

Calibration of a Microscopic Traffic Simulator

by

Mathew Kurian

Bachelor of Technology in Civil Engineering

Indian Institute of Technology, Madras, June 1997

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

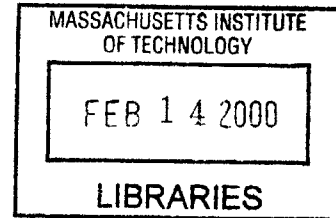
Master of Science in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000

ENG



© Massachusetts Institute of Technology 2000. All rights reserved.

Author
Department of Civil and Environmental Engineering
November 1, 1999

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

Certified by.....
Mithilesh K. Jha
Research Associate, Center for Transportation Studies
Thesis Supervisor

Certified by.....
Didier M. Burton
Research Associate, Center for Transportation Studies
Thesis Supervisor

Accepted by .
Daniele Veneziano
Chairman, Department Committee on Graduate Students

Calibration of a Microscopic Traffic Simulator

by

Mathew Kurian

Bachelor of Technology in Civil Engineering

Indian Institute of Technology, Madras, June 1997

Submitted to the Department of Civil and Environmental Engineering
on November 1, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science in Transportation

Abstract

A systematic calibration study was performed on a microscopic traffic simulator-MITSIM. An optimization based framework was developed for calibration. Car-Following model parameters were identified for calibration and experimental design methodology was used to determine the set of sensitive parameters. Calibration was performed by minimizing the deviation between the simulated and observed values of speed. Two different objective function forms were formulated for quantifying the deviation between the simulated and observed values. The search space and the optimum parameter values for the two objective function forms were compared. The effect of stochasticity in calibrating the parameter values was also studied. Stochasticity was found to have a significant impact on the optimal parameter values. It was found that though calibration is an intricate process, the performance of the simulator can be substantially improved by an appropriate calibration study.

Thesis Supervisor: Moshe E. Ben-Akiva

Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Thesis Supervisor: Mithilesh K. Jha

Title: Research Associate, Center for Transportation Studies

Thesis Supervisor: Didier M. Burton

Title: Research Associate, Center for Transportation Studies

Acknowledgements

I would like to thank my supervisors Prof. Moshe Ben-Akiva, Mithilesh Jha, and Didier Burton for their guidance and support at all stages of my research. I would also like to thank SAMTECH, for the evaluation version of Boss/Quattro. This software proved to be very useful for implementation and analysis.

I would also like to thank the Centre for Transportation Studies and the Central Artery/Tunnel project for providing funding for my research. I would also like to thank the Dept. of Civil Eng. for the opportunity to work as a Teaching Assistant for the course 1.00 for three semesters.

I would also like to thank Cynthia Stewart and her staff, and Leanne Russell for their support.

My special thanks are also due to Sydney Miller for her friendship, and all the folks at the CTS office, Janet, Nancy, and Mary, who made CTS so much more lively and enjoyable.

I also want to thank my officemates Bruno, Tomer, Masroor, Seong, Atul and Sreeram and also Vinay, Salal and Nadine. My special tons of thanks are due to my friends Debjit and Petros for making my stay at MIT so much more enjoyable.

I would like to thank Prof. Nigel Wilson for his support, guidance, understanding and his eagerness and willingness to help.

Finally I would like to thank my Dad, Mom and my sister Jinu for their understanding and support.

Contents

1	Introduction	10
1.1	Literature Review	11
1.1.1	Traffic Simulation Models	11
1.1.2	Microscopic Simulator	11
1.1.3	Calibration Studies	11
1.2	Calibration Framework	14
1.3	Thesis Objective	15
1.4	Thesis Outline	16
2	Simulator, Parameters and Data	17
2.1	MIcroscopic Traffic SIMulator-MITSIM	17
2.1.1	Main Simulation Models	19
2.2	Car-Following Model	19
2.2.1	Car-Following Parameters	20
2.3	Simulation Network and Data	22
2.4	Conclusion	24
3	Sensitivity Analysis	28
3.1	Responses	29
3.2	Experimental Design	29
3.2.1	2^k Factorial Design	30
3.2.2	Fractional Factorial Designs	32
3.2.3	Response Functions	33

3.2.4	Stochasticity	33
3.3	Results	34
3.4	Conclusion	37
4	Calibration	42
4.1	Response Selection	43
4.2	Objective Function Forms	43
4.3	Stochasticity	44
4.4	Optimization	47
4.4.1	Optimization Methodology	48
4.5	Results	48
4.5.1	Comparison of the Spread of the Objective Functions	49
4.5.2	Results for a Fixed Random Number Seed	50
4.5.3	Sampling Distribution of Optimal Parameter Values	57
4.6	Conclusion	66
5	Summary, Conclusions and Research directions	69
5.1	Summary and Conclusion	69
5.2	Research Directions	70
A	Boss/Quattro	72
A.1	Introduction	72
A.2	Structure	72
A.3	Using Boss/Quattro with MITSIM	75
A.3.1	Environment Setup	75
A.3.2	Studies	77
A.3.3	Sensitivity Study	78
A.3.4	Optimization study	80
A.4	Conclusion	80
B	MITSIM-Parameters	82

List of Figures

2-1	I-880 Network	23
2-2	Comparison between simulated and field data.	25
2-3	Contour plots of simulated and field data.	26
2-4	Comparison of simulated and field data	27
3-1	Plot of the responses versus the design points	36
4-1	Illustrating the effect of stochasticity in calibrating parameter values .	46
4-2	Convergence results for optimizing ψ	51
4-3	Convergence results for optimizing ϕ	51
4-4	Convergence results for optimizing ψ	52
4-5	Convergence results for optimizing ϕ	52
4-6	Correlation plots	54
4-7	Comparison of simulated and observed data for parameter set 1	58
4-8	Comparison of simulated and observed data for parameter set 2	58
4-9	Comparison of simulated and observed data for parameter set 3	58
4-10	Comparison of simulated and observed data for parameter set 4	58
4-11	Contour plots for simulated and field data for parameter set 1	59
4-12	Contour plots for simulated and field data for parameter set 2	59
4-13	Contour plots for simulated and field data for parameter set 3	60
4-14	Contour plots for simulated and field data for parameter set 4	60
4-15	Comparison of simulated and field data for parameter set 1	61
4-16	Comparison of simulated and field data for parameter set 2	61
4-17	Comparison of simulated and field data for parameter set 3	62

4-18	Comparison of simulated and field data for parameter set 4	62
4-19	Comparison of simulated and observed data for parameter set obtained by optimizing ψ	63
4-20	Comparison of simulated and observed data for parameter set obtained by optimizing ϕ	64
4-21	Contour plots for simulated and field data for parameter set 1 obtained by optimizing ψ	64
4-22	Contour plots for simulated and field data for parameter set 2 obtained by optimizing ϕ	65
4-23	Comparison of simulated and field data for parameter set 1 obtained by optimizing ψ	65
4-24	Comparison of simulated and field data for parameter set 2 obtained by optimizing ϕ	66
4-25	Comparison of calibrated and simulated objective function values . .	68
A-1	Boss/Quattro's interactions	73
A-2	Boss/Quattro's interactions with MITSIM	76
A-3	An example of a design of experiments run	79

List of Tables

2.1	Parameter estimates for acceleration, Source[21]	21
2.2	Parameter estimates for deceleration, Source[21]	22
2.3	Range of values for the parameters	22
2.4	Initial Values for the Parameters	23
3.1	Design matrix for a 2^3 factorial design	30
3.2	Design matrix for a 2^8_V factorial design	35
3.3	Screening for ψ_{flow}	38
3.4	Result for the final model for ψ_{flow}	39
3.5	Result for final model for ψ_{speed}	40
3.6	Result for final model for $\psi_{occupany}$	41
4.1	Comparing the variation for the two objective function forms	50
4.2	Local optimum values	53
4.3	Correlation coefficient	54
4.4	Correlation Coefficient	55
4.5	Parameter values obtained for optimizing ψ	56
4.6	Parameter values obtained for optimizing ϕ	56
4.7	Objective function values for initial parameter value	57
4.8	Optimal parameter value by optimizing objective function form ψ	63
4.9	Optimal parameter value by optimizing objective function form ϕ	63
B.1	Description of some of the parameters in MITSIM	85

Chapter 1

Introduction

The enormous growth of road traffic has resulted in heavy congestion in and around urban areas, leading to increased delays, pollution, and inefficiency. From the supply side, a solution to this problem would be to expand the road network; but this is not economically feasible. Moreover, on the demand side, social and individual behavior characteristics limit the constraints that can be applied to encourage travel by other modes. Hence efficient management of transportation infrastructure seems to be the most viable solution for managing congestion.

One proposed approach for traffic management is to use a set technologies called the Advanced Traffic Management Systems (ATMS), and Advanced Traveler Information Systems (ATIS). They form part of a suite of technologies known as the Intelligent Transportation System (ITS). ATMS and ATIS aim at applying advanced technologies in the areas of dynamic traffic management, traffic control, information and communication systems, and traffic modeling to better manage existing transportation infrastructure.

Traffic simulation, especially microscopic traffic simulation, is often used as a tool to evaluate the various ATMS/ATIS strategies before they are implemented in practice. The microscopic traffic simulators use several models like the car-following model, the lane-changing model etc. to simulate traffic. These models use various parameters, which in turn determine the results of the simulation. The values of these parameters should be carefully selected so that the simulation model replicates

real world traffic observations. Hence, parameter calibration is an essential component of any successful simulator application.

1.1 Literature Review

1.1.1 Traffic Simulation Models

Traffic simulators can be broadly classified into three categories: microscopic, macroscopic and mesoscopic. A microscopic simulator models and simulates the trajectories of individual vehicles using various disaggregate models. Some examples of disaggregate models are car-following model and lane-changing model. Macroscopic simulators on the other hand simulate vehicle movement using aggregate models. The traffic flows are approximated as fluid flows and the vehicles are moved based on assumed speed-density relationships. Mesoscopic simulation models are hybrid models of microscopic and macroscopic models.

1.1.2 Microscopic Simulator

A Microscopic Traffic Simulator—MITSIM has been developed at MIT by Yang [24, ch. 3] for modeling traffic flow. MITSIM represents networks at the lane level and simulates movements of vehicles using car-following, lane-changing, and traffic signal response logic. Various parameters (see Appendix B) determine the output generated by the simulator. The validity of these parameter values is crucial for a successful simulation application. Moreover, the parameter values depend on the traffic conditions and may need to be recalibrated in the future due to changing traffic conditions. This work focuses on a systematic framework to apply a non-linear optimization technique to calibrate a microscopic simulator. The simulator used for this study is MITSIM.

1.1.3 Calibration Studies

Two broad procedures for model calibration [22] are:

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

The first technique uses estimation techniques, usually econometric models, to directly estimate the individual parameter values. An example is the work done by Subramaniam [21] and Ahmed [1] wherein the parameters of the car-following and lane-changing model in MITSIM are estimated from real world data using a Maximum Likelihood Estimator. This estimation technique enables the parameter values to be provided as inputs to the simulation model. The disadvantage of this technique is the large amount of data required. For instance, the data required for estimating the parameters for the car-following and lane-changing model include position, speed, acceleration, and length of the vehicles [1].

The indirect technique uses the simulation model itself to predict the parameter values. This technique is especially useful when repeated re-calibration needs to be done for changing traffic conditions. Calibration is usually done by minimizing the deviation between the observed and simulated values by varying the parameter values. The advantage of this method is that calibration can be done using more readily available aggregate data. Flow, speed and occupancy values across sensor stations can be easily measured and used for calibration.

Usually, the number of parameters affecting a simulation model is too large. Hence, for a feasible calibration study, an appropriate subset of parameters should be selected. Moreover, a suitable objective function form should be defined for minimizing the deviation between the observed and simulated values. Also, to obtain meaningful parameter estimates, the stochasticity of the simulator should be taken into account.

There have been various studies involving indirect calibration of simulation models. Goodspeed [13] outlines the work done in the optimization of a hydrological catchment model. Various forms of objective functions were considered and also two different optimization methodologies were compared. But the simulation model for

hydrological catchment was deterministic and also the number of parameters was less compared to microscopic traffic simulators.

In the work done by R.L. Cheu et al.[8], a microscopic traffic simulator INTRAS was calibrated with a 30-sec interval, loop detector data set. The key parameters thought to influence traffic flow were identified, probably based on previous simulation experience. The parameters were calibrated sequentially; while the optimum value of one parameter was determined, the remaining parameters were kept constant. The simulated volume and occupancy were plotted against actual field volume and occupancy. The correlation coefficients and the slopes of the fitted straight lines that pass through the origin were used as performance measures. To take into account the stochasticity in the simulation model, for each parameter set, an average value from three simulation runs with different random number seeds was used. This study does not detail how the parameter set was determined and also does not investigate the effect of varying the forms of the objective function. Also the procedure adopted to consider the stochasticity, namely, averaging over three runs is inadequate and the sequential calibration methodology used may lead to a local optimum solution.

In a related work R.L. Cheu et al.[7] used a Genetic Algorithm to calibrate the microscopic simulator FRESIM. The free flow speeds and vehicle movement parameters were selected for calibration. The fitness function (similar to an objective function) form was based on the natural logarithmic base, raised to the absolute value of the difference between the observed and simulated values, summed over all time for a set of selected sensors. This work did not address the stochasticity involved in microscopic simulation and also did not discuss how the appropriate fitness function was selected. The advantage of using a Genetic Algorithm for optimization is that, irrespective of the nature of the search domain there is a higher likelihood of reaching an optimum point. But the search domain has never been previously explored to justify the use of robust, though computationally intensive Genetic Algorithm based methods.

From the literature review it is clear that a systematic calibration study is still lacking. Most of the studies do not address parameter selection, type of responses to be calibrated, form of objective function for calibration, and the issue of stochasticity

in the simulation model. The present work is the first step in this direction.

1.2 Calibration Framework

The calibration of a simulation model is an essential part of the larger evaluation framework consisting of both calibration and validation. If a model is calibrated on some data it will be expected that the model will be able to reproduce the same data. A very simple scheme for a successful calibration and validation study is a “dual sampling scheme.” In this procedure two separate data sets are used, one for model calibration and the other for model validation [22]. This research does not address the validation of the model.

The components of the calibration framework can be described as follows.

1. **Determining the network and data.** The first step in a successful calibration study is the initial values of input parameters. Usually in any complex simulator the number of parameters is so large that only a subset of them can be calibrated. Hence the successful application of the simulator depends on the data collected, regarding the network geometry and vehicle characteristics. Also the calibration data should have the field observations that the simulator is trying to match. In the present study a 5.9-mile stretch of I-880 near Hayward, California is used (see Chapter 2). The data the simulator is trying to match are the speed, flow and occupancy observations from 16 detector stations.
2. **Estimating the origin-destination matrix for the simulator.** Often the origin-destination (O-D) matrix cannot be determined directly from the field observations. Many O-D estimation algorithms need travel times as a necessary input and a preliminary simulation is required to determine them. In this study the O-D matrices used were estimated using the method of Ashok and Ben-Akiva [15]. It needs to be noted that the travel times obtained using the calibrated parameter values may be different from the values used for estimating the O-D matrices. Hence, subsequent iterations may be required for achieving

the required consistency.

3. **Selecting the set of parameters to calibrate.** From among the set of parameters affecting the performance of the simulator, a subset is often isolated based on previous experimental observations or intuitive reasoning. Once this is done a sensitivity study may be useful to identify and document the most sensitive parameters. Note that such a study makes sense only after a subset of response functions has been selected.
4. **Formulating an appropriate objective function.** The most widely used objective function is the minimization of the sum of squares of the difference between the observed field data and the simulated data [22]. Other objective functions noted were maximizing the correlation coefficients, minimizing the deviation of the slope of a straight line, drawn between observed and estimated values, from a 45 degree line [8], and also fitness function formulations for a Genetic Algorithm [7] based optimization study. The optimum parameter value will depend on the objective function used and it will be interesting to study the effect of different objective functions on the optimum parameter value estimates.
5. **Calibrating the parameter values.** Based on the sensitivity study and the nature of the objective function over the parameter domain, an appropriate optimization algorithm needs to be used to obtain the optimal parameter set.

1.3 Thesis Objective

The main objective of this thesis is to propose and implement a systematic calibration framework for a microscopic simulator. This includes discussions on

- the sensitivity of MITSIM to car-following parameters,
- the selection of a set of sensitive parameters,
- the effect of different objective function forms,

- the different approaches considered to tackle stochasticity, and
- calibrated parameters and comparison to starting values.

1.4 Thesis Outline

Chapter 2 discusses the car-following model in MITSIM and justifies the selection of the model parameters for calibration. It then describes the simulation network in detail and graphs are plotted to compare the initial simulated values to the field observations. In Chapter 3 experimental design methodology is explained and implemented to select a set of sensitive parameters from the parameters identified in Chapter 2. Chapter 4 discusses the selection of a simulation output for calibration and compares two objective function forms. Calibration is then performed and the results are discussed. Finally in Chapter 5 the conclusions and directions for future work are discussed.

Appendix A details how the study was performed using the Boss/Quattro analysis software. Appendix B lists all the parameters in MITSIM.

Chapter 2

Simulator, Parameters and Data

This chapter briefly discusses the microscopic traffic simulator—MITSIM. From among the many models affecting the simulator, the parameters in the car following model are chosen for calibration. The range of values over which the parameters are to be varied for the calibration study are also determined. Finally we discuss the network and the data for the calibration study.

2.1 Microscopic Traffic Simulator-MITSIM

A Microscopic Traffic Simulator (MITSIM) was developed at MIT by Yang [24, ch. 3] for modeling traffic flow in networks involving advanced traffic control and route guidance systems. MITSIM represents networks at the lane level and simulates movement of vehicles using car-following, lane-changing and traffic signal response logic. Various parameters determine the output generated by the simulator. The parameters have been classified into the following groups (See also appendix B).

- **Traffic flow Parameters:** These parameters are related to the traffic flow properties like the jam density, the default speed limit, the free speed etc.
- **Simulation Parameters:** These parameters are related to the simulation step size, the vehicle loading models etc.

- **Sensor Device Characteristics:** The working probability of the sensors is controlled by these parameters.
- **Travel Demand:** These parameters are related to the vehicle classes, the fleet mix and driver groups.
- **Vehicle Characteristics:** The performance characteristics of the vehicles is determined by these parameters.
- **Vehicle Routing:** These parameters influence the routing algorithms.
- **Vehicle Movements:** Parameters related to
 - Acceleration model
 - * Car following
 - * Merging
 - * Event Responding
 - Lane changing model
 - * Gap Acceptance
 - * Nosing and Yielding
 - Startup delays

Vehicle movement in MITSIM is controlled by the acceleration model and lane-changing model. The acceleration model is composed of the car-following, merging and event response models. The car-following model computes the acceleration or deceleration of a vehicle in terms of its relationship with and response to the leading vehicle. The merging model guides a vehicle as it moves into a merging area. The event response model captures drivers' responses to events and incidents. The lane-changing model involves gap-acceptance, nosing and yielding and it controls the details of lane switching.

The total number of parameters from the models discussed above was found to be more than 200. Moreover for the network under study, a single simulation run

takes more than twenty minutes. Hence for a feasible calibration study, a subset of parameters need to be identified.

2.1.1 Main Simulation Models

The car-following and lane-changing models can be considered to be the main models influencing the vehicle movements. For the network under study, our aim is to match the aggregate simulated and observed data. The aggregate data that we trying to match are the flow, occupancy and speed across all the sensor stations located in the network over the time period under study (see Figure 2-1). Since we are dealing with aggregate data it is reasonable to assume that the effect of the lane-changing model can be ignored.

Gazis et al. [11] have shown that at steady state, several proposed macroscopic traffic flow theories were equivalent to their microscopic counterparts. In addition, May and Keller [18] showed that by varying the parameter values in Herman's [10] car-following model, different macroscopic speed-density relations can be obtained. Hence it is intuitive to expect that by changing the car-following parameters the simulated speed-flow-density values could be made to follow different macroscopic relations and thereby a better fit could be achieved (See Figure 2-4). Hence in this study we have limited our interest to the car-following model, thereby restricting the number of parameters under consideration.

2.2 Car-Following Model

The car-following model calculates a vehicle's acceleration rate, taking into account its relation with the leading vehicle. The car-following model used in MITSIM draws upon previous research [10]. Depending on the relative magnitude of the current headway to the pre-determined headway thresholds h^{upper} —the upper headway threshold and h^{lower} —the lower headway threshold, a vehicle is classified into one of the following three regimes: *free flowing*, *car following*, and *emergency decelerating*.

Free flowing regime: If the time headway is larger than a pre-determined threshold

h^{upper} , the vehicle does not interact with the leading vehicle and is said to be in a free flowing regime. If its current speed is lower than its target speed, it accelerates at the *maximum acceleration rate* to achieve its target speed. If the current speed is higher than the target speed, the vehicle decelerates at the *normal deceleration rate* to slow down.

Emergency regime: If a vehicle has a headway smaller than a pre-determined threshold h^{lower} , it is in the emergency regime. In this case the vehicle uses an appropriate deceleration rate to avoid collision and extends its headway.

Car-following regime: If a vehicle has a headway between h^{lower} and h^{upper} , it is in the car-following regime. In this case the acceleration rate is calculated based on the following Herman's general car-following model [10]:

$$a_n = \alpha^\pm \frac{v_n^{\beta^\pm}}{(x_n - x_{n-1})^{\gamma^\pm}} (v_{n-1} - v_n) \quad (2.1)$$

where x_{n-1} and v_{n-1} are the position and speed of the leading vehicle, and x_n and v_n are the position and speed of the current vehicle, respectively. The distances are measured from the downstream end. a_n is the acceleration of the current vehicle. The model parameters α^+ , β^+ , and γ^+ are used in acceleration, while α^- , β^- , and γ^- are used in deceleration.

2.2.1 Car-Following Parameters

Based on the previous discussion, the following eight parameters were selected for calibration in the MITSIM car-following model.

cflowerbound: If the time headway is lower than this value then vehicles apply normal deceleration. Hence with a lower value of *cflowerbound*, even under relatively congested conditions a vehicle is more likely to be in the car-following mode.

minresponsedistance: The lower bound for the distance to normal stop for a vehicle is governed by *minresponsedistance*. If space headway is greater than the distance to normal stop, then the vehicle accelerates at the maximum rate and it is in the free flowing regime. This could mean that the lower the *minresponsedistance*, the higher

<i>Parameter</i>	<i>Estimate</i>	<i>Std. err.</i>	<i>t-stat</i>
α^+	9.21	0.141	1.237
β^+	-1.667	0.320	5.201
γ^+	-0.884	0.232	3.818

Table 2.1: Parameter estimates for acceleration, Source[21]

the likelihood that even with closer car spacing a free flow model could be achieved.

Herman’s model parameters: Separate parameters (six in all) are used for accelerating and decelerating vehicles. $(\alpha^+, \alpha^-, \beta^+, \beta^-, \gamma^+, \gamma^-)$

Parameter Value Range

Gazis et al. [11] integrated Herman’s general car-following model, assuming integer values of β and γ to obtain a macroscopic flow-density relationship. Based on a correlation analysis between the flow and density, Gazis et al. reported that the best correlation corresponds to values of β lying in the range of 1 and 3 and γ lying between -1 and 2. This estimation work was further extended by May and Keller [18]. They studied the application of non-integer values for β and γ to explain the macroscopic speed-density relationship. They suggested the following ranges for β and γ : $-1 < \beta < 3$, $-1 < \gamma < 4$. Subramanian [21] suggested a general framework for the estimation of the parameters of a car-following model. He estimated car-following parameters in MITSIM assuming different sensitivities for acceleration and deceleration. These values are given in Table 2.1 and 2.2. Kazi [1] estimated the coefficients of an extended car-following model. The signs of the parameter estimates agree with that of Subramanian’s estimates. However, the coefficient estimates were not used in this work because of the difference in the car-following model.

It is to be noted that for accelerating vehicles the parameters β and γ are negative. Hence, according to Equation 2.1, driver’s acceleration is inversely proportional to speed and proportional to headway. For decelerating vehicles, the parameters β and γ are positive, suggesting that the deceleration applied is proportional to speed and inversely proportional to headway. Based on these studies we selected ranges of

<i>Parameter</i>	<i>Estimate</i>	<i>Std. err.</i>	<i>t-stat</i>
α^-	15.24	0.140	4.282
β^-	1.086	0.278	3.901
γ^-	1.659	0.183	9.077

Table 2.2: Parameter estimates for deceleration, Source[21]

<i>Parameter</i>	<i>Lower Bound</i>	<i>Upper Bound</i>
<i>cflowerbound</i>	0.1-sec	1-sec
<i>minresponsedistance</i>	5-feet	30-feet
α^+	1-(MKS units)	30-(MKS units)
α^-	1-(MKS units)	30-(MKS units)
β^+	-4	-0.1
β^-	0.1	3.0
γ^+	-4	-0.1
γ^-	0.1	4.0

Table 2.3: Range of values for the parameters

parameter values which are presented in Table 2.3.

2.3 Simulation Network and Data

The data for this study was obtained from the Freeway Service Patrol Project ¹. The network used for the calibration study is the same as that used by Yang [24] in his validation study; namely a 5.9 mile stretch of I-880 freeway around Hayward, California. The network and the sensor locations are shown in Figure 2-1. The data considered was only for the north bound traffic. The network has 4 on-ramps and 6 off-ramps. The left-most lane is an HOV lane. The data set, consisting of traffic counts, speed, and occupancy averaged over 5 minute intervals, was acquired from the 16 sensor locations. Using the observed traffic counts and speeds during a 4-hour time period for a number of days, time-dependent Origin-Destination matrices were estimated using the method of Ashok and Ben-Akiva [15]. The data set for the 16

¹URL: <http://www.path.berkeley.edu/FSP/index.html>

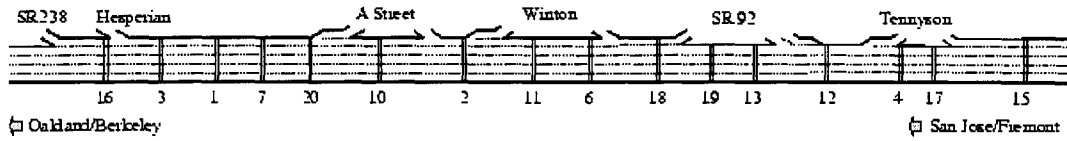


Figure 2-1: I-880 Network

<i>Parameter</i>	<i>Initial Value</i>
<i>cflowerbound</i>	0.5
<i>minresponsedistance</i>	15
α^+	2.15
α^-	1.55
β^+	-1.67
β^-	1.08
γ^+	-0.89
γ^-	1.65

Table 2.4: Initial Values for the Parameters

sensor stations for the day 2/19/93 was used for the calibration study. The simulation was done for the period from 6:50 to 9:10 am and the data was collected from 7:00 to 9:00 am. The parameters used for the base case are the same as those used by Yang [24] for his validation study. These parameter values are presented in Table 2.4.

The scatter plots for the simulated and field observations for the starting parameter values are shown in Figure 2-2. The simulated values were averaged over eight replications. The number of replications necessary was determined using the method given in Appendix C. The results show that the simulated flow fit the actual field data reasonably well while speed and occupancy show a poor fit.

The time-dependent nature of the simulated values is depicted in the contour plots given in Figure 2-3. Note that in this figure the sensor stations have been numbered consecutively with sensor 15 as 1 and sensor 3 as 15 (See Figure 2-1). The plots are drawn with time on the X-axis and the sensor numbers on the Y-axis. Each time unit represents 5 minutes of real time. The contours represent responses of equal value. For example, plot 2-3(a) shows the contour plot for field (plot on the left) and

simulated occupancies (plot on the right). By comparing the actual and simulated contour plots it is clear that in the field congestion builds up around sensor stations 2 and 3 and is then released, but the simulation is not able to replicate this build-up and release of congestion.

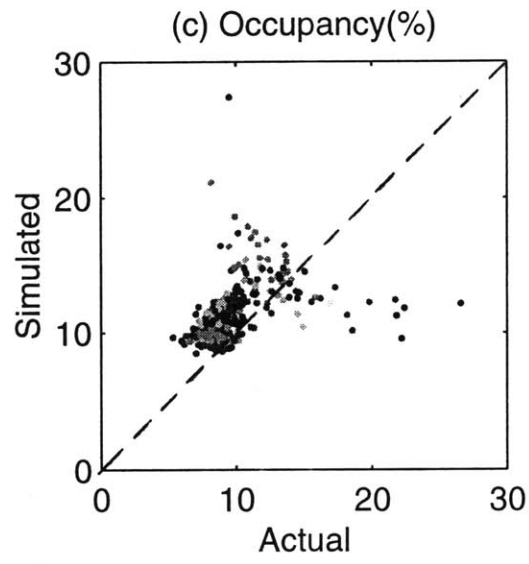
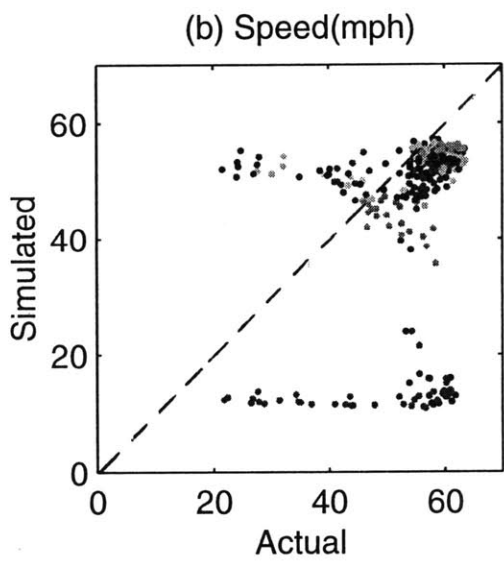
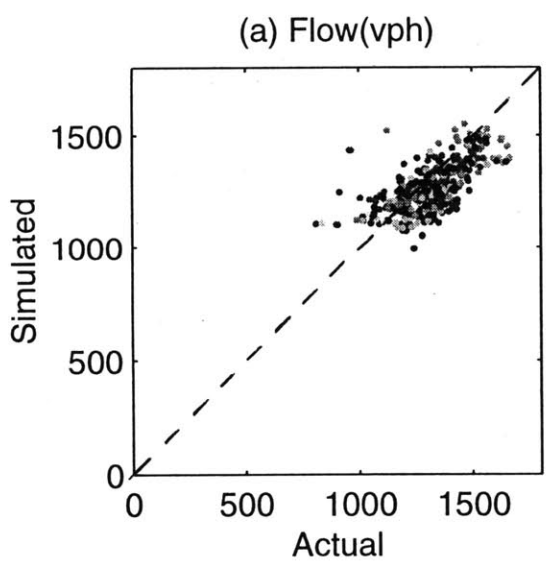
To compare the simulated and observed speed-flow-occupancy values, simulated values are drawn with a '.' and actual field data with a '+' in Figure 2-4. It was observed that in the field, the flow values at sensor stations (1,4,5,6,7,8,9,10,11) were on the left side of the maximum flow while congested flow was observed at sensor stations (2,3,12,13,14,15). However, as is clear from the figure, the simulated values do not show this behaviour.

From Figure 2-1 the congested flow observations correspond to the sensor locations in between the Tennyson on/off-ramps and the sensors in between the A street on-ramp and Hesperian off-ramp. From the speed-occupancy plots shown in Figure 2-4 it is clear that the simulated and observed speed-occupancy plots show substantial difference. As discussed in section 2.1.1 May and Keller [18] showed that by varying the car-following parameter values different speed-occupancy relations can be obtained. Hence, we would expect that the calibration procedure adopted namely, calibrating the car-following model parameters would better simulate the speed-occupancy relationship and the build-up and dissipation of congestion.

2.4 Conclusion

We discussed the microscopic traffic simulator—MITSIM and identified the car-following model parameters for calibration. The range of values for these parameters were also determined. The network under study was discussed and the field data was compared to the simulated data. The next chapter discusses the first step in the calibration process, namely, selecting a sub-set of parameter from the eight parameters (*cflowerbound*, *minresponsedistance*, α^+ , α^- , β^+ , β^- , γ^+ , γ^-) that we identified in this chapter.

Figure 2-2: Comparison between simulated and field data.



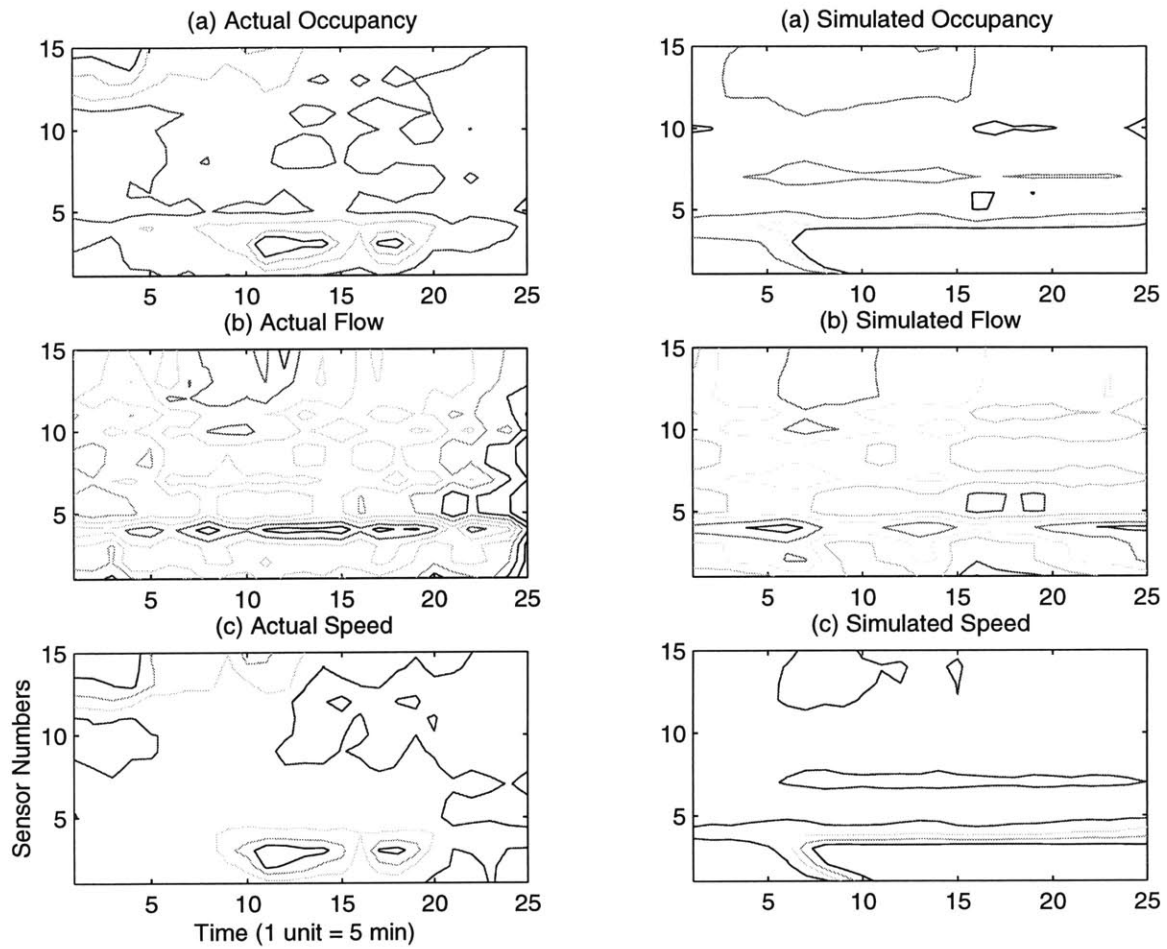


Figure 2-3: Contour plots of simulated and field data.

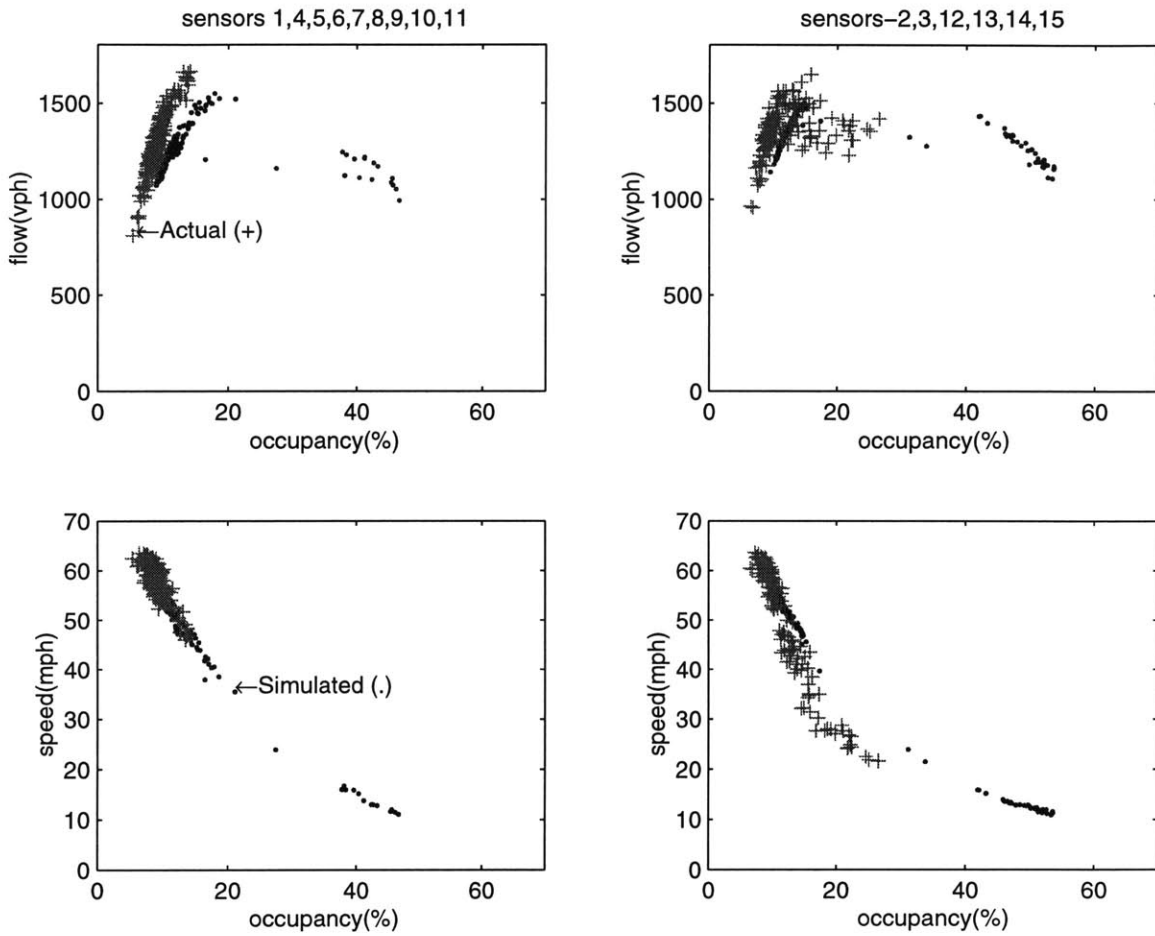


Figure 2-4: Comparison of simulated and field data

Chapter 3

Sensitivity Analysis

It was pointed out in the previous chapter that a single simulation run, of the network under consideration, takes more than twenty minutes of simulation time. Even after selecting the car-following model for calibration, we were left with eight parameters. It would be advantageous to still reduce the number of parameters and calibrate only the sensitive ones. Experimental design is a common methodology used to determine the sensitivity with respect to the given parameters with the least number of simulation runs.

In experimental design, the input parameters are called *factors* and the output performance measures are called *responses*. In our study, the simulation was the experiment, and the input parameters were the *factors*. Since minimizing the deviation between the simulated and actual observations was our goal, we considered different quantitative performance measures (responses) to select the 'best' among them. For a given response, the most sensitive parameters were determined using experimental design techniques.

This chapter discusses a few performance measures (responses) used, a common experimental design technique, and how the set of sensitive parameters were identified using this technique.

3.1 Responses

As discussed earlier, we need quantitative performance measures (responses) for conducting a successful experimental design. Since we had three output measures, namely flow, speed, and occupancy, we used separate performance measures for each of them. In literature, it is a common methodology to use the sum of squares of the deviations as a performance measure. Based on this the following performance measures were suggested.

$$\psi_{flow} = \sum_{all\ sensors} \sum_{all\ time} (simulated_{flow} - actual_{flow})^2$$

$$\psi_{speed} = \sum_{all\ sensors} \sum_{all\ time} (simulated_{speed} - actual_{speed})^2$$

$$\psi_{occupancy} = \sum_{all\ sensors} \sum_{all\ time} (simulated_{occupancy} - actual_{occupancy})^2$$

The three responses were selected for their simplicity. Note that each response corresponds to an output measure of interest. An important question had to be answered using the three different responses. If the sensitivity study showed that the sensitive parameters for the three responses were the same, then minimization of one response measure will affect the others. However, if the sensitive parameters were different, then each response can be calibrated separately.

The sensitive parameters were determined using experimental design methodology and the next section discusses the experimental design technique.

3.2 Experimental Design

If a simulation model has only one factor, then experimental design is simple: we just need to run the simulation at various values, or *levels*, of the factor and from the analysis of the responses, a conclusion can be made as to whether the factor is sensitive or not. However, if there is more than one factor (say k factors), then we also have to take into account the *interactions* between the factors, i.e. whether the effect of one factor depends on the level of other factors. One way to measure the effect of a particular factor would be to keep the level of all the other $k-1$ factors

Design Point	Factor-1	Factor-2	Factor-3	Response
1	-	-	-	R_1
2	+	-	-	R_2
3	-	+	-	R_3
4	+	+	-	R_4
5	-	-	+	R_5
6	+	-	+	R_6
7	-	+	+	R_7
8	+	+	+	R_8

Table 3.1: Design matrix for a 2^3 factorial design

fixed and vary the level of the factor of interest and see how the response changes. The whole process can be repeated for all the other factors. Such a process is not only inefficient but it also does not allow us to measure the interactions between the factors. A more efficient strategy is to use *factorial experimental design* techniques to decide on a systematic way of performing experiments at different factor levels, so that sufficient information can be gathered to identify the sensitive factors and their interactions.

3.2.1 2^k Factorial Design

In a 2^k factorial design, we choose two levels for each factor and then simulations are done for each of the 2^k possible factor-level combinations, which are also called design points. The form of the simulation experiments can be represented in a compact tabular form, also known as a *design matrix*, as shown in Table 3.1, for three factors ($k=3$). The '-' sign is associated with one level of a factor and the '+' sign represents the other. Since there are three factors, we have $2^3 = 8$ possible design points. The variables R_i for $i=1,2,\dots,8$ are the values for the response when running the simulation with the i th combination of the factor levels.

Typically, we are interested in two kinds of effects: the *main* and *interaction* effects. The main effect measures how each factor individually affects the response and the interaction effect measures whether the effect of one factor depends on the

level of other factors. It is easy to calculate the main and interaction effects from the design matrix.

The main effect of factor j is the average change in the response due to moving the factor j from its '-' level to its '+' level while holding all the other factors fixed. This average is taken over all the combinations of the other factor levels in the design. From Table 3.1 the main effect of factor 1 (e_1) is thus

$$e_1 = \frac{(R_2 - R_1) + (R_4 - R_3) + (R_6 - R_5) + (R_8 - R_7)}{4} \quad (3.1)$$

and for factor 2 is

$$e_2 = \frac{(R_3 - R_1) + (R_4 - R_2) + (R_7 - R_5) + (R_8 - R_6)}{4} \quad (3.2)$$

From Table 3.1 and the equations 3.1 and 3.2 a simple formula for the main effect can be devised. To compute e_j we simply apply the signs in the 'Factor-j' column to the corresponding R_i , add them up, and then divide by 2^{k-1} .

The interactions are measured using interaction effects. For a 3 factor design, we have two-factor and three-factor interaction effects. Again an expression for interaction effects can be easily determined from the design matrix. To compute e_{ij} (two-factor i-j interaction), for each row, we multiply the sign in the 'Factor-i' column with the sign in the 'Factor-j' column, with the convention that the product of like signs is a '+' and the product of opposite signs is a '-'. We apply this sign to the corresponding R_i and add them up and divide by 2^{k-1} . Using this formula the interaction effect e_{13} can be written as follows.

$$e_{13} = \frac{(R_1 - R_2) + (R_3 - R_4) + (R_6 - R_5) + (R_8 - R_7)}{4}$$

A similar formula can also be devised for determining the three-factor interaction effect e_{123} .

$$e_{123} = \frac{(R_2 - R_1) + (R_3 - R_4) + (R_5 - R_6) + (R_8 - R_7)}{4}$$

Note that the effects are symmetric; for example $e_{12} = e_{21}$, $e_{13} = e_{31}$, etc. and also $e_{123} = e_{231} = e_{312}$, etc.

3.2.2 Fractional Factorial Designs

In the previous section, an experimental design with three factors was examined. In more involved designs, a large number of factors would be present and the number of design points would become quite large; a design with 8 factors will involve $2^8 = 256$ design points. Hence, when the number of factors are large, a full factorial design is not practical. However, there are a few procedures that can be used to *screen* out unimportant factors.

Fractional factorial design provides a way to get good estimates of the main effects and two-way interactions at a fraction of the computational effort required by a full 2^k factorial design. In a fractional factorial design, we choose a subset (of size 2^{k-p} where $0 < p < k$) from the 2^k design points and run simulations for only these chosen points. If p is large, the number of runs is less, but we would get less information from the design. The information loss is understood in terms of *confounding* in 2^{k-p} designs. Since we are reducing the number of design points, it could happen that the formulas for two effects are the same. In this case the effects are said to be *confounded* with each other.

Resolution is the term used to quantify the effect of confounding in 2^{k-p} factorial designs. In a fractional factorial design, it is guaranteed that two effects are not confounded with each other if the sum of their 'ways' is less than the designs resolution; for this purpose the main effects are regarded as 'one-way' effects. For instance in a resolution *IV* (resolution four) design the main effects are not confounded with two-way interactions ($1 + 2 < 4$), but the two-way interactions are confounded with each other ($2 + 2 = 4$). Thus, resolution *IV* design allows us to obtain main effects but cannot provide reliable two-way effects. A resolution *V* design on the other hand allows us to obtain main and two-way effects, but three-way effects will be confounded with two-way effects.

Once we have determined the factors k and the desired resolution, a subset of

design points (i.e p in 2^{k-p}) has to be determined. The design matrix, for a given number of factors and desired resolution, can be obtained from experimental design literature [17].

3.2.3 Response Functions

A simulation model can be considered as a function with the responses depending on the input parameters. The form of the function is unknown, and response function methodology is a method used to approximate this function. Depending on the resolution of the experimental design technique, different functional forms can be fitted to the responses 3.3. The coefficients of the response function are usually estimated using least-square regression. Factorial designs, are actually based on regression meta-models and the regression least-square estimates are related to the effects estimates in experimental design. Note that if we are considering a two factorial design, since a variable has only two levels, the response function as given in 3.3 can be equivalently represented and estimated using a dummy variable for each parameter.

The methodology we followed, to find the sensitive parameters, was to first fit a polynomial function, with single order and all the two-factor interaction terms, to the response. Then the influence of each term on the response variation was analyzed, and less significant ones were dropped in the final model. Thus, the final model had only the most significant parameters.

3.2.4 Stochasticity

Since we were dealing with a stochastic simulator, we had to replicate the simulation at each design point. The simulation was replicated five times at each design point (See Appendix C), and the response function was fitted to the average values of the responses over the five replications.

3.3 Results

A fractional factorial design of resolution five: 2_{IV}^8 was conducted for the eight parameters identified in Chapter 2. The design matrix is given in Table 3.2. For each design point, the three responses as given in Section 3.1 were evaluated. The plot of the responses versus the iterations is shown in Figure 3-1. From this figure it is not evident whether the three responses are sensitive to the same set of parameters.

To determine this, each response was initially represented by a polynomial function, with all the one-way and two-way interaction effects. The form of the response function is given in 3.3

$$R = \mu + \sum_{i=1}^8 \beta_i x_i + \sum_{i=1, i < j}^8 \sum_{j=1}^8 \beta_{ij} x_i x_j \quad (3.3)$$

where μ is the constant and R is the response.

x_1 represents the variable *minrespdistance*, x_2 is *cflowerbound*, x_3 is α_+ , x_4 is β_+ , x_5 is γ_+ , x_6 is α_- , x_7 is β_- , and x_8 is γ_- .

The β are the coefficients to be determined from the analysis.

Finally the influence of each term on the response was analyzed and less significant ones were dropped in the final model

As an example consider the response function given by

$$\psi_{flow} = \sum_{all\ sensors} \sum_{all\ time} (simulated_{flow} - actual_{flow})^2$$

The response is approximated as shown in Equation 3.3. The first step in determining the sensitive variables is to screen out the less sensitive variables. For this, an analysis of variance is conducted and the influence of each term on the response variation is analyzed. The result of such an analysis using the BOSS/Quatro software program (See Appendix A) is given in Table 3.3

Each row in Table 3.3 represents a term in Equation 3.3. The SOURCES column represents the variables. As shown in Equation 3.3 we have eight variables, they are represented by the rows from 1 to 8. The next set of rows represent the two-way

Design Point	minrespdistance	cflowerbound	α_+	β_+	γ_+	α_-	β_-	γ_-
1	5	0.1	1	-4	-4	1	3	4
2	30	0.1	1	-4	-4	1	0.1	0.1
3	5	1	1	-4	-4	1	0.1	0.1
4	30	1	1	-4	-4	1	3	4
5	5	0.1	30	-4	-4	1	0.1	4
6	30	0.1	30	-4	-4	1	3	0.1
7	5	1	30	-4	-4	1	3	0.1
8	30	1	30	-4	-4	1	0.1	4
9	5	0.1	1	-0.1	-4	1	0.1	4
10	30	0.1	1	-0.1	-4	1	3	0.1
11	5	1	1	-0.1	-4	1	3	0.1
12	30	1	1	-0.1	-4	1	0.1	4
13	5	0.1	30	-0.1	-4	1	3	4
14	30	0.1	30	-0.1	-4	1	0.1	0.1
15	5	1	30	-0.1	-4	1	0.1	0.1
16	30	1	30	-0.1	-4	1	3	4
17	5	0.1	1	-4	-0.1	1	3	0.1
18	30	0.1	1	-4	-0.1	1	0.1	4
19	5	1	1	-4	-0.1	1	0.1	4
20	30	1	1	-4	-0.1	1	3	0.1
21	5	0.1	30	-4	-0.1	1	0.1	0.1
22	30	0.1	30	-4	-0.1	1	3	4
23	5	1	30	-4	-0.1	1	3	4
24	30	1	30	-4	-0.1	1	0.1	0.1
25	5	0.1	1	-0.1	-0.1	1	0.1	0.1
26	30	0.1	1	-0.1	-0.1	1	3	4
27	5	1	1	-0.1	-0.1	1	3	4
28	30	1	1	-0.1	-0.1	1	0.1	0.1
29	5	0.1	30	-0.1	-0.1	1	3	0.1
30	30	0.1	30	-0.1	-0.1	1	0.1	4
31	5	1	30	-0.1	-0.1	1	0.1	4
32	30	1	30	-0.1	-0.1	1	3	0.1
33	5	0.1	1	-4	-4	30	3	0.1
34	30	0.1	1	-4	-4	30	0.1	4
35	5	1	1	-4	-4	30	0.1	4
36	30	1	1	-4	-4	30	3	0.1
37	5	0.1	30	-4	-4	30	0.1	0.1
38	30	0.1	30	-4	-4	30	3	4
39	5	1	30	-4	-4	30	3	4
40	30	1	30	-4	-4	30	0.1	0.1
41	5	0.1	1	-0.1	-4	30	0.1	0.1
42	30	0.1	1	-0.1	-4	30	3	4
43	5	1	1	-0.1	-4	30	3	4
44	30	1	1	-0.1	-4	30	0.1	0.1
45	5	0.1	30	-0.1	-4	30	3	0.1
46	30	0.1	30	-0.1	-4	30	0.1	4
47	5	1	30	-0.1	-4	30	0.1	4
48	30	1	30	-0.1	-4	30	3	0.1
49	5	0.1	1	-4	-0.1	30	3	4
50	30	0.1	1	-4	-0.1	30	0.1	0.1
51	5	1	1	-4	-0.1	30	0.1	0.1
52	30	1	1	-4	-0.1	30	3	4
53	5	0.1	30	-4	-0.1	30	0.1	4
54	30	0.1	30	-4	-0.1	30	3	0.1
55	5	1	30	-4	-0.1	30	3	0.1
56	30	1	30	-4	-0.1	30	0.1	4
57	5	0.1	1	-0.1	-0.1	30	0.1	4
58	30	0.1	1	-0.1	-0.1	30	3	0.1
59	5	1	1	-0.1	-0.1	30	3	0.1
60	30	1	1	-0.1	-0.1	30	0.1	4
61	5	0.1	30	-0.1	-0.1	30	3	4
62	30	0.1	30	-0.1	-0.1	30	0.1	0.1
63	5	1	30	-0.1	-0.1	30	0.1	0.1
64	30	1	30	-0.1	-0.1	30	3	4

Table 3.2: Design matrix for a 2^8_V factorial design

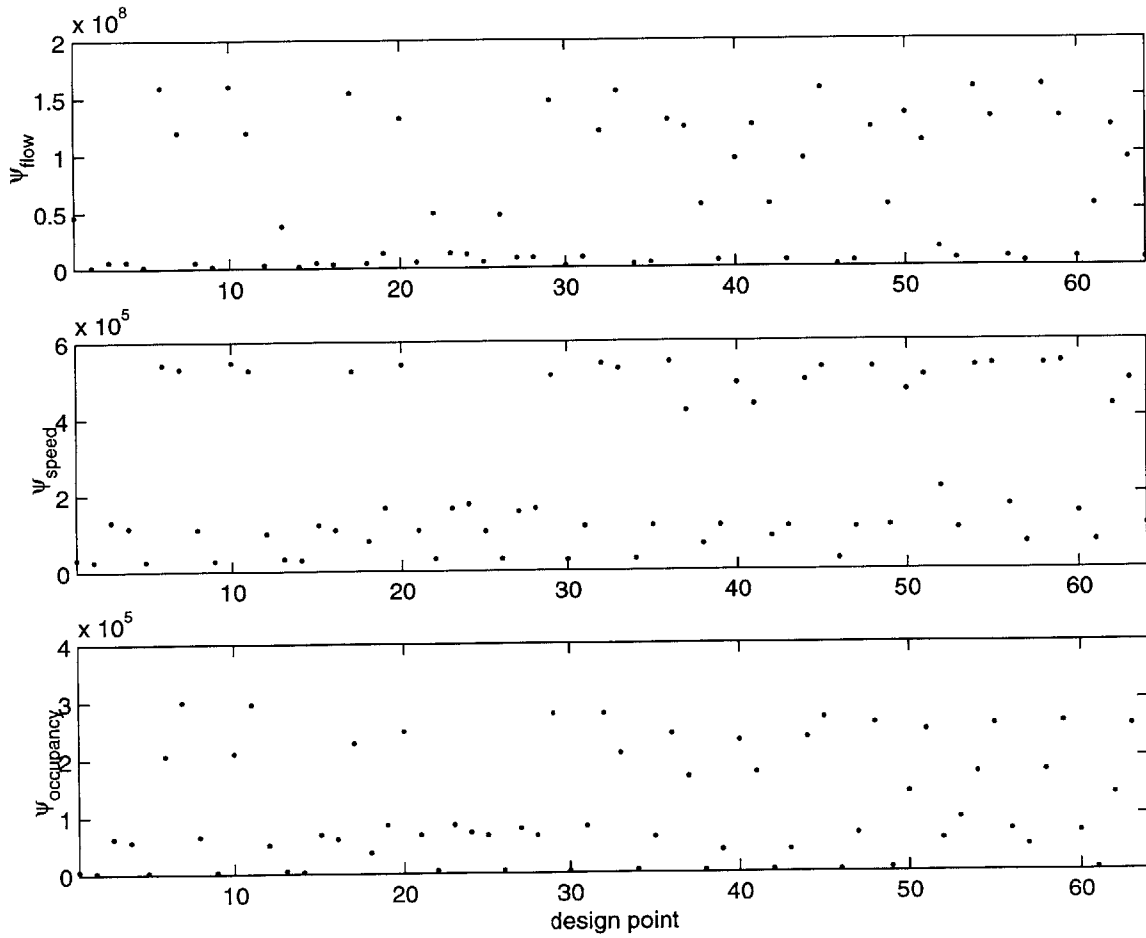


Figure 3-1: Plot of the responses versus the design points

interaction effects. For example, the ninth row with numbers 1 2 under the sources column represent the two-way interaction between the variables x_1 and x_2 . Since we have eight variables we have a total of $8C_2 = 28$ two-way interaction effects. The SUM OF SQUARES column gives the variation due to each variable. The residual variation is given in the second to last row, and the total variation is given in the last row. DF denotes the degree of freedom. MEAN SQUARED is an estimate of the variance. The residual mean squared is also given in the second to last row. The column F-LEVEL gives the result of the F-test: for each variable the ratio of its mean squared to the residual mean squared. Based on the F-test a conclusion is made as to a variable's significance.

We can see that the main effect of only a few parameters are significant. Also note that the interaction effects are significant only for those parameters with significant main effects. So, in the next step, the response function is modified to include only the significant terms, and the analysis is done again. The result is shown in Table 3.4. The same procedure is applied to the response ψ_{speed} and $\psi_{occupancy}$. The results for the final model are shown in Tables 3.5 and 3.6.

3.4 Conclusion

We can see that the four parameters ($c_{flowerbound}, \alpha^-, \beta^-, \gamma^-$) (see Chapter 2) were found to be sensitive for the three responses. Moreover, these four parameters affect the deceleration characteristics of a vehicle. It appears that the parameters affecting the deceleration characteristics are more sensitive to the current data set. Also note that since the sensitive parameters were found to be the same for the three responses, the minimization of one response will definitely affect others.

The results obtained depends on the range and levels of the parameters. The range of the paramater values should be correctly determined . As discussed in section 2.2.1 the values we used were based on previous estimation studies. Moreover, when we use a 2^k factorial design, we are assuming that the parameter effects vary linearly between the '+' and '-' values. This could be improved by using a 3^k factorial design,

though with increased computational costs.

The next chapter discusses the calibration of these parameters over the range of values determined in Chapter 2.

ANALYSIS OF VARIANCE

SOURCES	SUM OF SQUARES	DF.	MEAN SQUARED	F-LEVEL	CONCLUSION
1	0.1677211E+14	1	0.167721E+14	0.323415E-01	NO SIGNIFICANT
2	0.6921709E+16	1	0.692171E+16	0.133471E+02	HIGH. SIGNIFICANT
3	0.1128680E+14	1	0.112868E+14	0.217642E-01	NO SIGNIFICANT
4	0.1963264E+15	1	0.196326E+15	0.378574E+00	NO SIGNIFICANT
5	0.1371702E+15	1	0.137170E+15	0.264504E+00	NO SIGNIFICANT
6	0.1382006E+17	1	0.138201E+17	0.266491E+02	HIGH. SIGNIFICANT
7	0.4448516E+17	1	0.444852E+17	0.857803E+02	HIGH. SIGNIFICANT
8	0.1105315E+18	1	0.110531E+18	0.213137E+03	HIGH. SIGNIFICANT
1 2	0.3015633E+14	1	0.301563E+14	0.581502E-01	NO SIGNIFICANT
1 3	0.7470733E+13	1	0.747073E+13	0.144057E-01	NO SIGNIFICANT
1 4	0.1449746E+13	1	0.144975E+13	0.279553E-02	NO SIGNIFICANT
1 5	0.9150158E+14	1	0.915016E+14	0.176442E+00	NO SIGNIFICANT
1 6	0.2347503E+11	1	0.234750E+11	0.452667E-04	NO SIGNIFICANT
1 7	0.1427141E+14	1	0.142714E+14	0.275194E-01	NO SIGNIFICANT
1 8	0.1352122E+14	1	0.135212E+14	0.260729E-01	NO SIGNIFICANT
2 3	0.4943016E+12	1	0.494302E+12	0.953157E-03	NO SIGNIFICANT
2 4	0.2385160E+14	1	0.238516E+14	0.459928E-01	NO SIGNIFICANT
2 5	0.2293346E+14	1	0.229335E+14	0.442224E-01	NO SIGNIFICANT
2 6	0.3730828E+15	1	0.373083E+15	0.719412E+00	NO SIGNIFICANT
2 7	0.4156797E+16	1	0.415680E+16	0.801551E+01	HIGH. SIGNIFICANT
2 8	0.1148690E+14	1	0.114869E+14	0.221501E-01	NO SIGNIFICANT
3 4	0.1171587E+14	1	0.117159E+14	0.225916E-01	NO SIGNIFICANT
3 5	0.2205883E+13	1	0.220588E+13	0.425359E-02	NO SIGNIFICANT
3 6	0.4250261E+13	1	0.425026E+13	0.819574E-02	NO SIGNIFICANT
3 7	0.1626635E+13	1	0.162663E+13	0.313663E-02	NO SIGNIFICANT
3 8	0.1626996E+14	1	0.162700E+14	0.313732E-01	NO SIGNIFICANT
4 5	0.1655424E+15	1	0.165542E+15	0.319214E+00	NO SIGNIFICANT
4 6	0.3419700E+13	1	0.341970E+13	0.659418E-02	NO SIGNIFICANT
4 7	0.3463867E+10	1	0.346387E+10	0.667934E-05	NO SIGNIFICANT
4 8	0.3454109E+13	1	0.345411E+13	0.666053E-02	NO SIGNIFICANT
5 6	0.9820832E+13	1	0.982083E+13	0.189374E-01	NO SIGNIFICANT
5 7	0.1073757E+14	1	0.107376E+14	0.207052E-01	NO SIGNIFICANT
5 8	0.1830763E+12	1	0.183076E+12	0.353024E-03	NO SIGNIFICANT
6 7	0.9438494E+16	1	0.943849E+16	0.182002E+02	HIGH. SIGNIFICANT
6 8	0.1109459E+17	1	0.110946E+17	0.213936E+02	HIGH. SIGNIFICANT
7 8	0.1358164E+17	1	0.135816E+17	0.261894E+02	HIGH. SIGNIFICANT
Residual	0.1400204E+17	27	0.518594E+15		
Total	0.229213E+18	63			

Table 3.3: Screening for ψ_{flow}

ANALYSIS OF VARIANCE

SOURCES	SUM OF SQUARES	DF.	MEAN SQUARED	F-LEVEL	CONCLUSION
2	0.1382006E+17	1	0.138201E+17	0.207497E+02	HIGH. SIGNIFICANT
6	0.9438494E+16	1	0.943849E+16	0.141711E+02	HIGH. SIGNIFICANT
7	0.1109459E+17	1	0.110946E+17	0.166576E+02	HIGH. SIGNIFICANT
8	0.1358164E+17	1	0.135816E+17	0.203917E+02	HIGH. SIGNIFICANT
2 7	0.1105315E+18	1	0.110531E+18	0.165954E+03	HIGH. SIGNIFICANT
6 7	0.1358164E+17	1	0.135816E+17	0.203917E+02	HIGH. SIGNIFICANT
6 8	0.1109459E+17	1	0.110946E+17	0.166576E+02	HIGH. SIGNIFICANT
7 8	0.9438494E+16	1	0.943849E+16	0.141711E+02	HIGH. SIGNIFICANT
Residual	0.3663201E+17	55	0.666037E+15		
Total	0.229213E+18	63			

ALLOCATION OF VARIATION

% VARIATION	EQUALS
explained by 2	6.03%
explained by 6	4.12%
explained by 7	4.84%
explained by 8	5.93%
explained by 2 7	48.22%
explained by 6 7	5.93%
explained by 6 8	4.84%
explained by 7 8	4.12%
unexplained	15.98%

SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARE	F-LEVEL	CONCLUSION	R-SQUARED
REGRESSION	0.21403D+18	8	0.26754D+17	0.969D+02	HIGH. SIGNIFICANT	0.93376
RESIDUAL	0.15183D+17	55	0.27606D+15			
TOTAL	0.22921D+18	63				

Maximum deviation : 1136.02 %
 Minimum R-squared : 0.93376

Table 3.4: Result for the final model for ψ_{flow}

ANALYSIS OF VARIANCE

SOURCES	SUM OF SQUARES	DF.	MEAN SQUARED	F-LEVEL	CONCLUSION
2	0.1831540E+12	1	0.183154E+12	0.123742E+03	HIGH. SIGNIFICANT
6	0.1139058E+12	1	0.113906E+12	0.769565E+02	HIGH. SIGNIFICANT
7	0.1193230E+12	1	0.119323E+12	0.806165E+02	HIGH. SIGNIFICANT
8	0.2362965E+12	1	0.236297E+12	0.159646E+03	HIGH. SIGNIFICANT
2 7	0.1620284E+13	1	0.162028E+13	0.109469E+04	HIGH. SIGNIFICANT
6 7	0.2362965E+12	1	0.236297E+12	0.159646E+03	HIGH. SIGNIFICANT
6 8	0.1193230E+12	1	0.119323E+12	0.806165E+02	HIGH. SIGNIFICANT
7 8	0.1139058E+12	1	0.113906E+12	0.769565E+02	HIGH. SIGNIFICANT
Residual	0.8140719E+11	55	0.148013E+10		
Total	0.282390E+13	63			

ALLOCATION OF VARIATION

% VARIATION	EQUALS
explained by 2	6.49%
explained by 6	4.03%
explained by 7	4.23%
explained by 8	8.37%
explained by 2 7	57.38%
explained by 6 7	8.37%
explained by 6 8	4.23%
explained by 7 8	4.03%
unexplained	2.88%

SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARE	F-LEVEL	CONCLUSION	R-SQUARED
REGRESSION	0.26134D+13	8	0.32667D+12	0.853D+02	HIGH. SIGNIFICANT	0.92544
RESIDUAL	0.21054D+12	55	0.38281D+10			
TOTAL	0.28239D+13	63				

Maximum deviation : 285.01 %
 Minimum R-squared : 0.92544

Table 3.5: Result for final model for ψ_{speed}

ANALYSIS OF VARIANCE

SOURCES	SUM OF SQUARES	DF.	MEAN SQUARED	F-LEVEL	CONCLUSION
2	0.1660677E+11	1	0.166068E+11	0.999751E+02	HIGH. SIGNIFICANT
6	0.3959231E+11	1	0.395923E+11	0.238351E+03	HIGH. SIGNIFICANT
7	0.2288750E+11	1	0.228875E+11	0.137786E+03	HIGH. SIGNIFICANT
8	0.8045689E+11	1	0.804569E+11	0.484362E+03	HIGH. SIGNIFICANT
2 7	0.3602031E+12	1	0.360203E+12	0.216847E+04	HIGH. SIGNIFICANT
6 7	0.8045689E+11	1	0.804569E+11	0.484362E+03	HIGH. SIGNIFICANT
6 8	0.2288750E+11	1	0.228875E+11	0.137786E+03	HIGH. SIGNIFICANT
7 8	0.3959231E+11	1	0.395923E+11	0.238351E+03	HIGH. SIGNIFICANT
Residual	0.9135994E+10	55	0.166109E+09		
Total	0.671819E+12	63			

ALLOCATION OF VARIATION

	% VARIATION	EQUALS
explained by 2	2.47%	
explained by 6	5.89%	
explained by 7	3.41%	
explained by 8	11.98%	
explained by 2 7	53.62%	
explained by 6 7	11.98%	
explained by 6 8	3.41%	
explained by 7 8	5.89%	
unexplained	1.36	

SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARE	F-LEVEL	CONCLUSION	R-SQUARED
REGRESSION	0.61533D+12	8	0.76916D+11	0.749D+02	HIGH. SIGNIFICANT	0.91591
RESIDUAL	0.56491D+11	55	0.10271D+10			
TOTAL	0.67182D+12	63				

Maximum deviation : 1044.84 %
 Minimum R-squared : 0.91591

Table 3.6: Result for final model for $\psi_{occupany}$

Chapter 4

Calibration

In the previous chapter we identified (*cf lowerbound*, α^- , β^- , γ^-) as the set of sensitive parameters. The next step was to calibrate these parameters. In this thesis, calibration is done by optimizing an objective function criteria. However, some important issues had to be addressed. The first task was to identify the particular response to calibrate from among flow, speed and occupancy. Furthermore, there were different ways in which the sensor data across space and time could be combined to form a suitable objective function. Moreover, due to the stochasticity of the simulator, the calibrated parameter values were also stochastic. Hence, we had to address this issue while calibrating the parameter values. Lastly, the nature of the search space determined the effectiveness of the optimization methodology.

The following issues are discussed in this chapter before presenting the results

- the selection of a response for calibration,
- the effect of two different objective function forms in calibrating the parameters,
- the effect of stochasticity in calibrating the parameters, and
- the optimization methodology adopted.

Finally, simulation is done with calibrated parameter values and the simulation output is compared to the observed actual data.

4.1 Response Selection

When calibrating a simulator, our goal is to minimize the deviation between the observed and simulated values of flow, speed, and occupancy. In this thesis, we have adopted a simpler approach and have focused our attention on minimizing only the deviation in speed. This helps us focus on other aspects of the problem, namely the effect of using different objective function forms, stochasticity, and the nature of the search domain. Moreover, once we have a procedure to minimize the deviation in speed, the next step could be to study the effect of minimizing a weighted combination of all the three responses.

In determining the O-D matrices by the method of Ashok and Ben-Akiva [15], the simulated and observed flow values were already matched. Hence, we narrowed our investigation to minimizing the deviation in speed or occupancy. In a later section (see Section 4.5.2), it is shown that the correlation between the deviation in flow and speed, was higher than that between flow and occupancy. This justified our approach of minimizing the deviation in speed.

4.2 Objective Function Forms

Once a response was selected for calibration, the next step was to explore different objective function forms. We had sensor data from different locations and time periods. If we assume, that the parameters of the car-following model do not change with respect to geometry and traffic conditions, more efficient estimates could be obtained by pooling the sensor data.

Often an objective function as shown in Equation 4.1 is used for minimizing the deviation between the simulated and observed speed.

$$\psi_{speed} = \sum_{all\ sensors} \sum_{all\ time} (simulated_{speed} - actual_{speed})^2 \quad (4.1)$$

However, pooling of data adds new dimensions of difficulty. The observations may be correlated along both time and space. This is very significant in the case of a traffic

simulation model, where the speed values across sections are correlated because of the propagation of flow and shock waves. The build up and release of congestion will produce the correlation along time. In addition, there is also the problem of heteroscedasticity, wherein, the error associated with the estimated speed values will depend on its value.

When we formulate an objective function to minimize the deviation in the simulated speed, all the above factors have to be considered. It is difficult to account for the effect of correlation in devising an appropriate objective function form. Moreover, if correlations across both space and time are present then there could be situations in which pooling of data could lead to these effects canceling each other. However, if we suspect that the error associated with the speed depends on the value of speed then we can account for this by dividing each term in 4.1 by the speed value. Such an objective function is shown below.

$$\phi_{speed} = \sum_{all\ sensors} \sum_{all\ time} \left(\frac{simulated_{speed} - actual_{speed}}{actual_{speed}} \right)^2 \quad (4.2)$$

Different objective function forms can be formulated based on the assumptions regarding the error terms. In our work, we confine our attention to the two objective function forms. If our assumption about the error term is correct we would expect more efficient optimal parameter estimates when using the second objective function form.

4.3 Stochasticity

A common procedure to take into account stochasticity is to run several replications of the simulation, at each parameter value, and to take the mean value of the outputs. But even with the small network under consideration a single replication took twenty minutes of simulation time. Since multiple replication approach is expensive, we attempt to account for stochasticity by fixing the random number seed. But the validity of the parameter estimates obtained by such a technique is questionable. To understand the effect of stochasticity, we calibrate separately for each random number

seed. The parameter value obtained for a given random number seed is a particular realization of its actual value. Hence to obtain a better estimate we determine the sampling distribution of the calibrated parameters. The whole process is illustrated in the example given below.

Assume that we have two objective function forms given by χ and ω

$$\chi = \sum_{i=1}^N (z_{simulated}(p1, p2) - z_{observed})^2$$

$$\omega = \sum_{i=1}^N \left(\frac{z_{simulated}(p1, p2) - z_{observed}}{z_{observed}} \right)^2$$

where z is the response we are focusing our attention on, and N is the number of data points. Also assume that we have two parameter values to calibrate, namely $p1$ and $p2$. When we calibrate the values of $(p1, p2)$ we are essentially trying to determine that combination of $(p1, p2)$ which will minimize the respective objective function. Consider Figure 4-1(a)¹. It shows the objective function χ over the domain for a fixed random number seed. Note that when we fix the random number seed, for a given value of $(p1, p2)$ the response is always the same. Now the optimal calibrated parameter will be that value which minimizes the objective function.

Since we are dealing with a stochastic simulator; for a given value of $(p1, p2)$ the responses form a distribution. Hence if we run the simulator with different random number seeds we will get different surfaces and the family of such surfaces represent the true response function. This is indicated in Figure 4-1(b) where we have plotted the objective function for three different random number seeds. Note that the optimal values of the parameters could be different for each seed. The optimal parameters are thus stochastic and are distributed around a mean value, which we are trying to estimate. Now from a limited number of runs we can calculate the sampling distribution of these parameters.

In Figure 4-1(c) we have the objective function ω . Note that for this objective function, the distribution of the optimal parameters is different from our first one.

¹Note that these figures are not for the problem at hand; they are only for illustration.

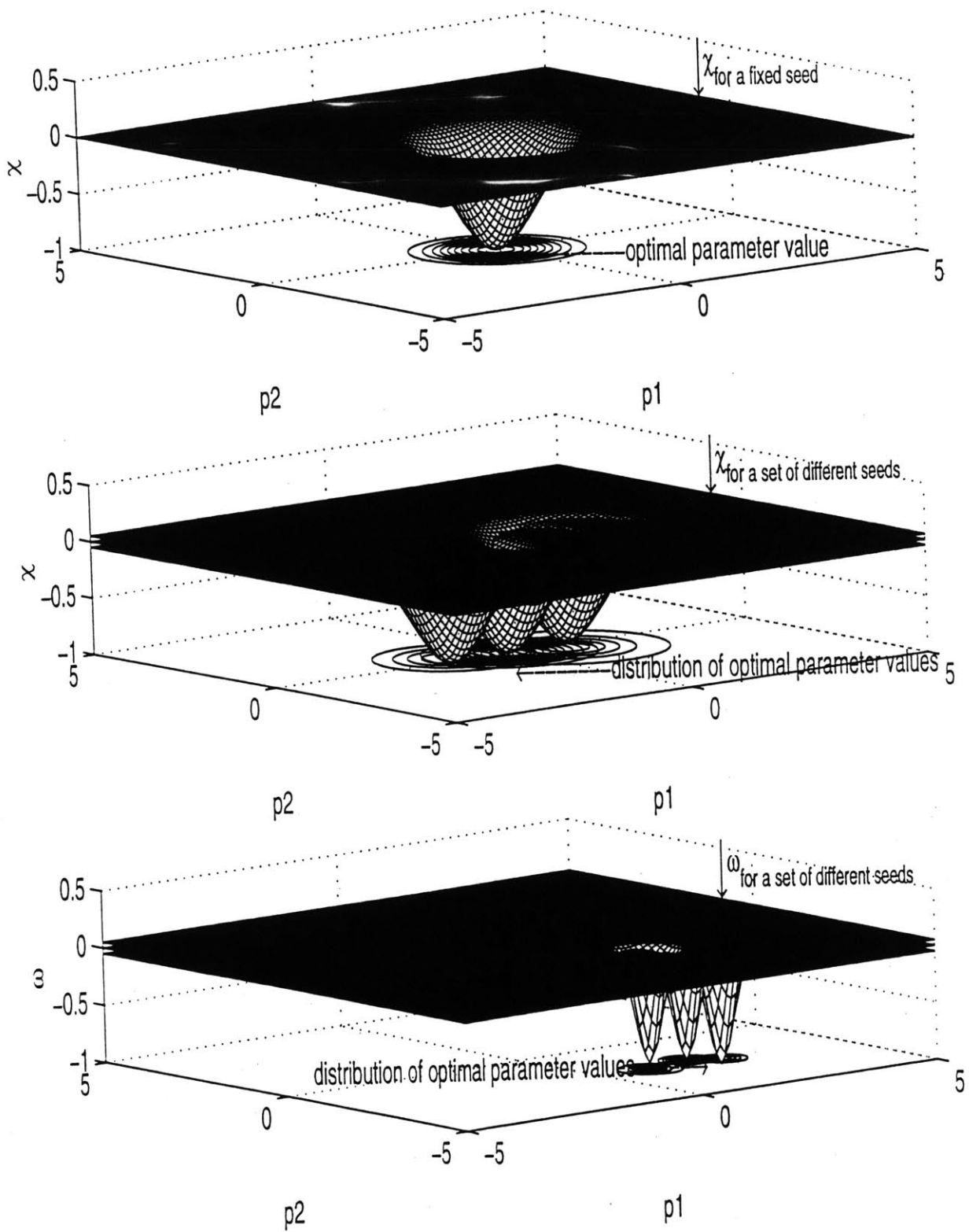


Figure 4-1: Illustrating the effect of stochasticity in calibrating parameter values

Using a different objective function form is equivalent to using a different estimator. Moreover, in simulation it is difficult to determine the properties of these estimators and we assume that both the estimators are consistent.

In the case of Linear Least Squares estimation it can be shown that the variance of the parameter estimates depends on the variance of the error term. Hence we would also expect that in our case (non-linear) the variance of our parameter estimates would depend on the error term variance. Since the objective function forms we have is actually a sum of the estimated errors, we would expect that a better objective function formulation i.e., the one accounting for heteroscedasticity, would give us better estimates of errors. If we have better error estimates we would also expect that the variation of the objective function would be smaller. Hence it would be interesting to see whether the objective function which we corrected for an assumed heteroscedasticity would have better properties than a simple sum of squares form.

4.4 Optimization

The calibration of a simulator by minimizing an objective function form is essentially a nonlinear estimation problem. Hence as discussed in [20] we could use direct search, or an optimization based methodology. In direct search the simulation needs to be done at all possible combinations of the parameter values. This is a very expensive process especially in traffic simulation. In an optimization-based methodology we start with an initial guess of parameter values, and based on a chosen direction of descent, the parameter values are successively altered until we get an optimal value.

In an optimization-based method there is no guarantee that the search process will converge, and even if does converge we cannot determine whether it is a local optimum or a global optimum. But this is a very generic problem in any optimization process and appropriate robust optimization methodology need to be used depending on the nature of the search domain. But in our case we first need to determine whether we have local optima in our search space. If this can be determined, then in future work it makes sense to use robust heuristic methods like Genetic Algorithms

for parameter calibration [7].

4.4.1 Optimization Methodology

The optimization problem that we are trying to solve is non-explicit and non-linear with respect to the parameter space. Hence we used an approximation to the objective function based on function evaluations (simulation runs). Once an approximation of the function is obtained, the optimal parameter values can be determined. However the actual function computed at this new point may not correspond to the value computed using the approximation. Hence an iterative process of evaluation, approximation and optimization is done until convergence is obtained. Note that we are trying to minimize a non-differentiable function using gradient-based methods. This may lead to troubles in the convergence of the optimization process. Moreover due to local optima we also need to explore the search space from different starting points.

For our optimization, we used the methodology provided by the Boss/Quattro analysis program (See Appendix A). We used the Method of Diagonal Quadratic approximation (MDQA), a second order algorithm, meaning it uses values from previous analyses for computing the diagonal terms of the hessian matrix (second order terms). To verify that the optimal points obtained are local optima we also ran a first order Convex Linearization based algorithm at each optimal point.

We first fixed the random number seed for the simulation and optimized the parameter values for different starting values. We then selected the best of these values obtained and repeated the whole process for different random number seed values. From the selected values of the optimal parameters, we finally determined the sampling distribution of the parameter estimates. We also compared the distribution obtained with the two different objective function forms.

4.5 Results

The variation of the two objective function forms were first compared. Then, the convergence property of the objective function forms, for a fixed seed, was studied.

The correlation of the objective function forms to the deviations in occupancy and flow was also determined. Optimal parameter values were then estimated for a fixed random number seed. The results obtained using these optimal parameters were compared to the one from the initial parameter value. Finally a sampling distribution of the optimal parameter values were determined and the results were compared to the starting value.

4.5.1 Comparison of the Spread of the Objective Functions

As discussed before, we had two different objective function forms under consideration (equations 4.1 and 4.2). Since the objective function forms we have are the sum of the estimated errors, we would expect that the function form corrected for heteroscedasticity will have lower variance. Hence it would be interesting to compare the variation of the two objective function forms. If the second form has lower variation then it shows that correction applied for heteroscedasticity was appropriate.

The variance of the estimated parameter values depend on the error term variance. The objective function variation also depends on the error term variance. Hence, an objective function with a lower variation would be expected to give more efficient parameter estimates.

We compare the variation of the two objective functions using the coefficient of variation, defined by

$$C_v = \sigma/\mu$$

where σ is the standard deviation, μ is the mean and C_v is the coefficient of variation. The two objective functions were evaluated at different points in the parameter space. For each of these points the variance of the objective function was estimated by replicating the simulation run twenty times. The properties for the objective function estimated at four different parameter points are given in Table 4.1 for illustration.

It was found that the variation depends on the point in the parameter space. Moreover it was found that, as expected, the variation of the objective function corrected for heteroscedasticity was at best lower or at worst equal to that of the sum

Point (<i>cflower</i> , α_- , β_- , γ_-)	Objective function	Mean	St. Deviation	C_v
(0.34,3.75,0.89,2.71)	ψ	34805	15800	0.4540
	ϕ	28.93	3.34	0.1154
(0.1,1,0.1,0.1)	ψ	44864	23883	0.5323
	ϕ	30.21	6.04	0.1999
(0.1,12.4,0.1,4)	ψ	30067	1276	0.0424
	ϕ	24.53	1.30	0.0530
(0.1,1.55,1.08,1.65)	ψ	101808	2766	0.0272
	ϕ	44.56	1.07	0.0240

Table 4.1: Comparing the variation for the two objective function forms

of squares form. Hence, we could expect more efficient parameter estimates using the second form of the objective function.

4.5.2 Results for a Fixed Random Number Seed

This section summarizes the result obtained for a fixed random number seed.²

Convergence

The convergence of the algorithm was found to depend on the starting value. The convergence for four different starting points while optimizing for the function ψ is given in Figure 4-2. The variation in the function ϕ while optimizing for ψ is also plotted. The convergence for optimizing the function ϕ is given in Figure 4-3³.

This process was repeated for other values of random number seed. Another set of plots is given in Figures 4-4 and 4-5⁴.

The convergence of the objective function was found to depend on the starting parameter values. Moreover, the converged parameter values were also dependent on the starting values. This indicates that the search domain is highly non-linear even for a fixed random number seed. In addition, changing the random seed significantly changes the convergence. This is because we are using a gradient based optimization

²In MITSIM random number seed is specified by using the -R xxxxxxxx option where each x is any number from 0 to 9

³using x=2 for all x's

⁴using x=5 for all x's

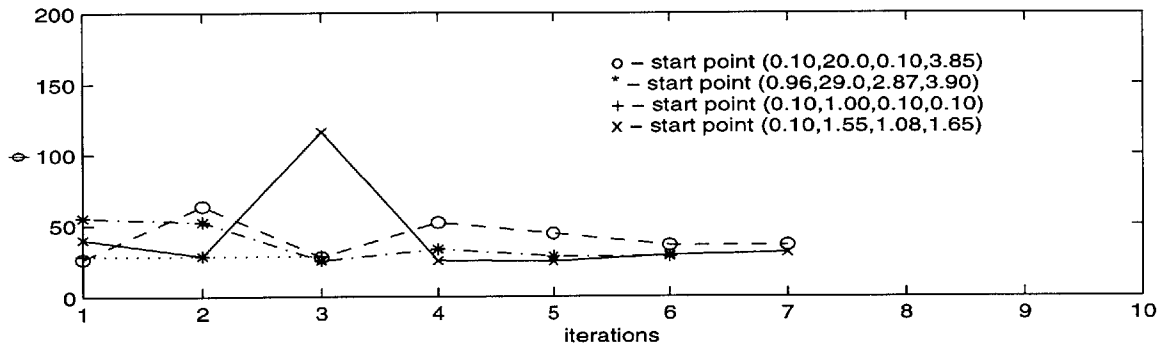
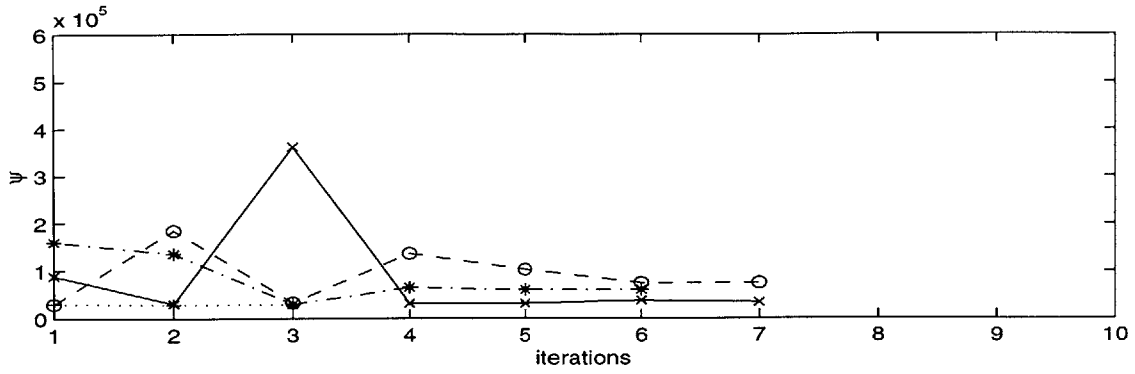


Figure 4-2: Convergence results for optimizing ψ

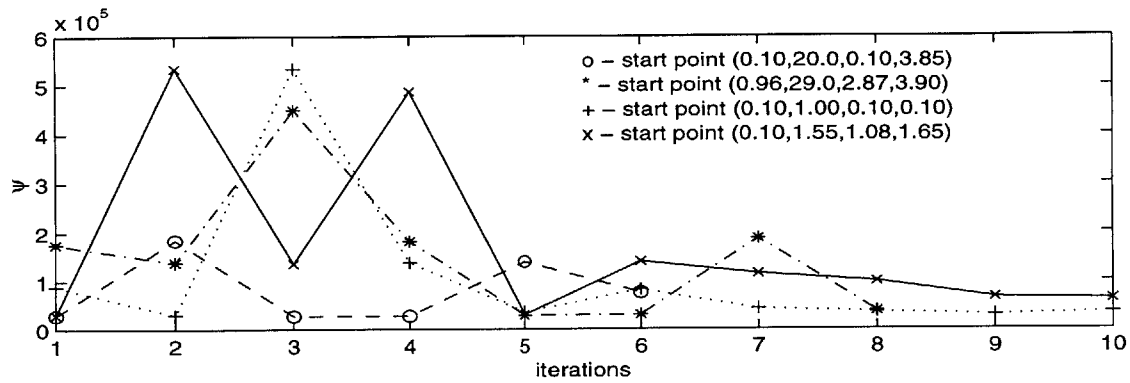
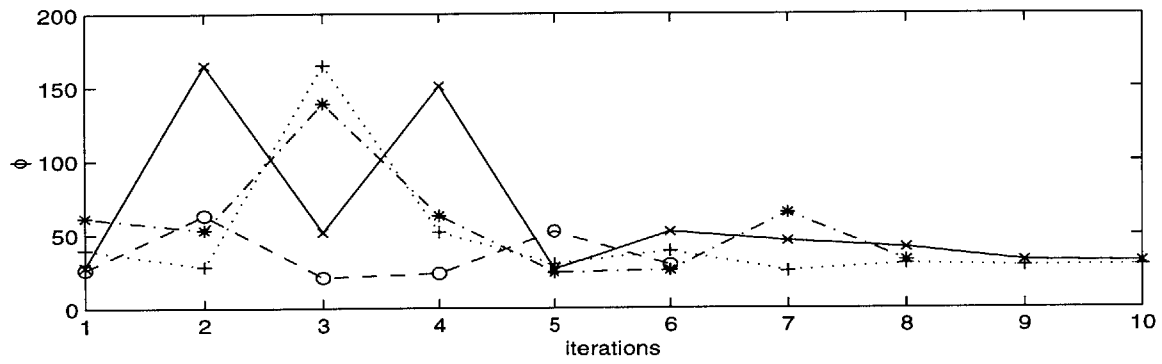


Figure 4-3: Convergence results for optimizing ϕ

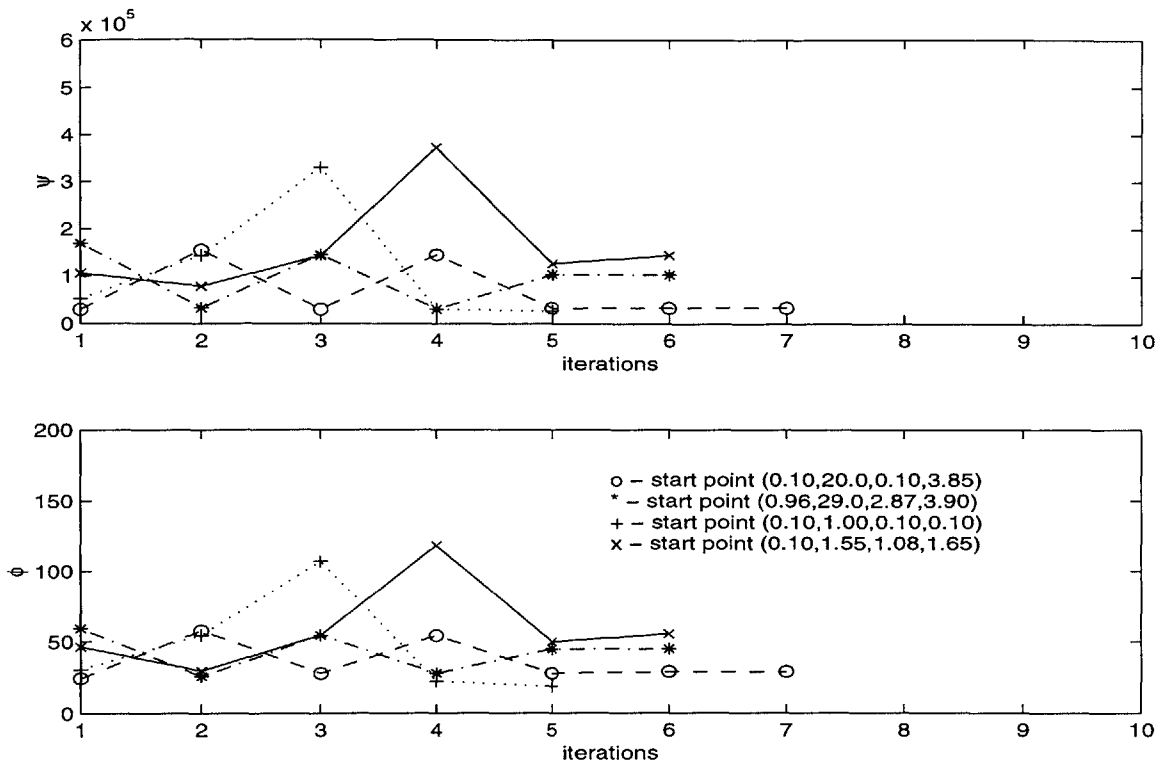


Figure 4-4: Convergence results for optimizing ψ

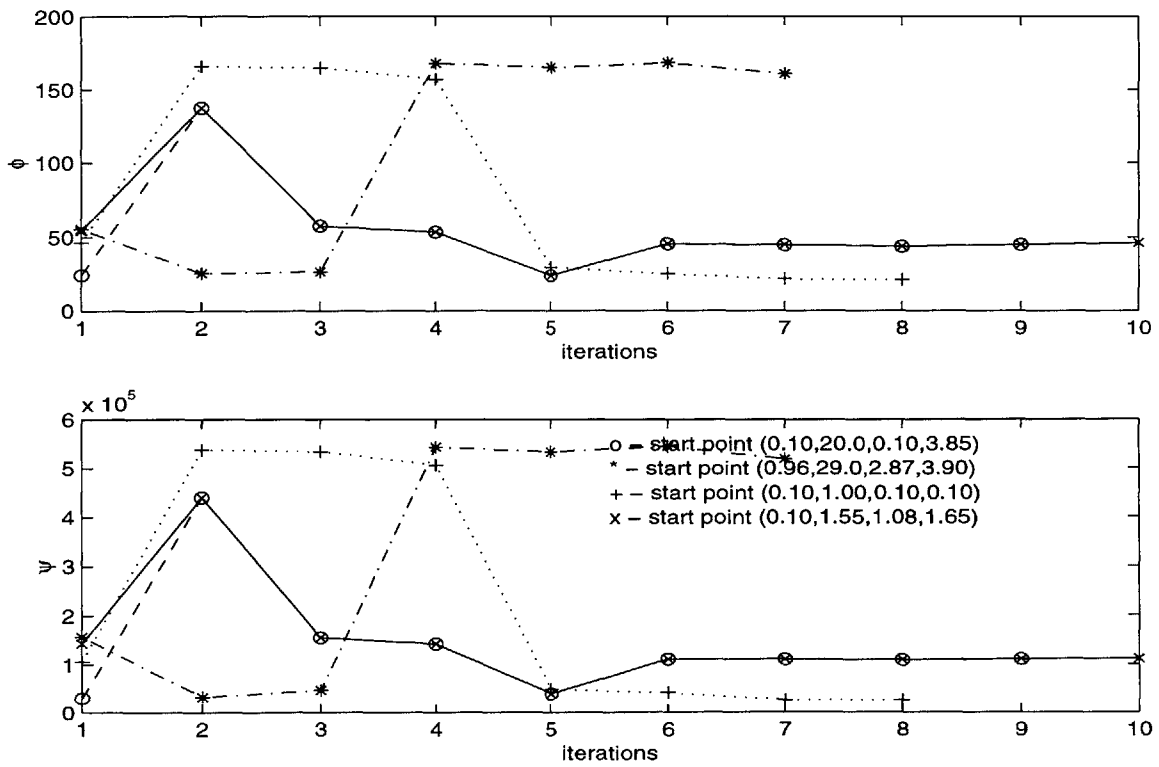


Figure 4-5: Convergence results for optimizing ϕ

Point (<i>cflower</i> , α_- , β_- , γ_-)	Function Values	
	ψ	ϕ
(0.1,1.56,2.77,3.95)	35075	29.23
(0.1,1.15,0.1,3.99)	31373	24.33
(0.1,2.11,2.42,3.8)	33167	30.62
(0.1,1,0.10.1)	30791	28.21
(0.1,1,3,4)	28297	28.19
(0.1,28.1,0.1,4)	27900	20.91

Table 4.2: Local optimum values

algorithm and the response function with a different random number seed could have a different gradient. This indicates a significant impact of stochasticity on calibration.

Moreover, there does not seem to be any significant difference between the convergence of the two forms of objective functions. This is intuitive because the sum of squares of the deviations and sum of squares of the percentage deviations are of the same degree. To obtain the function ϕ we are dividing each term of ψ by a constant. This will have an effect on scale but the degree of the function is preserved.

It was also observed that there were cases when the optimization algorithm failed to converge for around twenty iterations, and also where it started oscillating between particular values. Also note that the values converged to need not necessarily have the minimum objective function values.

From Figure 4-5 we can see that local optima exist. To explore whether local optima with comparable objective function values were present, points in the parameter space were selected with low objective function values, but different parameter values⁵ (Table 4.2) and a first order search was conducted in the neighborhood of these points. This verified that what we have is a local optima problem and not an identification problem. Local optima were also found to be present for both forms of the objective functions.

The previous discussion shows that even for a fixed random seed we have a highly irregular search space with many local optima. This was found to be true for both forms of the objective functions.

⁵for random number seed with all $x=2$

	Sum of Squared of deviation in			
	Flow	Speed(ψ)	Occupancy	Speed(ϕ)
Flow	1.0000	0.8357	0.7521	0.8479
Speed(ψ)	0.8357	1.0000	0.9571	0.9983
Occupancy	0.7521	0.9571	1.0000	0.9561
Speed(ϕ)	0.8479	0.9983	0.9561	1.0000

Table 4.3: Correlation coefficient

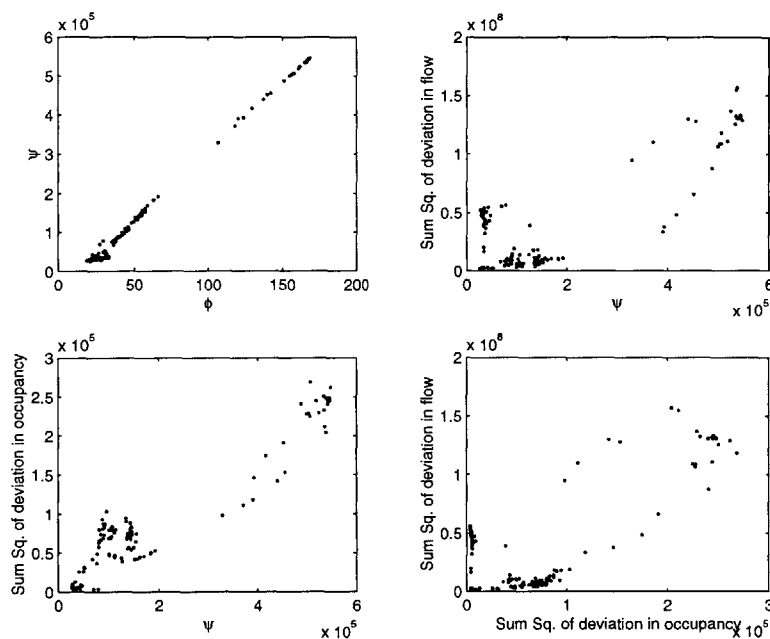


Figure 4-6: Correlation plots

Correlation

To find out whether we would be making any systematic errors by choosing one objective function form over the other, we determined the correlation of the two objective function values for different points in the search space ⁶. To understand the effect of minimizing the deviation in speed, on the deviation in occupancy and flow, we determined their correlation with the two objective function values. The result is given in Table 4.3. But because the correlation measure only linear dependencies, the plots are given in Figure 4-6.

Table 4.3 shows that the deviation in speed has a high correlation to that of flow and occupancy. This justifies our approach to minimize the deviation in speed. It

⁶as many as 150 different points and for different random seeds

	Sum of Squared of deviation in			
	Flow	Speed(ψ)	Occupancy	Speed(ϕ)
Flow	1.0000	0.2601	-0.2311	0.3333
Speed(ψ)	0.2601	1.0000	0.5648	0.4801
Occupancy	-0.2311	0.5648	1.0000	0.3381
Speed(ϕ)	0.3333	0.4801	0.3381	1.0000

Table 4.4: Correlation Coefficient

also shows that the values of the two objective functions under consideration are highly correlated. However, if we consider the values near the optimum (i.e. in the range (20-35) for ϕ), we can observe that the two different functions are not highly correlated. To substantiate this, the data was segmented based on the proximity to the optimal solution and the new correlation coefficients determined are given in Table 4.4.

This table shows that around the optimal parameter value, the two objective functions are not highly correlated and so the form of the function would influence the optimal parameter values. Hence, we need to have additional criteria to select the optimal parameter estimates. An appropriate criteria would be to see which of the optimal parameter estimates gives a low deviation for flow and occupancy. This also indicates that objective function forms weighted over the different output measures, may give better parameter estimates.

Optimal parameter values for fixed random number seeds

The calibration was done by fixing the random number seed, and was repeated for different starting parameter values. The best parameter values (which gave the lowest objective function values) were saved. The whole process was repeated for different random number seeds. Finally the parameter values obtained by minimizing the two different objective function forms were compared. However, for an effective comparison to be made, the average objective function values at these parameter values had to be determined. Table 4.5 compares the two best solutions obtained by optimizing the function ψ and Table 4.6 compares the two best solutions obtained by optimizing the function ϕ .

Parameter Values (cflowerbound, α_- , β_- , γ_-)	Fixed random number seed		Average over many seeds			
	ψ	ϕ	ψ		ϕ	
			Mean	Std. Dev	Mean	Std. Dev
(0.1,1,3,4)	28927	28.19	34376	3511.1	29.95	1.55
(0.1,10.71,0.1,4)	27075	18.66	43346	24736	27.48	6.63

Table 4.5: Parameter values obtained for optimizing ψ

Parameter Values (cflowerbound, α_- , β_- , γ_-)	Fixed random number seed		Average over many seeds			
	ψ	ϕ	ψ		ϕ	
			Mean	Std. Dev	Mean	Std. Dev
(0.1,28.1,0.1,4)	27900	20.91	28880	1333	23.87	1.62
(0.1,5.65,3,4)	30506	23.07	36264	5415.7	24.96	1.38

Table 4.6: Parameter values obtained for optimizing ϕ

By comparing the values of ψ (in the second column) in the two tables, we observe that the optimal values of ψ obtained are comparable irrespective of the form of the objective function we are optimizing. This also holds true for ϕ (compare the third column). However, note that a low value of ψ does not necessarily mean a low value of ϕ and vice versa. This is evident when we compare the first row of Table 4.5 and the second row of Table 4.6. This corroborates what we had discussed in the section on correlation (see 4.5.2), that near the optimal parameter values the correlation between ψ and ϕ is very low.

We are actually interested in the average value of ψ and ϕ over many random number seeds. The mean values and their standard deviations are given in the last four columns of the two tables. It can be noted that a low value of the objective function for a fixed seed need not guarantee a low mean value. However, as noted earlier the coefficient of variation of the function ϕ is lower than that for ψ . Hence we would expect that the quality of the solution obtained by fixing the random number seeds is better for the function ϕ . This can also be observed by comparing the solutions in Table 4.6 and 4.5.

Due to the nature of the search domain we are not able to conclusively recommend one objective function over the other. However, we are able to observe a high correlation between the two objective functions, suggesting that their bias is the same.

Parameter Values (cflowerbound, α_- , β_- , γ_-)	ψ		ϕ	
	Mean	Std. Dev	Mean	Std. Dev
(0.1,1.55,1.08,1.65)	101808	2766	44.56	1.07

Table 4.7: Objective function values for initial parameter value

Moreover, we observed that the coefficient of variation of one of them is lower. The function with the low coefficient of variation gave better parameter estimates for fixed random number seeds.

Comparison to initial parameter values

The starting parameter values and the average objective function values are given in Table 4.7. By comparing these values to the values in Table 4.5 and 4.6 we can see that there is a significant improvement in the objective function values. To graphically compare the solutions obtained above, for each parameter set, we plot the three figures as in Chapter 2. These are given from Figure 4-7 to 4-18.

Consider the scatter plots in Figures 4-7 to 4-9. Note that even though the objective function values are very close to each other the plots are very different. Moreover it can be seen that the plots for speed are very similar but the ones for occupancy and flow are substantially different. This is because we optimized only for the response-speed. Moreover, note that Figure 4-9 matches best the flow, speed, and occupancy. In addition, we can also observe from the contourplots that the one for the Parameter Set 3 (0.1,28.1,0.1,4) is able to simulate the build-up and release of congestion. This is also clear from the flow-speed-occupancy plots.

We can conclude that calibration is able to provide better parameter estimates, but based on our study we would expect that a better objective function definition—possibly using weights to combine responses—would be able to give better results.

4.5.3 Sampling Distribution of Optimal Parameter Values

The previous section discussed the perils involved in optimizing with a fixed random number seed. We could also account for stochasticity by obtaining the sample from

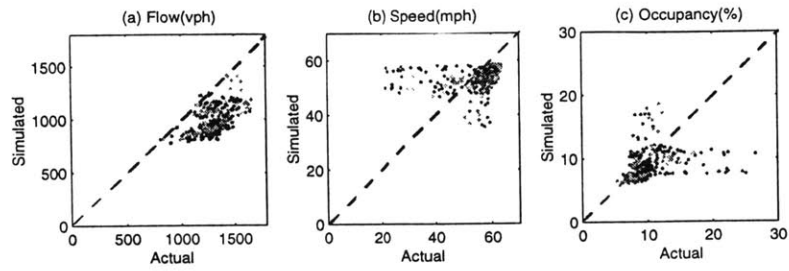


Figure 4-7: Comparison of simulated and observed data for parameter set 1

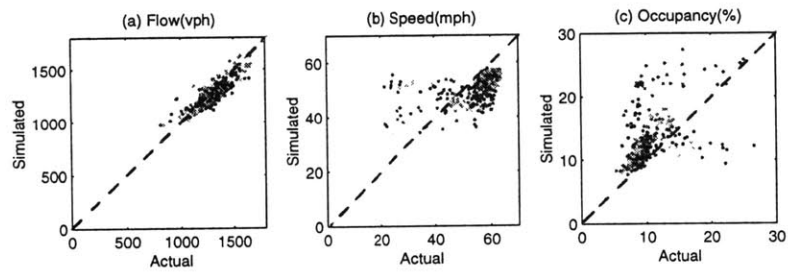


Figure 4-8: Comparison of simulated and observed data for parameter set 2

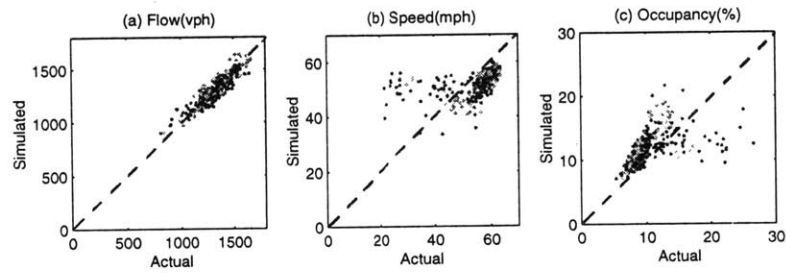


Figure 4-9: Comparison of simulated and observed data for parameter set 3

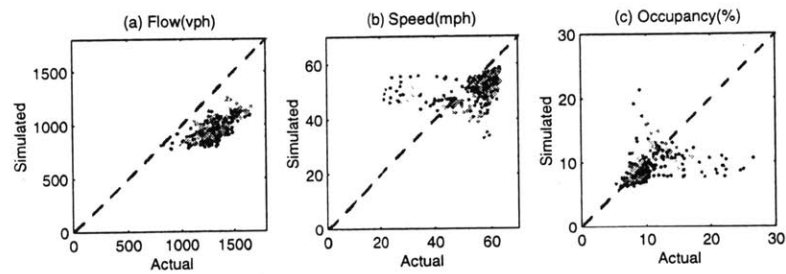


Figure 4-10: Comparison of simulated and observed data for parameter set 4

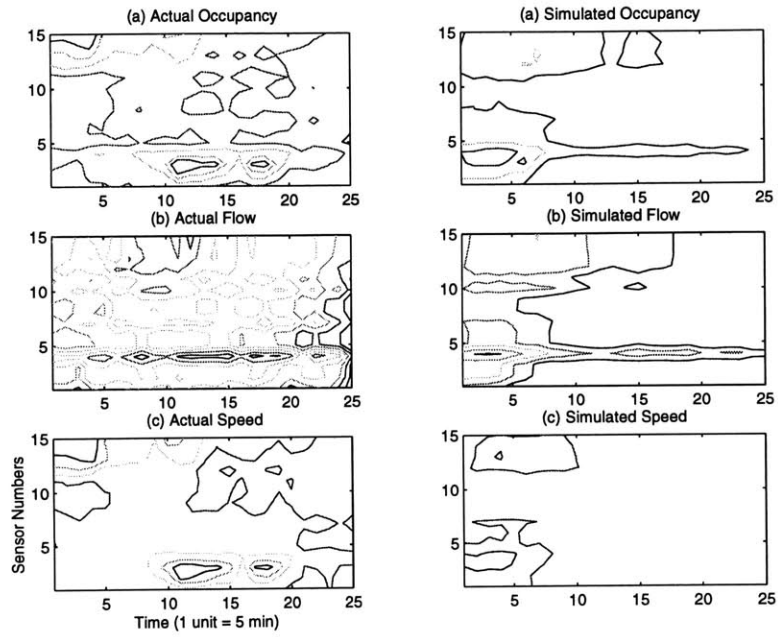


Figure 4-11: Contour plots for simulated and field data for parameter set 1

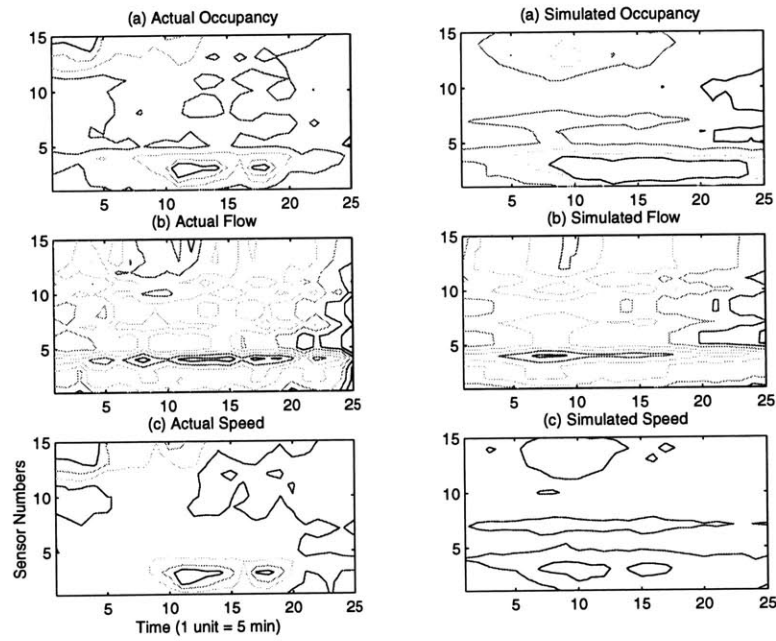


Figure 4-12: Contour plots for simulated and field data for parameter set 2

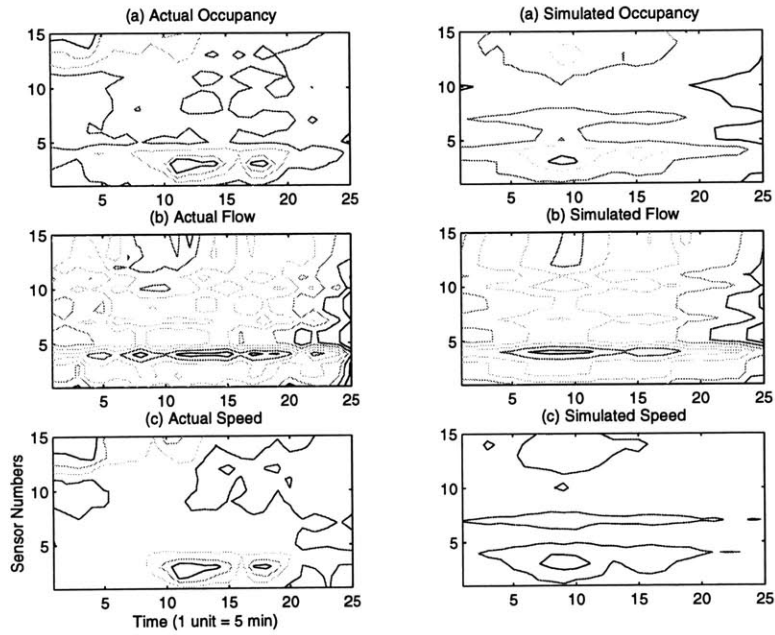


Figure 4-13: Contour plots for simulated and field data for parameter set 3

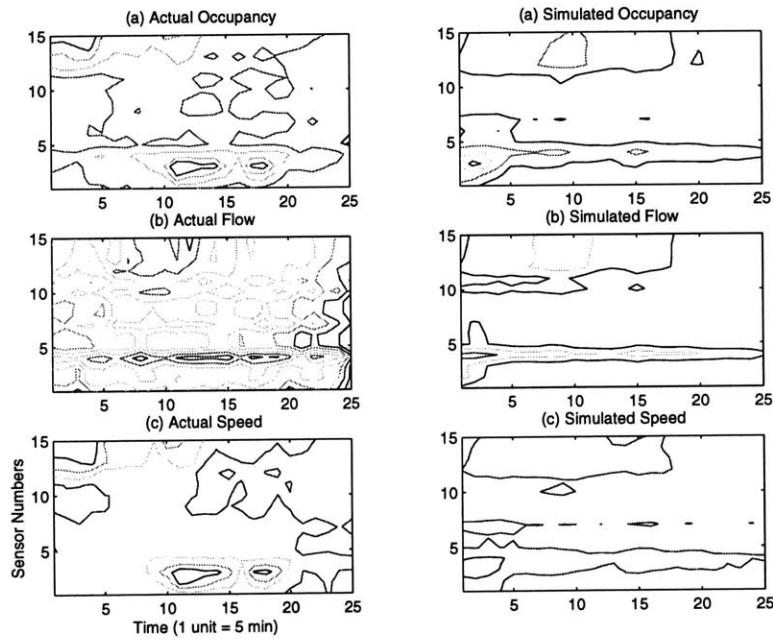


Figure 4-14: Contour plots for simulated and field data for parameter set 4

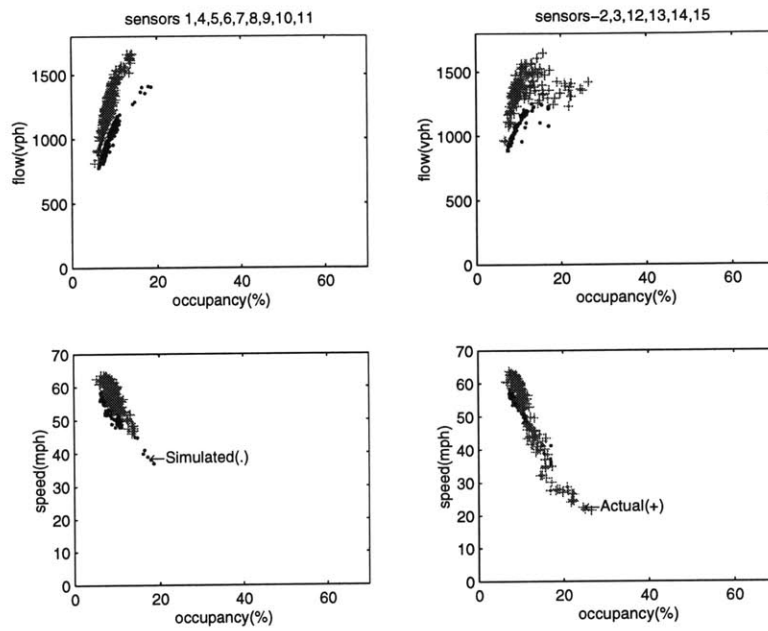


Figure 4-15: Comparison of simulated and field data for parameter set 1

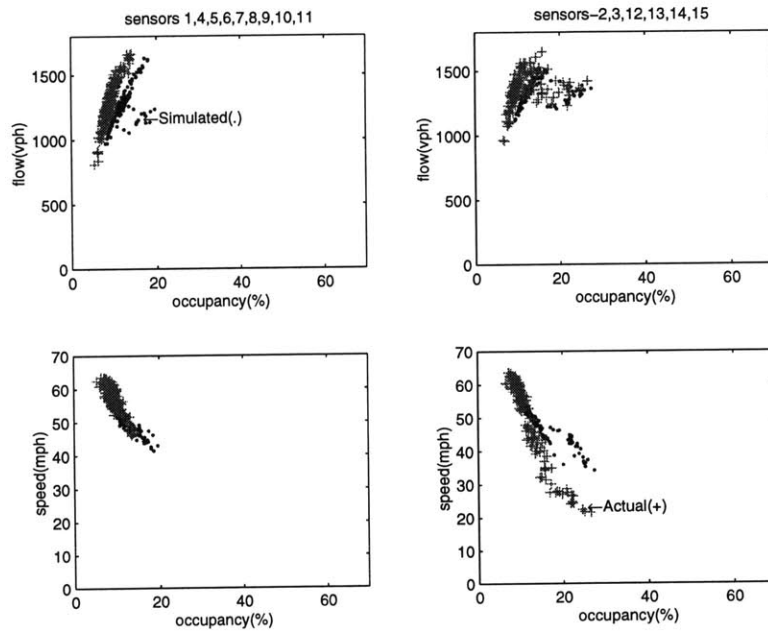


Figure 4-16: Comparison of simulated and field data for parameter set 2

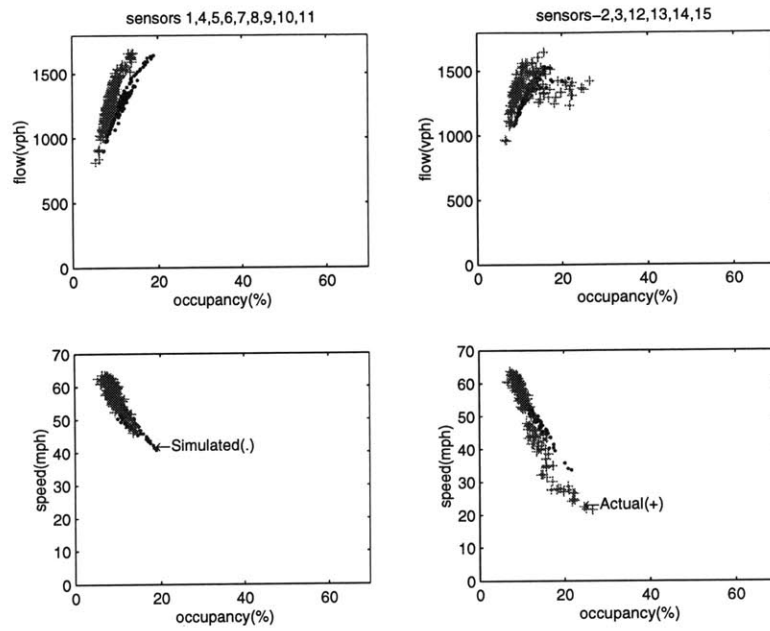


Figure 4-17: Comparison of simulated and field data for parameter set 3

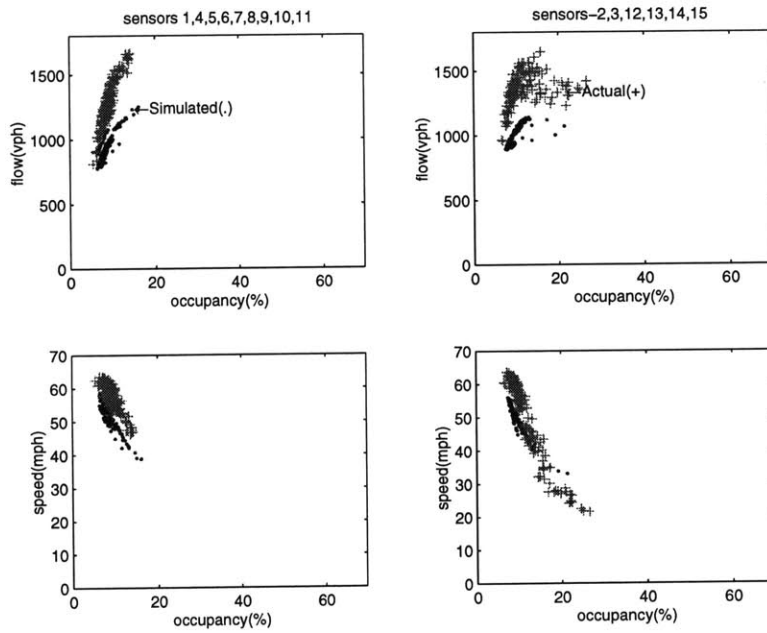


Figure 4-18: Comparison of simulated and field data for parameter set 4

Parameter	Mean	Std. Deviation
cflowerbound	0.1000	0.0000
α_-	9.1533	11.5443
β_-	1.2167	1.2496
γ_-	3.7803	0.2758

Table 4.8: Optimal parameter value by optimizing objective function form ψ

Parameter	Mean	Std. Deviation
cflowerbound	0.1078	0.0233
α_-	14.6967	13.857
β_-	1.7067	1.2881
γ_-	3.9478	0.0733

Table 4.9: Optimal parameter value by optimizing objective function form ϕ

the best parameter values; the best parameter values for each seed were saved, and the sampling distribution was determined by averaging over these saved values. The sampling distribution was determined separately for the two objective functions and the result is given in Tables 4.8 and 4.9.

The high standard deviations associated with the parameter estimates suggest that the estimates are not very reliable. This could be due to the nature of the search domain. Finally, the parameter values obtained by minimizing the two different objective function forms were compared by plotting the same set of figures as before. These are give in Figures 4-19 to 4-24.

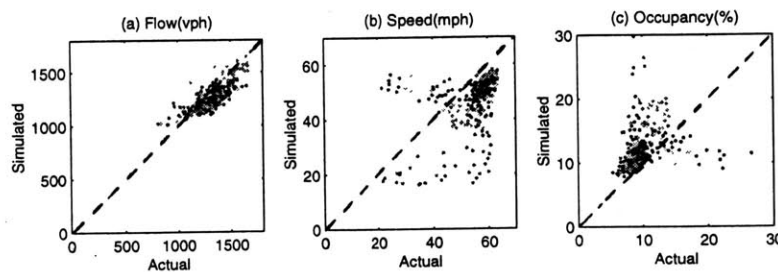


Figure 4-19: Comparison of simulated and observed data for parameter set obtained by optimizing ψ

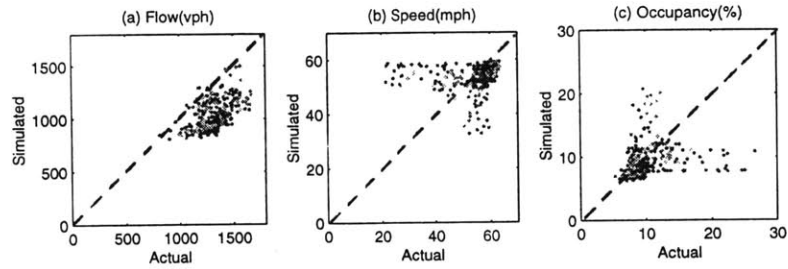


Figure 4-20: Comparison of simulated and observed data for parameter set obtained by optimizing ϕ

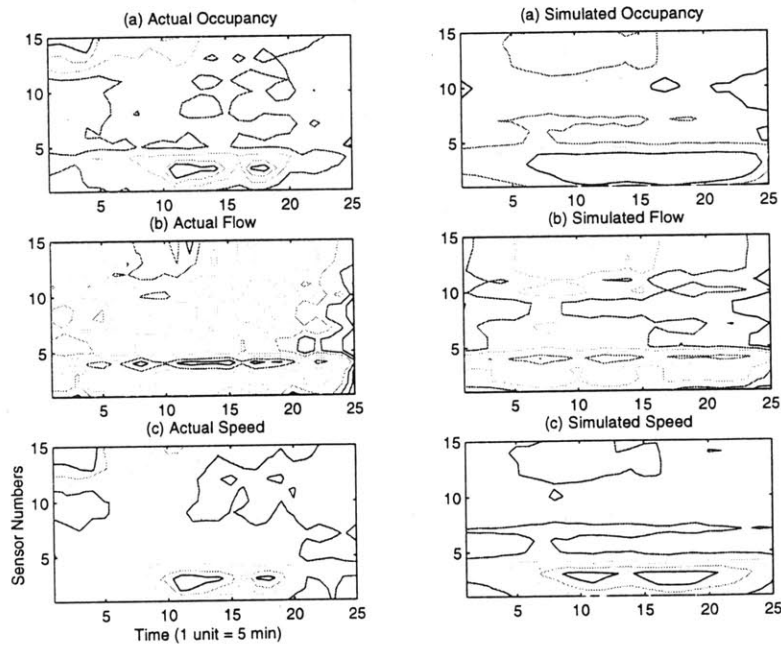


Figure 4-21: Contour plots for simulated and field data for parameter set 1 obtained by optimizing ψ

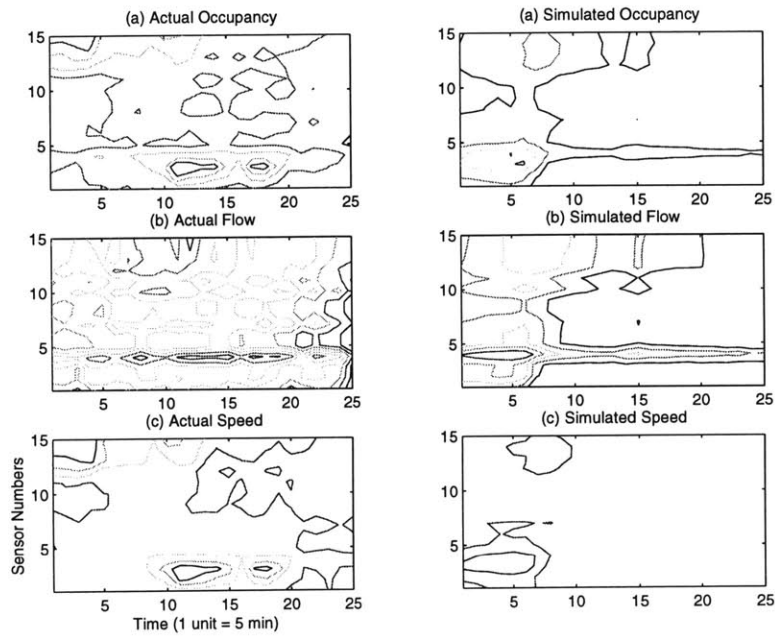


Figure 4-22: Contour plots for simulated and field data for parameter set 2 obtained by optimizing ϕ

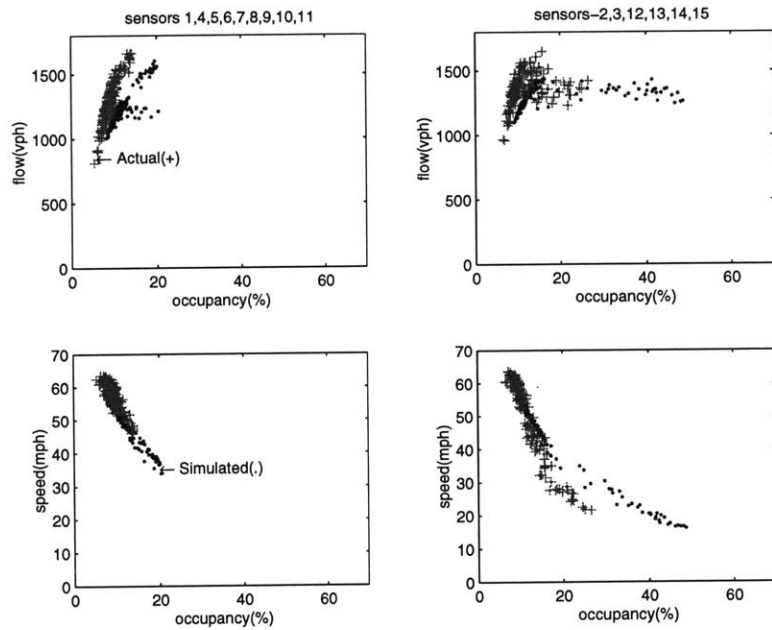


Figure 4-23: Comparison of simulated and field data for parameter set 1 obtained by optimizing ψ

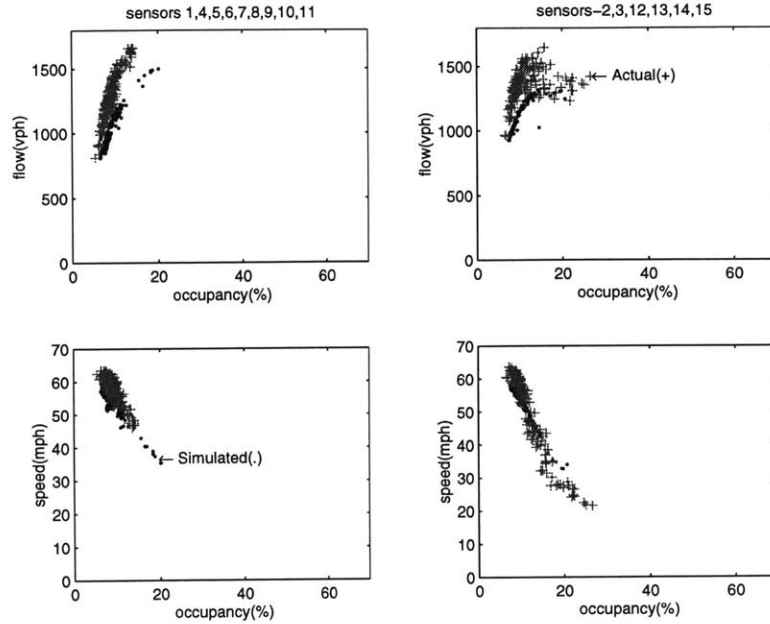


Figure 4-24: Comparison of simulated and field data for parameter set 2 obtained by optimizing ϕ

4.6 Conclusion

This chapter investigated the problems involved in calibrating a stochastic simulator. The results can be summarized as follows.

- Minimization of the deviation in speed was our objective function criteria. However by comparing the Figures 4-7 to 4-10 we can see that though the distribution of the speed values are roughly the same, the variation in flow and occupancy could be substantial. This suggests that objective function forms with weighted responses would give better results.
- The search domain was found to be highly non-linear. As a result finding a global optimum solution is likely to be intractable. The convergence and the solutions obtained were found to depend on the starting parameter values. This means that the optimization method discussed in Section 4.4 was able to find only local optimal solutions
- The parameter values obtained by ignoring the stochastic nature of the simulation could turn out to be an outlier as shown in Table 4.5 and 4.6. Hence

we need to evaluate the optimal parameter values for several random seeds and select the best from among them.

- The two objective function forms considered were highly correlated indicating that the bias is the same. The two function forms had different coefficients of variation. However it was not possible to make a final recommendation due to the nature of the search domain.
- The quality of the solution obtained by fixing the random number seed was found to be better for the objective function with lower coefficient of variation.
- Obtaining the sampling distribution of the parameters was not found to be a good strategy due to the presence of local optima.
- By comparing the objective function value at the optimal point to the starting parameter values, it was found that calibration has significantly improved the objective function values. Figure 4-25 compares the optimal values obtained with the starting value. The X-axis lists the four optimal parameter points obtained in the order listed in Tables 4.5 and 4.6. The Y-axis represents the objective function value. The straight line represents the starting value for the objective function.

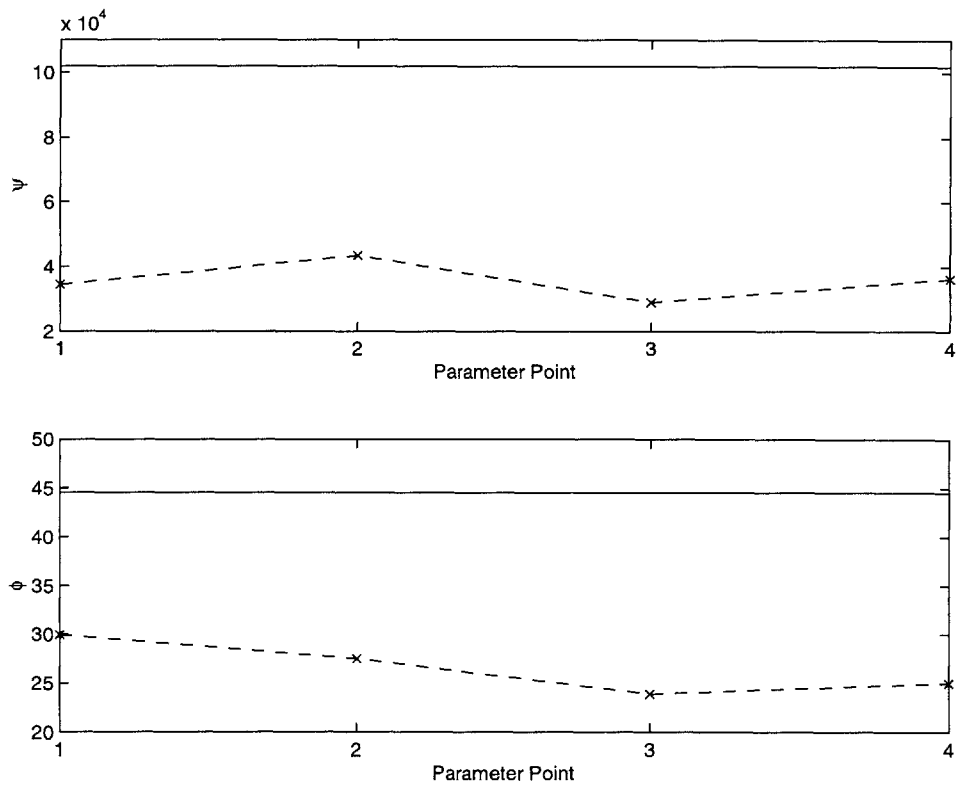


Figure 4-25: Comparison of calibrated and simulated objective function values

Chapter 5

Summary, Conclusions and Research directions

5.1 Summary and Conclusion

Based on our literature survey, a systematic study on calibrating microscopic traffic simulation models was found to be absent. In this work we developed the framework for the calibration of a stochastic microscopic traffic simulator. An optimization based framework was developed for calibration. The various issues affecting a calibration study, namely the selection of a response from among the different simulation outputs, the formulation of an appropriate objective function, the systematic identification of sensitive parameters, and the effect of stochasticity in calibrating the parameter values were discussed. All these issues were also quantified with respect to the simulator–MITSIM.

Car-following parameters were first selected for calibration and were found to be sensitive. Experimental design methodology was used to determine the set of sensitive parameters from among the car-following model parameters. This technique was able to substantially reduce the number of sensitive parameters. Car-following parameters related to the deceleration characteristics were found to be the most sensitive.

Two different objective function forms were considered for calibration. One function was found to have a lower coefficient of variation, implying that the correction

applied for heteroscedasticity was appropriate. The search space was explored and local optima were found to be present. The convergence property and the optimal parameter values were found to depend on the starting values of the parameters. This indicates that robust optimization based techniques, for example Genetic Algorithms, should be investigated for future studies.

The response speed was selected for calibration. The correlation between the deviation in speed and occupancy and between speed and flow was found to be high. However, different optimal solutions, with about the same deviation in speed but substantial difference in the deviation of flow and occupancy, were found . This indicates that other forms of objective functions—weighted over different responses could give better parameter estimates.

Stochasticity was found to have a very significant impact on the optimal parameter values. Due to the nature of the search domain, sampling distribution of the optimal parameter values could not give reliable parameter estimates. Different parameter values were obtained by optimizing the simulator for a fixed random number seed and with different starting values. The parameter values obtained by fixing the random number seed for the simulation were sometimes found to be outliers. The best parameter values were found by comparing the average value of the objective functions over many replications.

It was found that though calibration is an intricate process, the performance of the simulator could still be substantially improved by an appropriate calibration study.

5.2 Research Directions

- **A better objective function formulation:** A better objective function form can be formulated by considering some or all the points discussed below. In our study we did not consider weighted objective functions over all the responses. It was found that minimizing the deviation in speed need not necessarily improve the deviation in flow. Moreover the correlation in the responses, both in space and time, if accounted for may lead to better objective function forms.

- **Sensitive Parameters:** We used some empirical observations followed by design of experiments to come up with the sensitive set of parameters. Currently there are several other tools available for design of experiments study, that can accommodate more variables. If more sensitive variables can be determined then the calibration process could be that much more efficient.
- **A more robust search process:** We have determined that we have a local optima problem. In such cases it may be computationally more efficient to use heuristic optimization algorithms like genetic algorithms for calibration study. This can be investigated.
- **Stochasticity:** This study has shown that stochasticity does affect the optimal parameter estimates. Moreover the approach we took in optimizing was to fix the random number seed, optimize, and then determine the sampling distribution of the parameter values. A conventional approach is to replicate the simulation many times at each point in the parameter space. An interesting study would be to compare the results of these two approaches to calibration.
- **Response function methodology:** We used the design of experiments study to determine the set of sensitive parameters. The response functions obtained from this study can be optimized for the parameters. These parameter values can be used to determine the range of parameter values for a subsequent design of experiments study. Again the response function obtained can be optimized. This iteration between the design of experiments and optimization can be continued until convergence is obtained. The parameter values obtained by such a study can be compared to the values we have determined using gradient based optimization methods.

Appendix A

Boss/Quattro

A.1 Introduction

Boss/Quattro is a software environment that manages external applications. Using Boss/Quattro it is easy to analyze and optimize the influence of parameters on the responses yielded by external software applications. Boss/Quattro uses a *model* to get information about the parameters affecting an application output. Using a driver system, Boss/Quattro changes the values of these parameters and reads the results from the application run. New results can be defined, by combining the parameters, and the results directly read from the application run, using mathematical operators and functions. Boss/Quattro also provides built-in '*engines*' for Parametric Study, Monte-Carlo Study, Design of Experiments and Optimization. These engines can be mixed, allowing wider scope studies, and dependencies can be defined to match logical relationships between problem components. In addition a user can incorporate new optimization or analysis algorithms by writing specialized engines.

A.2 Structure

Applications are programs external to Boss/Quattro and a typical application run involves reading an input file, running an analysis, and finally processing the output from a set of result files. Applications that follow this input-analysis-output

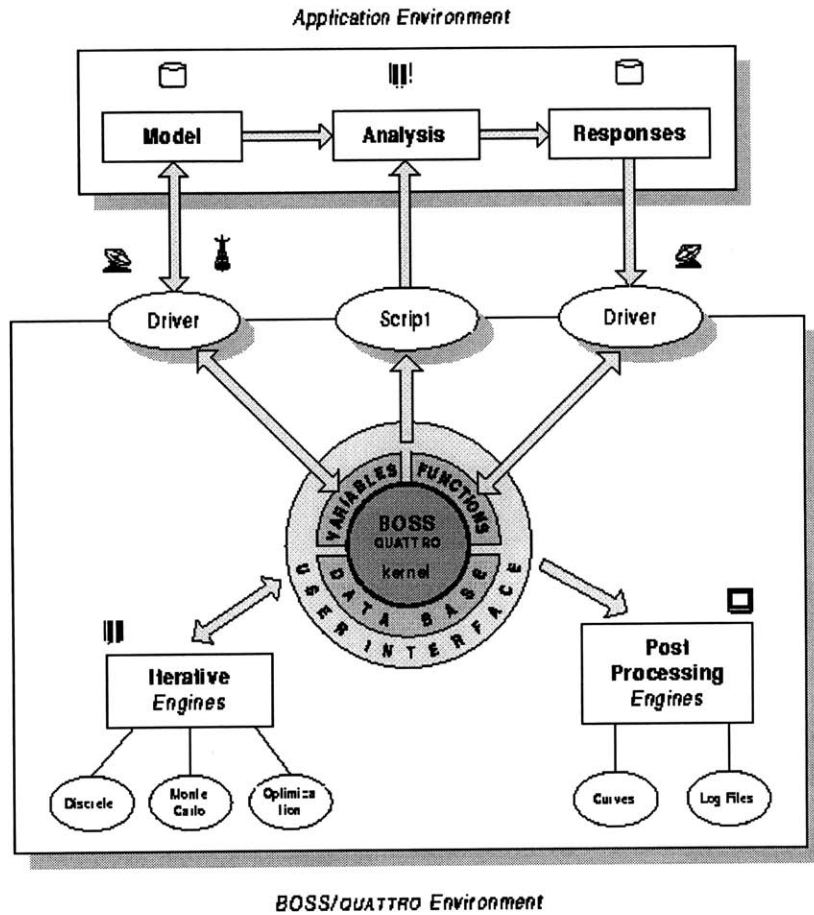


Figure A-1: Boss/Quattro's interactions

paradigm can be managed by Boss/Quattro. The interaction of Boss/Quattro with the application environment is represented in Figure A-1.

Quantities in the input can be symbolically represented using parameters. An input file is said to be parameterized when it is defined using these parameters instead of the values they represent. A parameterized input file is known as a *model*. The result of the application run, usually a set of output files, is called the result or the *analysis response*, and the application run itself is the *analysis*. The model, the analysis, and the response together form the application environment.

As shown in Figure A-1, Boss/Quattro controls the application environment using *drivers* and *scripts*. Drivers achieve the data exchange between Boss/Quattro and the application environment. Model and response files vary with the application

and a separate driver needs to be written for each file format. However there is a class of drivers called the *neutral drivers* which can be made independent of the external application input/output formats. If the model and response files can be modified to comply with the neutral format then neutral drivers can be used.

Boss/Quattro scripts contain commands that are specific to a local operating system (Bourne, C, or Korn shells for UNIX systems) and are used for running applications or other external programs. A model file (mandatory) and a result file are associated with a script file.

Boss/Quattro *engines* modify the parameter values in model files, run a sequence of analyses and collect the respective responses. The different engines differ in the way they generate or modify the parameter values. Some relevant engines in Boss/Quattro are

Parametric: Parametric engine helps to document the effect of variation in parameter values on the application output. Parameters come from the model files and can be chosen by the user. For each parameter the number of values can be specified and also the manner in which different parameter values are to be varied, whether serially or in parallel, can also be specified. Responses are read from the output files and new responses can also be defined.

Experiments: The experiments engine lets the user do a design of experiments study. The type of design can be chosen and aliases can be defined for fractional designs. The functions that need to be monitored can also be specified. Moreover response functions can be defined and a selection can be made from first order, mixed, second order or user defined functional forms. After the experiment run the response function model can be screened for less significant variables or a stepwise regression can be run to select the most significant ones.

Optimization: The optimization engine lets us optimize a response in a space determined by the set of parameters. Constraints can be also be incorporated. Selection can be made from among different optimization algorithms and convergence criteria can also be specified. Finally sensitivity computation can also be made at a specified point in the domain space.

Updating The updating engine allows to fit a function generated by a parameterized model to another target function. It uses an optimization algorithm to minimize the sum of squared differences between the computed and target function values. This can be used to minimize the deviation between a vector of computed application output values and some specified values.

A.3 Using Boss/Quattro with MITSIM

A.3.1 Environment Setup

As we had discussed previously, any application that fits in with the input-analysis-output paradigm can be managed using Boss/Quattro. MITSIM reads a set of input files, performs the simulation, and writes the sensor readings to a file. However, before undertaking any study, the application environment of MITSIM needs to be integrated with Boss/Quattro. The interaction of Boss/Quattro with the MITSIM application environment is represented in Figure A-2

As shown in the figure MITSIM takes as input three files namely, `paralib.dat`, `network.dat` and `master.mitsim`. We decided to use the neutral driver for exchanging data between Boss/Quattro and the MITSIM application environment. The input files were parameterized in the format required by the neutral driver. The parameterized input files (model files) were named `paralib.in`, `network.in` and `master.in` respectively. Since MITSIM could not directly read the model files, they had to be converted to their native form. Boss/Quattro provides scripts, called `Netural_CPP`, for converting the neutral format model files to their original form. However, these scripts had to be explicitly invoked by the user. Hence, we wrapped these in scripts files named `USERparalib`, `USERnetwork` and `USERmitsim` for the model files `paralib.in`, `network.in` and `master.in` respectively.

To successfully use Boss/Quattro we also had to facilitate data exchange between the application output and the neutral driver. The output file that we are interested in is `sensor.out`, which contains the simulated sensor readings. We could have either

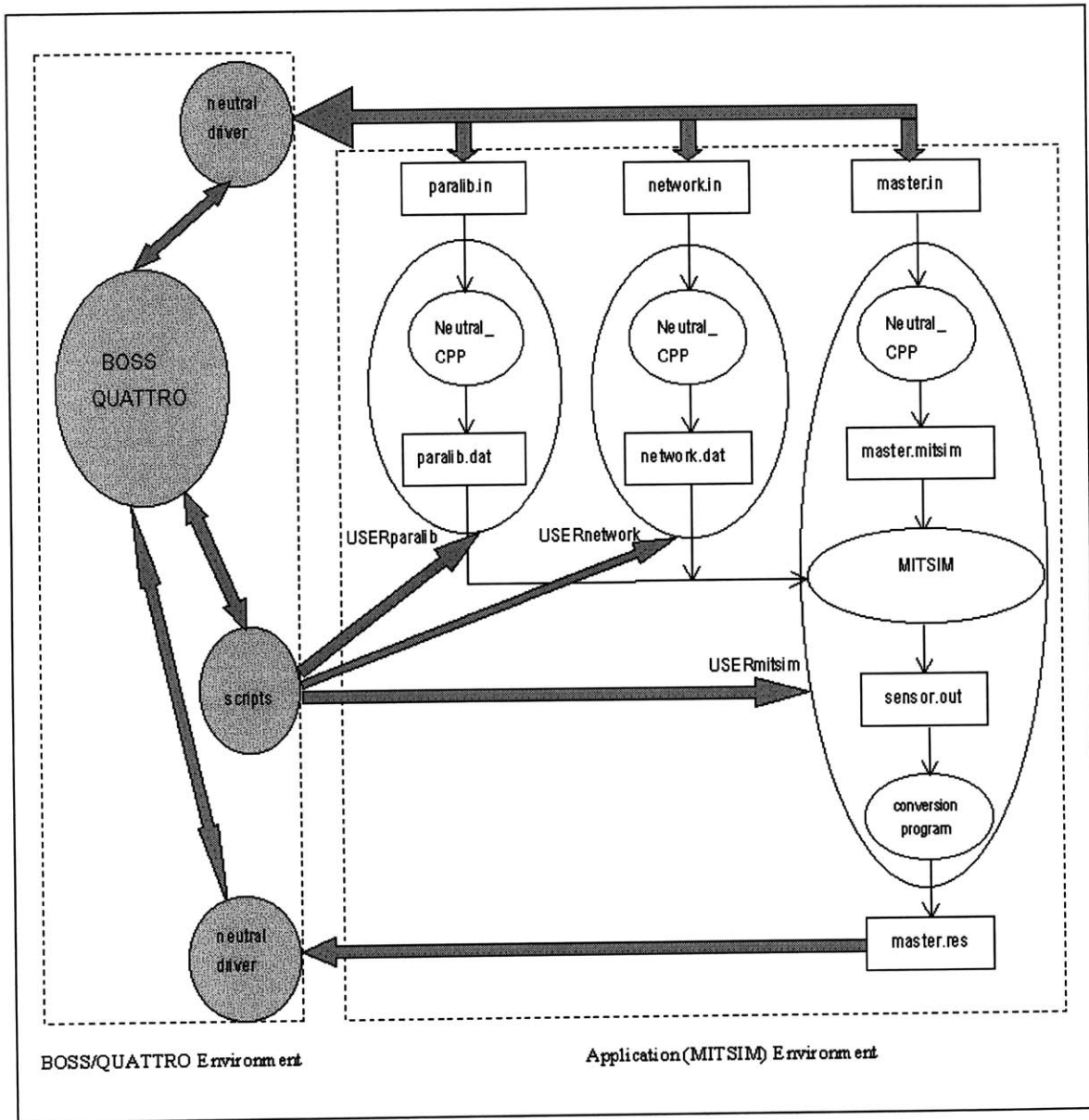


Figure A-2: Boss/Quattro's interactions with MITSIM

modified MITSIM to write the output in the format specified for the neutral driver, or we could have written a conversion program for converting the MITSIM output to the required format. We choose the second option, and wrote a simple C program. This executable program also had to be explicitly invoked by the user. The script file `USERmitsim` was hence modified to include the functionality of executing the application (running MITSIM) and then calling the conversion program.

The separate script files written and their respective tasks are explained below.

1. `USERparalib`: It takes the model file `paralib.in`, uses the `Neutral_CPP` script and produces the file `paralib.dat`. The file is then copied to the input directory of MITSIM.
2. `USERnetwork`: It takes the model file `network.in`, uses the `Neutral_CPP` script and produces the file `network.dat`. The file is also copied to the input directory of MITSIM.
3. `USERmitsim`: This takes the file `master.in` as the model file. Generates the file `master.mitsim` using the `Nuetral_CPP` script, and copies the file to the input directory of MITSIM. It then launches MITSIM, in the batch mode with the necessary command line parameters, and waits for the simulation run to terminate. Once the MITSIM run is complete it launches the conversion program to convert the file `sensor.out` to the format specified for the neutral driver. Finally the neutral format file is copied to the file `master.res` (the response file). The neutral driver reads the results from the file `master.res`.

Once MITSIM has been fully integrated with Boss/Quattro the various studies could be undertaken. The next two sections describe the sensitivity and optimization study performed on MITSIM.

A.3.2 Studies

To calibrate MITSIM, first a set of sensitive parameters needs to be identified (sensitivity study). To do this, in chapter 3 we proposed the use of design of experiments

methodology. As discussed in section A.2 design of experiments can be performed using the Experiments engine. Once a set of sensitive parameters has been determined an optimization study can be performed to do the calibration. The next two sections describe these two studies in detail.

A.3.3 Sensitivity Study

A sensitivity study using design of experiments involves the following

- Determining the design points
- Specifying the responses to monitor
- Running the simulator and calculating the response values at each point.
- Fitting a specified response function model to the observed response values.
- Screening for less sensitive variables.

Moreover, since we were dealing with a stochastic simulator we had to replicate the simulation run at each design point. The response value at each point was averaged over these runs.

The sensitivity study performed is shown in Figure A-3. Each rectangular item represents a task and the tasks involved in the study appear at the right pane of the window between the *Start* and *End* tasks. The Experiments task represents the experiments engine in Boss/Quattro and it manages the execution of all the tasks defined under it. The design type and aliases for fractional studies can be specified in this task. The design points are automatically generated once the above specifications have been made.

The responses that need to be monitored can also be defined in the experiments task. Since our objective was to fit the simulated data to observed data, the response function forms involve both the observed and simulated data. However, after each MITSIM run, we obtain only the simulated sensor data. Hence, to make the observed field data available for result definitions, three separate dummy scripts (one each for

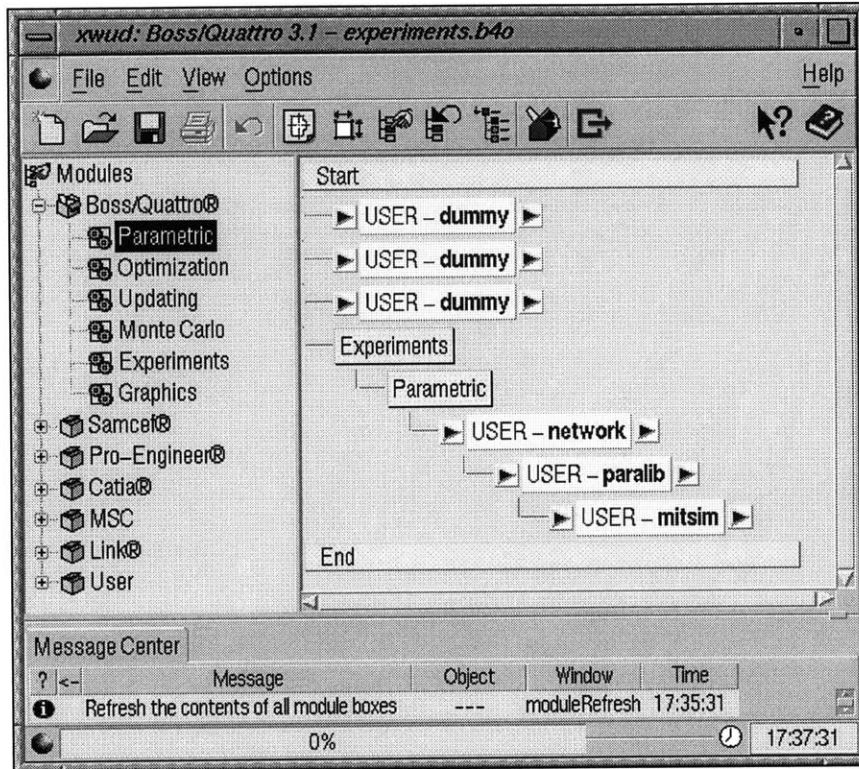


Figure A-3: An example of a design of experiments run

flow, speed and occupancy) were defined. These scripts have no executable code and are associated with empty model files. However, each one of them is associated with a result file. The result files contain the observed field data and were in the format required by the neutral driver.

The three tasks USERparalib, USERnetwork and USERmitsim executed one after the other constitute a single run of the simulator. Note that the task USERmitsim involves running the simulator. Hence, this task should be run only after the other two tasks are complete. This dependency in runs was accomplished by defining the three tasks one after the other as shown in Figure A-3.

Finally, in the experiments task the response function model can be specified and a selection can be made between screening for sensitive variables or running a stepwise regression.

Since MITSIM is a stochastic simulator, as discussed earlier we had to replicate the simulation run at each design point. This was accomplished by placing the Para-

metric task below the Experiments task. The Parametric task represents the parametric *engine*. A dummy variable was defined and the parametric engine was run for various values of the dummy variable. The parametric engine provides the functionality to average out observations over several runs. The response functions, defined in the experiments engine, used these averaged values. Thus the design of experiments with replication could be easily accomplished using Boss/Quattro.

A.3.4 Optimization study

The optimization study was conducted on the set of variables identified by the sensitivity study. The optimization study involved the following tasks.

- Defining the objective function
- Defining the variables and their ranges
- Choosing the optimization algorithm
- Identifying convergence criteria.

As for the sensitivity study a set of tasks are again defined for the optimization study. The optimization engine replaces the experiments engine as the top level task. The objective function definition and the range of the variables are specified in the optimization engine. The search algorithm type is chosen and convergence criteria are also specified. Thus the only set of tasks coming under the optimization engine will be the three tasks required for the MITSIM run. Note that we had performed the optimization after fixing the random number seed. Hence, we did not make use of the Parametric engine to simulate the replications necessary at each design point.

A.4 Conclusion

Boss/Quattro is a very versatile tool and by integrating Boss/Quattro with MITSIM many tasks that would otherwise take enormous amount of bookkeeping and

code writing can be easily accomplished. For a calibration study, the number of variables, the number of replications, the type of experimental design, the form of objective function and types of optimization algorithms etc. can be easily varied. Thus, Boss/Quattro integrated with MITSIM is a very powerful tool for any calibration study.

Appendix B

MITSIM-Parameters

The parameters have been classified into the following

- **Traffic Flow Parameters**

1. DefaultSpeedLimit
2. FreeSpeed
3. LaneSpeedRatio
4. JamDensity
5. MinimumSpeed

- **Simulation Parameters**

1. StepSize
2. UpdateStepSize—related to driver reaction time
3. Constant State Time
4. Loading Model
 - headway
 - coefficient for queue
 - speed at capacity.

- **Sensor Devices Characteristics**

1. DetectedMinimumSpeed

- **Travel Demand**

1. Vehicle classes and fleet mix (Table)
2. Distribution of driver groups (Table)
3. Distribution factor (default of zero-poisson interarrival times)

- **Vehicle Characteristics**

1. Maximum acceleration rate (Table)
2. Maximum deceleration rate (Table)
3. Normal deceleration rate (Table)
4. Limiting speed by class and grade (Table)

- **Vehicle Routing**

1. Path alpha
2. Route Choice fraction
3. Beta
4. Commonality factor
5. Freeway Bias
6. Diversion Penalty
7. Valid Path factor

- **Vehicle Movements (Driver Behaviour)**

- Car following
 1. Minimum response distance
 2. CF Lower bound
 3. CF parameters ($\alpha_+, \alpha_-, \beta_+, \beta_-, \gamma_+, \gamma_-$)
- Merging Model
 1. Length of Upstream area.
 2. length of Downstream area.
 3. Capacity of merging region in number of vehicles.
 4. Probability of aggressive merge from ramp.
- Event Responding
 - * Traffic Signals and signs
 1. Compliance rates
 2. Visibility Scaler
 - * Yielding
 1. Mandatory lane changing yielding model
 2. Lane changing yielding model.
- Lane Changing
- LC Entry lane check rightness probability
- Mandatory Lane Change (MLC)
 1. Distance Lower bound (x_0)
 2. delta (α_0)
 3. coef. for number of lanes (α_1)
 4. coef. for congestion level (α_2)
 5. Minimum time in lane
- Discretionary Lane Change (DLC)
 1. Step size for updating lane speed
 2. Prob of DLC given not DLC in last step
 3. Prob of DLC given DLC in last step
 4. Impatient lowerbound
 5. Impatient upperbound
 6. Slower lead threshold
 7. Speed threshold
 8. Acceleration threshold
 9. Speed ahead looking distance
 10. Minimum time in lane (same direction)
 11. Minimum time in lane (different direction)
 12. Maximum speed difference between neighbouring lanes
- Gap Acceptance
 - * Discretionary
 - Lead gap

- 1. Scale
 - 2. Gamma
 - 3. Alpha
 - 4. Beta1
 - 5. Beta2
 - Lag gap
 - same as in Lead gap
- * Mandatory
 - Lead gap
 - Lag gap
- Nosing and Yielding model
 - 1. Time horizon
 - 2. Constant (β_1)
 - 3. Coefficient for number of lanes.
 - 4. Coefficient for time in minutes since tagged
 - 5. Maximum probability for on ramp lane drop, incident.
 - 6. Maximum probability in connection to next link.
 - 7. Maximum yielding time.
 - 8. Maximum stuck time.
 - 9. Maximum distance for nosing.
 - 10. Minimum distance for nosing.
- Start Up Delays
 - 1. Maximum delay
 - 2. Position in queue
- Others
 - 1. Familiarity to the network
 - 2. Visibility Scaler
 - 3. Compliance Rates
 - 4. Headway variance

The Table B.1 gives a brief description of some of the parameter listed above and also provides the associated input file.

From this the total number of parameters identified are more than 200 and it would be computationally too expensive to do a full optimization over the whole set of parameters. In order to do a feasible computational study we need to select the subset of parameters from the above list.

Parameter	File	Description
DefaultSpeedLimit	network.dat	segment specific speed limit
FreeSpeed	network.dat	segment specific free speed limit
LaneSpeedRatio	paralib.dat	The variation of speed across lanes as a ratio of design speed.
JamDensity	paralib.dat	
MinimumSpeed	paralib.dat	
StepSize	master.mitsim	step size of simulation
UpdateStepSize	paralib.dat	specific for accelerating, decelerating and stopped vehicle
Constant state time	paralib.dat	
Loading model	paralib.dat	
Detected MinSpeed	paralib.dat	for calculating harmonic mean speed by sensor to avoid division by zero
Vehicle classes and fleet mix	paralib.dat	vehicle mix/ HOV probablilty (Table)
Distribution of driver groups	paralib.dat	
Distribution factor	paralib.dat	for the interarrival time Two classes of drivers with constant interarrival time or poisson interarrival time
Path alpha	paralib.dat	for combining historical and new travel times
RouteChoicefraction	paralib.dat	percentage of guided/unguided vehicles
Beta	paralib.dat	parameter in logit model
Commonality factor	paralib.dat	logit model
Freeway bias	paralib.dat	logit model bias (In route choice model)
Diversion penalty	paralib.dat	route switching penalty(In route switching model)
Valid path factor	paralib.dat	compares to shortest path (In route switching model)
Start up delays	paralib.dat	Based on poisiton in queue
Familiarity to network	paralib.dat	A constant based on the users familiarity to the network.
Visibility scaler	paralib.dat	A constant for the visibility
Compliance rates	paraib.dat	For various types of traffic controls
Headway variance	paralib.dat	two constants upper and lower bounds

Table B.1: Description of some of the parameters in MITSIM

Appendix C

Calculation of the Number of Replications

Since MITSIM is a stochastic simulator, the simulation output varies from one run to another. Each output represents a sample and a number of replications are required to get output values with a desired accuracy level.

Let y_r^s be an output from the r -th run of the simulator. Therefore, y_r^s is a realization of the random variable y^s . An unbiased estimator of y^s is the mean of the N observations of y_r^s from N different simulation runs. A 'different' simulation run is obtained by using a different random number seed. Mathematically,

$$\hat{y}^s = \frac{1}{N} \sum_{r=1}^N y_r^s$$

where,

\hat{y}^s is an estimate of y^s ,

N is the number of replications.

If we assume that, the N different realizations of the random variable y^s are distributed *iid* normal with an unknown variance. Then the number of replications required to obtain a certain accuracy at a certain level of significance is given by:

$$N_{reqd} = \left(\frac{st_{\alpha/2}}{\hat{y}^s e} \right)^2 \tag{C.1}$$

where,

s is an estimate of the standard deviation of y^s ,

e is the allowable error as a fraction,

α is the desired level of significance,

$t_{\alpha/2}$ is the critical value of the t-distribution at a level of significance α .

The output from the simulation includes speeds, flows and occupancies that have spatial as well as temporal dimensions. For each of these output for each time-space point, the required number of replications needs to be calculated. Then, the desired number of replications would be the most conservative value — in other words, the maximum of the number of replications required by each of the output elements.

Note that the number thus obtained could vary from one point to another in the parameter space. Hence, the required number of runs was determined at several points in the parameter space and the maximum over all the points was taken as the required number of replications.

For section 2.3 at that particular point in the parameter space with $\alpha = 0.1$ the fractional error after eight replications was found to be less than 0.2

For section 3.2.4 we determined the required number of runs as five using the equation C.1 with $\alpha = 0.1$ and $e = 0.4$. A higher value of fractional error was used due to the expensive computational requirements.

Bibliography

- [1] Kazi Iftekhar Ahmed. *Modeling Drivers' Acceleration and Lane Changing Behavior*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [2] Christos Alexopoulos and Andrew F. Seila. Advanced methods for simulation output analysis. In *Proc. 1998 Winter Simulation Conference*, pages 113–120, 1998.
- [3] Sigrun Andradottir. A review of simulation optimization techniques. In *Proc. 1998 Winter Simulation Conference*, pages 151–158, 1998.
- [4] Moshe E. Ben-Akiva and Lerman Steven R. *Discrete Choice Analysis: Theory and Application to Travel Demand*. The MIT Press, Cambridge Massachusetts, .
- [5] Yolanda Carson and Anu Maria. Simulation optimization: Methods and applications. In *Proc. 1997 Winter Simulation Conference*, pages 118–126, 1997.
- [6] Martha A. Centeno and Reyes M. Florencia. So you have your model:what to do next a tutorial on simulation output analysis. In *Proc. 1998 Winter Simulation Conference*, pages 23–29, 1998.
- [7] Ruey L. Cheu, X. Jin, K. C. Ng, Y. L. Ng, and D. Srinivasan. Calibration of FRESIM for singapore expressway using genetic algorithm. *Journal of Transportation Engineering*, 124(6):526–535, 1998/11.

- [8] Ruey L. Cheu, Wilfred W. Recker, and Stephen G. Ritchie. Calibration of INTRAS for simulation of 30-sec loop detector output. *Transportation Research Record*, 1457:208–215, 1994.
- [9] Gordon M. Clark. Use of common random numbers in comparing alternatives. In *Proc. 1990 Winter Simulation Conference*, pages 367–371, 1990.
- [10] D. C. Gazis, R. Herman, and R. Potts. Car-following theory of steady-state traffic flow. *Operations Research*, 7:499–595, 1959.
- [11] D.C. Gazis, R. Herman, and R. W. Rothery. Nonlinear follow-the-leader model of traffic flow. *Operations Research*, 9:545–567, 1961.
- [12] David Goldsman, James J. Swain, and David Withers. Single-replication simulation. In *Proc. 1990 Winter Simulation Conference*, pages 387–391, 1990.
- [13] M. J. Goodspeed. Problems in programmed optimization of a hydrological catchment model. In Institution of Engineers, editor, *Proc. 1975 Hydrology Symposium*, pages 127–130, Australia, 1975.
- [14] Sarah J. Hood and Peter D. Welch. The application of experimental design to the analysis of semiconductor manufacturing lines. In *Proc. 1990 Winter Simulation Conference*, pages 303–309, 1990.
- [15] Ashok Kalidas. Dynamic trip table estimation for real time traffic management systems. Master of science in transportation, Massachusetts Institute of Technology, Cambridge, MA, May 1992.
- [16] David W. Kelton. Statistical analysis of simulation output. In *Proc. 1995 Winter Simulation Conference*, pages 23–30, 1995.
- [17] Averill M. Law and David E Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Inc, 1982.
- [18] A.D. May, Jr. and H.E.M. Keller. Non-integer car following models. *Highway Research Board*, 199:19–32, 1967.

- [19] Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, 1984.
- [20] Robert E. Pindyck and Rubinfeld Daniel L. *Econometric models and Economic Forecasts*. McGraw-Hill Book Company, 1976.
- [21] Hariharan Subramanian. Estimation of car-following models. Master of science in transportation, Massachusetts Institute of Technology, Cambridge, MA, September 1996.
- [22] M. A. P. Taylor. Evaluating the performance of a simulation model. *Transportation Research*, 13A(3):159–173, 1979/06.
- [23] Xiujuan Wang. Validation and calibration needs of MITSIM. Master of science in transportation, Massachusetts Institute of Technology, Cambridge, MA, August 1997.
- [24] Qi Yang. *A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1997.