. Mithilesh Jha and Dr. Haris Koutsopoulos for their constant advice and encouragement throughout the course of my graduate studies. Their suggestions were invaluable and have helped me a lot in my research work. Dr. Mithilesh Jha has been a mentor, guide and a friend to me. He not only helped me in my research but also provided me guidance for my career. I am overwhelmed with gratitude for all his help. Dr. Haris Koutsopoulos helped me a lot by giving me ideas and feedback on my work.

I would also like to thank to Jacobs Civil Inc. for financially supporting my research and Center for Transportation studies for giving me an opportunity to pursue graduate studies at MIT.

My thanks go to Angus, Manish and Marge who helped me in getting started at the ITS Lab. I express especial thanks to Tomer, Rama, Srini and Kunal for their enormous help and advise in my research. I would like to thank Josef, Dan, Constantinos, Akhil, Joe and Zhili for their friendship and the great time I had at the ITS Lab.

Outside of lab, I would like to thank my friends for helping me understand that while research and science are both meaningful, we all have other aspects of our life as well. Deepak (Lala) and Dinesh have helped in innumerable ways and their friendship is something that I will always treasure. Nitesh and I go a long way back and I am ever grateful for his constant help, admonition as well as encouragement. Rahul, Sundar, Ullal and Supriya are great friends and I will always cherish the wonderful moments that I shared with them.

My thanks to the Singhvi family of Randolph, for their generous hospitality and love they have showered upon me. They have been like a home away from home.

I would like to acknowledge and thank my parents, brother and sister for their encouragement and support. They have always been a source of inspiration for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The use of traffic simulation models in traffic operations and traffic impact studies is increasing as technology provides traffic engineers and transportation planners with an ever-increasing variety of user-friendly software and fast computers. Traffic simulation models have drawn significant attention in evaluating the impact of changes in network infrastructure (e.g. adding a lane to an existing roadway or adding a road to a network), traffic control devices (e.g. retiming of traffic signal settings or installing a ramp metering scheme), and application of Intelligent Transportation Systems (ITS). Traffic simulation models provide several advantages over traditional analysis tools, such as the Highway Capacity Software (HCS), in that they can provide an analysis of entire roadway systems and a visual simulation of the results. However, the reliability of such models hinges on how field conditions are best captured by the parameters in the simulation models—achieving the best reproducibility of field conditions entails calibration of the simulation model, followed by validation of the calibrated system. Calibration is the process by which the parameters of various components of the simulation model are estimated so that the model will accurately replicate observed traffic conditions. Validation of the system refers to the verification of the model's capability to reproduce field conditions represented by the data.

Traffic simulators are broadly classified into three categories: macroscopic, microscopic and mesoscopic. Macroscopic traffic simulators simulate traffic at an aggregate level. Vehicles are moved according to average speed-density relationships. Macroscopic models are therefore capable of modeling large-scale networks, at the expense of sacrificing vehicle-to-vehicle or vehicle-to-traffic-control interactions. On the other hand, microscopic traffic simulators model individual vehicles and their trajectories at disaggregate level. Representation of car-following and lane-changing characteristics of each individual vehicle permits microscopic models to consider the non-homogenous flow of different vehicle types in individual lanes. The advantage of microscopic level of detail becomes significant if the models are utilized to evaluate various traffic control, ramp metering, variable message signs (VMS), and other type of ITS applications at the operational level. Mesoscopic simulation models are a hybrid of microscopic simulation models and macroscopic simulation models. This thesis focuses on microscopic traffic simulation models and calibration of the model parameters.

1.1 Overview of Calibration of Microscopic Simulation Models

Microscopic simulation models incorporate detailed modeling of merging, diverging, weaving, lane changing, turning and other behavioral characteristics reflecting real-life traffic conditions. There is extensive use of route choice models and driving behavior (car following and lane-changing) models to capture the network-specific traffic dynamics. In order to take the temporal variation of traffic conditions into account, the microscopic simulation models use time-dependent OD matrices. The microscopic simulator uses these models to predict drivers' behavior, such as selection of interspacing from the leading car, acceleration, deceleration, lane change decision, and route decision etc. in response to a stimulus.

Any evaluation framework for a microscopic traffic simulator would consist of calibration and validation. Calibration is the process of determining the modification of default model parameters so that the simulator output rightly reflects the local traffic conditions being modeled. Model validation is considered to be the process of determining the extent to which the model's underlying fundamental concepts and relationships capture the consequent behavior, as specified by the relevant theory and as recorded in field data records. Validation of a model aims to test the extent to which the calibrated model can accurately replicate traffic behavior.

Common calibration elements include the following:

- OD flows.
- Route Choice and Driving Behavior models.

- Steady State Travel Time.

Two broad procedures commonly used for model calibration (Kurian[1]) are:

1. Rational technique involving direct measurements of parameters.
2. Indirect technique in which the parameter values are inferred by comparing model outputs to the real world observations.

The first technique uses estimation methods based on econometric models to directly estimate the individual parameter values. The disadvantage of this technique is the large amount of detailed data required. For instance, the data required for estimating the parameters for the car-following and lane-changing models include position, speed, acceleration, and length of the vehicle. Ahmed [2] applied this technique to calibrate mandatory lane changing and discretionary lane changing model parameter of Microscopic Traffic Simulator (MITSIM). He used vehicle trajectories data at each second for calibration.

The indirect technique uses the simulation model itself to predict the parameter values. Calibration is performed by minimizing an objective function that captures the deviation between the observed and simulated values (flow, speed, density or combination of these values). The advantage of this technique is that it uses aggregate data that are more readily available. Flow, speed, occupancy data can be easily measured across a network at sensor locations and can be used for calibration.

1.2 Thesis Motivation

The output generated by a microscopic traffic simulator is determined by the various parameters governing the simulation models. The crux of the calibration problem thereby lies in the validity of these parameters. The OD matrices are also an important input to the simulation model. There are interactions between the OD matrices with the various models of the simulator and these interactions govern the traffic conditions depicted by the simulator. Hence it is important that these matrices are also estimated with the model parameters. Furthermore, in view of the increasingly important role that traffic simulation models play in the evaluation of ITS, it is important that a systematic procedure for calibration be developed.

The objective of this thesis is to develop a systematic procedure for calibration of microscopic traffic simulation model parameters and estimation of the OD matrices using aggregate data (such as speed, flow counts etc. at sensor locations) and seed OD matrices (usually at the planning level). The resulting output consists of a time-dependent OD matrix and calibrated parameters as depicted in Figure 1-1.

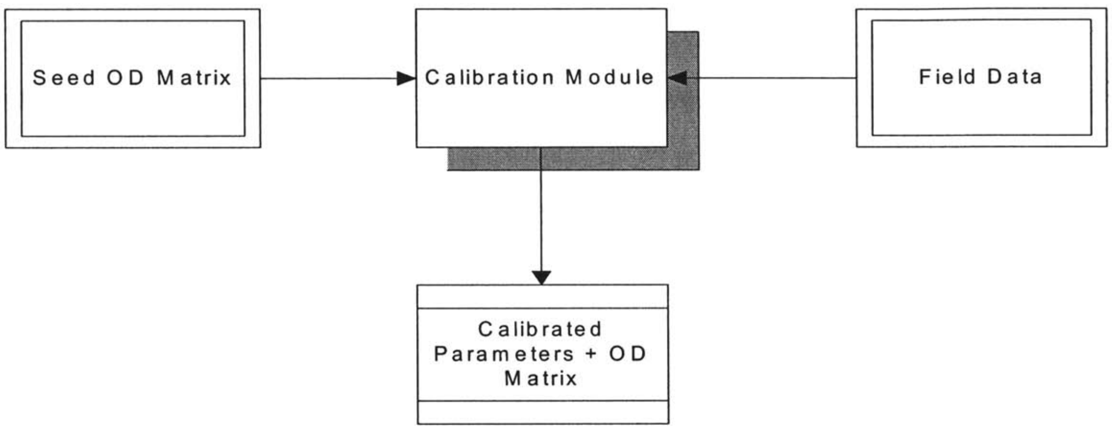


Figure 1-1: Inputs and Outputs of the Calibration Module

A challenging task in developing such a procedure is the use of appropriate solution methods. This is essential since optimization of model performance involves the selection of the "best" set of values for the parameters. Problems such as these pose a great challenge due to the large parametric space to be searched. The search is usually a multidimensional one where the values of the parameters can be conceived as coordinates and the "fitness" representing goodness of fit as a hilly surface. The process of seeking a good solution should therefore involve a systematic search method.

1.3 Thesis Outline

The thesis is organized into 6 chapters. Chapter 2 presents literature review in the area of calibration and OD estimation. Chapter 3 presents the overall framework of calibration and then discusses the details of OD estimation, steady state travel times computation and the algorithm used for calibration of parameters. Chapter 4 starts with a description of MITSIMLab, which is the microscopic traffic simulator used for the case studies. It discusses the various models in the simulator that need to be calibrated. Later in the chapter, the application of the proposed methodology to MITSIMLab is discussed. In Chapter 5, results from three case studies are presented. Finally in Chapter 6, the thesis contributions and the direction for future work are discussed.

Chapter 2

Literature Review

Calibration can be viewed as an optimization problem in which an objective function has to be minimized under certain constraints. The chapter reviews literature pertaining to calibration, OD estimation, and combined calibration and OD estimation.

2.1 Calibration Studies

A great chunk of the existing literature in the area of traffic simulation relates to the description of the models, their characteristics, assumptions, and the mathematical or logical relationships around which the models were developed. However, the literature concerning the calibration of these models is quite rare. Many of the parameters or distributions were calibrated using very limited data resources or derived for some specific conditions, and were hardly tested under general conditions. This section describes some of the work that was done for calibration of traffic models using indirect techniques.

Abdulhai and Ma [3] dealt with the calibration of microscopic traffic simulator parameters using a genetic algorithm approach. The software GENOSIM developed for the calibration purpose performs the role of obtaining traffic simulator parameters such as route-choice, lane changing parameters etc., assuming a given fixed input OD. Given an OD matrix and initial parameter estimates, the micro-simulator was run to produce simulated data. Based on simulated versus actual data and an objective function, a genetic algorithm approach identifies better parameter estimates using the so-called crossovers and mutations. This process of running the simulator and GENOSIM was carried until satisfactory results were obtained. This approach was then used to calibrate five parameters of a network with 470 nodes, 1270 links, and 109 junctions in Toronto, Canada during a morning peak period of 8:00 to 9:00 AM. While the methodology mentioned in the paper is promising in terms of using a systematic search technique to identify better parameter values, there are some issues that merit consideration. The genetic algorithm approach uses genetic parameters, such as the probability of crossovers, the probability of mutations, crossover and mutation options etc., to obtain new set of parameters for the traffic simulator. However, values for these genetic parameters for a general network are normally based on trial and error. Thus methods to obtain or "calibrate" these genetic parameters are important for practical use of the proposed approach.

Lee et al. [4] presented the application of genetic algorithm as an optimization tool for determining a suitable combination of PARAMICS parameter values. In their study, a freeway segment on I-5 southbound in California between Culver and Jamboree was selected for parameter calibration. The basic sub-models calibrated were the lane changing and the car following models. Both models are primarily affected by the mean headway and mean reaction time and hence these parameters were selected as the calibration elements in the model. The fitness function used in the study was a combination of average relative errors of 30-second flow and occupancy between simulation output and field data. To take stochasticity of the model into account, an average of three runs with different random seeds was used. However in order to simplify the potential impact brought by the OD, they chose a network with multiple origins and a single destination.

Cheu et al. [5] calibrated the microscopic traffic simulator INTRAS with a 30-sec interval, loop detector data set. The parameters were identified and then calibrated individually, i.e., optimal value of one parameter was obtained by keeping other parameters constant. Averaging over three runs was done to take care of the stochastic nature of the simulator. The disadvantage of his approach is that it does not capture the interactions between the parameters and hence the obtained solution might not be the best solution.

Cheu et al. [6] used a genetic algorithm to calibrate the microscopic simulator FRESIM for a Singapore expressway. The calibration was done based on the field data collected on weekdays over 5.6 KM segment of Ayer Rajar Expressway. The calibration parameters consisted of free-flow speeds and vehicle-movement parameters. The fitness function was a combination of average-absolute errors of 30 sec volume and speed between the FRESIM output and the field data. However, stochasticity associated with the simulation model was not addressed in this work.

Kurian [1] developed an optimization-based framework for the calibration of a stochastic microscopic traffic simulator. The various issues affecting a calibration study, namely the formulation of an appropriate objective function, the systematic identification of sensitive parameters, and the effect of stochasticity in calibrating the parameters values were discussed. The methodology started with an initial guess of the parameters and then updated the parameters based on the direction of descent. This methodology can lead to a local optimum; however, a more robust approach has to be adopted to achieve a global optimum.

2.2 OD-Estimation

One of the fundamental inputs to the simulation models is the OD matrix. In practice, it is very rare that a true OD matrix is known. Hence, it has to be estimated using some estimation techniques. OD matrices can either be static or dynamic in nature. A review of static OD estimation methods is presented first.

Static methods used for the estimation of OD flows include entropy-maximization or information-minimization approaches (Vanzuylen and Willumsen [7]) maximum likelihood methods (Maher [8]; Spiess [9]), and generalized least squares methods (Cascetta [10]; McNeil [11]; Bell [12]).

Cascetta [10] broadly classified the estimation techniques into the following three categories:

- Direct sample estimation
- Model estimation

- Estimation using traffic counts

In direct sample estimation, many surveys, such as home or destination interviews, roadside interviews, flagging techniques, etc. or a combination of them are used for estimation of the OD matrices using one of the classical sampling theory estimators. These estimates are often biased because of non-response bias and systematic measurement errors. In the model estimation method, OD matrices are estimated by applying a system of models that give the number of journeys made. The third method uses traffic counts for estimation of OD matrices. This method is less costly to access, as flow measurements are much easier to get as compared to the expensive surveys. Furthermore, traffic counts are repeatable so that the evolution of the phenomenon can be followed.

Ever-increasing urban traffic congestion has stimulated various attempts to estimate OD matrices from traffic counts where congestion effects are considered appropriately via incorporating user equilibrium into the estimation process. Nguyen [13] addressed the OD matrix estimation problem by seeking to reproduce the observed travel time when the OD is assigned to the network in a user-optimal manner. The resulting formulation was shown to be similar to the elastic-demand network equilibrium model for traffic assignment problems. Since such an OD matrix solution has been proven to be non-unique, a secondary objective function is needed to select the most desirable solution. LeBlanc and Farhangian [14] proposed a partial lagrangian method to choose the OD matrix closest to a reference among all feasible solutions.

Cascetta [10] proposed a generalized least squares estimator of the OD matrix, which involves combining trip table's direct or model estimators with the traffic counts via an assignment matrix. He considered two cases. The first one was a more general one in which the estimator was stochastically constrained to the observed flows that were considered to be random variables because of the measurement errors and the temporal fluctuations. In the second case the estimator was deterministically constrained to the observed flows. In both the cases, the variance of the obtained OD was proved to be lower than that obtained with the direct or model estimator.

In another paper, Cascetta and Nguyen[15] presented an in-depth study of a methodology for estimating or updating OD matrices form traffic counts. He formulated the trip matrix estimation or updating as an optimization problem in which the objective function of the latter depends on the statistical inference technique adopted. *Maximum Likelihood* (ML) estimators, *generalized least-squares* (GLS) estimators were examined and contrasted to Bayesian methods for estimating the OD matrices. For each approach, a generic optimization formulation of the estimation problem using as abstract traffic assignment operator was described, and the computational issues were discussed.

The literature discussed so far was related to static OD estimation. The disadvantage of such static models is that they do not capture the dynamics of the traffic flow, and they assume that the observed link flows represent a steady-state situation that persists over a block of time. Several researchers have investigated the problem of dynamic OD estimation.

Cremer and Keller [16] proposed a method for dynamic OD estimation. The OD flows were treated as time-variant variables, which depend on time varying exit and entrance counts. Dynamic models were developed for complex intersections using four different methods: method of cross-correlation matrices, method of constrained optimization, recursive estimation, and estimation using Kalman filtering. Nihan and Davis [17] proposed recursive error methods to estimate the dynamic OD matrices from the entrance and the exit counts. Recursive error method is a recursive formulation of ordinary least squares where all the past observation are taken into account, to estimate the OD matrices from the entrance and the exit counts. Both these works ignored the effect of route choice. This restricts the use of these models only to intersections or small segments of a freeway where the travel time between each OD pair is roughly constant.

Bell [12] determined split parameters for input-output network relationships that represented traffic flows at intersections of road networks or along small freeway segments. He proposed two methods for dynamically estimating the OD flows. In the first method it was assumed that travel time is geometrically distributed and the model made use of the recurrence model of platoon dispersion. This method is suitable for a single intersection because geometric distribution of time is appropriate only over short distances. In the second method, no assumption was made about the travel time distribution, and hence it can be applied to the freeway networks. The disadvantage of these types of models is that they cannot be applied to a general network and are restricted to intersections, junctions or small segments of network corridors.

Chang and Tao [18] proposed a cordonline (hypothetical closed curve that intersects with a set of links and divides the network into two parts: inside and outside the encircled sub network) model for estimation of dynamic OD estimation. The models were developed using Kalman filter approach. The key features of the model are: it does not rely on any prior OD information and it takes into account the effect of signals while computing travel time, and hence can be applied to urban signalized networks. However, the disadvantage of this model is that it requires traffic counts at every entry and exit location, which is a very highly unreasonable demand in case of general networks.

Cascetta et al. [19] in a later work extended his generalized least-squares approach to estimate dynamic OD flows. Two alternative estimation procedures, simultaneous estimation and sequential estimation, were discussed in his paper. The simultaneous approach estimates the OD flows for all the time intervals in one step, using the observed link counts for all the intervals. On the other hand, the sequential estimation technique estimates OD flows for each time interval sequentially. For estimating the OD flows for a time interval, it uses the counts relating to that interval and the previous interval and also the estimated OD of previous intervals. The simultaneous approach gives slightly better results than sequential, but the computational feasibility of sequential estimation makes it more attractive.

Ashok and Ben-Akiva [20] proposed a Kalman filter approach for dynamically updating the OD matrix. The state vector consists of OD flow deviations from the prior estimates that are based on historical data rather than the OD flows themselves. The consideration of deviation variables better supports the normality assumptions but requires knowledge of the historical OD flows in order to estimate the OD trip tables. Furthermore, the measurement equation assumes knowledge of an assignment map that gives the fraction of each OD flows, departing from its corresponding origin during each time period, that would reside on each link during each subsequent period.

Ashok and Ben-Akiva [21] pointed out that the assignment matrix that are used for estimation of OD matrices are themselves estimates and thus prone to have random error. The assignment matrix depends upon the underlying travel time and route choice fractions. Each of these inputs are themselves realization of random events. To incorporate the stochasticity of the assignment matrices in the model, they proposed two approaches. One of them being, adding randomness to the assignment matrix by means of additional equations. The disadvantage of this is that it increases the computation burden. In the other approach, they defined assignment matrix in terms of travel time and route choice fractions. Additional equations were added for underlying travel time and route choice fractions. Again, this increased the computation burden.

2.3 Joint Calibration of Parameters and Estimation of OD Matrices

Even though there are some articles pertaining to calibration and many related to estimation of OD matrices, articles that address both the issues together are rare.

Liu and Fricker [22] tried to estimate an OD matrix and the travel-cost coefficient of the logit-based route-choice probabilities from link traffic counts on uncongested networks. They proposed a two-stage heuristic search method to sequentially estimate the OD and the travel-cost coefficient. In the first stage, the travel-cost coefficient is fixed and the OD is obtained by minimizing the differences between the modeled and the observed link flows. In the second stage, the link flows from the first stage act as the input to calibrate the travel-cost coefficient using the maximum likelihood method. Iterations are repeated until the first derivative of the likelihood value approaches zero. However, apart from its ignorance of congestion effects, implementation of this model requires a complete set of link traffic counts.

Yang et al. [23] proposed an optimization model for simultaneous estimation of an OD matrix and a travel-cost coefficient for the congested network in a logit-based stochastic user equilibrium. The model was formulated in the form of a standard differentiable, nonlinear optimization problem with analytical stochastic user equilibrium. Explicit expression of the derivatives of the stochastic user equilibrium constraints with respect to

OD demand, link flow, and travel-cost coefficient were derived and computed efficiently through a stochastic network-loading approach. A successive quadratic-programming algorithm using the derivative information was applied to solve the simultaneous estimation model. This work however does not use dynamic OD estimation and estimates a static OD. Also, due to the inherent non-convex property of the problem, the technique used might lead to local optima.

Ben-Akiva et al. [24], calibrated MITSIMLab for a mixed urban-freeway network in Brunnsviken area in north of Stockholm under congested conditions. Aggregate data such as speed, flow measurements at sensor locations were used. They used an iterative procedure to take the interactions between the model parameters and the OD flows into account. Figure 2-1 depicts the methodology.

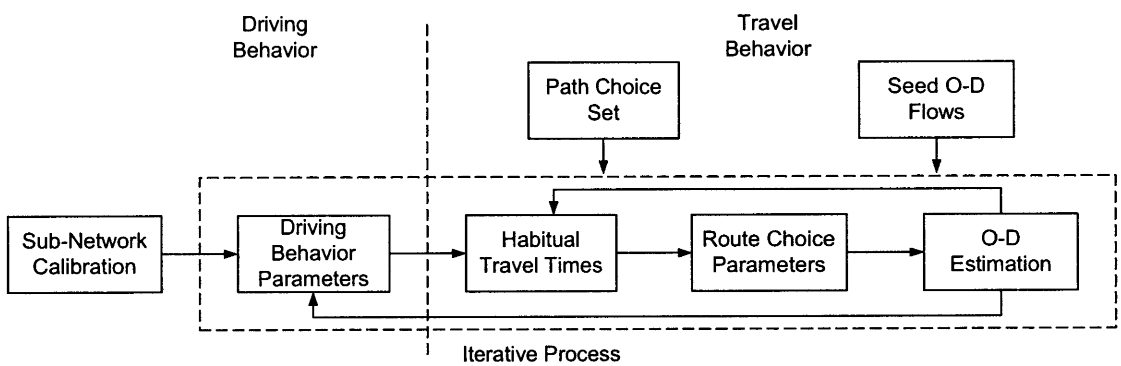


Figure 2-1: Aggregate Calibration of Microscopic Traffic Simulators (Ben-Akiva et al. [24])

MITSIMLab was calibrated for driving behavior parameters and travel behavior parameters (route choice parameters and OD flows). An optimization approach was used for calibration. A sequential approach was used for estimation of OD flows. The results of this work were promising.

2.4 Summary

This chapter discussed the literature related to the calibration of parameters of microscopic traffic simulators, techniques for OD estimation and methods for the joint calibration of microscopic traffic simulators with OD estimation. The current work is intended to overcome the problems of previous approaches and arrive at an automated robust approach for simultaneous estimation of dynamic OD flows and calibration of parameters for a generic traffic simulator.

Chapter 3

Calibration Issues and Methodology

This chapter discuses various issues in calibration, followed by the methodology for calibration of parameters along with estimation of OD matrices. The rest of the chapter is organized as follows: First the issues related to calibration are discussed and then calibration problem is formulated followed by methodology adopted to solve it. The details of the individual components and the algorithms are then presented.

3.1 Issues in Calibration

3.1.1 Parameter Set

The first step in calibration is identification of the parameters that need to be calibrated. The number of parameters in a microscopic simulator is very large. With the increase in the number of parameters to calibrate, the computation time also goes up significantly. Thus, it is crucial to identify the set of parameters that are most sensitive to simulations outputs and therefore, efforts should be made to fine-tune these critical parameters, instead of selecting a large set of parameters.

3.1.2 Computational Issues

As discussed above, a large number of parameters govern the output of simulation. The computation time is highly dependent on the number of parameters that are calibrated. In this research, along with the calibration of model parameters, estimation of OD flows is also performed. Estimation of OD flows by itself is a fixed-point optimization problem in which the OD estimation is dependent on the assignment matrix, which maps the OD flows to the sensor counts. The size of the assignment matrix rapidly increases with the number of OD pairs to be estimated and consequently, the computation time of the OD estimation algorithm also increases exponentially. Hence, an appropriate technique that addresses these computation issues needs to be adopted.

3.1.3 Stochastic Issues

Usually simulators are stochastic in nature and the outputs depend on the random number generation used within the simulator. As discussed earlier, calibration is an optimization problem in which the objective function is minimized. The value of this objective function depends on the output of the simulator. So while computing this objective, the stochastic nature of the simulator should be incorporated.

3.1.4 Measure of Performance

Simulation models can generally provide many measures of performance, including: flow, speed, density, queue lengths, link travel time, total travel time between a given OD pair, average fuel consumption, tailpipe emissions of hydrocarbons, carbon monoxide etc. The choice of the appropriate set of measures of performance would be influenced by the model capabilities, available data and study objectives. Another aspect to be considered is the level of accuracy that is desired. The choice of measure of performance along with the desired accuracy would greatly affect the results of the calibration.

3.2 Problem Formulation

Calibration can be considered as an optimization problem in which an objective function is minimized under various constraints. For calibration of traffic simulation models, the objective function is generally the deviation of the output of the simulator from the observed field values. The deviation can be considered between flow, speed, density or any combination of these values, depending upon the available data. The overall problem is depicted in Figure 3-1.

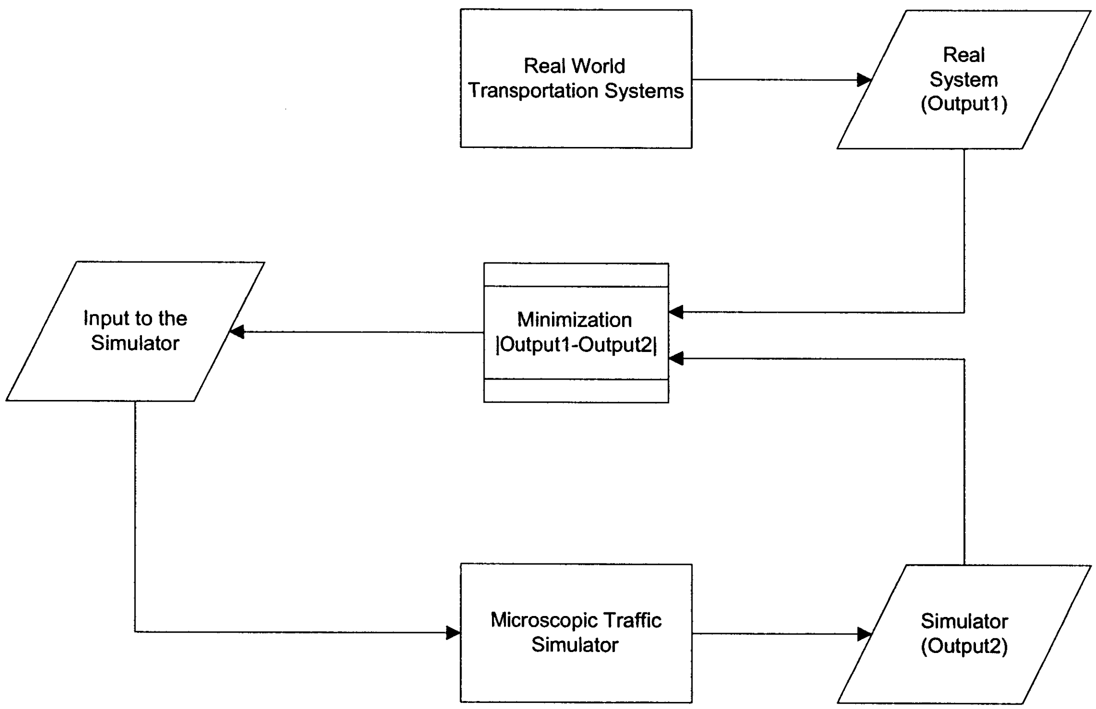


Figure 3-1: Overall Problem

Minimization of the objective function is achieved by estimating OD flows and finding the optimal values of the model parameters namely route choice parameters, driving behavior parameters and steady state travel times. However, there are circular dependencies among all variables listed above. For example, estimation of OD flows requires as an input the assignment matrix, which maps OD flows to counts at sensors, as an input variable. The assignment matrix depends on the steady state travel times, which in turn is dependent on the OD flows. Hence there is circular dependence between the

OD flows and the steady state travel times. If the model parameters are changed, the estimates of OD will also change and vice versa. These interdependencies are depicted in Figure 3-2.

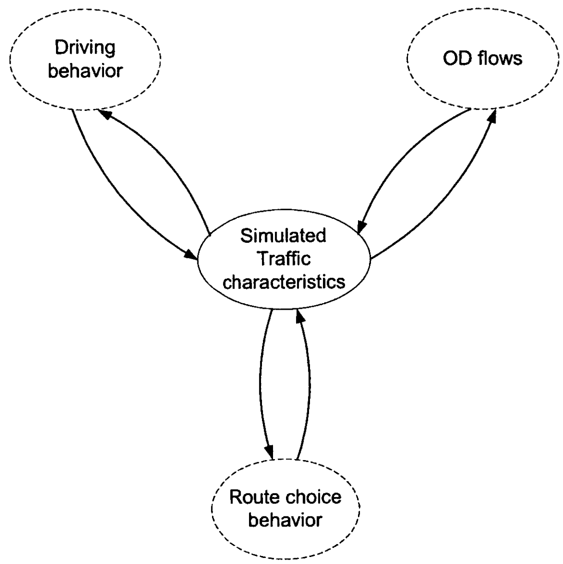


Figure 3-2: Interdependencies between Various Models (Ben-Akiva et al.[24])

Thus, overall calibration problem is a series of fixed-point problems, whose solution is an OD matrix that, once assigned to the network, reproduces flows and travel times consistent with the values used to compute the assignment fractions.

Mathematically, calibration can be formulated as:

$$\underset{\alpha,\beta,OD}{Minimize} \| Output - Observed \|$$
$$Output \xleftarrow{Simulation \quad Model} f(\alpha,\beta,OD,tt^{eq})$$
$$A \longleftarrow h(\alpha,\beta,OD,tt^{eq})$$
$$OD = \arg \underset{x}{Min} \| AX - Y \|$$
$$tt^{eq} = g(\alpha,\beta,OD)$$

Where:

α : Route Choice Parameters

β : Driver Behavior Parameters

OD : Estimated OD flows

tt^{eq} : Steady State Travel Times

A : Assignment matrix

Y : Observed counts

X : OD flows

In the above formulation, the function $f(.)$ is the simulation model. The value of the function is the output of the simulator. The function $h(.)$ is a subroutine that uses the output of the simulator to obtain the assignment matrix. From the above formulation, it is to be noted that the estimation of the OD and the assignment matrix is a fixed-point problem.

3.3 Overall Approach

As is evident from the above formulation, individual effects cannot be isolated and there are complex interactions between various elements of calibration. The iterative calibration methodology outlined in Figure 3-3 takes these interactions into account (Ben-Akiva et al.[24]).

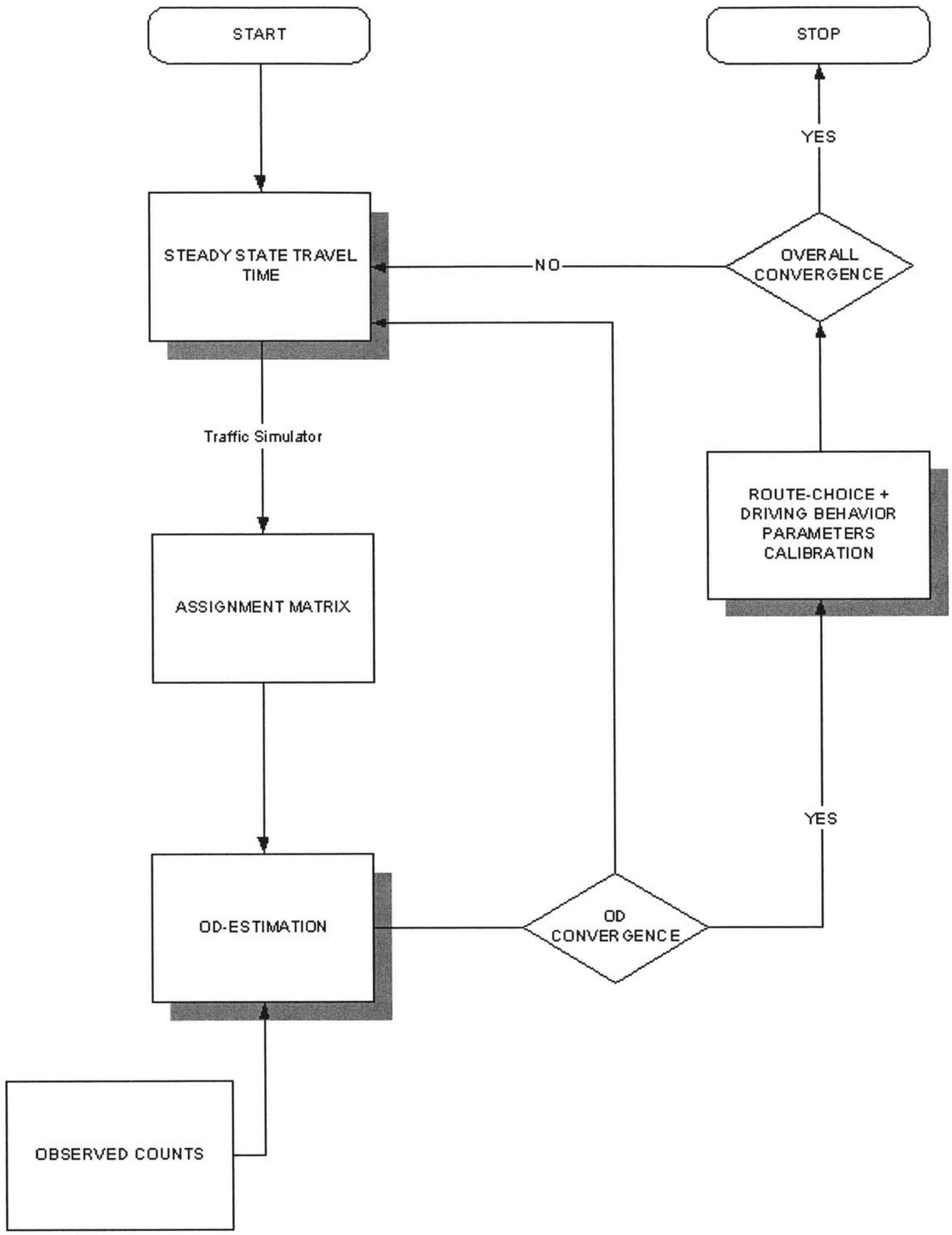


Figure 3-3: Overall Approach for Calibration and OD-Estimation

An initial set of values for the parameters in the route choice and driving behavior models is assumed. The traffic simulator is then used to obtain the assignment matrix. The route choice model uses the current simulated travel times to compute path choice fractions. The assignment matrix and the corresponding OD flows are the inputs to OD estimation module. The module estimates an OD matrix by minimizing the deviation between the observed and simulated link counts. The convergence of the OD matrix is then verified. If the OD does not converge then the new steady state travel times are calculated and fed into the simulator to get the new assignment matrix. This process continues till convergence of OD flows is achieved. The next stage involves changing the values of the calibration parameters and repeating the above steps. The iterative procedure continues until the objective criteria, such as the minimization of deviation between simulated and observed counts and/or speeds is achieved.

There are three modules in the above framework, namely:

- OD-Estimation Module
- Steady State Travel Time Table Module
- Parameter Calibration Module

Since the overall problem is an optimization problem, the optimization approach is discussed first, and then the following sections would describe the above-mentioned modules in detail.

3.4 Optimization Approach

Since the optimization has to be performed on outputs from the stochastic simulator it can be classified as a stochastic optimization problem. Stochastic optimization problems are difficult to solve because the value of the objective function is not explicitly known and cannot be written as a mathematical expression. A summary of the key difficulties in simulation optimization is provided by Banks et al. [25]: "Even when there is no uncertainty, optimization can be very difficult if the number of design variables are large, the problem contains a diverse collection of design variable types, and a little is known about the structure of the performance function. Optimization via simulation adds an additional complication because the performance of a particular design cannot be evaluated exactly, but instead must be estimated. Because we have the estimates, it may not be possible to conclusively determine if one design is better than another, frustrating optimization algorithms that try to move in improving directions. In principle, one can eliminate this complication by making so many replications, or such long runs, at each design point that the performance estimate has essentially no variance. In practice this could mean that very few alternative designs will be explored due to the time required to simulate each one."

A simple way of solving stochastic optimization problem is to use an optimization procedure that needs only function values and not first or second derivatives. In this case, the simulation model can be used to get the function values. To speed up the procedure, a more sophisticated procedure that utilizes the gradient information can be used. However, the simulation program does not typically provide this information. The method that uses the normal output is called black-box method. On the other hand, the method that uses additional information like the gradient of the function is classified as white-box method (Pflug [26]).

Black-box methods (Figure 3-4) are easy to implement. They consist of a simulation module, which is responsible for providing estimates for the objective function F and an optimization module, which uses these values to find the minimizer by iteration. The optimization module must use a method that does not require derivatives.

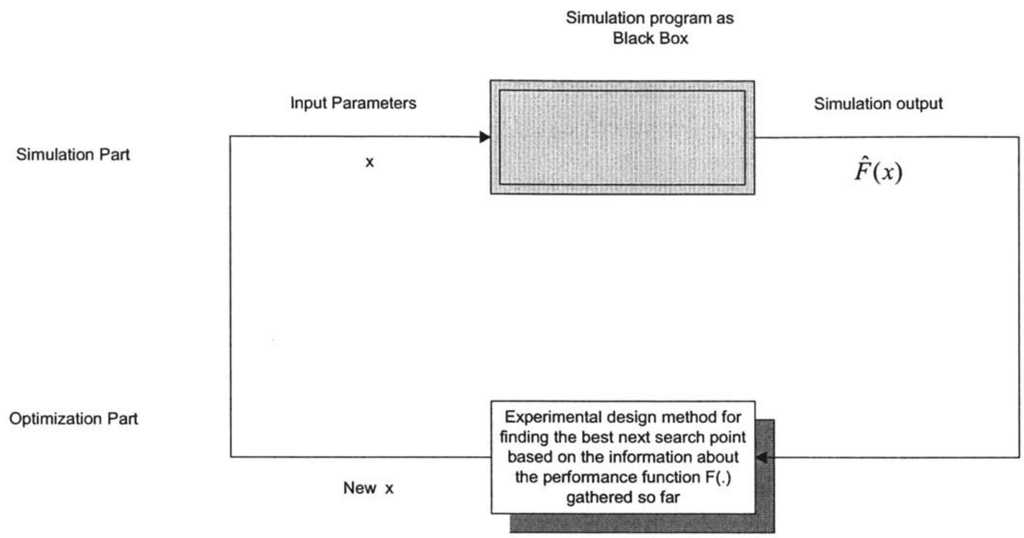


Figure 3-4: The Black Box Approach (Pflug[26])

In contrast, the white-box method uses the gradient, sub-gradients or higher derivatives of the objective function. Optimization with the gradient information has a better rate of convergence; this method is better than black-box method. However, applicability of this approach is limited in the present context, because typical traffic simulation software cannot provide estimates of gradient because the functional form is not known.

3.5 OD Estimation

As discussed earlier, OD matrices are important inputs to the simulator. They are the fundamental input for the problems regarding the planning and management of transportation systems. In practice, the "true" OD matrix is rarely known. As discussed in the literature review, there are broadly three techniques for estimation of OD matrices: direct estimation, model estimation and estimation from traffic counts. Estimation from traffic counts is one of the most feasible methods for obtaining OD matrices.

The inputs to this module are:

- A priori seed matrix.
- Traffic counts from field.

- Assignment matrix obtained from the simulator.

While estimating the new OD flows, the values of the parameters and the steady state travel time is fixed. The basic framework is depicted in Figure 3-5.

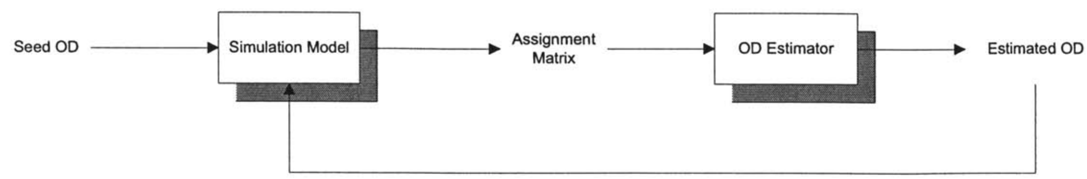


Figure 3-5: OD Estimation Framework (Ben-Akiva et al.[24])

The OD estimation problem can be stated as a constrained optimization problem in which the link counts, seed-OD matrix and an assignment matrix generated by the simulation model are used to get the new estimates of the OD flows. The constraint being imposed is that OD flows are non-negative.

Let the total study period H be divided into n_h intervals $h = 1, \ldots, n_h$ which, without any loss of generality, can be assumed to have equal length T. Let the network have n_l links with sensors and n_r nodes. Let x_{rt} represent the demand between OD pair r that leaves their origin during interval t, x_t the column vector obtained by arranging OD flows. Let x_{rt}^H denote the a priori information, or an initial estimate of the "true demand" x_{rt}, and x_t^H the corresponding vector. For each interval h, a link flow y_{lh} is associated with each link l of the network and y_h is the corresponding vector for interval h. The estimate of the demand is denoted by $\hat{x}$.

There are two key equations that are used in OD estimation (Cascetta [19]):

$$y_{lh} = \sum_{p=h'}^{h} \sum_{r=1}^{n_{OD}} A_{hp}^{l} x_{rp} + v_h^l$$

$$\hat{x} = x + \eta_h$$

A_{hp} is the assignment matrix, which maps OD flows from time period p to the link counts of time period h. A_{hp}^l is the fraction of travelers from r^{th} OD flow which are detected by sensor l during the period h. $h\text{-}h'+1$ is the maximum number of time intervals needed to travel between any origin and destination in the network. v and η are the errors associated with the counts and demand vector respectively.

Cascetta et al.[19] has shown that dynamic estimators can be obtained by solving a constrained optimization problem of the form:

$$\hat{x}_1,...\hat{x}_{n_h} = \arg\min_{x\geq 0} F_1(x_1....x_{n_h}, x_1^H...x_{n_h}^H) + F_2(x_1....x_{n_h}, y_1...y_{n_h})$$

The two functions *F₁(.,.)* and *F₂(.,.)* can be seen as two deviation measures. *F₁(.,.)* is with respect to the deviation from the seed-OD and *F₂(.,.)* is the deviation between the link flows resulting from the assignment of the demand vector and link counts. The functional form of the two functions depends on the assumptions made on the used estimator and the probability distribution of v and η.

As explained earlier, in case of congested networks, the OD estimation problem is a circular problem because the link flow estimates $y(\hat{x})$ and the resulting costs $c(y(\hat{x}))$, are obtained by assigning the estimated demand vector $\hat{x}$ to the network, which in turn is estimated by using the link costs and the link flows (Cascetta and Postorino [27]). So, the problem has to be solved by imposing the condition that the flows correspond to the equilibrium assignment of the demand vector. The equilibrium assignment model can be expressed as a fixed-point problem:

$$y(\hat{x}) = y; \qquad y = A[c(y)]x$$

Cascetta et al.[19] has proposed two techniques for dynamic OD estimation.

- Simultaneous estimation
- Sequential estimation

3.5.1 Simultaneous Estimation

In this approach, the entire OD matrix for all the intervals is estimated in one step. All unknown n_h demand vectors $(x_1, x_2, ...x_h, ..., x_{n_h})$ are placed together as a single vector. Similarly, the flow vector consists of vectors of flows for all intervals, written together, i.e, $(y_1, y_2, ...y_h, ..., y_{n_h})$.

The GLS estimator for the problem is expressed as:

$$(\hat{x}_1 \ldots \hat{x}_{n_h}) = \underset{x_1 \geq 0 \ldots x_{n_h} \geq 0}{\arg\min} \sum_{\mathrm{h}=1}^{\mathrm{n_h}} \left[(\mathrm{x_h} - x_h^H)^T W_h^{-1} (\mathrm{x_h} - x_h^H) + \delta_h^T V_h^{-1} \delta_h \right]$$

where

$$\delta_h = \left(\sum_{t=1}^{h} A_{ht} x_t \right) - y_h$$

Where V_h and W_h are the error covariances associated with link counts and OD flows, respectively.

3.5.2 Sequential Estimation

In this approach, the OD demand vector for a single interval h is estimated at each time interval. The counts of period h are expressed as linear (stochastic) functions of the unknown demand of the same period only. This is achieved by equating the demand relative to previous periods to the already computed estimates $\hat{x}_t$. The GLS formulation of the problem is:

$$\hat{x}_h = \underset{x_1 \geq 0 \ldots x_{n_h} \geq 0}{\arg\min} \left[(\mathrm{x_h} - x_h^H)^T W_h^{-1} (\mathrm{x_h} - x_h^H) + \delta_h^T V_h^{-1} \delta_h \right]$$

where

$$\delta_h = \left(\sum_{t=1}^{h-1} A_{ht} \hat{x}_t \right) + A_{hh} x_h - y_h$$

Sequential estimation reduces the computational complexity by breaking down a large optimization problem into several smaller ones. The other advantage is that the estimates obtained for an interval can be used as initial estimates in the subsequent estimations.

3.5.3 Error Covariance Matrix

In both the formulations discussed above, the error covariance matrices of the sampling errors and the assignment and measurement errors are required. Ashok [28] in his work has suggested a method to obtain the error covariance matrices from historical data. In Ashok's approach, the error covariance matrices are obtained from historical data. The matrix W can be obtained by an OLS regression on deviation of the OD flow in period h

with corresponding historical OD flows. The $(i,j)^{th}$ element of this matrix could be approximated by

$$W_{ij} = e_i^{'} e_j / n$$

Where e is the OLS residual vector (vector of error between the OD flow in period h with corresponding historical OD flow) and n is the number of sample observations

Similarly, matrix V can be obtained from the residuals of measurement equation. Here the residual e_h would be obtained from computing the difference $\left(\sum_{t=1}^{h} A_{ht} x_t\right) - y_h$ over many days. Each day would yield one value for every residual vector e_h. From the values of the residuals e_h over several days, the variance-covariance matrices V can be calculated. Both V and W matrix can be assumed constant for the study period.

3.6 Steady State Travel Times

After completing the assignment matrix consistency loop, optimum OD matrices are obtained for the given set of parameters. The next step in the process is to generate the steady state travel times for all the links in the network. The steady state travel times are generated through iterations. In this process, the OD and parameter values are kept fixed. A smoothing of the previous travel times with the latest travel times is performed to obtain a more reliable travel time matrix. The smoothed travel time is fed back into traffic simulator and the above-mentioned steps are iteratively performed until convergence.

Mathematically it can be expressed as:

$$TT_{it}^{k+1} = \lambda^k tt_{it}^k + (1 - \lambda^k) TT_{it}^k$$

where

TT_{it}^{k+1} : Expected travel time on k^{th} iteration, on link i, during time period t

λ^k : Weight parameter of k^{th} iteration

tt_{it}^k : Simulated (experienced) travel time on k^{th} iteration

3.7 Parameter Calibration

In the next stage of the calibration approach, an enhanced set of parameters is to be obtained. As discussed in Chapter 1, the objective of the work involves determining a systematic search method to avoid ad-hoc selection of the values of parameters and ensure robustness of results. In this step, the OD flows and the steady state travel time are kept fixed.

A sequential search technique based on the Box complex algorithm (Box[29]) is used to determine the optimal values of these parameters. The advantage of using Box algorithm is that, it just requires the value of the objective function and does not depend upon the gradient of the objective function and hence fits well into our black-box approach. It proceeds by continuously dropping the worst point from among the points in the simplex and adding a new point determined by the reflection of this point through the centroid of the remaining vertices.

The objective function, which can be the sum of the squares of the differences between observed flows and/or speeds and simulator-generated flows and/or speeds, is to be minimized. The objective function is first evaluated for a user-specified number of points (i.e. combinations of values of the parameters to be calibrated) and the procedure attempts to move the point yielding the highest objective function value closer to the centroid of the remaining points. The point with the highest objective function value is 'corrected' during each iteration until the objective function value converges to the minimum. A flow diagram detailing the procedure is shown in Figure 3-6.

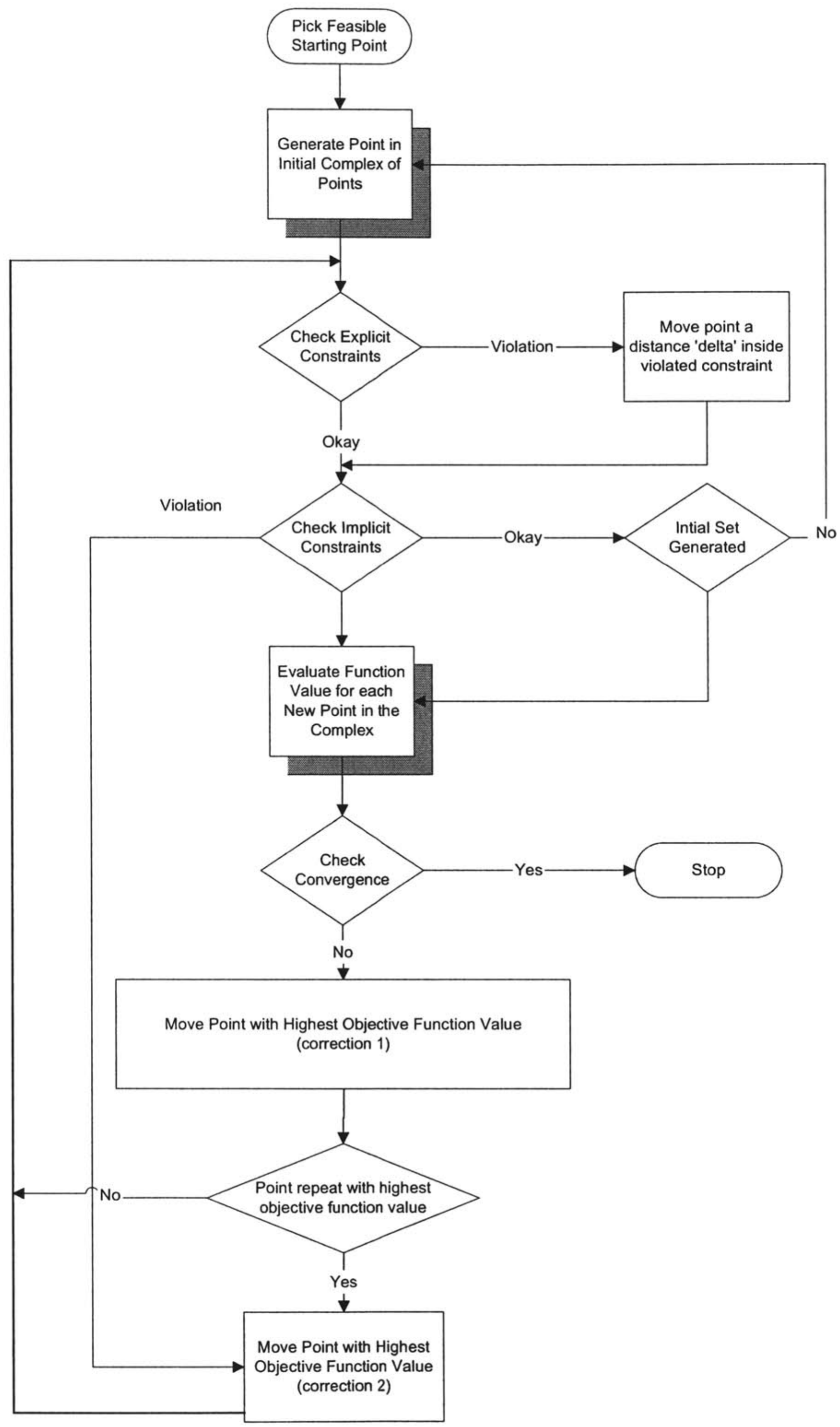


Figure 3-6: The Box Algorithm

Box Algorithm (Kuester and Mize [34]):

The algorithm finds the minimum of a multivariate, nonlinear function subject to nonlinear inequality constraints:

Minimize $\quad F(X_1, X_2, \ldots\ldots\ldots, X_N)$

Subject to $\quad G_k \le X_k \le H_{k,} \qquad k = 1, 2, \ldots, M$

The implicit variables $X_{N+1}, \ldots X_M$ are dependent functions of the explicit independent variables $X_1, X_2, \ldots, X_N$. The upper and lower constraints H_k and G_k are either constants or functions of the independent variables.

Let the objective function be some function $F(X_1, X_2, \ldots\ldots\ldots, X_N)$. It is desired to find the values of $X_1, X_2, \ldots, X_N$ for which $F(X_1, X_2, \ldots\ldots\ldots, X_N)$ is a minimum. A *point* is any combination of values $X_1, X_2, \ldots, X_N$ for which the objective function value will be computed. The algorithm enlists a *complex* of K *points* to search for the minimum of the objective function, where K is an input parameter specified by the user. The basic algorithm is as follows:

Step 1: An original "complex" of K > N + 1 points is generated consisting of a feasible starting point (specified by the user) and K-1 additional points generated from random numbers and constraints for each of the independent variables:

$$X_{i,j} = G_i + r_{i,j}(H_i - G_i),$$

$$i = 1, 2, \ldots, N$$

and

$$j = 1, 2, \ldots, K\text{-}1$$

where $r_{i,j}$ are random numbers between 0 and 1

Step 2: The selected points must satisfy both the explicit and implicit constraints. If at any time the explicit constraints are violated, the point is moved a small distance δ inside the violated limit. If an implicit constraint is violated, the point is moved one half of the distance to the centroid of the remaining points

$$X_{i,j}(new) = (X_{i,j}(old) + \overline{X}_{i,c})/2$$

$i = 1, 2, \ldots, N$

where the coordinates of the centroid of the remaining points, $\overline{X}_{i,c}$, are defined by

$$\overline{X}_{i,c} = \frac{1}{K-1}\left[\sum_{j=1}^{k} X_{i,j} + X_{i,j}(old)\right], \quad i = 1, 2, \ldots, N.$$

This process is repeated as necessary until all the implicit constraints are satisfied.

Step 3: The objective function is evaluated at each point. The point having the highest function value is replaced by a point which is located at a distance α times as far from the centroid of the remaining points as the distance of the rejected point on the line joining the rejected point and the centroid (correction 1):

$$X_{i,j}(new) = \alpha\left[\overline{X}_{i,c} - X_{i,j}(old)\right] + \overline{X}_{i,c}$$

$$\mathrm{i} = 1, 2, \ldots, \mathrm{N}$$

Box recommends a value of α =1.3

Step 4: If a point repeats in giving the highest function value on the consecutive trials, it is moved one half the distance to the centroid of the remaining points (correction 2).

Step 5: The new point is checked against the constraints and is adjusted as before if the constraints are violated.

Step 6: Convergence is assumed when the objective function values at each point are within β percentage for γ consecutive iterations.

To calibrate the parameters, the algorithm proceeds as described above. The module accepts as input the measured flows, speed etc. values from sensors in the real network. Each time the objective function is evaluated, the simulator is called and the outputs from the simulator are used in the objective function. To take care of the stochasticity related to the simulator, the simulator can be run for a number of times and then the average value of the objective function over these runs can be used in the box algorithm.

3.8 Convergence Criterion

There are four points where the convergence has to be verified:

- OD convergence
- Steady state travel time convergence
- Parameter convergence within Box algorithm
- Parameter convergence in the main iteration loop

The termination criterion used for checking the convergence of the OD matrix estimation module is based upon the maximum percent difference between the elements of the new estimated OD (x^k) with the old OD (x^{k-1}).

$$\max\left|\frac{(x_{ij}^k - x_{ij}^{k-1})}{x_{ij}^k}\right| < \varepsilon$$

where:

ε : *Tolerance*

The iterations end when this tolerance criterion is met or when maximum number of iterations has been performed.

The termination criterion for checking the converges of steady state travel time is exactly the same as in the case of OD-estimation, where the OD flows are replaced by the link travel times.

Within the Box algorithm, the convergence is achieved if the difference between the maximum value of the objective function and the minimum value of the objective function is within β percent of the minimum function value, for γ (which is specified by user) consecutive iterations or when maximum number of iterations has been performed.

The convergence of parameters in the main iterative loop is again on the same lines as that of the OD convergence. The convergence criterion is based on the elements of the new estimated parameter value (α^k) with the old value of the parameter (α^{k-1}):

$$\max\left|\frac{(\alpha_i^k - \alpha_i^{k-1})}{R_i}\right| < \mu$$

where:

R_i : Difference between Upper and lower bound of the parameter

μ : Tolerance

The iterations end when this tolerance level is achieved, or when maximum number of iterations has been performed.

3.9 Summary

In this chapter, we formulated the calibration problem and then proposed an iterative methodology to take the interactions between the model parameters and the OD matrices into account. Details of each individual module were then presented. The next chapter focuses on implementing the developed module in a Microscopic Traffic Simulator Laboratory (MITSIMLab).

Chapter 4

MITSIMLab and Implementation

This chapter gives an overview of MITSIMlab, which was the simulator used for testing the module and performing the case studies. The chapter discusses the various models of MITSIMLab and then describes the implementation of the methodology with MITSIMLab.

4.1 Overview of MITSIMLab

MITSIM (MIcroscopic Traffic SIMulation Laboratory) has been developed at MIT by Yang for modeling traffic flows. MITSIMLab is a simulation-based laboratory that was developed for evaluating the impacts of alternative traffic management system designs at the operational level and assisting in subsequent refinement (Yang[30]). Examples of the systems that can be evaluated with MITSIMLab include advanced traffic-management systems (ATMS) and route-guidance systems.

MITSIMLab is a synthesis of a number of different models and has the following characteristics:

- Represents a wide range of traffic management system designs;
- Models the response of drivers to real-time traffic information and controls;
- Incorporates the dynamic interaction between the traffic management system and the drivers on the network.

The various components of MITSIMLab are organized in three modules:

- Microscopic Traffic Simulator (**MITSIM**)
- Traffic Management Simulator (**TMS**)

- Graphical User Interface (**GUI**)

The interactions among the various MITSIMLab modules are shown in Figure 4-1. A microscopic simulation approach, in which the movements of individual vehicles are represented, is adopted for modeling traffic flow in the traffic flow simulator (MITSIM). This level of detail is necessary for an evaluation at the operational level. The Traffic Management Simulator (TMS) represents the candidate traffic control and routing logic under evaluation. The control and routing strategies generated by the traffic management module determine the status of the traffic control and route guidance devices. Drivers respond to the various traffic controls and guidance while interacting with each other.

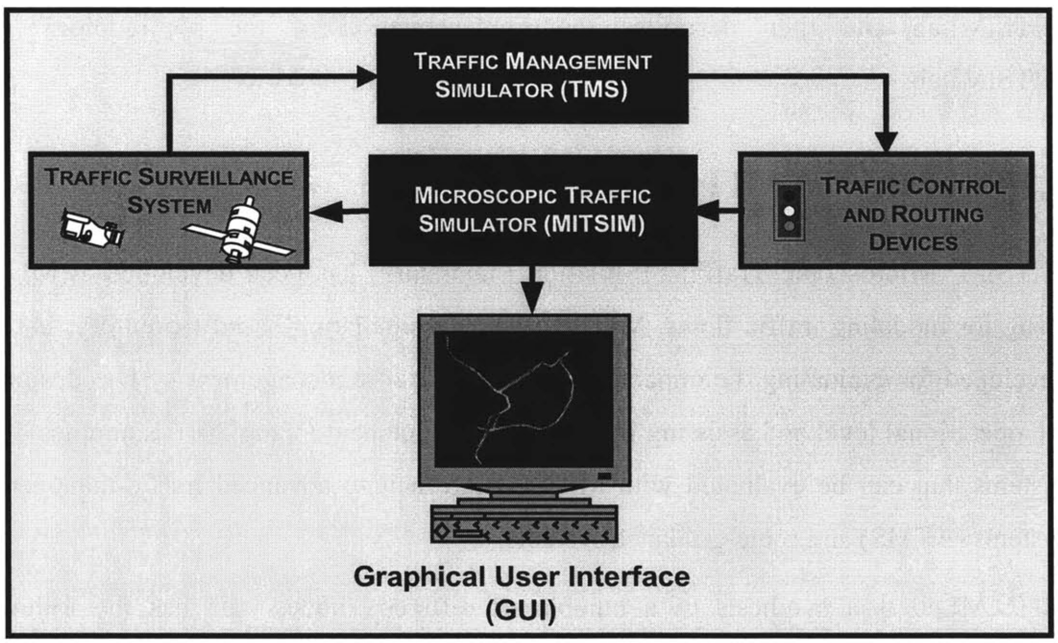


Figure 4-1: Elements of MITSIMLab and Their Interactions

4.1.1 Traffic Flow Simulator (MITSIM)

The role of MITSIM is to represent the "world". The traffic and network elements are represented in detail in order to capture the sensitivity of traffic flows to the control and routing strategies. The main elements of MITSIM are:

- ***Network Components:*** The road network along with the traffic controls and surveillance devices are represented at the microscopic level. The road network consists of nodes, links, segments (links are divided into segments with uniform geometric characteristics), and lanes.

- ***Travel Demand and Route Choice:*** The traffic simulator accepts as input time-dependent origin to destination trip tables. These OD tables represent either expected conditions or are defined as part of a scenario for evaluation. A probabilistic route choice model is used to capture drivers' route choice decisions.
- ***Driving Behavior:*** The origin/destination flows are translated into individual vehicles wishing to enter the network at a specific time. Behavior parameters (such as desired speed, aggressiveness, etc.) and vehicle characteristics are assigned to each vehicle/driver combination. MITSIM moves vehicles according to the car-following and lane-changing models. The car-following model captures the response of a driver to conditions ahead as a function of relative speed, headway and other traffic measures. The lane-changing model distinguishes between mandatory and discretionary lane changes. Merging, drivers' responses to traffic signals, speed limits, incidents, and tollbooths are also captured.

4.1.2 Traffic Management Simulator (TMS)

The traffic management simulator mimics the traffic control system in the network under consideration. A wide range of traffic control and route guidance systems can be simulated, such as:

- Ramp control
- Freeway mainline control
- Lane control signs (LCS)
- Variable speed limit signs (VSLS)
- Portal signals at tunnel entrances (PS)
- Intersection control
- Variable Message Signs (VMS)
- In-vehicle route guidance

TMS has a generic structure that can represent different designs of such systems with logic at varying levels of sophistication (from pre-timed to responsive).

4.1.3 Graphical User Interface (GUI)

The simulation laboratory has an extensive graphical user interface that is used for both, debugging purposes and demonstration of traffic impacts through vehicle animation.

4.2 MITSIMLab Model Descriptions

4.2.1 Driving Behavior Models

In MITSIMLab vehicles move according to the following models (Ahmed[2]) (which are a function of the traffic environment around the subject vehicle, vehicle's individual characteristics, and driver's characteristics):

- General acceleration
- Lane changing and gap acceptance
- Merging
- Forced merging
- Intersection models

General acceleration:

A vehicle accelerates (decelerates) in order to:

- React to the vehicles ahead;
- Perform a lane changing or merging maneuver;
- Respond to events (e.g. red signals and incidents).

The most constraining of these situations determines the acceleration (deceleration) rate to be implemented in the next simulation cycle. Depending on the degree of interaction with the vehicle ahead, the subject vehicle can be in free-flowing, car-following, or emergency regime. The degree of interaction is determined by the time headway between the two vehicles. The acceleration in the free-flowing regime is a function of the vehicle's desired speed, while in the car-following and emergency regimes, the acceleration is a function of traffic conditions and relative position and speed of the two interacting vehicles.

Free-flowing regime: In the free-flowing regime, the vehicle accelerates if its current speed is different from the driver's *desired speed.* The acceleration applied by a driver in this regime is assumed to have the following functional form:

$$\alpha_n^{ff}(t) = \lambda^{ff}\left[V_n^*(t-\tau_n) - V_n(t-\tau_n)\right] + \varepsilon_n^{ff}(t)$$

Where,

$\alpha_n^{ff}(t)$: acceleration of driver n at time t

λ^{ff}: parameter

$V_n^*(t)$: desired speed of the driver

$V_n(t)$: speed of subject vehicle at time t

$\varepsilon_n^{ff}(t)$: error term

Car-following: The car-following model is used for calculating a vehicle's acceleration or deceleration rate in various cases such as:

- Car-following relationship with the leading vehicle;
- Competition with other vehicles if two or more lanes merge into a single downstream lane;
- Yielding to another vehicle shifting into the same lane.

The car following model is a generalization of the non-linear GM model. Furthermore, the parameters of the model can be different for acceleration and deceleration situations. The general structure of the model is shown in Figure 4-2:

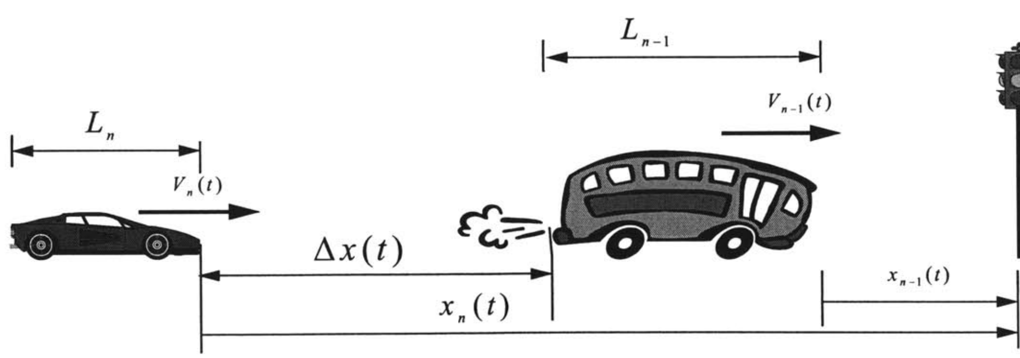


Figure 4-2: Car-Following Model

The car-following model can be expressed mathematically as:

$$a_n^{cf}(t) = \alpha \frac{V_n(t-T)^{\alpha}}{[\Delta x(t-T)]^{\beta}} k^{\delta} [V_{n-1}(t-T) - V_n(t-T)]^{\gamma} + \varepsilon_n^{cf}(t)$$

Where,

$a_n^{cf}(t)$: acceleration of vehicle n at time t;

$\Delta x(t)$: gap between vehicles at time t;

k: density of traffic in the vicinity of the vehicle;

$V_n(t)$: speed of vehicle n at time t;

α, β, γ,δ: parameters

$\varepsilon_n^{cf}(t)$: error term

Emergency regime: In the emergency regime, the vehicle uses an appropriate deceleration rate to avoid collision. The deceleration rate depends on the state of the front and subject vehicles. In all cases though, the applied rate guarantees that the subject vehicle will always decelerate to extend the headway to a safe range.

Lane Changing and Gap Acceptance Models

Lane Changing Model

The lane changing model is implemented in three steps: (a) checking if a change is necessary and defining the type of the change; (b) selecting the desired lane; and (c) executing the desired lane change if the available gaps are acceptable.

Lane changing may be mandatory (MLC) or discretionary (DLC). Mandatory lane changing is performed when the current lane ceases to be an option (due to, for example, lane use regulations, incidents, and need to take exit ramps), and thus the driver must move to another lane. Discretionary lane changing is performed when a driver is not satisfied with the driving conditions in the current lane (due to, for example, average speed of the lane as compared to the driver's desired speed, and existence of heavy vehicles).

The Gap Acceptance Model

The gap acceptance model captures drivers' assessment of gaps as acceptable or unacceptable. Drivers are assumed to consider only the adjacent gap. An adjacent gap is defined as the gap in between the lead and lag vehicles in the target lane (see Figure 4-3). For merging into an adjacent lane, a gap is acceptable only when both lead and lag gaps are acceptable.

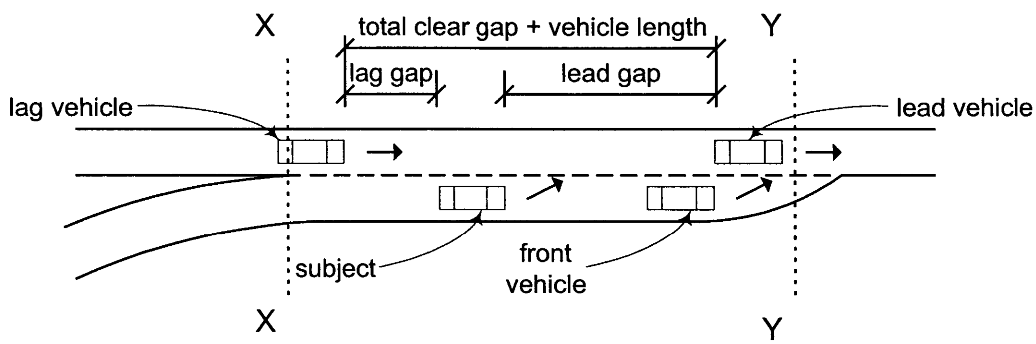


Figure 4-3: Gap Acceptance Model

Drivers are assumed to have minimum acceptable lead and lag gap lengths (lead and lag critical gaps respectively). These critical gaps vary not only among different individuals, but also for a given individual under different traffic conditions. The value of the critical gap is a function of traffic density, distance to the point by which the driver has to complete a mandatory lane change, etc.

Forced Merging Model

In heavily congested traffic, gaps for merging and lane changing are difficult to find. In these situations the driver creates a gap by forcing another vehicle to yield. The probability of forced merging is a function of traffic conditions and characteristics of the subject drivers.

4.2.2 Route Choice Model

In MITSIMLab, drivers make route choice decisions either pre-trip or en-route (based on information received from Variable Message Signs (VMS) or in-vehicle devices). A probabilistic route choice model is used to capture route choice decisions and it has two forms: path- or link-based. The path-based model is path-size logit model (Ramming[31]). The link-based model calculates the probabilities of choosing an outgoing link at each intersection using the formula (see Figure 4-4):

$$p(l \mid j, t) = \frac{\exp\left[\beta\left(\hat{c}_l(t) + \hat{C}_k\left(t + \hat{c}_l(t)\right)\right)\right]}{\sum\limits_{\hat{C}_k(t+\hat{c}_l(t)) \leq \hat{C}_j(t)} \exp\left[\beta\left(\hat{c}_l(t) + \hat{C}_k\left(t + \hat{c}_l(t)\right)\right)\right]}$$

Where:

$\hat{c}_l(t)$: expected time to traverse link l for a vehicle that enters that link at time t;

$\hat{C}_k(t)$: expected shortest travel time from node k to the destination for a vehicle that arrives at k at time t;

β: model parameter.

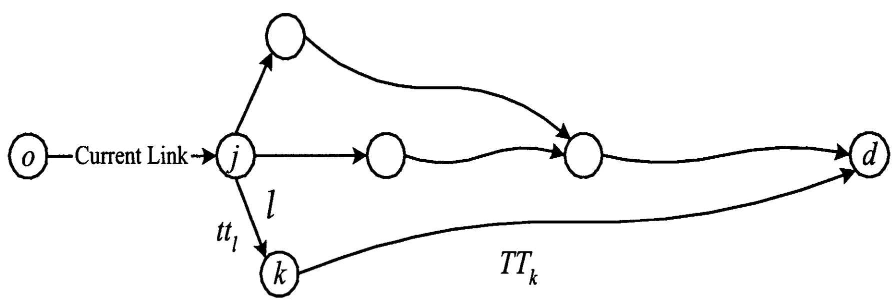


Figure 4-4: Linked-based Route Choice Model

The expected travel time to one's destination for each alternative downstream link at an intersection can be time dependent. The expected travel time depends on the type of information the driver has access to. If no information is available habitual travel times are used.

4.3 Implementation

In order to fit MITSIMLab into the module, certain handshake routines had to be written that converted the MITSIMLab generated output into the format that is taken by the module. Figure 4-5 shows an overall picture of MITSIMLab integrated with the calibration module.

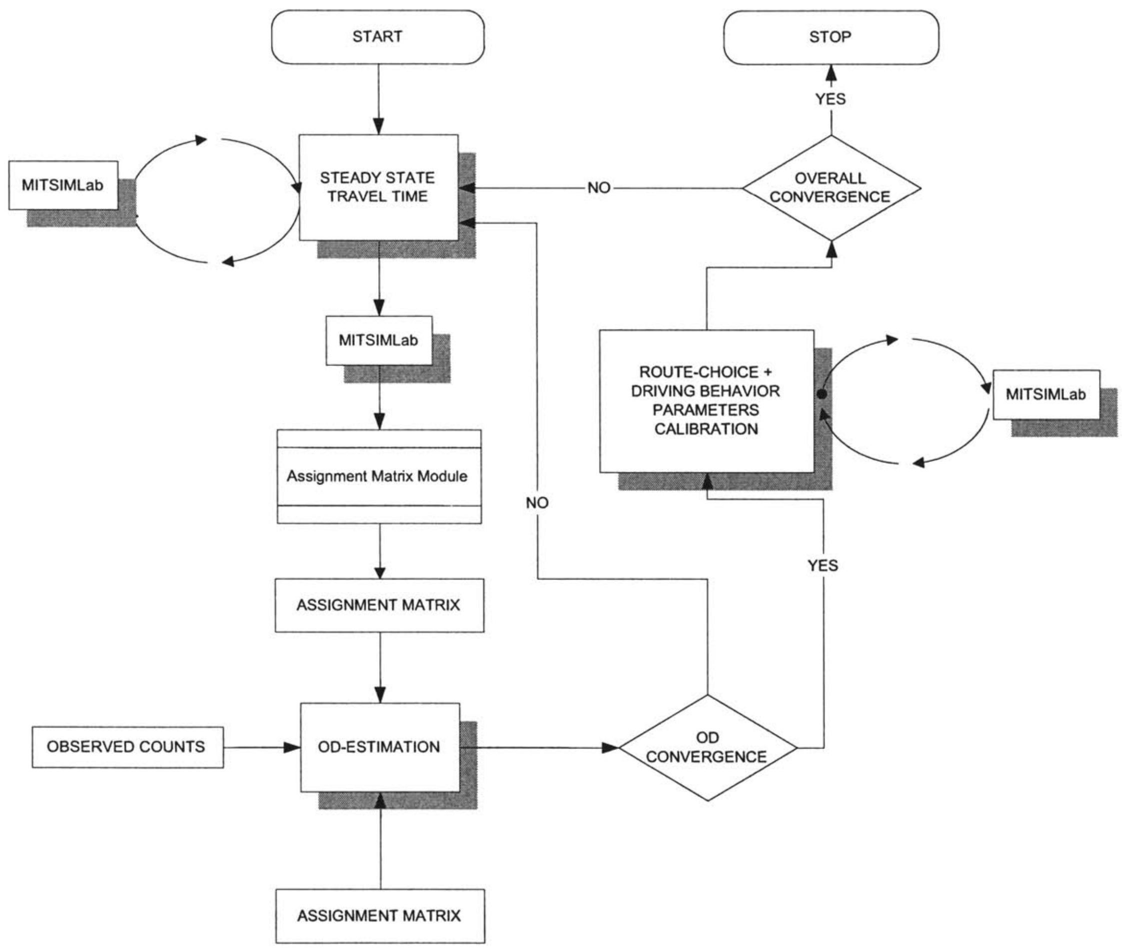


Figure 4-5: MITSIMLab in Calibration Module

4.3.1 Assignment Matrix Module

MITSIMLab is used to generate the assignment matrix. The sensor counts generated using MITSIMLab are translated into an assignment matrix using a matlab program. MITSIMLab produce an output file called "assignment_matrix.out" which has to be processed in matlab to give the assignment matrix. Figure 4-6 depicts the methodology.

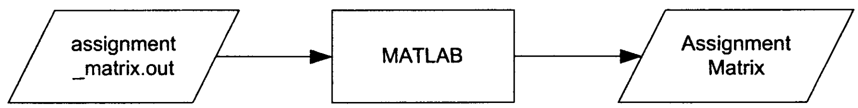


Figure 4-6: Assignment Module

4.3.2 OD Estimation Module

The OD estimation module gives the new estimates of the OD flows in terms of a vector. This output cannot be directly fed into MITSIMLab. Hence, a routine in matlab was written to translate the new estimates of the OD flows into the input that is compatible with MITSIMLab (Figure 4-7).

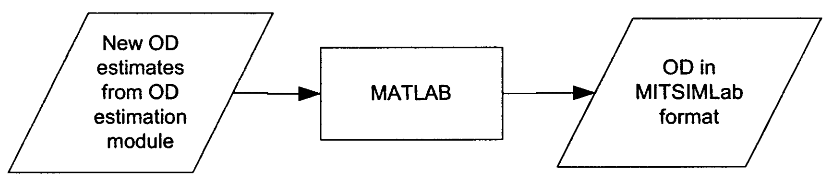


Figure 4-7: OD Estimation Module

4.3.3 Steady State Travel Time

Steady state travel times are generated using MITSIMLab. MITSIMLab outputs the travel time experienced on the network into an output file (linktime.out). This file is fed back into MITSIMLab to generate new estimates of travel times. The process of smoothening is done using a matlab program which takes the input link travel time file and the output link travel time file and does a weighted averaging, to give a new travel time file which than becomes the input for the next iteration. This process is repeated until we get a converged link travel time. Figure 4-8 shows this process.

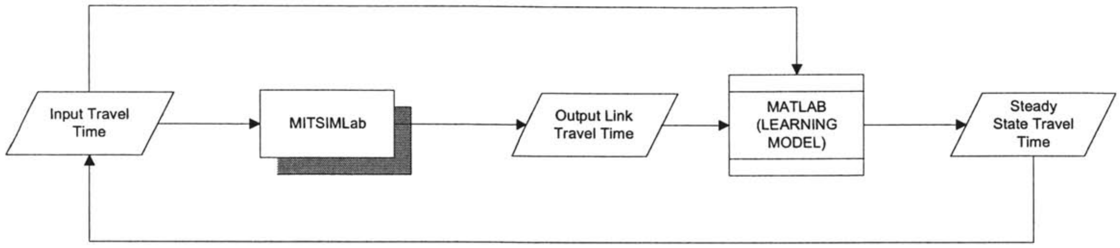


Figure 4-8: Steady State Travel Time

4.3.4 Objective Function

The box complex algorithm evaluates the value of the objective function for a given set of parameters. To evaluate the value of the objective function, the outputs from MITSIMLab are required. Generally the objective function is to minimize the deviation between counts, speed (or a combination of these values) generated by MITSIMLab and the field values of counts, speed etc. The outputs generated from MITSIMLAB have to be processed to give the desired values in the desired format. Appropriate routines are written in matlab to convert the output from MITSIMLab to required format (Figure 4-9).

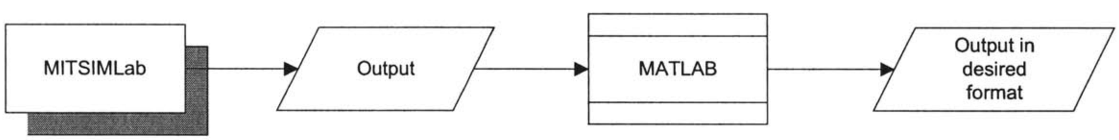


Figure 4-9: Output Processing

The other issues to be considered is that, box algorithm generates user-specified number of sets of parameters and than as it goes along, it drops the set with the worst objective function and generates a new set of parameters. In order to evaluate the objective function associated with the different set of parameters generated by box algorithm, it is necessary to input these set of parameters in MITSIMLab. So it is required to make routines that can send the set of parameters from the box algorithm output to the MITSIMLab input. Furthermore, box algorithm needs the value of the input parameters, so again a routine is required which can read these values from MITSIMLab and pass it to box algorithm. Most of the parameters that are changed in MITSIMLab are written in a parameter file (paralib.dat). Appropriate routines were written in matlab to parse the initial value of the initial set of parameters and than pass it to the box algorithm and routines which can write the set of values generated from box algorithm into the parameter file (paralib.dat). The process is depicted in Figure 4-10.

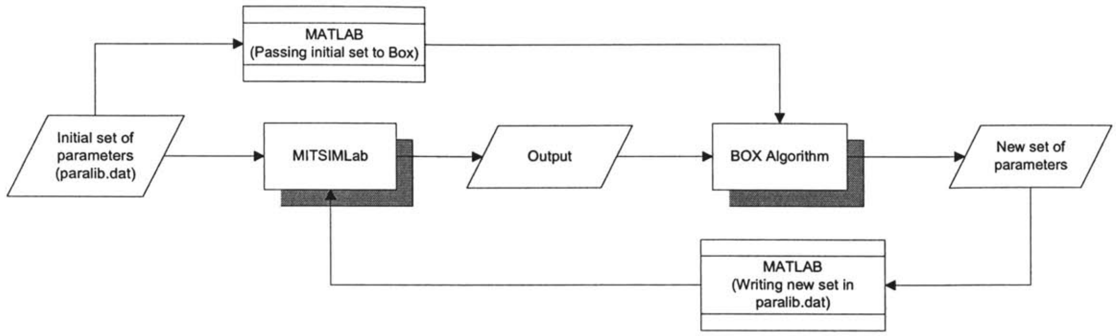


Figure 4-10: MITSIMLab with Box Algorithm

The other issue to be addressed is the stochasticity associated with the simulator. To evaluate the value of the objective function for a given set of parameters, the simulator can be run for a user defined number of times, with the objective function evaluated for each run. Finally, while returning the value of the objective function to the box algorithm, an average of these objective functions can be returned.

4.4 Summary

In this chapter an overview of MITSIMLab and its models was provided. The implementation of calibration framework in MITSIMLab was discussed. The next chapter uses the implementation in MITSIMLab for the calibration of three networks.

Chapter 5

Case Studies

This chapter presents three case studies performed using the module. The simulator used in these studies is MITSIMLab.

5.1 Case Study 1: The HCQS Network

5.1.1 Project and Network Description

The Highway Capacity and Quality of Service (HCQS) committee of Transportation Research Board has outlined a test case to be used to address the calibration and validation issues for simulation and the capacity analysis of various traffic simulation models. The purpose of this is to serve as a common ground for exhibitors/demonstrators to showcase their products' latest features in the HCQS exposition of simulation models. The dataset comprised calibration data (speed, density etc.), traffic volumes, signal timings and information about the basic network geometry.

The network consists of a 5-mile freeway with two interchanges, and is shown diagrammatically in Figure 5-1. There are altogether four signalized intersections with pre-timed signals.

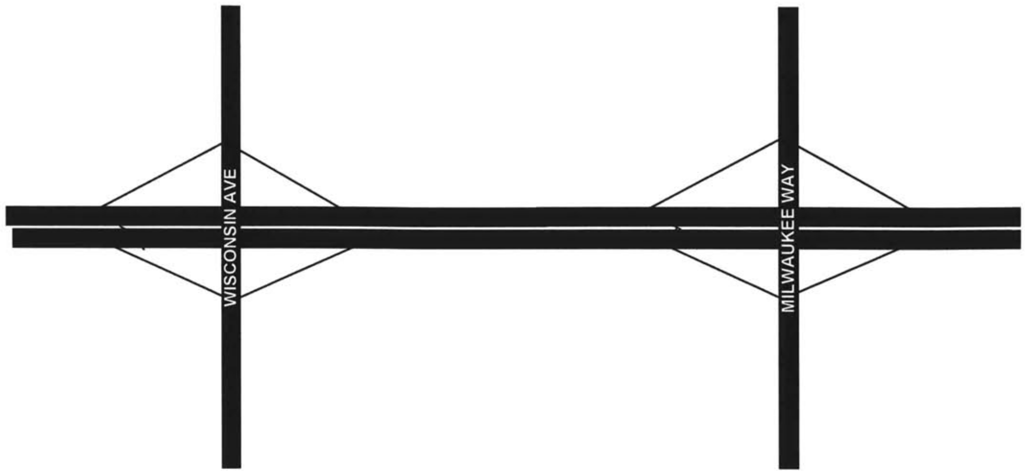


Figure 5-1: HCQS Network

5.1.2 Data

Data from 5:30-5:45 AM was available for calibration. The available data comprised:

- Average speeds at mid-sections of the freeway
- Average speed on the arterial segments
- Average density on the freeway segments
- Average Travel Times on the freeway segments
- Average Travel Times on the arterial segments

The simulation was performed from 5:00 AM to 6:00AM. The OD flows were estimated using traffic volumes on the freeway along with entrance counts on the on and off ramps. All signals were pre-timed signals, with a constant 90-second cycle length and differing offsets. The volume splits at the intersections and the signal timings were provided. In the weaving section, 10% of on-ramp traffic exits at the next off-ramp (ramp-ramp movement). It was given that there is 95% peak hour factor for all the volumes, adjusted for the third 15-minute period.

5.1.3 Input Files for MITSIMLab

The network was coded in MITSIMLab using the information provided by HCQS committee and the signals and sensors were placed at the appropriate locations. The pre-timed signals were coded accordingly. The link volumes for the entire network and the splits at the interchanges were used to generate the dynamic OD matrix.

5.1.4 Implementation of the Module

Choice of parameters: Since there is no route-choice associated with this network, only the car-following parameters were selected for calibration. In the car-following model, the scaling factor α (Chapter 4) captures the effect of unobserved variables that are not captured by coefficients of the explanatory variables. Hence α was chosen as a calibration parameter. There are two values of α; one is associated with acceleration and the other is associated with deceleration. Both these parameters were chosen for calibration. Along with these parameters a scaling factor for scaling vehicle-desired speeds was used. The lower and upper bounds of these parameters were set as shown in Table 5-1. The range was decided based on the default value of these parameters.

	Alpha (Acc)	*Alpha (Dec)*	*Scaling Factor*
Lower Bound	*0*	*-0.1*	*0.8*
Upper Bound	*0.1*	*0*	*1.2*

Table 5-1: Upper and Lower Bound for Chosen Parameters

Objective Function: Based on the available the data, the objective function chosen is given by equation shown below:

$$\sum_{i=1}^{l}(v_i - v_i^{'})^2 + \omega\sum_{j=1}^{m}(k_j - k_j^{'})^2$$

Where:

v_i : Observed Field Speed.

$v_i^{'}$: Speeds from MITSIMLab.

l : Number of sensors.

k_j : Observed Field density for segment j.

$k_j^{'}$: Density for segment j from MITSIMLab.

m : Number of segments for which densities are reported.

ω : Relative weight of density function with respect to speed function

In this study, the deviations in observed and simulated densities and speeds were treated equally and hence, $\omega = 1$ was employed. The Box algorithm module was run to generate six sets parameters and consequently better sets of parameters. To take care of the stochasticity associated with MITSIMLab, for each set of parameters, MITSIMLab was run three times and an average value of the objective function over these three runs was returned to the box algorithm.

5.1.5 Results

The different sets of parameters generated by the Box algorithm and the values of the objective function associated with them are given in Table 5-2.

	Alpha(Acc)	*Alpha(Dec)*	*Scaling Factor*	*Function Value*
		Initial Set		
1	*0.0400*	*-0.0420*	*1.0000*	*459.7*
2	*0.0665*	*-0.0110*	*0.8764*	*556.4*
3	*0.0870*	*-0.0265*	*0.9690*	*595.9*
4	*0.0010*	*-0.0313*	*1.1400*	*24674.2*
5	*0.0137*	*-0.0654*	*0.9961*	*17862.9*
6	*0.0819*	*-0.0834*	*1.1264*	*466.1*
		Iteration		
1	*0.0999*	*-0.0644*	*0.8001*	*455.0*
2	*0.0999*	*-0.0195*	*0.9002*	*589.0*
3	*0.0654*	*-0.0668*	*0.9038*	*484.0*
4	*0.0328*	*-0.0977*	*0.9949*	*4180.0*
5	*0.0518*	*-0.0756*	*0.9681*	*531.0*
6	*0.0686*	*-0.0322*	*0.9089*	*583.0*
7	*0.0697*	*-0.0429*	*0.9251*	*518.0*
8	*0.0613*	*-0.0646*	*0.9547*	*486.0*
9	*0.0702*	*-0.0482*	*0.9332*	*474.0*
10	*0.0660*	*-0.0590*	*0.9480*	*484.0*

Table 5-2: Parameters Sets and Corresponding Objective Function Value

It is evident from Table 5-2 that the Box algorithm searches the multidimensional search space and tries to reach the global optimum. . The box algorithm initially generates six sets of parameters. It then evaluates the objective function value associated with each set of parameters. Then it moves the point returning the highest function value, towards the centroid of the remaining five points. For e.g. in Table 5-2, out of the 6 parameter sets generated, the fourth set yielded the highest objective function value of 24674.2. Therefore, in the first iteration the fourth set was replaced by a new set of parameters. This replaced set of parameters yielded an objective function value of 455.0 as shown in Table 5-2 (corresponding to the first iteration). Thus, now among the six sets of parameters, the set with the objective function value of 17862.9 is the highest and is subsequently replaced by the box algorithm (as shown in iteration 2). This process continues until all the points converge towards the minima. Table 5-3 summarizes the highest and lowest values of the objective function obtained after each iteration.

	Objective Function Value	
	Highest	Lowest
Initial Set	24674.2	459.7
	Iteration	
1	17862.9	455.0
2	595.9	455.0
3	589.0	455.0
4	4180.0	455.0
5	556.4	455.0
6	583.0	455.0
7	531.0	455.0
8	518.0	455.0
9	486.0	455.0
10	484.0	455.0

Table 5-3: Highest and Lowest Values of the Objective Function

In the initial iterations, the Box algorithm tries to search the entire search space, so the value of the objective function fluctuates, but it starts to converge after a few iterations (after iteration 5).

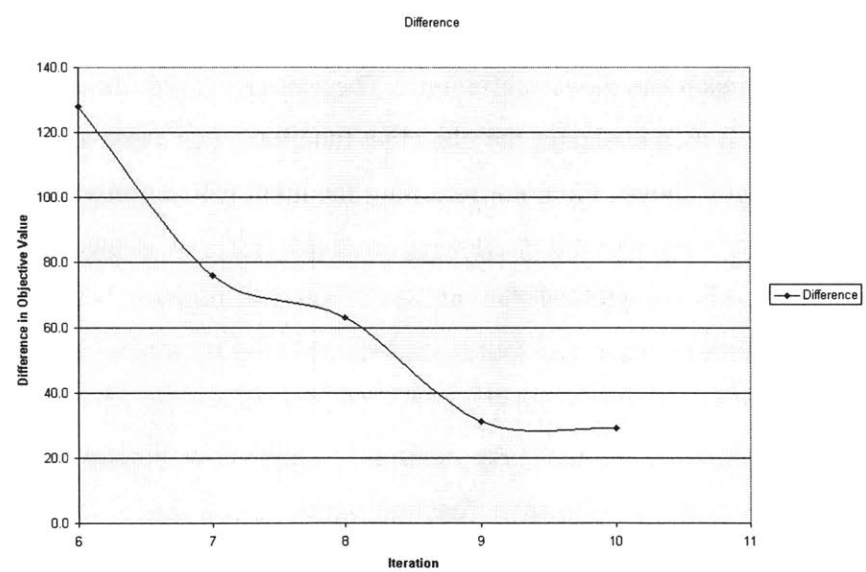
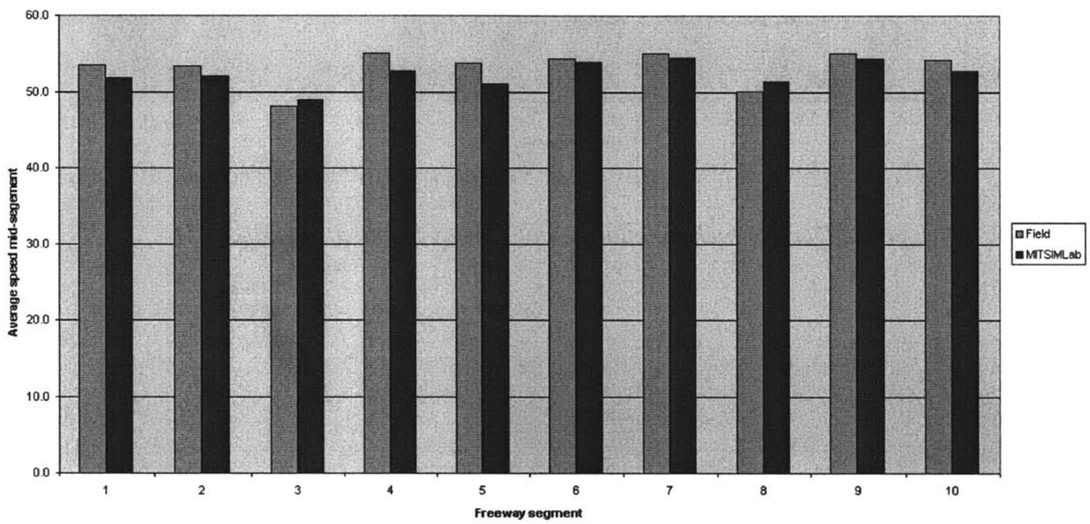


Figure 5-2: Differences in Highest and Lowest Objective Function Value

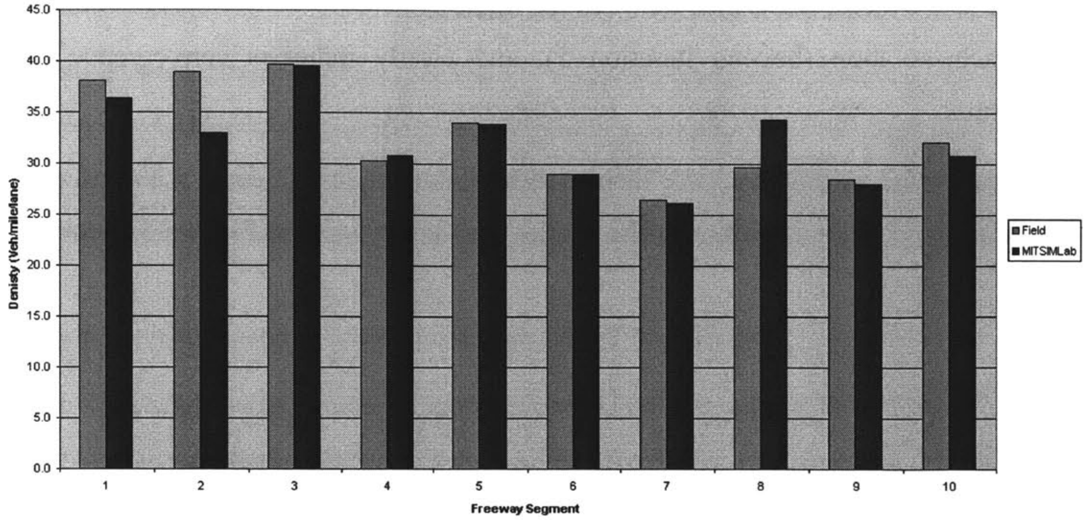
Figure 5-2 shows the difference between the highest and lowest objective function values after each iteration (beyond iteration 5), and clearly indicates convergence to the minimum.

These calibrated parameters were then used to obtain the speed, density and travel time on freeway and arterial locations. The results are shown below in Figures 5-3 through 5-5.

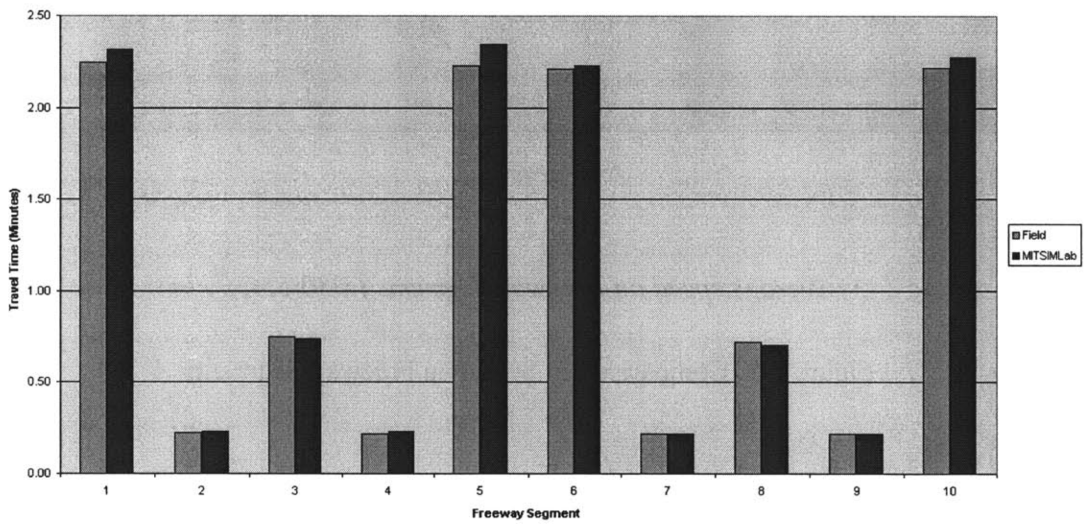


Average speed on Freeway Segments (5:30-5:45)

Figure 5-3: Comparison of Speed on Freeway Segments

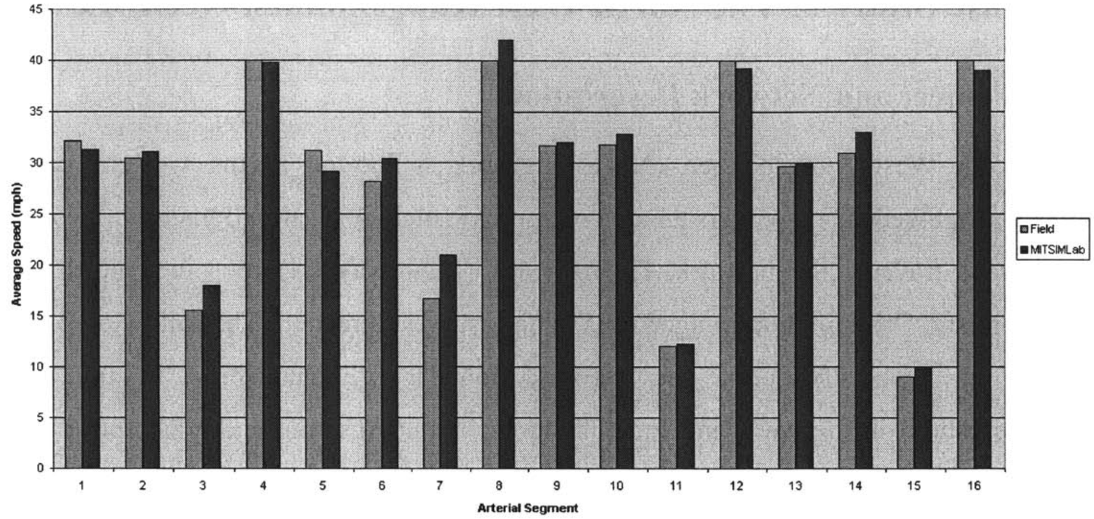


Average Density (veh/mile/lane) on Freeway Segments (5:30-5:45)

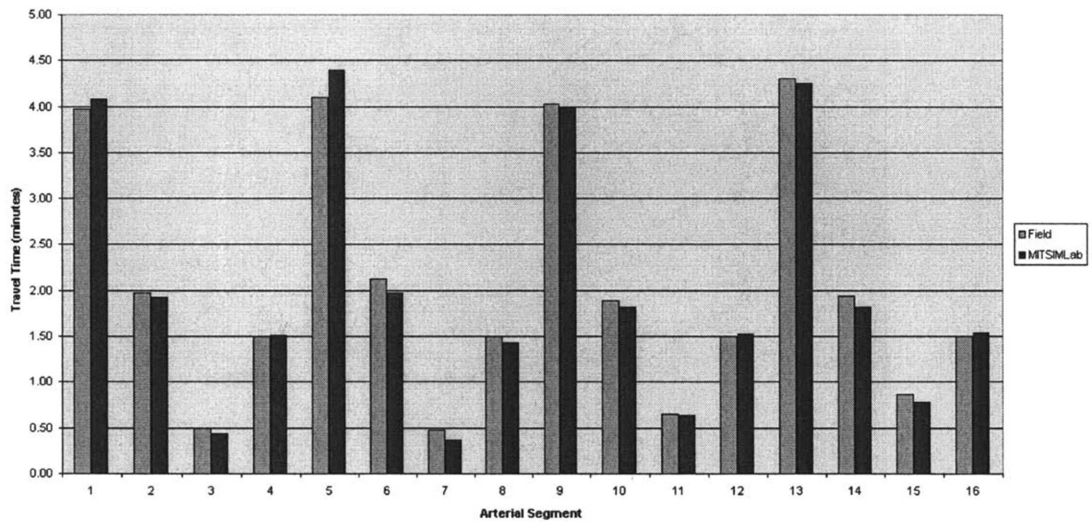


Average Travel Time on Freeway Segments (5:30-5:45)

Figure 5-4: Comparison of Density and Travel Time on Freeway Segments



Average Speed on Arterial Segments (5:30-5:45)



Average Travel Time on Arterial Segments (5:30-5:45)

Figure 5-5: Comparison of Speed and Travel Time on Arterial Segments

The results show a good match with the field data. The maximum percentage difference between the field data and MITSIMLab results of speed, density, and travel time on freeway was 5%, 5% and 15% respectively. Along the arterial segments, the maximum difference between the field and MITSIMLab speed and travel time was less than 27% and 20% respectively.

5.2 Case Study 2: The North West Kungsholmen Network

5.2.1 Project and Network Description

The North West Kungsholmen (NWK) network is located in the western part of Stockholm, Sweden, and covers an area of approximately 4 square kilometers. Two main motorways traverse the network. Essingeleden (Motorway E4) runs in a north-south direction, and Drottningholmsvagen runs in an east-west direction. The following bridges ("bron" in Swedish) define the network boundary: Ekelundsbron in the north, Tranebergsbron in the west, and Fredhällsbron, Mariebergsbron, and Västerbron in the south. Central Stockholm is located 2 kilometers east of the network. This section of Stockholm consists of industrial and residential developments. The area is highlighted in Figure 5-6 below.

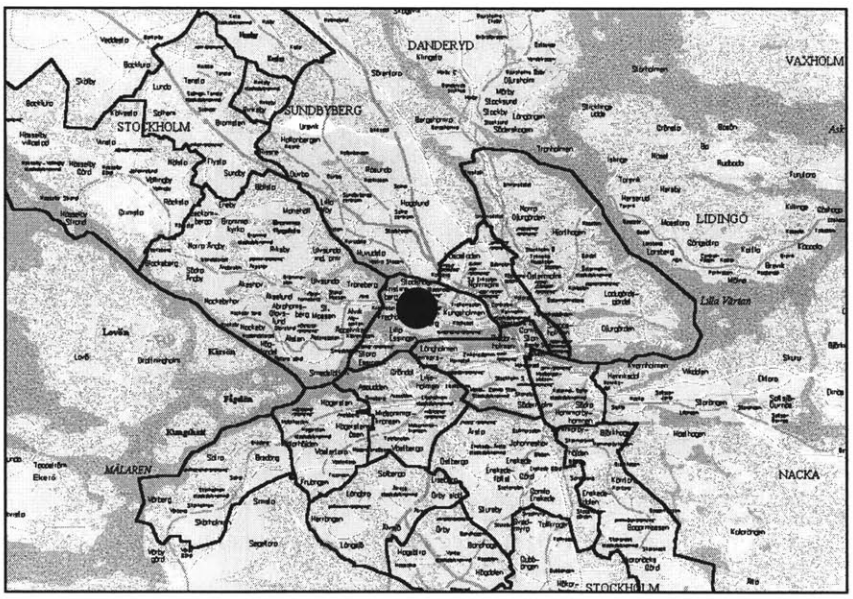


Figure 5-6: Network Location

Public Transportation in Stockholm is provided by Stockholm Localtrafik (SL). The system has a fleet of blue and red buses, and a subway system through the city. Signal priority for bus operations is used in the network, along Lindhagensgatan. Three signals operate under PRIBUSS, a signal priority strategy developed for use in Stockholm

(Cortes[32]). Figure 5-7 shows a diagram of the area, including the transit routes and the signal priority locations.

● PRIBUSS Signal Priority Locations

Figure 5-7: Network and Location of Signals

5.2.2 Data

Daily traffic volumes for year 1996 on selected links were provided by GFK (Cortes[32]). Preliminary OD data and directional volumes for the year 2000 for the morning peak hour (7 AM - 8 AM) were obtained from EMME/2 output and provided by SCC. The GFK data was thus normalized to the year 2000 for analysis, using growth rates determined by annual population statistics for the Stockholm area. As the morning peak hour was simulated for the NWK network, peak hour volumes were extracted from the daily GFK volumes using a peak hour traffic/daily traffic ratio (peak hour factor) of 10%, the value used by GFK. Traffic volume distributions over a typical weekday, when available,(such as at the Kristineberg and Fredhall junctions), was combined with peak hour factor to reflect a more accurate ratio. After this preliminary procedure, traffic volume data was thus estimated for the weekday morning peak hour in the year 2000 at various point locations throughout the network.

5.2.3 Input Files for MITSIMLab

The network was coded in MITSIMLab using the information provided by GFK, including construction drawings, transit maps and schedules, aerial photographs, and initial studies. Roadway geometry was determined from the drawings and photographs where possible; elsewhere, minor roadways within the network were assumed to consist of one lane in each direction. The network has a total of 122 links and 64 nodes. As previously described, three PRIBUSS signals are included in the network along with a total of 10 SL bus routes. Figure 5-8 shows the simulation network as represented in MITSIMLab.

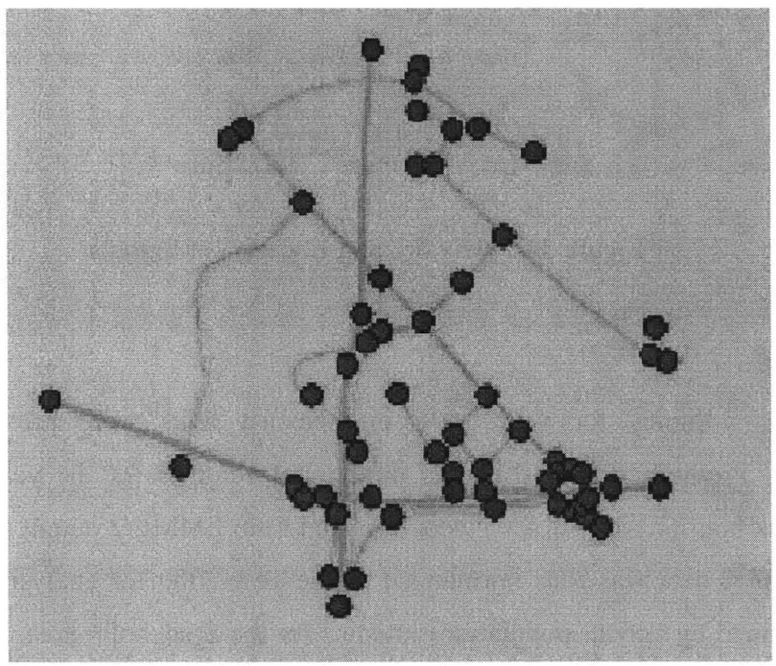


Figure 5-8: Network in MITSIMLab

5.2.4 Implementation of the Module

MITSIMLAB has been previously calibrated and validated for the Stockholm drivers (Ben-Akiva et al. [24]). Hence in this study, parameters were not calibrated and only OD-estimation was done using the counts at various sensor locations. Since not much data were available, the error covariance matrices of both counts and the OD flows were initially assumed to be identity and then were adjusted. The module started by computing the steady state travel time (5 iterations) and then iterations for OD estimation (5 iterations) were performed. After each iteration, the new set of steady state travel times was computed and fed back to MITSIMLab.

5.2.5 Results

The results of OD estimation are shown in Figure 5-9 they show a very good match with the field data.

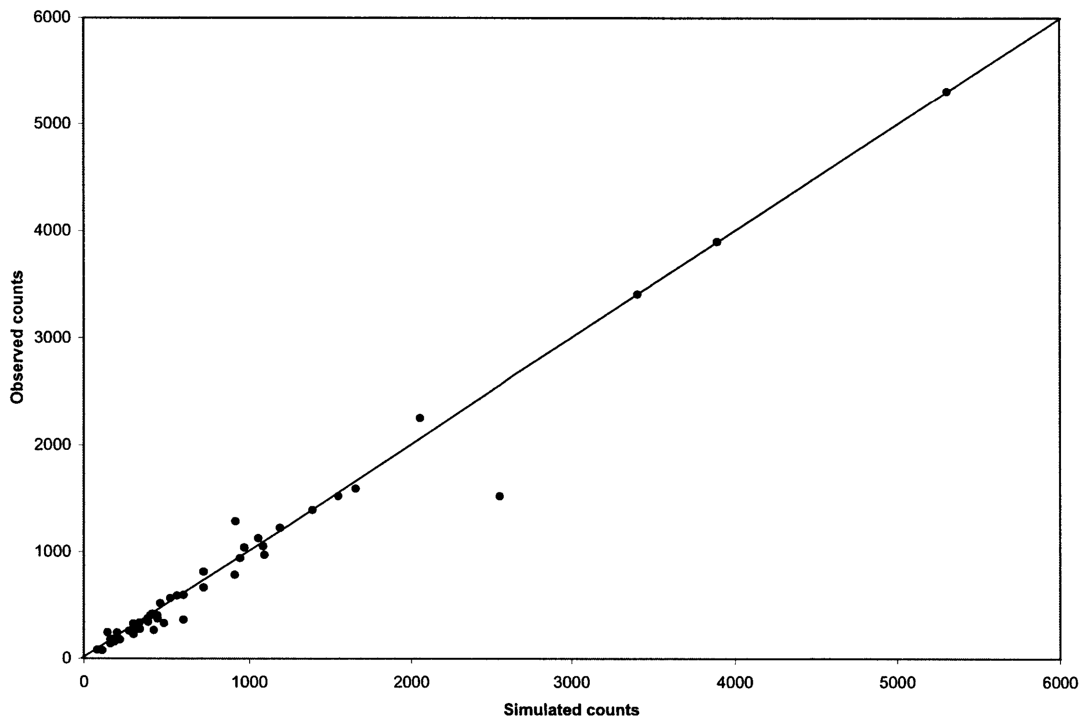


Figure 5-9: Comparison between Observed and Simulated Counts at Sensor Locations

5.3 Case Study 3: The Irvine Network

The data used in this research were collected from Irvine, a part of District 12 in Orange County, California, USA.

5.3.1 Network Description

The study network (Figure 5-10) is comprised of three major freeways and a dense network of urban arterials. The I-5 and I-405 Interstates, along with State Route133, define a wedge that is intersected by several major arterials. The city of Irvine is located in Orange County, just outside Los Angeles. It is a major commercial and business center, and serves as an important regional airport. Irvine is also home to several schools and universities. The city therefore attracts a varied mix of commuters and travelers.

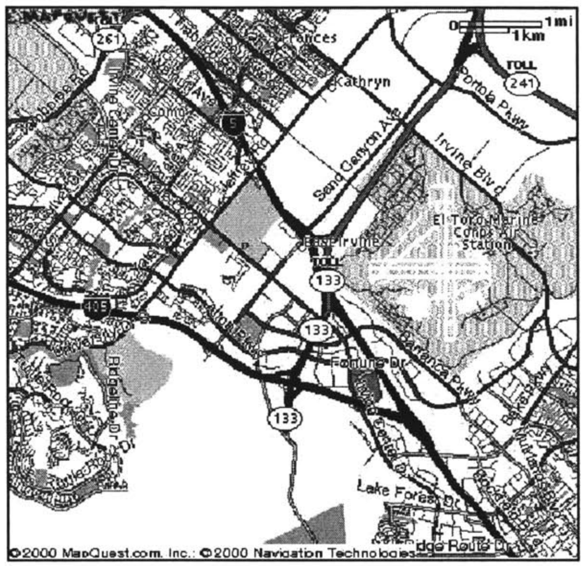


Figure 5-10: The Irvine Network

The network is represented as a set of 298 nodes connected by 618 directed links in MITSIMLab. These links represent the physical links on the network, and are further subdivided into 1373 segments to model the changing link section geometry. Almost all of the 80 intersections within the study area are signalized, and are controlled by vehicle-actuated signal logic. A high fraction of the signals along the primary arterials (Barranca Park-way, Alton Parkway and Irvine Center Drive) are coordinated.

5.3.2 Data and Analysis

The data was derived primarily from four sources:

- PARAMICS network files
- OD flows from OCTAM planning study
- Time-dependent detector data
- Signal timing and coordination plans

Information regarding the network was contained in a set of input files created for the PARAMICS traffic simulation system. These files included descriptions of network geometry, link and lane connectivity, sensor locations and signal timing plans.

The OCTAM planning study generated a static matrix of OD flows covering the morning peak period. While this matrix contained every possible OD combination from 61 zones, several of these flows were zero. In this study, a subset of 655 primary OD pairs with non-zero flows was extracted from the static matrix. A review of the static OD matrix showed that freeway-based OD pairs contributed to a major proportion of the total demand. Figures 5-11 indicates the primary OD pairs. The thickness of the lines connecting the origins and destinations is a measure of the magnitude of the OD flow.

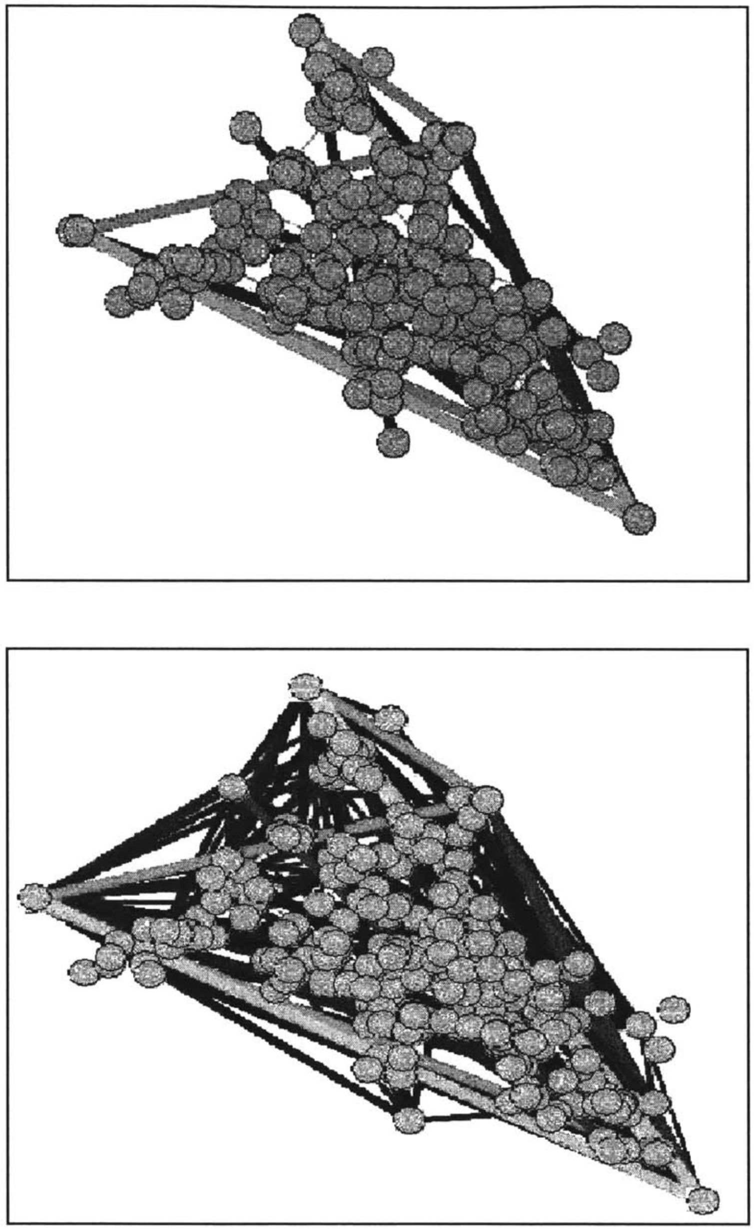


Figure 5-11: Primary OD pairs

Time-varying freeway and arterial detector data recorded over 5 working days was available from California Department of Transportation (Caltrans). This data consisted of counts and occupancies measured by lane-specific sensors on freeway links and lane-group-specific sensors on arterial links. While the freeway detectors reported data every 30 seconds, the arterial detectors were aggregated over 5 minute time slices. It should be noted that only 68 out of 225 sensors reported usable data. The remaining detectors were either located outside the study area, or their data files were inconsistent. Out of these 68 usable sensors, 30 sensors were located on freeway and ramps, and 38 on arterial.

The data collected by the surveillance system consisted of vehicle counts and occupancies. The counts were aggregated into common intervals of length 15 minutes. Further analysis indicated that there was not much day-to-day variability in the sensor data, even though the data was collected on 5 different days of the week (Figures 5-12 and 5-13 (Balakrishna [33])).

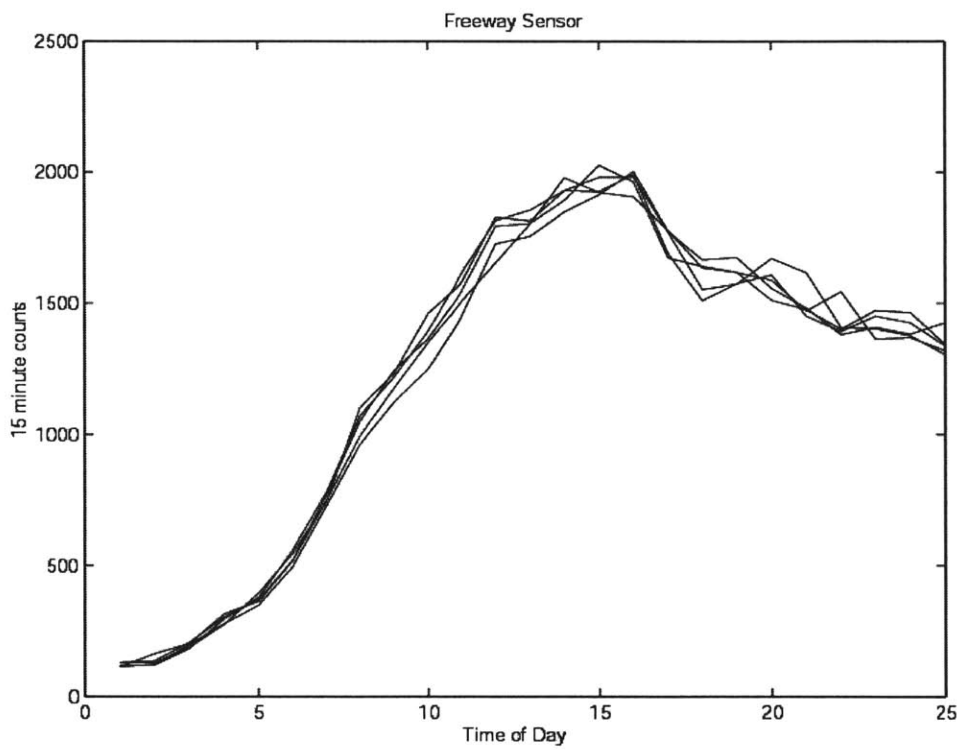


Figure 5-12: Counts Variation Across Days: Freeway Sensor

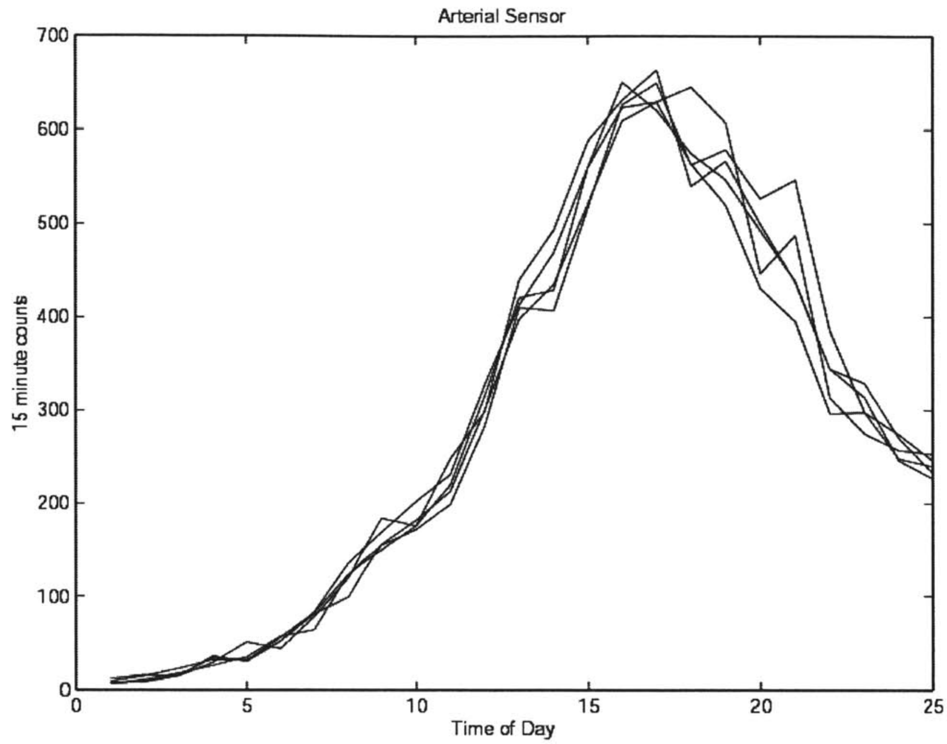


Figure 5-13: Counts Variation Across Days: Arterial Sensor

Signal timing and coordination charts from the City of Irvine specified the details regarding signal phasing, timing, actuation and coordination.

5.3.3 Input files for MITSIMLab

The network file describes the locations of nodes, links, segments and sensors, and defines the connectivity between individual lanes in the network. Other then the network file, MITSIMLab requires signal files as an input. Generic controllers were coded using the signal plans provided.

The OD demand file specifies the time-varying OD flows for each OD pair in the network that has a non-zero flow. between it. A seed OD matrix constructed for calibration of DynaMIT (Balakrishna[33]) was used as the seed for OD estimation process.

It is necessary that a good set of paths for each OD pair of interest is generated. A suitable path set generated for calibration of DynaMIT was used in this calibration study. All unique paths shorter than twice the shortest distance were included in the final path set. Manual inspection confirmed that most of the practical alternatives had been selected. The final set contained a total of 9036 paths.

5.3.4 Implementation

The calibration was performed for AM peak period, i.e., 6:00AM to 7:30AM. However, the calibration was started from 4:00AM (sensor records indicated that the network is almost empty) to ensure that the first interval that we estimate has minimal interference from vehicles departing in earlier time intervals. The study interval was divided into equal subintervals of 15 minutes. This discretization was based on probe vehicle data that indicated that maximum travel times in this network were approximately 10 minutes.

OD Estimation: The module requires error covariance matrices of both counts and the OD flows. These matrices were developed for calibration of DynaMIT (Balakrishna [33]) and were used in this calibration study. Since day-to-day variation in the available data was not significant, three days of data was used together for estimating the OD matrices. The deviation of the simulated counts was taken against the observed counts on all three days.

Calibration Parameters: The Irvine network has multiple paths for a given OD pair. Hence route choice parameters become very important in this case. Therefore the travel time coefficients (β) were calibrated along with freeway bias parameter. Furthermore, scaling factor alpha (α) of car-following model was also included. Table 5-4 summarizes the range of the selected parameters. The value of freeway bias is between 0 and 1. All travel times on the arterials are divided by freeway bias to account for driver's preference for freeway. The range of other parameters was set based on the default values.

	β(Guided)	*β(Unguided)*	*Freeway Bias*	*Alpha (Acc)*	*Alpha (Dec)*
Lower Bound	*-0.5*	*-0.5*	*0*	*0*	*-0.1*
Upper Bound	*0*	*0*	*1*	*0.1*	*0*

Table 5-4: Upper Bound and Lower Bound of Route Choice and CF Parameters

However, visual results of the simulation indicated that parameters of lane changing model also needed to be calibrated. Hence lane-changing coefficients were included in the calibration process. Scale parameters in the lane changing gap acceptance model (β0) and nosing models (b) were chosen for calibration. Table 5-5 summarizes the range of the selected parameters.

	β0 (Discretionary, Lead)	*β0 (Discreationary, Lag)*	*β0(Mandatory, Lead)*	*β0(Mandatory, Lag)*	*b*
Lower Bound	*0*	*1*	*0*	*0*	*-3.5*
Upper Bound	*1*	*3*	*0.7*	*1*	*-2.5*

Table 5-5: Upper Bound and Lower Bound of Lane Changing Parameters

Objective function: The objective function chosen for the calibration of parameters was a sum of squares of differences between counts generated by MITSIMLab and field values of counts (for all three days) at sensor locations. Mathematically,

$$\sum_{j=1}^{3}\sum_{i=1}^{l}(y_i - y_i^j)^2$$

where:

y_i : MITSIMLab count at sensor i.

y_i^j : Field count at sensor i on day j.

l : Number of sensors.

5.3.5 Results

The module was first used to generate steady state travel time, ans then to calibrate the route-choice and car-following parameters. All other parameters and OD matrices were fixed in this stage. After calibrating the route-choice and car-following parameters, calibration of lane changing parameters was performed by fixing all other parameters and the OD matrices.

Table 5-6 and 5-7 summarize the different sets of parameters generated by Box algorithm and their corresponding function values; the best set of parameters is highlighted.

	B(Guided)	***β(Unguided)***	***Freeway Bias***	***Alpha (Acc)***	***Alpha (Dec)***	***Function value***
Initial Set						
1	***-0.24***	***-0.24***	***0.80***	***0.040***	***-0.042***	***226297415***
2	*-0.49*	*-0.27*	*0.33*	*0.064*	*-0.043*	*242244647*
3	*-0.05*	*-0.47*	*0.43*	*0.021*	*-0.021*	*241647149*
4	*-0.40*	*-0.01*	*0.23*	*0.038*	*-0.094*	*238283048*
5	*-0.35*	*-0.21*	*0.58*	*0.078*	*-0.040*	*248156817*
6	*-0.17*	*-0.29*	*0.76*	*0.068*	*-0.095*	*293804289*
Iteration						
1	*-0.17*	*-0.31*	*0.42*	*0.005*	*-0.084*	*286525542*
2	*-0.22*	*-0.28*	*0.46*	*0.025*	*-0.072*	*227398667*
3	*-0.22*	*-0.27*	*0.64*	*0.057*	*-0.077*	*247731893*
4	*-0.24*	*-0.26*	*0.57*	*0.517*	*-0.068*	*245146859*
5	*-0.26*	*-0.26*	*0.54*	*0.049*	*-0.064*	*239320797*
6	*-0.38*	*-0.26*	*0.42*	*0.055*	*-0.051*	*246238874*

Table 5-6: Route Choice and CF Parameters and Their Corresponding Values

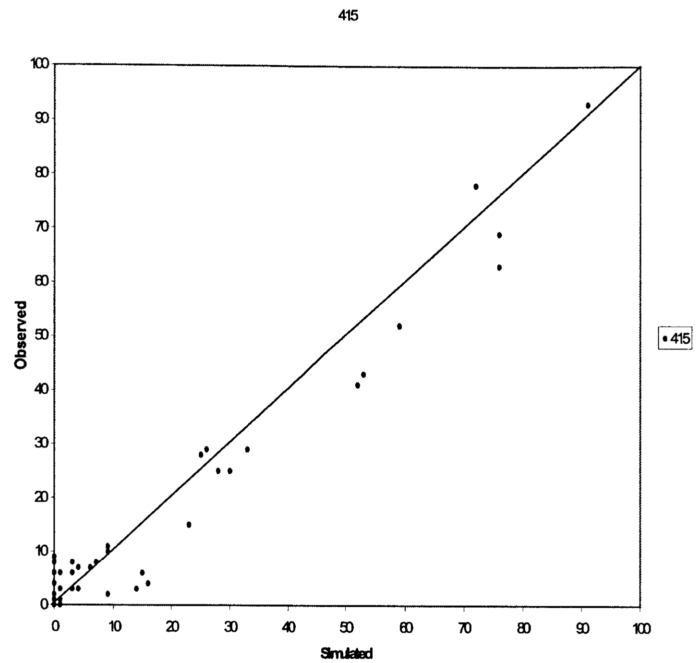
	β0 (Discretionary, Lead)	*β0 (Discreationary, Lag)*	*β0 (Mandatory, Lead)*	*β0 (Mandatory, Lag)*	*b*	*Function value*
Initial Set						
1	*0.0508*	*2.02*	*0.384*	*0.587*	*-3.159*	*231286412*
2	*0.231*	*1.037*	*0.517*	*0.893*	*-3.297*	*236725304*
3	*0.607*	*2.643*	*0.123*	*0.058*	*-3.301*	*224907573*
4	*0.486*	*1.889*	*0.284*	*0.353*	*-2.896*	*234212906*
5	***0.891***	***2.231***	***0.655***	***0.813***	***-3.228***	***224367123***
6	*0.762*	*2.584*	*0.642*	*0.0099*	*-3.301*	*226519434*
Iteration						
1	*0.999*	*2.999*	*0.289*	*0.0001*	*-3.021*	*230547553*
2	*0.999*	*2.999*	*0.593*	*0.217*	*-3.5*	*232747358*

Table 5-7: Lane Changing Parameters and Their Corresponding Values

Finally, the parameters were fixed at the calibrated values and OD estimation was performed (with calibrated parameters). After each iteration of OD estimation, a new set of steady state travel times were obtained by fixing the OD and the parameters. OD estimation was done iteratively till a convergence criterion was achieved.

There were 655 OD pairs and 14 time slices. The field observation was given for 68 sensor locations. Hence, for each time interval 68 count readings were to be matched and 655 OD pairs were to be estimated. The resulting assignment matrix was of size 952 × 9170. The module was able to handle such a large matrix and gave new estimates of the OD in less than 5 minutes.

The final results after OD–estimation are shown in Figures 5-14 through 5-20.



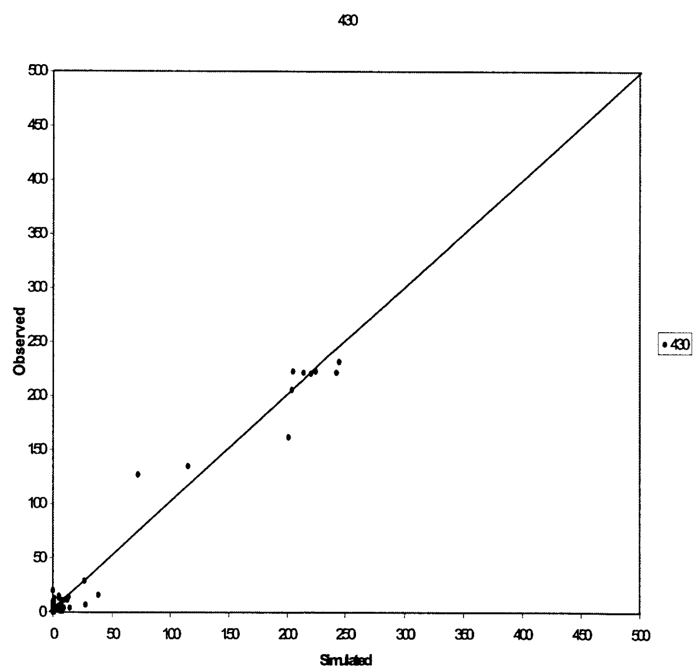
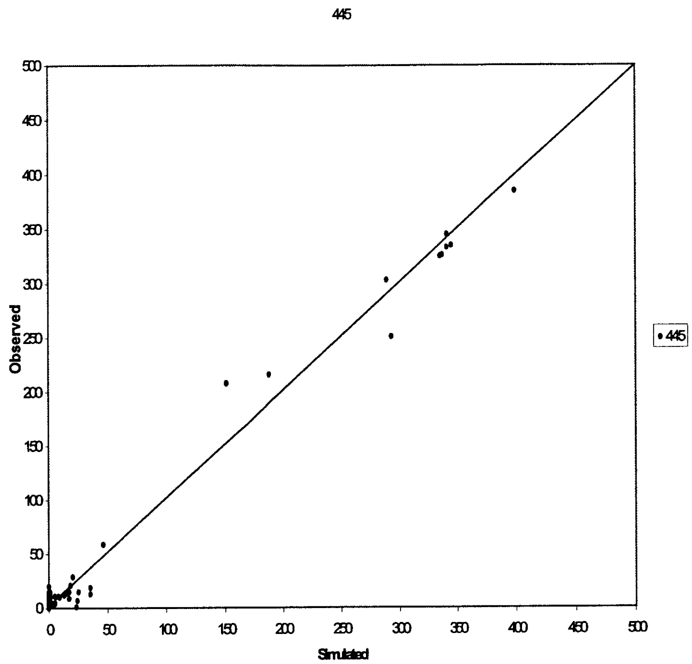


Figure 5-14: Observed vs. Simulated Counts at 4:15 AM and 4:30 AM



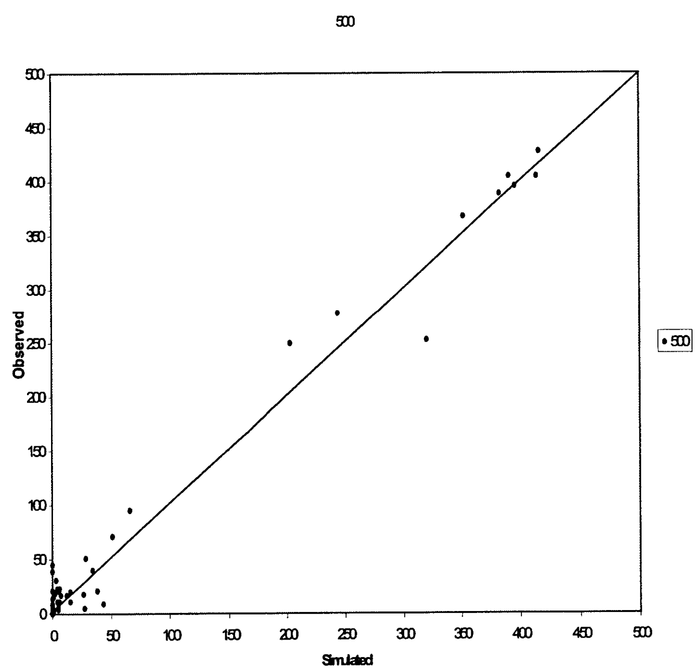
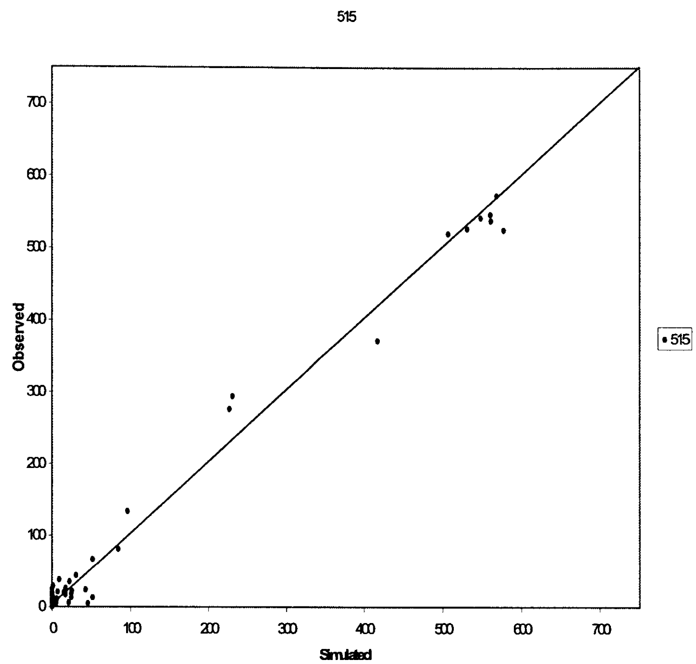


Figure 5-15: Observed vs. Simulated Counts at 4:45 AM and 5:00 AM



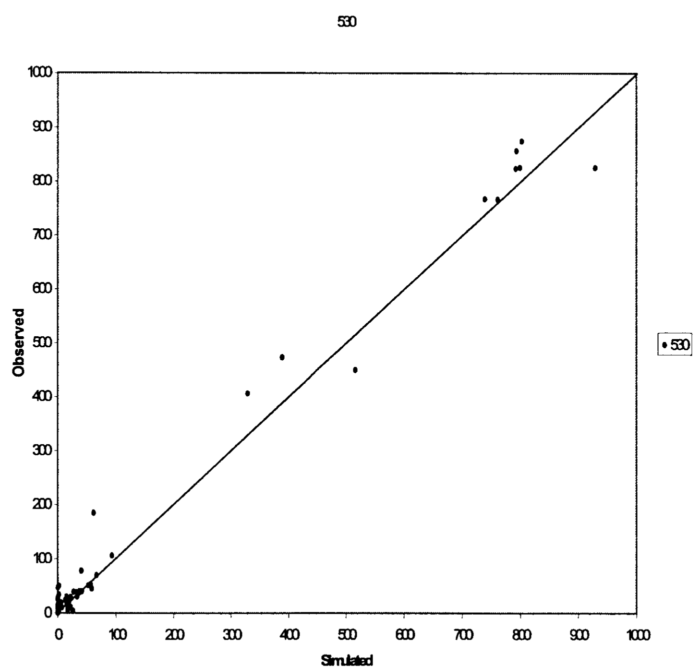
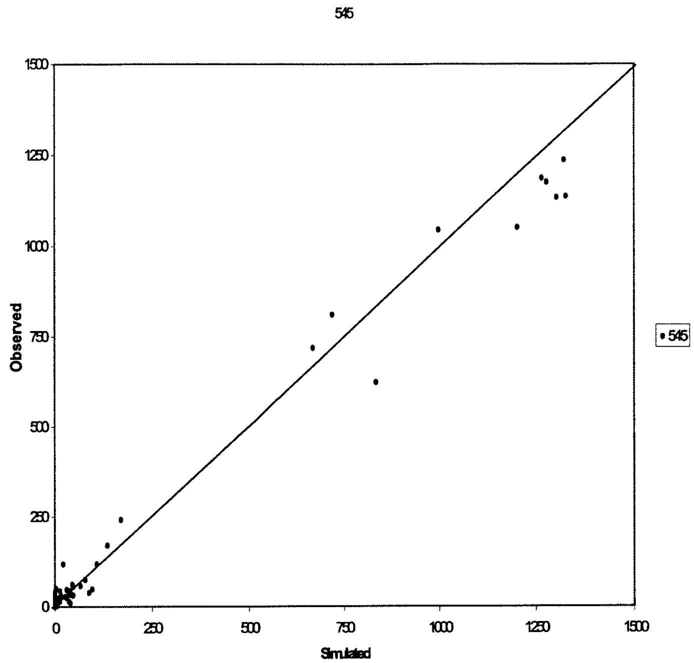


Figure 5-16: Observed vs. Simulated Counts at 5:15 AM and 5:30 AM



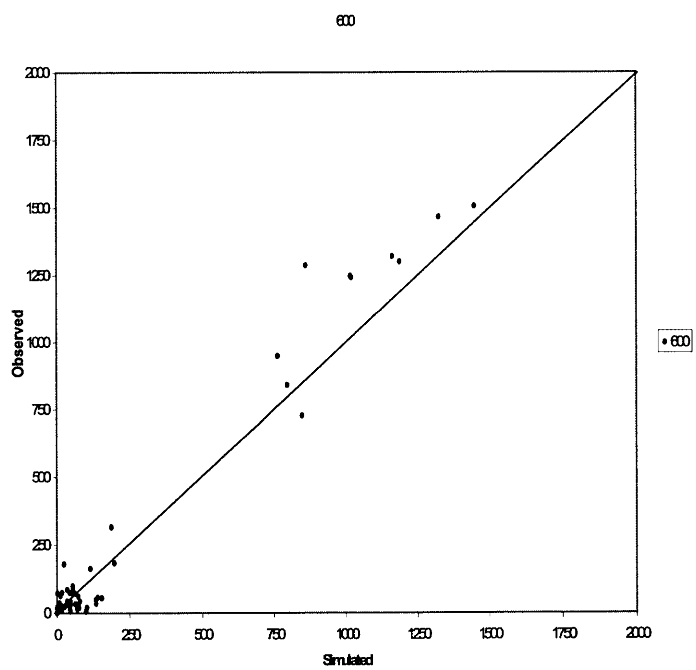
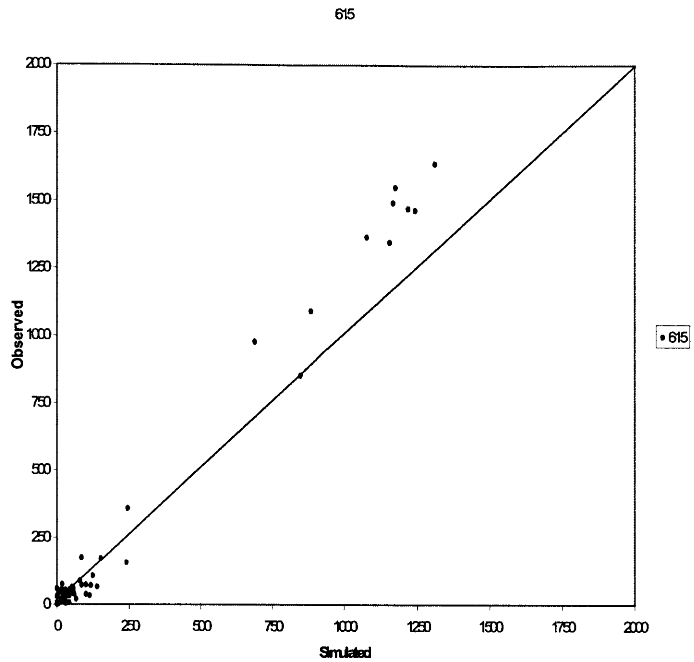


Figure 5-17: Observed vs. Simulated Counts at 5:45 AM and 6:00 AM



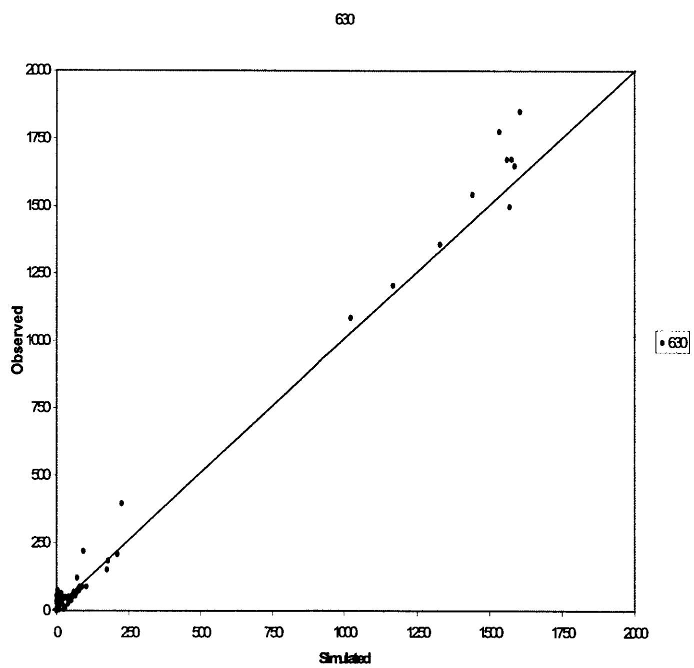
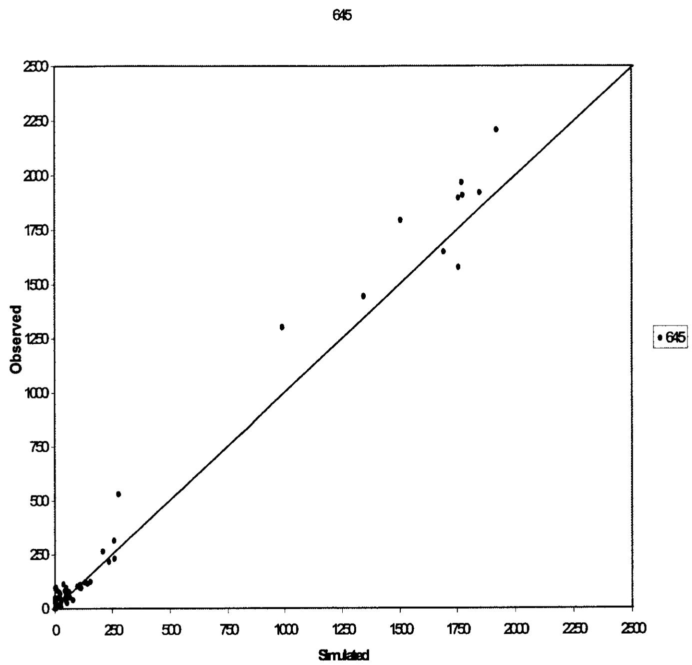


Figure 5-18: Observed vs. Simulated Counts at 6:15 AM and 6:30 AM



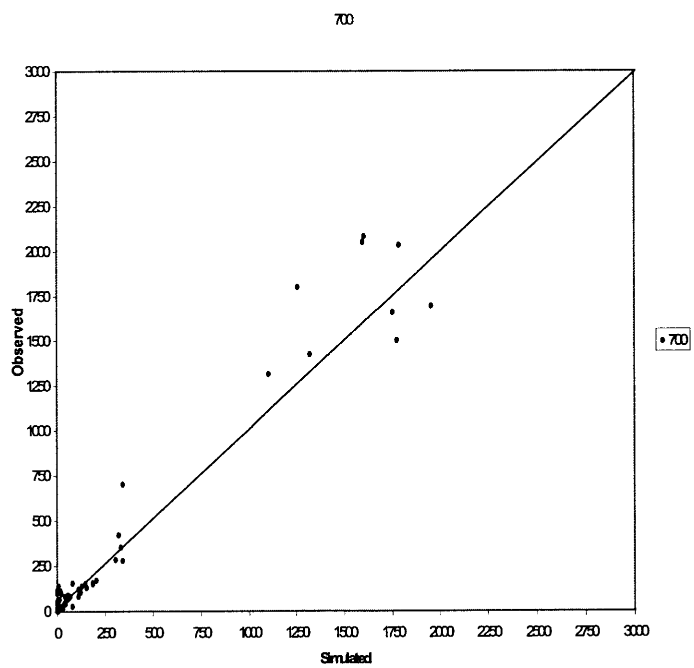
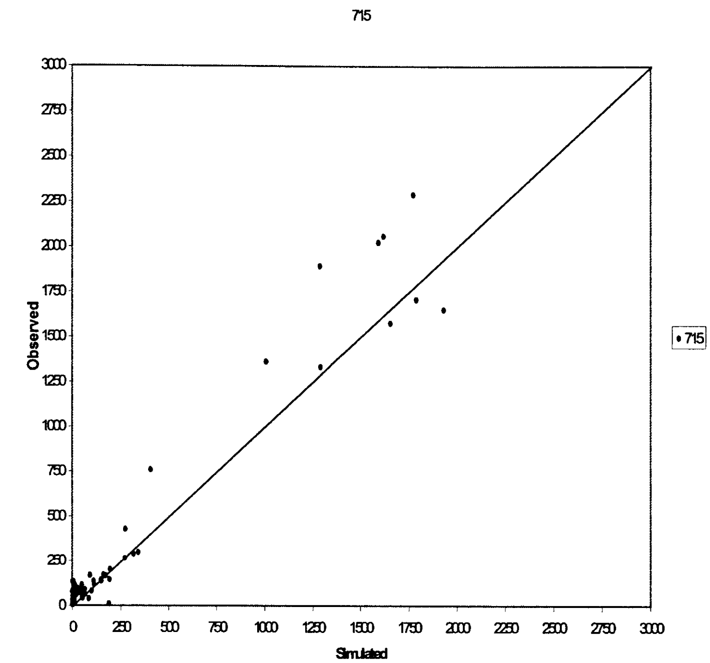


Figure 5-19: Observed vs. Simulated Counts at 6:45 AM and 7:00 AM



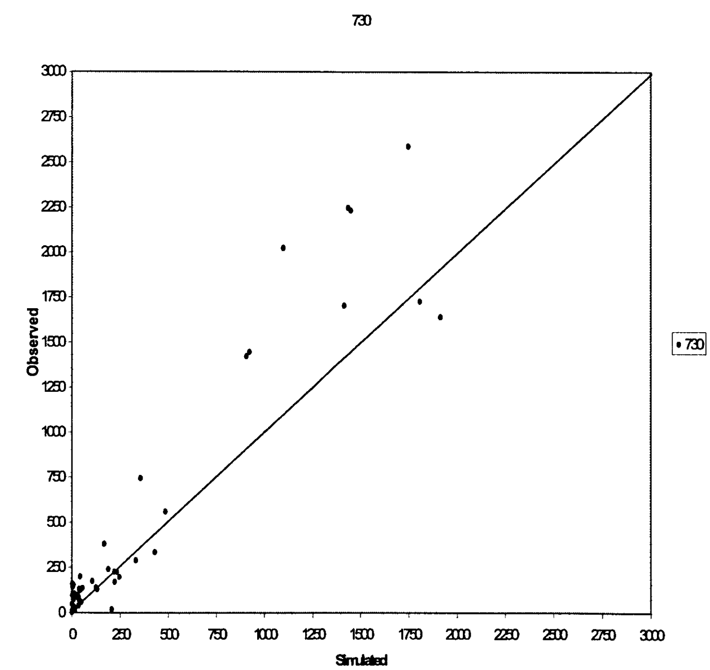


Figure 5-20: Observed vs. Simulated Counts at 7:15 AM and 7:30 AM

One of the biggest problems in calibrating this network was the computation time. Due to this limitation, we were constrained to do less number of iterations. The accuracy of these results can be further improved by increasing the number of iterations.

5.4 Summary of Results

Based on the three case studies, the following conclusions are drawn:

- For the simple HCQS network, the calibration approach was successful in matching the observed speeds, densities and travel times on the freeways and the arterials effectively. The validity and the potential of the Box algorithm to be used for the calibration of various parameters were aptly demonstrated.

- In the next case study on the North West Kungsholmen network, the capability of the calibration framework to estimate OD flows to match field sensor counts was tested. Again, the approach was successful in estimating OD flows that closely replicated field counts for an AM peak period.

- The final case study on the Irvine network was aimed to demonstrate the joint calibration of the model parameters with OD estimation. The results were promising. However, it was clear that several iterations are to be performed to closely replicate field data. Due to constraints imposed by the large running time involved in running the simulator and the large number of parameters to be calibrated, the optimal values of the parameters was not ascertained.

- Importantly, the case studies highlighted that the approach used for calibration of the model parameters based on the Box algorithm and the procedure used for OD estimation can yield accurate results.

Chapter 6

Conclusions

This chapter discusses the thesis contributions and provides directions for future research.

6.1 Thesis Contributions

In this thesis, a systematic approach for the calibration of microscopic traffic simulation models along with estimation of dynamic OD matrices was developed. The framework takes into account the interactions between various models in the simulator and OD flows. An optimization-based methodology was developed for calibration of the parameters and estimating OD matrices. Furthermore, the methodology addressed the issue of obtaining a global optimum in a multidimensional search space using the Box algorithm. The algorithm generates random set of parameters in the entire search space and then tries to converge all the generated points towards the minimum. Steady state travel times can also be computed and is an integral part of the module. Other issues addressed in this thesis are the computational issues related to OD estimation. As the size of OD matrices increase, the computational burden increases significantly. To address this issue of computation burden, a sequential OD estimation approach was implemented for application to cases where the simultaneous OD estimation approach would be infeasible.

The framework developed in this thesis can handle any objective function, and calibrate any user-specified set of parameters. In order to take care of the stochasticity associated with the simulator, the function can be written in such a way such that it returns the averaged value of objective function over pre-defined number of runs. The number of parameters to be calibrated is constrained only by computational time. As the number of parameters to be calibrated increases, the number of points initially generated by Box algorithm increases, which indirectly implies that more runs of the simulator are needed to evaluate the value of objective function associated with these sets. Furthermore, more iteration may be required to achieve the optimum value of the parameters.

The module was tested on three networks. For the HCQS network, the car-following parameters and the scaling factor for speeds of the driver groups were calibrated. The results showed that as the number of iterations increase, the Box algorithm tries to bring all the sets of parameters towards the minimum. In the North West Kungsholmen network case study, the simultaneous estimator was used for estimating the OD flows, and it produced promising results. In the third case study, the overall framework of the module was tested. In this case study, route choice parameters, car-following parameters and some lane changing parameters were calibrated along with estimation of OD matrices and estimation of steady state travel times. Since the number of OD pairs was quite large, the sequential approach was utilized. However because of the computation time of MITSIMLab, the number of iterations was reduced.

6.2 Future Research

Sensitive Parameters: The focus of this thesis was mainly to develop a generic tool that can be used for calibration purpose; the sensitivity of various parameters was not tested. Hence one good direction of research would be to identify the set of parameters that are most sensitive to the simulation results. Kurian ([1] partially addressed this issue. Also, the range of variation of these parameters should be identified so that they can be used for defining the upper bound and lower bound of the parameters. Calibration of these parameters would be computationally more efficient.

Search Algorithm: The Box complex algorithm was implemented in this module. Although this algorithm tries to take the solution towards the global minima, the final solution depends upon the number of initial sets of parameters generated and also on the random numbers generated. Hence, the performance of Box algorithm should be compared with other available algorithms.

Appendix A

Calibration Module

This appendix gives details of each component of the module.

The entire module has been written in MATLAB and the following assumptions are made:

General Assumptions

- Counts data is available at the sensor locations.
- The sensor data report interval is same as the OD interval.

MITSIMLab Input Files Assumptions:

- All input and output files are in the same directory.
- The OD file is called "od.dat".
- The sensor output file is called "sensor.out".
- The assignment matrix file is called "assignment_matrix.out".
- The input historical travel time file is called "linktime.dat".
- The output link travel time file is called "linktime.out".
- The parameter file is called "paralib.dat".

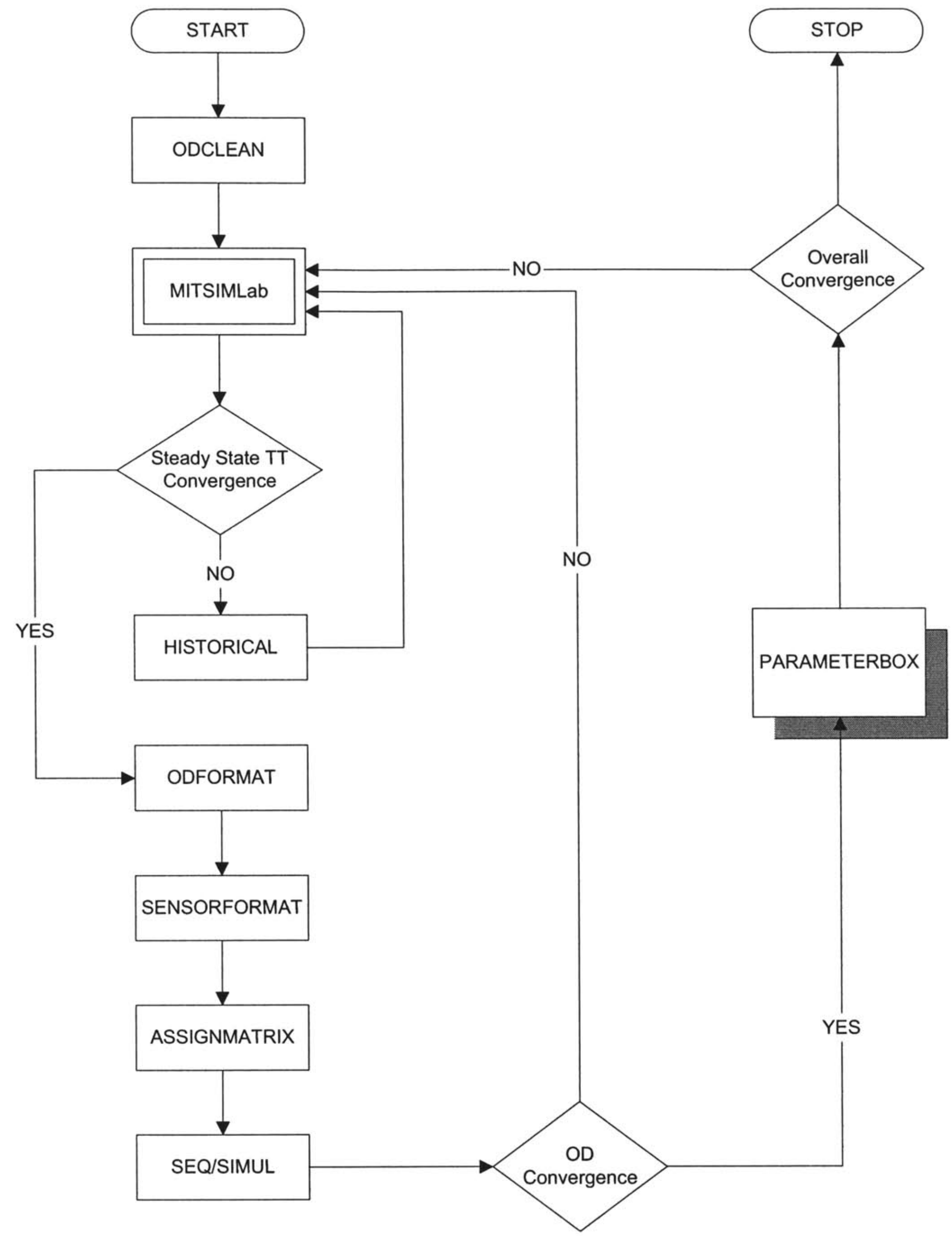


Figure A-1: Main Program of the Module

The module needs the following input files:

- Observed counts file (**counts.dat**).
- Weights file (**weight.dat**)
- Seed OD file (**seed-od.dat**)
- Any other file required for calculating the objective function value.

The main program of the module is depicted in Figure A.1:The module has many sub-modules and a description of each of the sub- modules is as follows:

ODCLEAN: This module cleans the seed OD (**seed-od.dat**) file by removing comments from the file and writing the pertinent data to **od.dat**. This makes processing easier.

ODFORMAT: This module covert the OD file (**od.dat**) into a matrix format which makes it easier to manipulate.

```
27000 0 1
{
{ 1   4   2409 }
{ 1   7   453 }
{ 1   10   411 }
}
```

Format of OD

```
27000  1  4  2409
27000  1  7  453
27000  1  10  411
```

Format after ODFORMAT module

SENSORFORMAT: Similar to ODFORMAT module, this module coverts the sensor output file (**sensor.out**) into matrix format.

```
27900 {
      0 1 372
      1 1 440
      2 1 250
      3 1 42
}
```

Format of sensor.out

```
27900  0  372
27900  1  440
27900  2  250
27900  3  42
```

Format after SENSORFORMAT

ASSIGNMATRIX: This module generates the Assignment Matrix, which maps OD flows to sensor counts. Assignment Matrix represents the fraction of demand for a particular OD contributing to sensor counts by time period. It uses the '**assignment_matrix.out**' generated by MITSIMLab and the output files generated by ODFORMAT and SENSORFORMAT modules to generate the Assignment Matrix.

SEQ: This module uses the Assignment Matrix generated by ASSIGNMATRIX module, output file of ODFORMAT module, field sensor count file (**counts.dat**) and the weights file (**weights.dat**) to estimate the OD matrices. Constrained OD estimation is performed sequentially and the estimated OD matrices are written back in MITSIMLab compatible format to **od.dat**.

SIMUL: This module is similar to the SEQ module. The only difference is that it estimates the OD matrices simultaneously.

HISTORICAL: This module calculates the weighted average of link travel time generated by MITSIMLab (**linktime.out**) and input travel time (**linktime.dat**) to produce a new historical travel time file (overwrite **linktime.dat**).

PARAMETERBOX: This module applies the Box algorithm and returns optimized value of the parameters. This module calls the BOX module, which in turn calls the CHECK module, INITIALPARAMETERS module, CHANGEPARAMETERS module and FUNC module.

BOX: This is the module where the Box algorithm is implemented. It generates the initial (K) sets of parameters. It calls the INITIALPARAMETERS module to get the initial value of the parameters specified in the parameter file. It then evaluates the function value associated with each set of parameters. To implement this, it first writes these values in the parameter file using the CHANGEPARAMETER module and then calls the FUNC module to evaluate the function value associated with them. Then based on the Box algorithm, it generates new set of parameters and repeats the above-mentioned steps till the convergence criterion is met.

INITIALPARAMETERS: This module passes the initial set of value from the parameter file to CONSX module. This module needs to be modified based on the chosen parameters. In this module the user has to specify a method to extract the initial set of parameters (chosen for calibration), from the parameter file (**paralib.dat**).

CHANGEPARAMETER: This module writes the new set of parameters into the parameter file (**paralib.dat**). This module will also change depending upon the parameters chosen for calibration. The user has to specify a method in this module to write the values of chosen parameters into the parameter file.

FUNC: This module returns the function values. It calls the simulator and the outputs of the simulator are then processed in this module to give the value of the objective function.

Bibliography

1. Kurian, M., *Calibration of a microscopic traffic simulator, Thesis (S.M.)-- Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering, 2000.* 90 leaves.

2. Ahmed, K.I., *Modeling drivers' acceleration and lane changing behavior*. 1999. 189.

3. Abdulhai, B. and T. Ma. *GENOSIM: A Genetic Algorithm-Based Optimization Approach and Generic Tool for the Calibration of Traffic Microscopic Simulation Paramters*. in *Transpostation Research Board*. 2002.

4. Der-Horng Lee, X.Y., P. Chandrasekar. *Parameter Calibration for Paramics Using Genetic Algorithm*. in *Transportation Research Board*. 2001.

5. Cheu, R.L., W.W. Recker, and S.G. Ritchie, *Calibration of INTRAS for simulation of 30-sec loop detector output.* Transportation Research Record, 1994. **1457**: p. 208-215.

6. Cheu, R.L., et al., *Calibration of FRESIM for Singapore Expressway using genetic algorithm.* Journal of Transportation Engineering-ASCE, 1998. **124**(6): p. 526-535.

7. Vanzuylen, H.J. and L.G. Willumsen, *The Most Likely Trip Matrix Estimated from Traffic Counts.* Transportation Research Part B-Methodological, 1980. **14**(3): p. 281-293.

8. Maher, M.J., *Inferences on Trip Matrices from Observations on Link Volumes - a Bayesian Statistical Approach.* Transportation Research Part B-Methodological, 1983. **17**(6): p. 435-447.

9. Spiess, H., *A Maximum-Likelihood Model for Estimating Origin Destination Matrices.* Transportation Research Part B-Methodological, 1987. **21**(5): p. 395-412.

10. Cascetta, E., *Estimation of Trip Matrices from Traffic Counts and Survey Data - a Generalized Least-Squares Estimator.* Transportation Research Part B-Methodological, 1984. **18**(4-5): p. 289-299.

11. McNeil, S. and C. Hendrickson, *A Regression Formulation of the Matrix Estimation Problem.* Transportation Science, 1985. **19**(3): p. 278-292.

12. Bell, M.G.H., *The Estimation of Origin-Destination Matrices by Constrained Generalized Least-Squares.* Transportation Research Part B-Methodological, 1991. **25**(1): p. 13-22.

13. Nguyen, S., *Estimating an OD matrix from network data: A network equilibrium approach.* 1977, University of Montreal: Montreal.

14. Leblanc, L.J. and K. Farhangian, *Selection of a Trip Table Which Reproduces Observed Link Flows.* Transportation Research Part B-Methodological, 1982. **16**(2): p. 83-88.

15. Cascetta, E. and S. Nguyen, *A Unified Framework for Estimating or Updating Origin Destination Matrices from Traffic Counts.* Transportation Research Part B-Methodological, 1988. **22**(6): p. 437-455.

16. Cremer, M. and H. Keller, *A New Class of Dynamic Methods for the Identification of Origin-Destination Flows.* Transportation Research Part B-Methodological, 1987. **21**(2): p. 117-132.

17. Nihan, N.L. and G.A. Davis, *Recursive Estimation of Origin-Destination Matrices from Input Output Counts.* Transportation Research Part B-Methodological, 1987. **21**(2): p. 149-163.

18. Chang, G.-L.T.X., *Estimation of Dynamic O-D Distributions for Urban Networks.* Transportation and traffic theory : proceedings of the 13th International Symposium on Transportation and Traffic Theory, Lyon, France, 1996: p. xv, 758.

19. Cascetta, E., D. Inaudi, and G. Marquis, *Dynamic Estimators of Origin-Destination Matrices Using Traffic Counts.* Transportation Science, 1993. **27**(4): p. 363-373.

20. Ashok, K. and M.E. Ben-Akiva, *Dynamic Origin-Destination Matrix Estimation and Prediction for Real-Time Traffic Management Systems.* Transportation and traffic theory : proceedings of the 12th International Symposium on the Theory of Traffic Flow and Transportation, Berkeley, California, USA,, 1993: p. xvii, 596.

21. Ashok, K. and M.E. Ben-Akiva, *Alternative approaches for real-time estimation and prediction of time-dependent Origin-Destination flows.* Transportation Science, 2000. **34**(1): p. 21-36.

22. Liu, S.S. and J.D. Fricker, *Estimation of a trip table and the Theta parameter in a stochastic network.* Transportation Research Part a-Policy and Practice, 1996. **30**(4): p. 287-305.

23. Yang, H., Q. Meng, and M.G.H. Bell, *Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium.* Transportation Science, 2001. **35**(2): p. 107-123.

24. Burghout, W., et al. *Calibration and Evaluation of MITSIMLab in Stockholm.* in *Transportation Research Board.* 2002.

25. Banks, J., J.S. Carson, B.L. Nelson, D.M. Nicol, *Discrete Event Systems Simulation.* 3rd ed. 2000, Englewood Cliffs, NJ: Prentice Hall.

26. Pflug, G.C., *Optimization of stochastic models : the interface between simulation and optimization*. 1996, Boston, Mass.: Kluwer Academic. xiv, 382.

27. Cascetta, E. and M.N. Postorino, *Fixed point approaches to the estimation of O/D matrices using traffic counts on congested networks.* Transportation Science, 2001. **35**(2): p. 134-147.

28. Ashok, K., *Estimation and prediction of time-dependent origin-destination flows, Thesis (Ph. D.)--Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering, 1996.* p. 155.

29. Box, E.P., *A new method for constrained optimization and a comparison with other methods.* The Computer Journal, 1956. **8**: p. 42-52.

30. Yang, J., *A simulation laboratory for evaluation of dynamic traffic management system*. 1997. 193.

31. Ramming, M.S., *Network knowledge and route choice, Thesis (Ph. D.)--Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering,* 2001.

32. Cortes, M.T., *Benefits of Emerging Transportation Technologies: Simulation Analysis and Policy Issues, Thesis, (S.M.)--Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering,* 2002.

33. Balakrishna, R., *Calibration of the Demand Simulator in a Dynamic Traffic Assignment System, Thesis (S.M.)--Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering,* 2002.

34. Kuester, J.L. and Mize J.H., *Optimization Techniques with Fortran.* 1973, New York, NY: McGraw-Hill.