

Calibration of Traffic Simulation Models using SPSA

by

Ioulia Markou

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Geoinformatics
School of Rural and Surveying Engineering
at The National University of Athens
Athens, 2014



Certified by:

Associate Professor Constantinos Antoniou, research supervisor, NTUA
Lecturer Ioanna Spyropoulou, NTUA
Lecturer Constantinos Kepaptsoglou, NTUA

Calibration of Traffic Simulation Models using SPSA

by

Ioulia Markou

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Geoinformatics
School of Rural and Surveying Engineering
at The National University of Athens
Athens, 2014



Certified by:

Associate Professor Constantinos Antoniou, research supervisor, NTUA
Lecturer Ioanna Spyropoulou, NTUA
Lecturer Constantinos Kepaptsoglou, NTUA

© Ioulia Markou 2014

All Rights Reserved

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Associate Professor Constantinos Antoniou for his advice and guidance. Without his help I would not be able to place this research in perspective. He is a constant source of knowledge and inspiration.

I would also like to thank the members of the Singapore-MIT Alliance for Research and Technology (SMART), and especially Stephen Robinson and Dr. Francisco Pereira, for the opportunity they gave me to work for six months on this challenging project.

Financial support from the Eugenides Foundation is gratefully acknowledged.

I am also grateful to my friends for their endless encouragement. Special thanks to Constantinos for his wholehearted support and patience.

At last, but definitely not least, I would like to thank my family, for their endless support and constant encouragement, especially my parents Babi and Panagiota and my sister Konstantina.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	xi
ABSTRACT	xiii
CHAPTER	
I. Introduction	1
II. Background Theory	5
2.1 Calibration Methodology	5
2.2 Calibration Algorithms	7
2.2.1 State-space formulation	7
2.2.2 Kalman-Filter	9
2.2.3 Iterated Extended Kalman Filter	12
2.2.4 Uncented Kalman Filter	13
2.2.5 Limiting Extended Kalman Filter	17
2.2.6 SP - Extended Kalman Filter	18
2.2.7 Optimization Algorithms	19
2.2.8 Stochastic approximation	24
2.3 Calibration of traffic simulation models	31
2.3.1 Car-Following Models	31
2.3.2 Large-scale simulation systems	37
III. Methodology	55
3.1 Calibration variables	55
3.2 Summary display of SPSA	56
3.3 Choice of non-negative coefficients	59

3.4	Objective Function	59
3.5	Validation process	60
IV.	Case Studies	63
4.1	Objectives	63
4.2	A simple static assignment problem	63
4.2.1	SPSA run following Spall's guidelines	65
4.2.2	Sensitivity analysis of parameter c	68
4.2.3	Sensitivity analysis of parameter a	71
4.2.4	Sensitivity analysis of parameter A	75
4.2.5	Sensitivity analysis of parameter gamma	78
4.2.6	Sensitivity analysis of parameter alpha	80
4.2.7	Final set of parameters	82
4.3	Another experiment using a car-following model	84
4.3.1	Transmodeler's Car-Following model overview	84
4.3.2	Model calibration using SPSA	85
V.	Towards complete DTA calibration	91
5.1	Objectives	91
5.2	Model and parameters	92
5.3	Experimental setup	94
5.3.1	Network description	95
5.3.2	Supply parameters' determination	96
5.3.3	Speed and flow data	96
5.4	Data cleaning	97
5.5	On-Line Calibration in the state estimation process	99
VI.	Conclusion	105
6.1	Summary	105
6.2	Future research directions	106
BIBLIOGRAPHY	108

LIST OF FIGURES

Figure

2.1	Example of stochastic optimization algorithm minimizing loss function $L(\theta_1, \theta_2)$ (Source: (<i>Spall</i> , 1998b))	25
2.2	A psycho-physical car following model Source (<i>Olstam and Tapani</i> , 2004)	34
2.3	Car- following notation (<i>Olstam and Tapani</i> , 2004)	35
3.1	Relative paths of SPSA and FDSA in a 2 parameter problem	58
4.1	Network for Stochastic Approximation Analysis (<i>Ozguven and Ozbay</i> , 2008)	64
4.2	RMSN values using the guidelines of <i>Ozguven and Ozbay</i> (2008)	66
4.3	RMSN values using the guidelines of <i>Ozguven and Ozbay</i> (2008)	66
4.4	New RMSN values using the guidelines of <i>Ozguven and Ozbay</i> (2008)	68
4.5	RMSN values using different values of c	70
4.6	RMSN values using different values of a	72
4.7	RMSN values using different values of c and $a=10$	74
4.8	RMSN for different values of A , $c=1$ and $a=10$	76
4.9	RMSN for different values of A , $c=1$ and $a=10$	77
4.10	RMSN for different values of γ , $A=10$, $c=1$ and $a=10$	79
4.11	RMSN for different values of α , $\gamma=0.01$, $A=10$, $c=1$ and $a=10$	81
4.12	Final results of RMSN	83
4.13	Range of speeds (m/s) (<i>Papathanasopoulou</i> , 2012)	86
4.14	Initial values of car following model parameters	87
4.15	Fitted values of Car-Following model's parameter "alpha"	88
4.16	Necessary sets of iterations per acceleration value	89
4.17	No. of necessary iterations sets for SPSA termination	89
5.1	Singapore Expressway Network (OpenStreetMap)	95
5.2	Example of cameras that measures traffic flow on Singapore expressways (photo taken by Stephen Robinson)	97
5.3	Map showing performance of each detector (Percentage Error)	98
5.4	A GLS approach is used in the current DynaMIT-R to calibrate OD values	99
5.5	Process flow of the SP-EKF method	100

List of Algorithms

II.1	Kalman Filter	10
II.2	Extended Kalman Filter	12
II.3	Unscented Transformation	14
II.4	Unscented Kalman Filter	16

LIST OF TABLES

Table

4.1	Summary of first implementations	67
4.2	Summary of first implementations	73
4.3	Summary of implementations (for different A)	78
4.4	Summary of implementations (for different gamma)	80
4.5	Experiments changing “a”, “A” “gamma” and “alpha” parameters .	82
4.6	Characteristics of data packages	86

ABSTRACT

Calibration of Traffic Simulation Models
using SPSA

by

Ioulia Markou

Supervisor: Associate Professor Constantinos Antoniou

Traffic simulation models, with the default parameter values, are not suitable for all studies. The characteristics of the network, the drivers behavior etc. are constantly changing. Each model represents this diversity through its parameters. When the basic format of the model is known, and there is enough data available, calibration is the procedure that aims at estimating the parameters that will lead to results, as close as possible to the observed ones in the field. The calibration is a crucial step for the successful application of traffic simulation models in transportation planning and real traffic management.

In this thesis, the calibration methodology using the algorithm Simultaneous Perturbation Stochastic Approximation (SPSA) is studied. The behavior of the algorithm is analyzed through various applications. Several particular behaviours are identified for which solutions are suggested. The algorithm was prepared to apply even in a quite demanding calibration workload, that of the expressway network of Singapore using the DymaMIT. Results indicate that SPSA successfully manages to reach the optimal solution, however with different speed and efficiency each time. The

parameter values were proved to be very important, especially the parameters “a”, “alpha” and “c”. Mismatches could even lead to the final failure of the algorithm’s convergence to the optimal solution.

CHAPTER I

Introduction

Mathematical models have become a very useful tool for representing various phenomena encountered in daily life. The demand for these models is increased as they are the main component of microscopic and macroscopic simulation software. Such programs are widely used in the transportation sector for the management, control and safety of traffic (capacity analysis, accident analysis, effects of traffic studies, intersection design, etc.). The modeling of driver behavior has contributed significantly to the development of intelligent systems (Intelligent Transportation Systems (ITS)) (*Koutsopoulos and Farah, 2012*).

The traffic models, with the default parameter values (or parameter values that the manufacturer gives), are not suitable for all studies. Most models and systems available make assumptions about the behavior of drivers, which are based on data collected under normal conditions and do not reflect the behavior of drivers under pressure. Their behavior can be affected by many parameters, such as the characteristics of drivers (age, sex, etc.), the perception of the driver (psychological state, stress, etc.), characteristics of the route (trip purpose, familiarity with the network etc.) and the environment (weather, behaviour of other drivers, emergencies).

The specifics of the environment are introduced indirectly to a model, through the values of their parameters. Prices change according to the requirements of each

research and the background. When the basic format of the model is known, and there is enough data available, calibration is the procedure that aims at estimating the parameters that will lead to results, as close as possible to the observed ones in the field. It shows which might be the “correct” combination of parameter values that will improve the agreement between model output and observed data; it is a widely used technique in the analysis of engineering systems. For the calibration procedure, either optimization algorithms or state-space representations could be used, depending on the nature and the requirements of the problem.

The main goal of this thesis is the calibration of traffic simulation models using the algorithm Simultaneous Perturbation Stochastic Approximation (SPSA). The behavior of the algorithm is analyzed through various applications. Several particular behaviours are identified for which solutions are suggested. The algorithm was prepared to apply even in a quite demanding calibration workload, that of the expressway network of Singapore.

At the **first part** of the paper, there is a bibliographic overview of calibration methodology. What follows, is a description of the various algorithms used. Afterwards, some simulation models are referenced, such as car-following models, which are utilized in a later chapter, to investigate the behavior of the SPSA.

At the **second part** begins by outlining the various model inputs and parameters. The SPSA algorithm is thoroughly analyzed, including methods of evaluating the utilized results.

At the **third part**, the two basic traffic model calibration case studies are listed. Sensitivity analysis is implemented and all the results of the algorithm are presented in detail. Furthermore there is a mention on certain difficulties which occurred and how they were resolved.

At the **fourth part**, the on-line calibration approach using the a DTA system is demonstrated. The DynaMIT (Dynamic network assignment for the Management

of Information to Travelers) is presented in detail. Afterwards the roads network of Singapore is described as well as the difficulties encountered in the use of the macroscopic simulator. To conclude, the main steps that demand the simulation and DynaMIT parameters calibration using SP-EKF are presented.

Conclusions and directions for further research are outlined in the **final part** of this thesis.

CHAPTER II

Background Theory

This chapter reviews the main principles of calibration methodology, which are followed in several transportation researches. A variety of optimization algorithms and state-space models are described. Furthermore, the systematic calibration of simulation based DTA systems and other traffic simulation models is presented.

2.1 Calibration Methodology

Mathematical models have a wide-spread use in transportation researches. All forms of traffic phenomena, like the formation of traffic jams, have almost universal properties that can be included in a plethora of traffic simulation algorithms. As a result, the availability of sufficiently accurate macroscopic and microscopic models is important for the design and the testing of modern freeway traffic control strategies. Their performance is largely independent of network's initial condition, data that could be incorporated in a model through different parameter values or small methodology changes.

When the basic format of the model is known and there is enough data available, calibration is the procedure that aims at estimating the parameters, that will lead to results close as possible to the observed ones in the field. It shows which might be the “correct” combination of parameter values that will improve the agreement between

model output and observed data. It is a widely used technique in the analysis of engineering systems.

The majority of the analytical methods use linear relationships in one way or another (*de Souza and Junqueira*, 2005). The examination of a calibration function for linearity is an important performance figure in validating an analytical method, as well as an everyday task in routine analytical operations. Additionally, an estimation of the uncertainty bounds around the optimum is useful. This uncertainty should be adequately accounted for in subsequent model applications (*Janssen and Heuberger*, 1995).

The calibration framework is a need which is continuously highlighted through several researches. *Allström et al.* (2014) presented a framework for calibrating a highway travel time estimation model, based on a two-stage process. Initially, the fundamental diagrams of links were calibrated, and in a second stage, the best possible model parameters were defined using a search method.

Ma et al. (2014) proposed a model calibration approach when on-road or in-lab instantaneous emission measurements are not directly available. Their final calibrated model was validated on several road networks with traffic states generated by the same microscopic traffic simulation model.

Antoniou et al. (2014) explored the use of distributions of collected data (such as accelerations, using opportunistic sensors, such as smart-phone accelerometers) for calibration purposes. It was noted that the collected data need to be appropriately pre-processed.

A microscopic calibration and validation of Car-Following Models was also implemented by *Treiber and Kesting* (2013). A special emphasis was given to the data requirements and preparation, as many authors conclude that there is an unsurmountable barrier for the rms error resulting in a stalemate when determining the “best” model. To conclude *Figueiredo et al.* (2014) presented which calibration parameters

errors tend to affect more the simulated results, and what are adequate precision levels to be achieved in a calibration process.

2.2 Calibration Algorithms

For the calibration procedure, either optimization algorithms or state-space representations could be used, depending on the nature and the requirements of the problem. At the first section, linear state-space models are described, and more specifically the Kalman Filter and its modified methodologies. At the second section, some significant optimization algorithms are presented, giving greater emphasis to the solution approach of Simultaneous Perturbation Methods.

2.2.1 State-space formulation

Calibration using a state-space model aims at improving the estimation accuracy and at maintaining the forecasting power of a model. Transition and measurement equations are used for the final problem solution. The state vector, which is defined as “the minimal set of data that is sufficient to uniquely describe the dynamic behavior of the system” (*Antoniou*, 2004) is fundamental for the state-space model description.

For the OD estimation and prediction problem, *Ashok and Ben-Akiva* (1993) proposed the use of *deviations*. As it is desirable to incorporate into the formulation as much historical information as possible, the use of *deviations* will bring several advantages to the application. The model formulation would indirectly take into account all the available *a priori* structural information. Traffic flow variables have skewed distributions, but the corresponding deviations on the other hand, have symmetric deviations and hence are more amenable to approximation by a normal distribution (*Antoniou*, 2004).

2.2.1.1 Set of parameters

The behavior of the studied system can be described with a set of parameters which compose the *state vector* \mathbf{x}_h . It includes the parameters π_h that need to be calibrated during the time interval h:

- OD flows,
- Speed-density relationship parameters and
- Segment capacities

It can be represented as:

$$\pi_h = \begin{bmatrix} x_h & p_h & c_h \end{bmatrix}^T = \begin{bmatrix} x_h & \gamma_h \end{bmatrix}^T \quad (2.1)$$

where x_h represents the vector of OD flows departing their origins during interval h, p_h the vector holding the values of parameters of the speed-density relationship models during interval h, c_h the vector of segment capacities for interval h, and γ_h summarizes the supply models' parameters.

The evolution of the *state vector* over time is captured by *transition equations*. Their general formulation is:

$$\pi_{h+1} = T(\pi_h, \pi_{h+1}, \dots \pi_{h-p}) + \eta'_h \quad (2.2)$$

where T is a function capturing the dependence of the parameter vector π_{h+1} during interval $h + 1$ on the values of the parameter vector during the past several intervals, p is the number of past parameter vectors that are considered and η'_h is a vector of random error terms.

2.2.2 Kalman-Filter

The formulation of several traffic simulation models is not linear, due to indirect measurement equations. However, it is useful to review a recursive solution to the discrete data linear filtering problem that R.E. Kalman described (*Kalman (1960)*). The Kalman Filter is the optimal mean square error (MMSE) estimator that can be obtained through linear estimators.

A form of feedback control is used for the Kalman filter's estimations. Firstly the filter estimates the process state at a certain time, and then obtains feedback in the form of (noisy) measurements. Therefore, it presents two groups of equations:

- Time update equations
- Measurement equations

The first group is responsible for projecting the current situation in the future and the second group for the feedback - i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate (*Welch and Bishop (1995)*).

The main steps of the Kalman Filter are as follows:

Algorithm II.1 Kalman Filter

Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \quad (2.3)$$

$$\mathbf{P}_{0|0} = \mathbf{P}_0 \quad (2.4)$$

for h=1 to N **do**

Time update

$$\mathbf{X}_{h|h-1} = \mathbf{F}_{h-1} \mathbf{X}_{h-1|h-1} \quad (2.5)$$

$$\mathbf{P}_{h|h-1} = \mathbf{F}_{h-1} \mathbf{P}_{h-1|h-1} \mathbf{F}_{h-1}^T + \mathbf{Q}_h \quad (2.6)$$

Measurement update

$$\mathbf{G}_h = \mathbf{P}_{h|h-1} \mathbf{H}_h^T (\mathbf{H}_h \mathbf{P}_{h|h-1} \mathbf{H}_h^T + \mathbf{R}_h)^{-1} \quad (2.7)$$

$$\mathbf{X}_{h|h} = \mathbf{X}_{h|h-1} + \mathbf{G}_h (\mathbf{Y}_h - \mathbf{H}_h \mathbf{X}_{h|h-1}) \quad (2.8)$$

$$\mathbf{P}_{h|h} = \mathbf{P}_{h|h-1} - \mathbf{G}_h \mathbf{H}_h \mathbf{P}_{h|h-1} \quad (2.9)$$

end for

where \mathbf{X}_0 is a starting estimate of the state and \mathbf{P}_0 is the initial state of the variance-covariance matrix. During the time update, the above values are projected from time step h-1 to step h (Equations 2.5 and 2.6). The measurement update phase incorporates the new information about the measurement vector \mathbf{Y}_h and generates an *a posteriori* state estimate (Equation 2.8). The final step is to obtain an *a posteriori* error covariance estimate via Equation 2.9. In the last important process, key role plays the Kalman gain \mathbf{G}_h , which is computed as per Equation 2.7.

All the above procedure is repeated N times, with the previous *a posteriori* estimated used to project or predict the new *a priori* estimates. It is one of the very appealing features of the Kalman filter, as it makes practical implementations more feasible that for instance an implementation of a Weiner filter which is designed to operate on all on all of the data directly for each estimate (*Welch and Bishop (1995)*).

More information about the the Kalman filter, can be found in many texts, including *Maybeck* (1982), *Chui and Chen* (1999) and *Grewal and Andrews* (2011).

2.2.2.1 Extended Kalman Filter

As it is already mentioned, the original Kalman filter theory applies to linear systems. In many studies though, non-linear models have been designed and analyzed. Solutions for these interesting problems lead to the development of modified Kalman Filter methodologies. Good approximations are achieved by the **Extended Kalman Filter (EKF)** via first order Taylor series expansion (linearization) of the appropriate equations. In more details:

The EKF algorithm includes an intermediate linearization step (Equation 2.14), where it is necessary to use numerical derivatives. Assuming the use of central derivatives, it is necessary to evaluate the function $2n$ times, where n is the dimension of the state vector. (If forward derivatives are used, then this number drops to $n+1$ evaluations.) Each such evaluation implies one run of the simulator. Therefore, it becomes apparent that this process of linearization dominates the computational complexity of the algorithm.

More details are presented in *Julier and Uhlmann* (1997), *Chui and Chen* (1999) and *Liang-Qun et al.* (2005).

Algorithm II.2 Extended Kalman Filter

Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \quad (2.10)$$

$$\mathbf{P}_{0|0} = \mathbf{P}_0 \quad (2.11)$$

for h=1 to N **do**

Time update

$$\mathbf{X}_{h|h-1} = \mathbf{F}_{h-1} \mathbf{X}_{h-1|h-1} \quad (2.12)$$

$$\mathbf{P}_{h|h-1} = \mathbf{F}_{h-1} \mathbf{P}_{h-1|h-1} \mathbf{F}_{h-1}^T + \mathbf{Q}_h \quad (2.13)$$

Linearization

$$\mathbf{H}_h = \left. \frac{\partial h(x^*)}{\partial x^*} \right|_{x^* = \mathbf{X}_{h|h-1}} \quad (2.14)$$

Measurement update

$$\mathbf{G}_h = \mathbf{P}_{h|h-1} \mathbf{H}_h^T (\mathbf{H}_h \mathbf{P}_{h|h-1} \mathbf{H}_h^T + \mathbf{R}_h)^{-1} \quad (2.15)$$

$$\mathbf{X}_{h|h} = \mathbf{X}_{h|h-1} + \mathbf{G}_h (\mathbf{Y}_h - \mathbf{H}_h \mathbf{X}_{h|h-1}) \quad (2.16)$$

$$\mathbf{P}_{h|h} = \mathbf{P}_{h|h-1} - \mathbf{G}_h \mathbf{H}_h \mathbf{P}_{h|h-1} \quad (2.17)$$

end for

2.2.3 Iterated Extended Kalman Filter

When the linearization of the measurement equation about the present best estimate of the state vector \mathbf{X} is involved, the resulting filter is called the **Iterated Extended Kalman Filter (Iterated EKF)**. When this step is completed a presumably superior estimate $\mathbf{X}_{h|h}$ is available which could then be used to linearize the measurement equation and repeat the update step. These iterations could be repeated as many times as deemed necessary, while assessing the numerical derivative. The last process burdens fairly the overall runtime of the algorithm by an amount

equal to the EKF algorithm (*Antoniou (2004)*).

2.2.4 Uncented Kalman Filter

Another good approximation of non-linear transformations is the **Uncented Kalman Filter (UKF)**, proposed by *Julier and Uhlmann (1997)*. It uses a deterministic sampling approach (Unscented Transformation, UT) to represent a random variable using a number of deterministically selected sample points (often called *sigma points*). These sample points completely capture the true mean and covariance of the Gaussian random variable (GRV), and when propagated through the true non-linear system, captures the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any nonlinearity (*Wan and Van Der Merwe (2000)*). The UT performs the following (Algorithm 3):

Algorithm II.3 Unscented Transformation

Generation of *sigma points*

$$X_{0|h} = x_h \quad (2.18)$$

for i=1 to n **do**

$$X_{i,h} = x_h + (\sqrt{(n + \kappa)\mathbf{P}_{x,h}})_i \quad (2.19)$$

end for

for i=n to 2n **do**

$$X_{i,h} = x_h - (\sqrt{(n + \kappa)\mathbf{P}_{x,h}})_i \quad (2.20)$$

end for

Generation of weights

$$W_0^m = \kappa / (n + \kappa) \quad (2.21)$$

$$W_0^c = \kappa / (n + \kappa) + (1 - a^2 + b) \quad (2.22)$$

for i=1 to 2n **do**

$$W_0^m = W_0^c = 1 / [2(n + \kappa)] \quad (2.23)$$

end for

The UKF firstly uses the $2n+1$ *sigma points* that the Unscented Transformation has calculated for the time update step. The prior estimate of the state vector and the state covariance is computed as a weighted sum of the propagated sigma points (Equation 2.25 and 2.26). In a second phase, the sigma points are transformed into a vector of respective measurements, through the Equation 2.27. The measurement vector is computed as a weighted sum of the generated measurements (Equation 2.28).

The covariance of the state and the measurement vectors play a significant role in the calculation of the Kalman gain (Equations 2.29, 2.30). The weights that have been obtained from the Unscented Transformation during the initialization step, are used in the calculations. Equation 2.32 introduces the measurement vector \mathbf{y}_h and uses the Kalman gain to correct the state estimate x_h . Finally, through the Equation 2.33, the state covariance is updated.

Algorithm II.4 Unscented Kalman Filter

for h=1 to N **do**

Generate *sigma points* and weights using the Unscented Transformation (Algorithm 3)

Time update

$$X_{h|h-1} = \mathbf{f}(X_{h-1}) \quad (2.24)$$

$$x_{h|h-1} = \sum_{i=0}^{2n} W_i^m X_{i,h|h-1} \quad (2.25)$$

$$\mathbf{P}_{x,h|h-1} = \sum_{i=0}^{2n} W_i^c (X_{i,h|h-1} - x_{h|h-1})(X_{i,h|h-1} - x_{h|h-1})^T + \mathbf{Q}_h \quad (2.26)$$

$$y_{i,h|h-1} = \mathbf{h}(X_{i,h|h-1}) \quad (2.27)$$

$$y_{h|h-1} = \sum_{i=0}^{2n} W_i^m y_{i,h|h-1} \quad (2.28)$$

Measurement update

$$\mathbf{P}_{y,h} = \sum_{i=0}^{2n} W_i^c (y_{i,h|h-1} - y_{h|h-1})(y_{i,h|h-1} - y_{h|h-1})^T + \mathbf{R}_h \quad (2.29)$$

$$\mathbf{P}_{xy,h} = \sum_{i=0}^{2n} W_i^c (X_{i,h|h-1} - x_{h|h-1})(y_{i,h|h-1} - y_{h|h-1})^T \quad (2.30)$$

$$\mathbf{G}_h = \mathbf{P}_{xy,h} \mathbf{P}_{y,h}^{-1} \quad (2.31)$$

$$x_h = x_{h|h-1} + \mathbf{G}_h (y_h - y_{h|h-1}) \quad (2.32)$$

$$\mathbf{P}_{x,h} = \mathbf{P}_{x,h|h-1} - \mathbf{G}_h \mathbf{P}_{y,h} \mathbf{G}_h^T \quad (2.33)$$

end for

2.2.5 Limiting Extended Kalman Filter

A special case of the Extended Kalman Filter, which improves the computational performance of the algorithm, was proposed in *Antoniou* (2004) and in *Antoniou et al.* (2007a). The **Limiting Extended Kalman Filter** overtakes the step of the linearization of the measurement equation and consequently the computation time is considerably decreased. In more detail:

- The *limiting* Kalman Filter replaces the Gain matrix \mathbf{G} with its limit \mathbf{G} , called the *limiting* (or *stable*) *Kalman gain matrix*. The matrix is computed off-line, as the average of a number of available Kalman gain matrices:

$$\mathbf{G} = \frac{\sum_{m=1:M} \mathbf{G}_m}{\mathbf{M}} \quad (2.34)$$

where \mathbf{G}_m is the Kalman gain obtained from EKF during interval m and \mathbf{M} is the total number of available Kalman gain matrices.

- Using the same principle as above, the time-dependent matrix \mathbf{H}_h is also replaced with the average \mathbf{H} of a number of available matrices:

$$\mathbf{H} = \frac{\sum_{m=1:M} \mathbf{H}_m}{\mathbf{M}} \quad (2.35)$$

where \mathbf{H}_m is the matrix obtained from EKF during interval m and \mathbf{M} is the total number of available matrices.

- The resulting matrix \mathbf{H} is used then to update the state covariance (Equation 2.17).

Antoniou (2004) additionally mentioned that, weighted averages could also be considered, instead of the average of all the available Kalman gain matrices from the

off-line computations. For example lower weight values could be assigned to older gain matrices.

The performance of that approximate algorithm proved very close to the “exact” EKF and raised interest for the approach. Two characteristic case studies were implemented by *Antoniou et al.* (2013). Through the first case scenario in a synthetic network, it was proved that LimEKF provides comparable to that of the best algorithm (EKF) corrections and minimizes the discrepancy between the simulated and observed traffic conditions. Additional experiments undertaken in a real-world, large-scale network in Stockholm, Sweden further validated these findings. The noise that creeps into the computation of the individual Kalman gain matrices was reduced and this may be one of the key reasons for its good performance.

2.2.6 SP - Extended Kalman Filter

A modified Kalman Filter methodology for non-linear models has been developed by *Antoniou et al.* (2007b). More specifically, the integration of simultaneous perturbation in the step of linearization was presented. The proposed methodology is based on the Extended Kalman Filter, which has already been described in Section 2.2.1.

The innovation of SP is that each of the elements are not perturbed individually, but instead all elements are perturbed simultaneously. Only two functions are necessary for the gradient approximation. In the proposed algorithm of SP-EKF the linearization step:

$$\mathbf{H}_h = \left. \frac{\partial h(x^*)}{\partial x^*} \right|_{x^*=X_{h|h-1}} \quad (2.36)$$

is replaced by the simultaneous perturbation approximation:

$$\widehat{g}(\theta^i) = \frac{z(\theta^i + c^i \Delta_i) - z(\theta^i - c^i \Delta_i)}{2c^i} \begin{bmatrix} \Delta_{i1}^{-1} \\ \Delta_{i2}^{-1} \\ \vdots \\ \Delta_{iK}^{-1} \end{bmatrix} \quad (2.37)$$

The use of the simultaneous perturbation was proved far more efficient than the usual numerical derivatives. All on-line calibrated sets of parameters result in significant improvements. It is therefore interesting and challenging to see how SP-EKF could be applied on DynaMIT.

2.2.7 Optimization Algorithms

Optimization algorithms are significant for the design, analysis and control of most engineering systems. A state-space equivalent formulation is the so-called direct optimization formulation. *Ashok* (1996) has discussed the connection between the state-space and the Generalized Least Square direct minimization formulations in general, and Kalman Filter and least square estimation as their respective solutions in specific.

Optimization algorithms can be classified into pattern search, path search, and random search techniques. Commonly adopted optimization algorithms also need an explicit objective function. A small description of some optimization methods follows.

2.2.7.1 Partial least-squares (PLS) modeling

Partial least-squares (PLS) modeling is a statistical tool that has already been applied to the quantitative analysis of various data. It is a set of algorithms developed by Word for use in econometrics (*Manne*, 1987). They have in common that no a priori assumptions are made about the model structure. *Geladi and Kowalski* (1986)

provided a tutorial about this regression method. In order to make the calculations easier, it is suggested to tailor the data in the calibration set. The average value for each variable is calculated from the calibration set and then subtracted from each corresponding variable.

Dependent variables and the independent ones can be scaled differently because the sensitivities absorb the differences in scaling. They are treated according to the following three ways (*Geladi and Kowalski*, 1986):

- No scaling is needed when all the variables in a block are measured in the same units
- Variance scaling is used when the variables in a block are measured in different units; scaling is accomplished by dividing all the values for a certain variable by the standard deviation for that variable, so that the variance for every variable is unity.
- One can decide that certain variables are of less importance and hence should not influence the model very much; so they are given a smaller weight.

PLS software has been made available by several Fourier transform infrared (FT-IR) instrument manufacturers for quantitative spectral analyses. PLS is capable of being a full-spectrum method and therefore enjoys the signal averaging advantages of other full-spectrum methods such as PCR and CLS.

2.2.7.2 Principal Components Analysis (PCA)

Principal Components Analysis (PCA) is “a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences”, as *Smith* (2002) noted. It is a powerful tool, because it can be hard to find patterns in data of high dimension. When the appropriate ones are found, data could be compressed by reducing the number of dimensions, without much loss of information.

A detailed tutorial on the specific methodology of calibration and data analysis could be found in *Smith* (2002).

PCA has been proposed as a method for traffic anomaly detection (*Ringberg et al.*, 2007). Traffic measurements from two IP backbones (Ablilene and Geant) across three different traffic aggregations (ingress routers, OD flows, and input links) were analyzed. The final product was a detailed inspection of the feature time series for each suspected anomaly.

Lakhina et al. (2004) made its application very popular in the networking community. The method detected when a volume anomaly is occurring, identified the underlying OD flow which is the source of anomaly, and accurately estimated the amount of traffic involved in the anomalous OD flow.

However, PCA is very sensitive to its parameters settings (*Ringberg et al.*, 2007). The authors have reported about instability problems encountered when using PCA, however, they failed in providing precise reasons for their observation. *Brauckhoff et al.* (2009) mentioned that PCA does not consider the temporal correlation of the data.

2.2.7.3 Generalized Likelihood Uncertainty Estimation (GLUE)

The GLUE procedure recognizes the equivalence or near-equivalence of different sets of parameters in the calibration of distributed models. It is based upon making a large number of runs of a given model with different sets of parameter values, chosen randomly from specified parameter distributions. Noted that “on a basis of comparing predicted and observed responses, each set of parameter values is assigned a likelihood of being a simulator of the system. That likelihood may be zero when it is considered that the set of parameter values gives a behavior that is not characteristic of the system, either because of the direct comparison with the available data, or because of conditioning on the basis of some a priori knowledge about the system”.

All the simulations with a likelihood measure significantly greater than zero (0) are retained for consideration. If a rescaling of the likelihood values is chosen, then the sum of all likelihood values equals one (1) yields a distribution function for the parameter sets. Under these conditions, the optimal solution during the calibration search is given a likelihood of one (1) and all the others are set to zero (0). According to *Beven and Binley* (1992), GLUEs procedure requirements are:

- A formal definition of a likelihood measure or set of likelihood measures
- A formal definition of a likelihood measure or set of likelihood measures.
- An appropriate definition of the initial range of distribution of parameter values
- A procedure for using likelihood weights in uncertainty estimation
- A procedure for updating likelihood weights recursively as new data become available
- A procedure for evaluating uncertainty such that the value of additional data can be assessed.

2.2.7.4 Nelder-Mead method

The Nelder-Mead method or downhill simplex method or amoeba method is a commonly used nonlinear optimization technique, which is a well-defined numerical method for problems for which derivatives may not be known. It was proposed by *Nelder and Mead* (1965) and is a technique for minimizing an objective function in a many-dimensional space.

The method uses the concept of a simplex, which is a special polytope of $N+1$ vertices in N dimensions. It approximates a local optimum of a problem with N variables when the objective function varies smoothly and in unimodal.

The Nelder-Mead algorithm was used by *Spiliopoulou et al.* (2014) through the calibration procedure, for the calculation of the optimal parameter values, and the minimum value of the objective function. It was also used by *Brockfeld et al.* (2004) to calibrate a small set of supply parameters in a wide range of microscopic and macroscopic traffic models. Others report on the successful application of genetic algorithms (GA) for the calibration of select parameters in various microscopic traffic simulation tools (*Abdulhai et al.*, 1999; *Lee et al.*, 2001; *Kim and Rilett*, 2004).

An improved method, called Globalized Bounded Nelder-Mead (GBNM) algorithm, is particularly adapted to tackle multimodal, discontinuous optimization problems, for which it is uncertain that a global optimization can be afforded (*Luersen et al.*, 2004). It does not need sensitivities and constructively uses computer resources up to a given limit. More details could be found in *Luersen et al.* (2004) and *Ghiassi et al.* (2008).

2.2.7.5 Box-Complex method

The Box-Complex method follows some of the principles of the Nelder-Mead approach. More specifically, they both begin with a set of randomly-selected feasible points that span the search space. The point with the highest objective function value in each iteration is replaced by its reflection about the centroid of the remaining points. If the result gets worse, the studied point may be moved closer to the centroid using some contraction scheme. More details could be found in *Box* (1965).

Balakrishna (2006) proposed the combination of the Box-Complex and SNOBFIT optimization algorithms as a solution approach of the off-line calibration problem. The first algorithm is able to cover the feasible space effectively, while the SNOBFIT algorithm utilizes the information from every function evaluation to systematically search for local and global minima.

2.2.7.6 BOSS Quattro package

The Boss Quattro package is an optimization tool that has been used in several projects, such as the calibration of the MITSIMLab model. It is a general-purpose design program that includes several “engines”: optimization, parametric studies, ”Monte-Carlo” studies, “design of experiments” and updating. According to *Radovic and Remouchamps* (2002), the classes of problems that could be solved using BOSS/Quattro are:

- Linear and nonlinear convex constrained problems with a large number of variables and/or design functions. The objective function is supposed to be monotonic. The proposed algorithm is ConLin algorithm.
- Problems with mixed discrete and continuous variables.
- General nonlinear optimization with continuous functions The proposed algorithms are MDQA (method of diagonal quadratic approximations), GCMMA (globally convergent method of moving asymptotes by Svanberg) and SQP (sequential quadratic programming).
- Updating problems for models that have not precisely set parameters.
- General nonlinear problems with practically unavailable derivatives

BOSS-Quattro’s open architecture allows a wide spectre of possible design optimization approaches from m parametric studies and various simulations to sensitivity analyses (*Dolšak and Novak*, 2011).

2.2.8 Stochastic approximation

The mathematical representation of the simultaneous perturbation methods is the minimization (or maximization) of some scalar-valued objective function with respect

to a vector of adjustable parameters (*Spall*, 1998b). The proposed methodology is a step-by-step procedure for changing the adjustable parameters from some initial guess to a value that offers an improvement in the objective function (Figure 2.1).

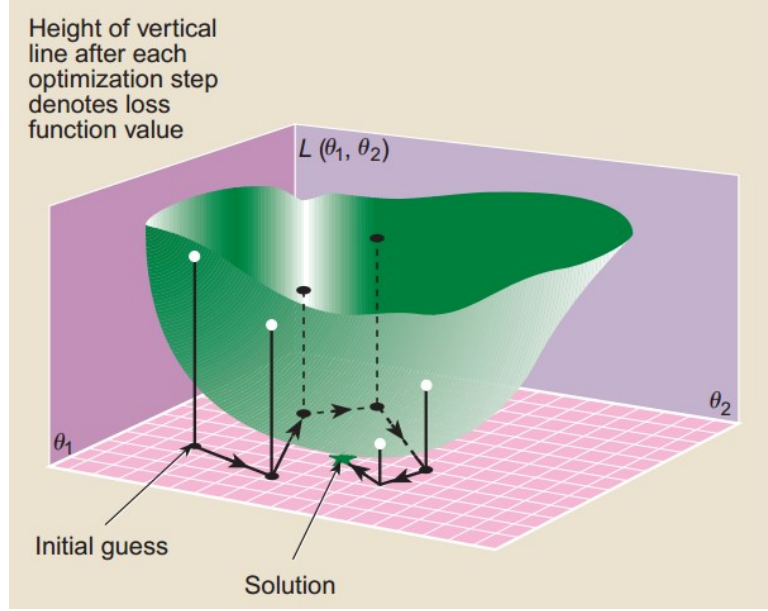


Figure 2.1: Example of stochastic optimization algorithm minimizing loss function $L(\theta_1, \theta_2)$ (Source: (*Spall*, 1998b))

There has been a growing interest in recursive optimization algorithms that do not depend on direct gradient information or measurements (such as R-M SA algorithm), but they are based on an approximation of the gradient formed from (generally noisy) measurements of the loss function.

Spall (1998b) mentioned that, as a general rule, when speed is measured in number of iterations, the gradient-based algorithms converge faster to the optimal solution than those using loss functionbased gradient approximations. On the basis of asymptotic theory, the second methodology requires additional information.

The SPSA and FDSA procedures are in the general recursive SA form:

$$\theta^{i+1} = \theta^i - \alpha^i \hat{g}(\theta^i) \quad (2.38)$$

where θ^i is the parameter vector at the beginning of iteration i ; $\widehat{g}(\theta^i)$ is a current estimation of the gradient and α^i is a gain sequence of step sizes. Under appropriate conditions, the iteration in equation 2.38 will converge to θ^* in some stochastic sense.

2.2.8.1 Finite Difference Stochastic Approximation (FDSA)

The gradient approximation $\widehat{g}(\theta^i)$ is the essential part of the Equation 2.38. One-sided gradient approximations involve measurements $y(\widehat{\theta}_k)$ and $y(\widehat{\theta}_k + \text{perturbation})$, while two-sided approximations involve measurements of the form $y(\widehat{\theta}_k \pm \text{perturbation})$. The classical method for gradient free stochastic optimization is the Kiefer-Wolfowitz finite-difference SA (FDSA) algorithm (*Kiefer et al.*, 1952).

The two-sided FD approximation for use with Equation 2.38 is

$$\widehat{g}(\theta^i) = \begin{bmatrix} \frac{y(\widehat{\theta}_i + c_i \xi_1) - y(\widehat{\theta}_i - c_i \xi_1)}{2c_i} \\ \vdots \\ \frac{y(\widehat{\theta}_i + c_i \xi_p) - y(\widehat{\theta}_i - c_i \xi_p)}{2c_i} \end{bmatrix} \quad (2.39)$$

where ξ_i denotes a vector with a 1 in the i th place and 0's elsewhere and $c_k > 0$ defines the difference magnitude. The pair a_i, c_i are the gains (or gain sequences) for the FDSA algorithm.

A total of $2p$ function evaluations are required per symmetric gradient computation. The effort per iteration increases each time linearly with the problem size.

2.2.8.2 Simultaneous Perturbation Stochastic Approximation (SPSA)

The SPSA methodology reduces significantly the run times from days to minutes or hours compared other applicable algorithms. For example, Robbins-Monro algorithm (*Robbins and Monro*, 1951), which concludes when an unbiased estimator of the gradient is used to perform the parameter updates, requires detailed knowledge of

the model being calibrated. Consequently, similar work is considered impractical in a demanding environment such as the DTA models which have large-scale and stochastic nature.

SPSA algorithm approximate the gradient with just two functions evaluations:

$$\widehat{g}(\theta^i) = \frac{z(\theta^i + c^i \Delta_i) - z(\theta^i - c^i \Delta_i)}{2c^i} \begin{bmatrix} \Delta_{i1}^{-1} \\ \Delta_{i2}^{-1} \\ \vdots \\ \Delta_{iK}^{-1} \end{bmatrix} \quad (2.40)$$

where Δ_i is a K-dimensional perturbation vector and Δ_{iK} is the i th component of the Δ_k vector; the common numerator in all p components of $\widehat{g}(\theta^i)$ reflects the simultaneous perturbation of all components in θ^i in contrast to be component-by-component perturbations in the standard finite-difference approximation.

SPSA's convergence is highly dependent on the choice of gain sequences, as well as the distribution of the perturbations Δ_i . The optimal solution should be approached at rates that are neither too high nor too low, and the objective function must be several times differentiable in the neighborhood of θ^0 .

2.2.8.3 The Weighted SPSA algorithm

Through a case study on the whole Singapore expressway system, it was found that although SPSA algorithm has been successfully applied and it kept its advantage of highly computationally efficient, the accuracy performance deteriorated when the scale of the DTA calibration problem became much larger, in terms of network size and number of considered intervals. It was found that the source of this problem was not the stochasticity in the DTA models, or the inconsistency in the observed data due to measurement error, but the way SPSA estimated gradients. The algorithm tries to

find a direction and amplitude for each parameter value by comparing the influences to the system caused by perturbing each of the parameter value in two opposite directions. The influences are determined by the sum of all the distances between model outputs and corresponding observed measurements. As all the parameters are perturbed at the same time, the change in a measurement value may or may not be caused by this specific parameter. When each parameter is highly correlated to most of the measurements, then this may not be a significant issue. However, in a real-world traffic system correlations between model parameters and measurements are often sparse, in terms of both the spatial and temporal dimensions.

In order to improve the performance of SPSA algorithm on DTA calibration, some successful attempts have been made. *Balakrishna and Koutsopoulos (2008)* incorporated transition equations of OD flows in the objective function. The proposed method does not rely on assignment matrices and it avoids the problem of solving expensive fixed-point problems. *Cipriani et al. (2011)* proposed a modification of the basic SPSA path search optimization method by introducing the assumption that the starting point is "near" the optimal one. They introduced a generation constraint, which allows the algorithm to avoid overestimation problems while the normalization of link terms of the OF is useful mainly in guarding against underestimation cases.

All the researches however, targeted only the estimation of OD flows. The concern of algorithm noise could not be addressed with these modifications. It is important to incorporate into the demand - supply calibration framework the knowledge about existing correlations in a the system (*Lu, 2014*).

In the calibration problem of a DTA model there are two types of correlations:

- Spatial correlations: They are mainly decided by the network topology, traffic conditions and drivers' route choice behavior.
- Temporal correlations: They are mainly determined based on departure times and travel times.

The structure which can integrate into the DTA model the above factors between each parameter and measurements, is a matrix W_e . The features and the structure of the network is known. Assume the time period of interests can be divided into intervals $h = 1, 2, \dots, H$. Within an interval, $p = 1, 2, \dots, P$ is the index of model parameter and $m = 1, 2, \dots, M$ is the index of measurement.

$$W_e = w_{p+Px(h_1-1), m+Mx(h_2-1)} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,M} & \cdots & w_{1,MxH} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,M} & \cdots & w_{2,MxH} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_{P,1} & w_{P,2} & \cdots & w_{P,M} & \cdots & w_{P,MxH} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_{PxH,1} & w_{PxH,2} & \cdots & w_{PxH,M} & \cdots & w_{PxH,MxH} \end{bmatrix} \quad (2.41)$$

where $w_{p+Px(h_1-1), m+Mx(h_2-1)}$ is the relative correlation between the p_{th} model parameter in interval h_1 and the m_{th} measurement in interval h_2 , $\forall p, \forall m, \forall h_1 \in h, \forall h_2 \in h$.

The weight matrix is a combination of the above matrix and the matrix W_h (Equation 2.42), which represents the distances between the current parameter values and the reliable historical parameter values in the objective function.

$$\mathbf{W} = [\mathbf{W}_e \mathbf{W}_h] \quad (2.42)$$

The most challenging part of the calculations is the determination of the influencing measurements. The weight matrix can be calculated before the calibration process, and as a result W-SPSA maintains SPSA's advantage of high computational efficiency. It can be described as a normalized Jacobi matrix where each element in the matrix is the normalized value of $\frac{\partial m_i}{\partial p_j}$ where m_i is the i_{th} measurement and p_j

is the j_{th} parameter starting from the first interval.

In this approach, measurements are considered in a weighted manner based on their relevance to a parameter:

$$\widehat{g_{ki}}(\widehat{\vartheta_k}) = \frac{\sum_{j=1}^D w_{ji} [(e_{Mkj}^+)^2 - (e_{Mkj}^-)^2]}{2c_k \Delta_{ki}} = \frac{1}{2c_k \Delta_{ki}} W_i' \begin{bmatrix} (e_{Mk1}^+)^2 - (e_{Mk1}^-)^2 \\ \vdots \\ (e_{MkD}^+)^2 - (e_{MkD}^-)^2 \end{bmatrix} \quad (2.43)$$

Where w_{ji} is the element at the j_{th} row and i_{th} column of a D x P weight matrix. W_i' is the i_{th} column of the matrix. D is the number of deviations (measurements plus historical parameter values) and P is the number of parameters.

The frequency of this process depends on the nature of the problem, the stage of calibration process, and the amount of available computational power. More details about the WSPSA's gradient approximation could be found in *Lu (2014)*.

2.3 Calibration of traffic simulation models

Each traffic simulation model should be able to produce results close to reality. Accurate solutions would be achieved through proper calibration of the simulation parameters as well as validation. Therefore, the calibration algorithms could be applied to simple equations, such as the speed-density relationship, but also to traffic simulators. In this section, certain models where the calibration algorithms could be applied, are described.

2.3.1 Car-Following Models

A car following model controls the behavior of drivers in relation with the preceding vehicle in the same lane. A vehicle is limited by the movement of the vehicle in front of it, because driving at the desired speed can lead to a collision. When a vehicle is unrestricted by a preceding vehicle, it is assumed that it moves freely at the drivers desired speed. The actions of a vehicle which follows another are defined by the acceleration of the vehicle, even though in some models like that of *Gipps* (1981), the actions of the vehicle following are based on its velocity. Some car following models describe the behavior of the drivers, only in the case that they are following some other vehicle, while they include all other situations. Every car following model must define the state of the vehicle as well as the actions performed in each situation.

Most car following models describe the behavior of drivers in various situations. A common classification is the following: One situation corresponds to unrestricted driving state, another to the case where a vehicle follows some other car and one more for emergency deceleration. The vehicles in unrestricted state have no limitations and attempt to achieve the desired velocity, while the vehicles which follow another are limited by its speed. The vehicles in emergency deceleration status, are subject to sudden braking in order to avoid a collision (*Olstam and Tapani*, 2004).

2.3.1.1 Categories of Car-Following Models

According to *Olstam and Tapani* (2004) the car following models are categorized by their underlying logic. Those categories are Gazis-Herman-Rothery models, the models of safety distance, psycho-physical models and the models of fuzzy logic.

The Gazis-Herman-Rothery (GHR) models, or car following models in general, first appeared in 1958 and have undergone significant improvements since. Those models check only for the actual behavior of drivers who follow another vehicle. The relationship between the preceding vehicle and the one following is a function similar to action-reaction. The acceleration of the superseding vehicle is proportionate to the speed of the preceding vehicle, the difference between their respective velocities and the distance between them (*Brackstone and McDonald*, 1999). The model also includes three parameters which control the proportions. The model, can be symmetrical in the case where the values of the parameters used, are the same for both the acceleration and deceleration states, or asymmetrical in the case where different values are used.

In the safety distance models, the driver of the superseding vehicle should always keep a safety distance relative to the vehicle in front. According to *Hoogendoorn and Bovy* (2001), a simple example of a safety distance model is the following: A rule for the subsequence of a vehicle in a safe distance, requires a distance between the two vehicles, equal to the length of a vehicle for every ten miles per hour. However, the safety distance is usually defined by Newtons laws of motion. In some models, the aforementioned distance is described as the necessary distance in order to avoid a collision if the preceding vehicle brakes abruptly. The first such model was presented by Kometani and Sasaki in 1959 (*Brackstone and McDonald*, 1999). In 1981, Gipps presented an improved version of the original model; in this version, the vehicle that follows is not going to collide with the preceding vehicle if their temporal distance is equal or larger than $3 T / 2$ (where T is the reaction time) and the estimate of the

driver of the last vehicle -regarding the deceleration of the vehicle in front-is equal or larger than the actual deceleration.

In 1963 Michaels presented a new approach of car following models (*Brackstone and McDonald*, 1999). The models which use this approach are referred to as psycho-physical. The aforementioned GHR models, assume that the driver of the last vehicle reacts to small changes in the relative speed, and in any action of the vehicle in the front, even if the relative distance is too large or the reaction does not happen as soon as their relative speed becomes zero. This can be further corrected, either by extending the GHR models by adding extra states of driving or by using a psycho-physical model.

The psycho-physical models use the lesser bounds or points at which the driver changes his behavior; the drivers can react to changes in distance or relative speed, only when those bounds are met (*Leutzbach*, 1988). The lowest boundaries and the states that they define, are presented in a diagram of distance relative to velocity for a pair of vehicles. One such example is shown in Figure 2.2. Great examples of psycho-physical models are those developed by *Wiedemann and Reiter* (1992) and *Fritzsche* (1994).

The fuzzy logic models use fuzzy sets in order to quantify concepts such as very close; such sets are inserted in logical rules, e.g. if very close, then use emergency braking. In previously mentioned models, the exact speed, distance and everything else required are assumed to be known; in fuzzy logic models, it is instead assumed that the drivers are capable of judging if the speed of the preceding vehicle is low, medium, high or very high. The fuzzy sets are likely to overlap and in such cases, a probabilistic density function is used as to decide e.g. if the driver believes that the speed of the vehicle ahead is low or medium. There have been numerous attempts at creating and using one such model, as for example that by *Al-Shihabi and Mourant* (2003).

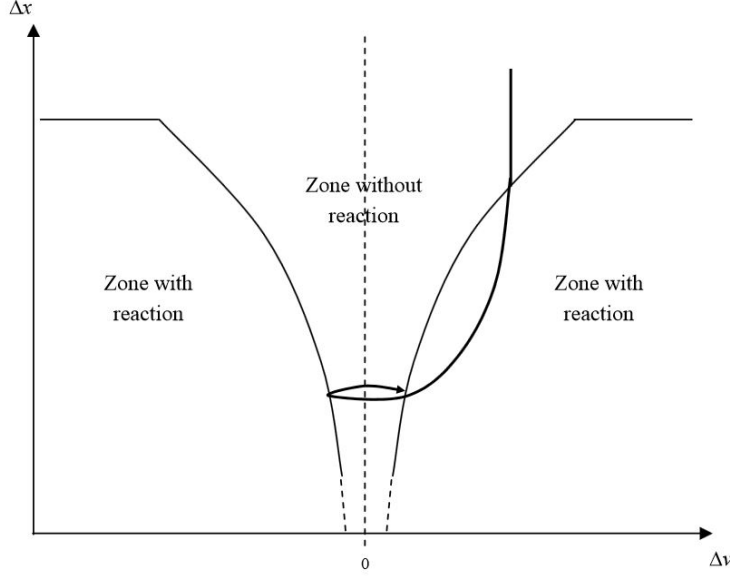


Figure 2.2: A psycho-physical car following model Source (*Olstam and Tapani, 2004*)

The Gipps model

One of the most widespread models is that of Gipps (1981), which relies on safety distance. It is used daily as it constitutes the basis of various pieces software of micro simulation, such as AIMSUN (*Barceló and Casas, 2005*), DRACULA (*Liu, 2010*), SIGSIM (*Silcock, 1993*), and SISTM (*Abou-Rahme et al., 2000*). According to this model, the speed of a vehicle (n) that follows another ($n-1$) is subject to three limitations. Firstly, the speed of the vehicle n cannot surpass the speed at which its driver wants to move (V_n). Furthermore, the vehicle accelerates fast initially, as to approach the desired speed, and after achieving it, its acceleration declines to zero. In the case where two vehicle are apart one from another, namely in free flow state, the two aforementioned conditions can be summarized in the following formula:

$$u_n(t + \tau) \leq u_n(t) + 2.5 \cdot a_n \cdot \tau \cdot \left(1 - \frac{u_n(t)}{V_n} \cdot \sqrt{(0.025 + \frac{u_n(t)}{V_n})}\right) \quad (2.44)$$

The third condition is applied in traffic, where vehicles move closer to each other,

and it defines the behavior of the vehicle in the back while braking. It is assumed that its driver will adapt its speed, so that a safety distance will be kept, even if the preceding vehicle suddenly stops. Thus, according to that formula, in the case where the following vehicles speed is limited by that of its preceding one:

$$u_n(t + \tau) \leq b_n \cdot \tau + \sqrt{(b_n^2 \cdot \tau^2 - b_n[2 \cdot [x_{n-1}(t) - s_{n-1} - x_n(t)] - u_n(t) \cdot \tau - \frac{u_{n-1}(t)^2}{\hat{b}}])} \quad (2.45)$$

As is apparent from the Equation 2.45, the speed of the following vehicle is affected by the drivers reaction time, the distance between the vehicles, as well as their respective velocities and braking rates. Additionally, *Gipps* (1981) notices that it is necessary to add an extra safety space θ to the drivers reaction time, which will assure the timely braking of the vehicle, in case where the drivers reaction is delayed. The safety timeframe is constant and equal to half of the reaction time ($\tau/2$) and is included in the Equation 2.45 (*Rakha et al.*, 2010). Also, the driver of the vehicle n , adapts its speed according to the maximum anticipated braking that the $n-1$ vehicles driver can induce, but is able to decelerate even more if necessary. As a result, even if the leading drivers will to break is underestimated, the model is still valid.

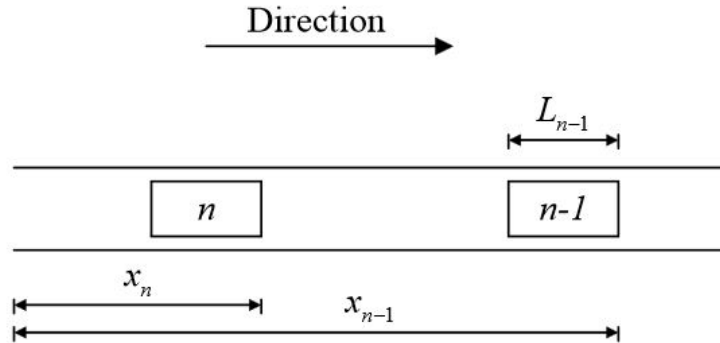


Figure 2.3: Car- following notation (*Olstam and Tapani*, 2004)

Many researchers have attempted to modify the aforementioned model; notewor-

thy are the the examinations of *Wilson* (2001) and *Rakha and Wang* (2009). However, the already described remains the most widely used. Its widespread usage is due to its clear content and definite assumptions by *Punzo et al.* (2012). A more in depth analysis of the model and its evolution until today, has been conducted by *Punzo et al.* (2012), who consider necessary the calibration of the models parameters.

Additional researches

Konishi et al. (2000) proposed a coupled map (CM) car-following model to describe the dynamical behavior of an open flow. They also introduced a decentralized delayed-feedback control scheme in the model for the suppression of traffic jams.

In 2013, A new anticipation optimal velocity model (AOVM) was proposed by considering anticipation effect on the basis of the full velocity difference model (FVDM) for car-following theory on single lane (*Peng and Cheng*, 2013).

Ge et al. (2014) proposed a control method to suppress two-lane traffic congestion. Their methodology was extended to the FVD (Full velocity difference) car-following model for two-lane traffic flow. They noted that without control signals, lane changing behaviors would break the steady states of traffic flow.

Considering high speed following on expressway or highway, an improved car-following model was developed by *Jia et al.* (2014). They introduced the parameter of “variable safety headway distance”.

From the above short sample, but also from other numerous studies, it is understood that many extended traffic models have been developed and will continue to evolve. It is therefore important to develop a comprehensive methodology that will allow quick and efficient calibration of models parameters. The simultaneous multiple parameters calibration is helpful, therefore the SPSA could be a fairly promising algorithm.

2.3.2 Large-scale simulation systems

Dynamic Traffic Assignment (DTA) models include traffic analysis tools capable of evaluating travel activities and dynamic network performance for a specific period of time, or for extended daily hours (*Chiu et al.*, 2011). The range of their possibilities is quite large and it includes off-line assessment of incident management strategies, on-line support of real-time emergency response efforts and optimization of the operation of Traffic Management Centers (TMC) through the provision of real-time predictions (*Antoniou*, 2004). More details concerning DTA models could be found in Chapter V.

DTA models' value (particularly large-scale simulation systems) depends on their ability to replicate specific conditions accurately. *Balakrishna* (2006) noted that “while advanced DTA models provide realistic abstractions of actual demand and supply processes, their outputs are governed by a large set of inputs and parameters that must be estimated before the models are applied. Well-calibrated models are therefore critical to the success of any DTA application”.

DTA models involve a large amount of parameters that need to be calibrated using actual traffic data. A successfully calibrated model is able to accurately predict traffic conditions. *Off-line calibration* typically results in the creation of a historical database that ensures the correct replication of average conditions, covering a wide range of factors such as day of the week, month, season, weather conditions and special events. Its results should be adjusted in real-time to be sensitive to the variability of traffic conditions from their average values. *On-line calibration* uses these results as a priori estimates. It requires accurate real-time predictions of traffic conditions on a given day, which are also impacted by factors such as weather, road surface conditions, traffic composition and incidents.

Parameters of simulation models that need to be calibrated could be divided into driving behavior parameters (acceleration, lane changing, and intersections models)

and travel behavior parameters (origin destination (O-D) flows, route choice model) (*Toledo et al.*, 2003).

Until today a great number of approaches have been developed and successfully applied for the systematic calibration of demand and supply components in DTA models. Most of the proposed approaches used a few selected calibration parameters because of the large total number of combinations and the complexity of the optimization. Two main categories are:

- The iterative demand-supply joint calibration
- The simultaneous demand-supply joint calibration

The first framework combines the independent demand and supply estimation into a consistent framework. It attempts to consider the interaction between the two DTA components by iteratively updating them during the calibration process. One such model was applied on a portion of the City of Calgary road network in Alberta, Canada by *Mahut et al.* (2004). The parameters in supply model, together with other global parameters, were adjusted manually to fit a set of one-hour turning counts. Hourly OD flows are estimated by matching turning movement counts at major intersections with their simulated values. *Balakrishna* (2006) noted that, such detailed counts, collected for this case study through an extensive survey, are rarely available.

2.3.2.1 Dynamic traffic assignment framework

Traffic flow phenomena are complex, nonlinear and they are discovered on free-ways all over the world. Due to the individual reactions of human drivers, vehicles do not collaborate simply by following the laws of mechanics, but rather show phenomena such as traffic breakdown, hysteresis, stop-and-go traffic and synchronized flow. Drivers in such systems need to be in constant control of their vehicle. It is crucial

to make fast, continuous decisions, relating to speed, acceleration and deceleration, route and lane choice, merging and response to information and control messages.

Congestion is increasing in many urban areas where populations and city economies are growing and it is likely to continue to increase. Transportation systems nowadays are trying to manage the urban traffic congestion with the better management and utilization of existing infrastructure on a cost-effective basis. Challenges arising from several emerging policy, real-time applications and planning, have led recent researches to focus on developing traffic modes which could explain and reproduce traffic flow phenomena. A great number of cities have placed on key points sensors, who daily collect and archive time-varying traffic data, forming in this way an integrated surveillance system of the road network of the region. Within the above context, the use of detailed simulation-based Dynamic Traffic Assignment (DTA) models has begun. DTA models include traffic analysis tools capable of evaluating travel activities and dynamic network performance for a specific period of time, or for extended daily hours (*Chiu et al.* (2011)). The range of their possibilities is quite large and it includes off-line assessment of incident management strategies, on-line support of real-time emergency response efforts and optimization of the operation of Traffic Management Centers (TMC) through the provision of real-time predictions (*Antoniou* (2004)).

Existing DTA models are generally classified into two broad categories: analytical models and simulation-based ones. The analytical models can be further categorized as mathematical programming, optimal control, or variational inequality models. Most analytical DTA models present a number of limitations, because through some simplifications that inevitably are involved, they cannot sufficiently capture the true dynamics of traffic conditions such as congestion buildup and dissipation (*Kamga et al.* (2011)).

DTA models take into account complex interactions between supply and demand

in a transportation network. They implement each time a specific set of modules at specified frequencies (time-based) or when certain events occur (event-based) (*Yang and Koutsopoulos (1996)*).

Real-time DTA systems typically comprise three main functions (*Ben-Akiva et al. (2002)*):

- Estimation of the current state of a transportation network
- Prediction of future traffic conditions
- Provision of continuous and up-to-date information to the travelers

They are designed to be an integral part of a TMC. They utilize simultaneously historical data of the traffic conditions in the network and real-time data from a surveillance system which is formatted accordingly to the requirements of the system. A detailed treatment of the demand-supply interactions within a state-of-the-art DTA system can be found in *Ben-Akiva et al. (2002)*.

For the the first phase, the basic components are (i) historical data which include time dependent Origin-Destination matrices or link travel times, (ii) a detailed description of the studied road network and (iii) real time traffic counts from a surveillance system. Through an iterative simulation of demand-supply interaction, realtime observations from the surveillance system are reproduced.

During the next phase of prediction-based information generation, consistent and unbiased traffic information for dissemination to travelers is generated. Information based on predicted network conditions (i.e. anticipatory information) is likely to be more effective than information based on current traffic conditions because it accounts for the evolution of traffic conditions over time which is what travelers will experience.

More details about the demand and supply representation and interaction for the purposes of estimation and prediction can be found in *Ben-Akiva et al. (2002)*.

2.3.2.2 History

There was always a tendency to explain the dynamics of traffic flow on a complex network. Many researchers were interested in exploring possible ways of control to alleviate traffic congestion, a phenomenon that is commonly seen in developed and developing countries. The first proposed macroscopic traffic flow model is the LWR model proposed by Lighthill, Whitham, and Richard in 1955 (*Lighthill and Whitham (1955)*). This model, and later the more sophisticated ones (LWR-extended models) could replicate phenomena such as traffic waves and breakdown. However, they display a lack for the mechanics of producing stop-and-go traffic (*Guan et al. (2012)*). Subsequently, early microscopic models, also known as following-the-leader models, were developed in the 1960s. Two decades later, the cellular automata (CA) model was presented, as another type of microscopic model.

Mathematical programming DTA models, which represent the problem in a discretized time-setting, have started to be formulated by Merchant and Nemhauser in the late 1970s. The formulation was limited to the deterministic, fixed-demand, single-destination, single-commodity, system optimal case. Extensive studies have shown that the mathematical models have significant limitations for developing deployable models for general networks. According to *Peeta and Ziliaskopoulos (2001)*, these DTA formulations tend to present many obstacles in:

- the use of link performance and/or link exit functions
- holding-back of traffic
- efficient solutions for real-time deployment in large-scale traffic networks and

- a clear understanding of solution properties for realistic problem scenarios

Within the last decades optimal control and variational inequality models have also been developed. The first category includes DTA formulations, where the O-D trip rates and the link flows are sought as continuous functions of time, and the second category consists of mechanisms which address equilibrium and equivalent optimization problems. A comprehensive review about these two categories can be found in *Peeta and Ziliaskopoulos (2001)*.

The fourth and most relevant category with the subject of this research is that of simulation-based DTA models. Studies have been carried out with the use of these specific models for more than 40 years. Considering the dynamic nature of the network under time-varying demands, simulation-based DTA models are able to compute the spatio-temporal path for every vehicle, while accounting for real-time driver behavior (*Kamga et al. (2011)*). For the implementation of these tasks, a set of vehicles and their travel paths are required. The simulator provides the link travel times, which are then used for the calculation of the time-dependent shortest paths. Taking into consideration the completed computations, the vehicles are loaded onto the network. This process is repeated for as many times as necessary, as reported by the user-specified convergence criterion.

There have been many efforts to develop simulation models for studying networks under Intelligent Transportation Systems (ITS). An important first attempt was made with INTEGRATION (*Van Aerde (1999)*), a model developed at the Queens University. It is a fully microscopic simulation model that tracks the longitudinal and lateral movements of individual vehicles to the resolution of a deci-second. Its car-following algorithm takes into consideration the individual vehicle speeds based on the macroscopic parameters of free-flow speed, speed at capacity and jam density. INTEGRATION uses up to five different driver/vehicle types and in this way, it has

the ability to represent different routing behavior or access privileges to real-time traffic conditions.

MITSIM (MIcroscopic Traffic SIMulator) has been proposed by Massachusetts Institute of Technology as a new alternative for modeling traffic flows in networks involving advanced traffic control and route guidance systems (*Yang and Koutsopoulos (1996)*). It can simulate the state of the network in detail and for the vehicle movements it uses car following, lane changing signal and event responding logic. It has been validated in a number of studies, such as in Stockholm, Sweden and in Boston. (*Ben-Akiva et al. (2010)*, *Toledo et al. (2003)*). It accepts traffic controls and routing information as input from traffic management systems and maintains the state of traffic signals and signs in the simulated network.

Another important effort to find solutions for O-D demand with fixed departure times was made by Mahmassani and Peeta using DYNASMART (Dynamic Network Assignment Simulation Model for Advanced Road Telematics), a mesoscopic traffic simulator which was designed at the Center of Transportation Research at the University of Texas. It is developed to be used as an assignment and simulation model for ITS. Traffic signals, ramp meters and incidents can be studied, using all of the tools provided by DYNASMART. It has already been used to study the core network of Austin, Texas and the network of Anaheim, California. Further development of this model has led to the development of DYNASMART-X a traffic assignment and optimization tool (*Boxill and Yu (2000)*). It performs traffic routing functions in different modes, including predictive, decentralized reactive (when local network controllers route vehicles by reacting to events such as incidents), and hybrid (a combination of the centralized and decentralized approaches) (*Balakrishna (2006)*). DYNASMART-P is a similar variant of the real-time DYNASMART-X system.

Some of the main problems that this deterministic DTA models present is the need for a-priori knowledge of O-D demands for the whole area of interest, and the

assumption of users, who respond ideally to the information provided, such as messages from the Advanced Traveler Information Systems (ATIS). Based on the above, *Mahmassani* (1993) managed to introduce more categories of users, with different classes of information, availability, information supply strategy, and driver response to the information provided.

In the mid 1990s other research studies followed which developed rolling horizon DTA models as an effort to include real-time variations in network conditions. They contributed significantly to the increase of computational efficiency for the sake of real-time tractability (*Peeta and Ziliaskopoulos* (2001)).

Ben-Akiva et al. (1997) proposed DynaMIT (Dynamic Network Assignment for the Management of Information to Travelers) as a real time dynamic traffic assignment system that provides traffic predictions and travel guidance. Utilizing information from the traffic surveillance system, it achieves the creation of user-optimal guidance which takes into account estimated network conditions and traveler response to information. A detailed description of the its methodology, will be presented in ??.

DynaMIT-R, developed for real-time applications, synthesizes estimates of current network conditions from historical information along with real-time surveillance data. Future network state, which is represented by OD predictions, is assigned using a mesoscopic supply simulator to assess network performance in the near future.

A commercial network planning tool equipped to perform iterative simulations towards a dynamic user equilibrium solution is the developed by INRO, Dynameq (Dynamic Equilibrium). Innovative microscopic traffic models and route-choice algorithms were used to identify the best paths on the network for each origin-destination pair (*Mahut et al.* (2005)).

In 2000 Ziliaskopoulos and Wallet introduced an internet-based GIS system which incorporates data and models into the same framework. In their proposed system RouteSIM is being used, a mesoscopic model which focuses on traffic distribution.

The simulation-based DTA model is able to capture many details of the real network, such as traffic signals, by using time-dependent cell capacities and saturation flow rates.

2.3.2.3 Emerging / Future perspectives

The current versions of simulation-based traffic prediction state models, such as DynaMIT, DYNASMART and DynusT, provide well-developed tools that can be readily deployed for a variety of applications. Next generation real-time DTA models will be enriched with more information such as the way that trips are generated, and the modes (e.g. public transit) are represented. Additionally, dynamic pricing of the transportation network, incorporation of commercial vehicles, and the response to anticipated personal devices as source and distributors of information will be incorporated.

Milkovits et al. (2010) proposed DynaMIT 2.0 as the next generation real-time dynamic traffic assignment system. Accurate state estimation and prediction could be achieved in multiple dimensions.

Focusing on the activity-based modeling, DynaMIT 2.0 moves from static OD-based demand to activity-based demand. Each agent can generate a daily activity plan, which consists of multiple activities, into a single trip or tour. Consequently, during a day, the agent can change an activity choice, based on the traffic conditions, without affecting previous or future trips-decisions during a day. A characteristic example could be the fact that someone may not make a shopping trip in the morning when traffic is heavy, but instead run the errand when they go for lunch. It is a significant advantage to allow higher flexibility in traveler choice, including not making the trip at all. *Bowman and Ben-Akiva* (2001) proposed a model designed to capture interactions among an individual's decisions throughout a 24 h day by explicitly representing tours and their interrelationships in an activity pattern. An interesting

example of an activity-based transportation model, developed by researchers at ETH in Zurich is MATSIM (www.matsim.org). An important detail is the fact that it does not use real-time data.

DynaMIT 2.0 will also include the parameter of multi-modal network. MILA-TRAS (Microsimulation Learning-based Approach for Transit Assignment) is a characteristic example of this criterion, as it is a high-fidelity agent-based transit choice model, using ArcGIS and Paramics, which provides the transit network and stochastic rider experience respectively. Alternative travel options, such as public transportation and non-private auto transport modes should be an integral part of supply and demand modeling (*Milkovits et al. (2010)*), because they represent different interactions with traffic, weather and passengers demand.

Personal smart-phones and/or vehicle navigators are an integral part of everyday life for the majority of people. Many emerging smart-phone applications require position information to provide location-based or context-aware services. GPS is often preferred over its alternatives such as GSM/WiFi based positioning systems because it is known to be more accurate. All these data could be used as increased traffic and mobility data, because traditional self-reported travel surveys typically suffer from problems such as limited sample size, under-reporting of total completed trips, imprecision of trip start/end times etc. (*Chen et al. (2010)*). Additionally, smart-phone could be used as a mobility advisor. Travelers could be alerted to traffic levels and incident locations. DynaMIT 2.0 will response to all these described tools, increasing with this way significantly its accuracy.

Finally, DynaMIT 2.0 will take into account the commercial vehicles, as they account for almost 10% of Vehicle Kilometer Traveled (VKT) in an urban network, and the reaction to congestion pricing, as accurate prediction is critical for setting the appropriate price points (*Milkovits et al. (2010)*).

2.3.2.4 Estimation of supply models

DTA systems have macroscopic or microscopic supply models. But especially the mesoscopic systems, such as DynaMIT, have microscopic demand models and macroscopic supply models. The second one focuses mainly on capacities and link-based speed-density models. The typical data used for this task are sensor records of at least two of the three primary traffic descriptors: speeds, flows (or counts) and densities.

Until today many methodologies of supply calibration have been developed and studied. *Leclercq* (2005) used data from arterial segments in Toulouse, France for the estimation of the four parameters of a two-part flow-density function. He worked on the optimization of the fit to observed sensor flows, with the fitted flows obtained from an aggregate relationship comprised of a parabolic "free-flow" part and a linear congested regime. *Van Aerde and Rakha* (1996) calibrated the speed-flow profiles by fitting data from loop detectors on I-4 near Orlando, Florida.

Some other researchers focused on the independent estimation of subsets of supply parameters, such as *Muñoz et al.* (2004), who analyzed a calibration methodology for a modified cell transmission model (MCTM), applied to a 14-mile westbound stretch of the I-210 freeway in southern California. Free-flow speeds, congestion-wave speeds and jam densities were determined, using a least-squares data fitting approach. *Yue and Yu* (2000) succeeded in adjusting the free-flow travel times and turning fractions to match detector count data through the calibration of the EMME/2 and QRS II models for South Missouri City, a small suburban network outside the city of Houston, TX.

Kundie (2002) introduced the implementation of SPSA (Simultaneous Perturbation Stochastic Approximation) for the approximation of the gradient of the objective function through finite differences, and the calibration of the supply models within a mesoscopic DTA system. *Balakrishna et al.* (2007), exploiting the above methodol-

ogy, presented a systematic offline DTA calibration methodology for all demand-and-supply inputs and parameters simultaneously. Optimization algorithms were implemented successfully on complex and large-scale calibration problems, using real-world sensor count and speed data from a large urban network in Los Angeles, California.

2.3.2.5 OD Estimation and Prediction

The dynamic traffic management systems include dynamic driver information and adaptive traffic control. A DTA system integrates historical data and information in order to estimate the current state of the studied network and provide prediction-based information to the travelers. Intelligent transportation systems (ITS) are being widely deployed in the context of achieving the above objectives. Fair use of advanced traffic management systems (ATMS), advanced traveler information systems (ATIS) and advanced public transportation systems (APTS) allows the maximization of capacity and efficiency of current transportation modes, minimization of delays and general improvement of system reliability. An interesting summary of studies of ATIS time savings can be found in *Toledo and Beinhaker* (2006). Study results with real-world data from Los Angeles, California show that ATIS-based routing may lead to travel times saving of up to 14%, and reduction of travel time variability by up to 50%, depending on the type of information.

The ability of full-time surveillance of the current state of a road network is supported by sophisticated sensor networks which automatically and routinely collect and archive time-varying traffic data. Data collection technologies provide the opportunity for acquiring traffic data at a variety of resolutions to match the needs for both traffic management and control applications. Traffic sensors are categorized according to their functionality as point, point-to-point and area-wide (*Antonioni et al.* (2011)). Any traffic modeling application is able to accept and process a wealth of information from different traffic data types. As a result, the accuracy of each model

could be significantly enhanced. Point to point travel times are valuable for assessing and validating travel times, while point flow measurements are beneficial for the provided accurate flow data. Video cameras, placed at various intersections can inform in collaboration with an image processing software about the number of vehicles which cross a studied area.

Recently, wireless communication infrastructures and navigation technologies have modified in a significant degree the manner by which we conceive data collection and data coverage. These technologies collect vehicle positions, infer relevant information concerning vehicular kinematic characteristics and congestion and supply updated traffic information to all drivers (*Marfia et al. (2013)*). The processed data is disseminated to users using various media, such as variable message signs (VMS), web sites or through wireless communication with the in-vehicle navigation systems.

The procedure of estimating and predicting OD flows benefits from surveillance data. Many effective and theoretically consistent methodologies using aggregate data such as traffic counts and/or OD demand counts have been proposed. The effectiveness of each application depends on the network configuration as well as the sensor infrastructure. Relevant information can yield a system which is based on video cameras placed at various intersections and software with image processing capabilities to match the plate numbers of vehicles as they cross equipped intersections. A research which utilizes the data collected from the described plate matching system was implemented by *Ma and Koutsopoulos (2008)* in Stockholm, Sweden for the on-line estimation and prediction of travel times in urban areas. **OD estimation** Traffic

state estimation task resolves the problem where the number of traffic flow variables to be estimated may be much larger than the number of variables that are directly measured. Previous work on inference for OD matrices from link count data has led to two basic directions, the static and the dynamic OD estimation. In the first case the

average OD flows are estimated taking into account average traffic flows measured in the same period over some network links. *Maier* (1983) considered using a Bayesian approach to combine prior information on an OD matrix with current observations of traffic flows on monitored links. *Cascetta* (1984) presented a generalized least squares estimator of an OD matrix. *Hazelton* (2000) has developed a methodology for estimating origin-destination matrices from link flows on uncongested network using a statistical model of the transport system with Poisson distributed OD flows. In another research *Sherali et al.* (2003) exploited shortest path network flow sub-problems for the estimation of an OD trip-table, given incomplete link flow information on the network. *Li* (2005) using the Bayesian approach estimated, reconstructed and predicted the population means of traffic flows.

More attention is being paid though to the need of determining a "dynamic" O-D matrix. Using time-varying traffic counts many researchers have tried to propose solutions to the problem of estimating time-varying OD flows or average OD flows. In the first case, OD matrices, each corresponding to a time interval, are used for various predictive network control problems such as route guidance and congestion prevention. DYNA was one of the first research projects which used time dependent matrices, for the prediction of traffic conditions in real time on part of the Dutch motorway network. The model system contained a statistical model for noisy traffic data filtering, a dynamic traffic assignment model for 15-60 min. predictions, and a real-time OD matrix prediction model, which was merged with the second one, when the time horizons overlap (*Ben-Akiva et al.* (1995)). Another important implementation was the study of *Cascetta et al.* (1993), who proposed a simultaneous and sequential estimator (namely Generalized Least Squares) utilizing the Italian Brescia-Padua motorway. It was proved that estimates without a priori information could give significant results. Additional relevant examples of the least square models could be found in the studies of *Sherali and Park* (2001) and *Lin and Chang* (2007).

The simultaneous estimation of OD flows involves the calculation, storage and inversion of a large augmented assignment matrix. Several studies (*Cascetta and Russo* (1997), *Bierlaire and Crittin* (2004)) have shown that the implementation is too computationally intensive on large networks. *Toledo et al.* (2003) proposed a sequential estimation technique, which exploits the sparse structure of the assignment matrix to successfully deal with this limitation.

The *sequential* estimator optimizes fir the unknown OD flows one interval at a time:

$$\hat{x}_h = \arg_{x_h} \min[f_1(x_h, x_h^a) + f_2(y_h, \hat{y}_h)] \quad (2.46)$$

where x_h is the current best solution; x_h^a are *a priori* flows (extracted from other studies, or set to \hat{x}_{h-1}); \hat{y}_h are the fitted counts obtained by assigning x_h to the network; $f_1(\cdot)$ and $f_2(\cdot)$ are functions that measure the distance between the estimated or fitted quantities from their *a priori* or observed values.

A linear *assignment matrix* mapping is used when a measurement equation links the OD flows x_h with the counts \mathbf{y}_h :

$$\mathbf{y}_h = \sum_{p=h-p'}^h \mathbf{a}_h^p \mathbf{x}_p + \mathbf{v}_h \quad (2.47)$$

where the elements of \mathbf{a}_h^p specify the fractions of each OD flow in \mathbf{x}_p (departing during interval \mathbf{p}) that arrive at every sensor location during interval h . \mathbf{v}_h represents the error term. The symbol p' indicates the number of necessary intervals for the longest trip on the network. It depends on the network topology and the congestion levels.

Since the sequential estimator constrains the flows in prior intervals to their best estimates, the measurement equation may be re-written as:

$$\bar{\mathbf{y}}_h = \mathbf{y}_h - \sum_{p=h-p'}^h \mathbf{a}_h^p \hat{\mathbf{x}}_p = \mathbf{a}_h^h \mathbf{x}_h + \mathbf{v}_h \quad (2.48)$$

Consistent with the GLS formulation, Equations 2.46 and 2.48 yield the following estimator:

$$\hat{\mathbf{x}}_h = \arg_{x_h} \min[(x_h - x_h^a)' \mathbf{W}_h^{-1} (\mathbf{x}_h - \mathbf{x}_h^a) + (\bar{\mathbf{y}}_h - a_h^h x_h)' \mathbf{R}_h^{-1} (\bar{\mathbf{y}}_h - a_h^h x_h)] \quad (2.49)$$

where \mathbf{W}_h^{-1} and \mathbf{R}_h^{-1} are error variance-covariance matrices. If there are no reliable estimates of them, it is proposed to set them to identity matrices of appropriate dimensions. Finally, the above optimization is constrained so that $x_h \geq 0$. **OD pre-**

diction The prediction of origin-destination flows represents an important step of better management of the current road network. This process is one of the key components of dynamic traffic assignment which will contribute to the efficient utilization of existing capacities (*Ashok and Ben-Akiva (2000)*). Since the early 1980s, several modeling approaches have been used. *Wang et al. (2006)* presented RENAISSANCE, a real-time freeway network surveillance tool, which suggests the use of Kalman Filter for the state prediction. Similar methodology followed *Liu et al. (2006)* for the prediction of urban arterial travel time. Literature has also shown that neural networks are a promising alternative for modeling and predicting traffic parameters, because almost any function can be approximated. *Vlahogianni et al. (2005, 2008)* concentrated on the development of optimized neural network models to forecast traffic flow in highly congested urban signalized arterials. Their proposed multilayer traffic pattern recognition strategy is able to identify transitional traffic conditions and generate clusters of traffic patterns with similar statistical characteristics. *Dunne and Ghosh (2011)* demonstrated the effect of data aggregation level on forecasting model per-

formance, because it has been found that it eliminates variation in data and alters most properties, including non-stationarity and nonlinearity, that exist at the disaggregated level (*Vlahogianni and Karlaftis (2011)*). More details about the Neural Network and Artificial Intelligence (AI) applications could be found in *Vlahogianni et al. (2014)*; *Adeli (2001)*).

CHAPTER III

Methodology

This chapter begins by outlining the various model inputs and parameters. Furthermore, the SPSA algorithm is thoroughly analyzed. Some methods of SPSA's performance evaluation are presented.

3.1 Calibration variables

Most of the problems that need to be calibrated are highly non-linear. The desired outcome is usually a result of multiple algorithms' and models' processes. For a simple static assignment problem, i.e. the calibration of the speed-density relationship::

$$v = v_{max}[1 - (\frac{k - k_{min}}{k_{jam}})^{\beta}]^{\alpha} \quad (3.1)$$

the calibrated variables, that will be included in the vector θ are: (i) v_{max} = the speed on the segment under free-flow traffic conditions, (ii) k_{min} = the minimum density beyond which free-flow conditions begin to break down, (iii) k_{jam} = the jam density, and (iv) α, β = segment-specific coefficients. A-priori estimates of the parameter values are necessary and could be obtained by fitting the speed-density relationship to an initial set of data using non-linear least squares (*Antonioni et al.*,

2007b).

For a DTA model the variables collection is more complicated. The critical set of data may be grouped into demand and supply-side parameters. Travel time, fraction of freeway links, number of left turns, number of signalized intersections and the frequency of freeway-arterial changes are some coefficients that represent the basic demand parameters in a route-choice model (*Antoniou*, 2004). The supply variables for microscopic models are more complex and may include car-following and lane changing parameters, thus explaining individual driver decisions and maneuvers. For mesoscopic and macroscopic models, link / segment capacities and speed-density relationships are more likely to be included.

3.2 Summary display of SPSA

The SPSA algorithm is a member of the iterative stochastic optimization algorithms family. When the objective function has not an analytical form, the SPSA is an ideal solution for optimization problems. It iteratively traces a sequence of parameter estimates that converge the objective functions gradient to zero.

The following step-by-step summary display how SPSA iteratively produces a sequence of estimates:

1. The process is initialized ($i = 0$) so that $\theta^i = \theta^0$, a K-dimensional vector of apriori values. Additionally, SPSA algorithm's non-negative coefficients α , A , c , a and γ are picked according to the characteristics of the problem. Some guidance on picking these coefficients in a practically effective manner is provided by Spall (*Spall*, 1998a).
2. In this step, the number of gradient replications **grad_rep** for obtaining the average gradient estimate at θ^i is chosen.

3. The iteration counter is increased by one unit and the step sizes α^i and c^i are calculated as $\alpha^i = a/(A + k + i)^\alpha$ and $c^i = c/(k + i)^\gamma$.
4. A K-dimensional vector Δ^i of independent random perturbations is generated by Monte Carlo. Each element $\Delta_{ik}, k = 1, 2, \dots, K$, is drawn from a probability distribution that is symmetrically distributed about zero, and satisfies the conditions that both $|\Delta_{ik}|$ and $E|\Delta_{ik}^{-1}|$ are bounded above by constants. A simple choice for each component of Δ_{ik} , is to use the Bernoulli ± 1 with equal probability. Note that the inverse moment condition above precludes the use of the uniform or normal distributions.
5. The loss function is evaluated at two points, by obtaining two measurements based on the simultaneous perturbation on "either side" of θ^i . These points correspond to $\theta^{i+} = \theta^i + c^i \Delta_i$ and $\theta^{i-} = \theta^i - c^i \Delta_i$. Each point is checked if it is between the lower and upper bound constraints before function evaluation.
6. The K-dimensional gradient vector is approximated as

$$\hat{g}(\theta^i) = \frac{z(\theta^{i+}) - z(\theta^{i-})}{2c^i} \begin{bmatrix} \Delta_{i1}^{-1} \\ \Delta_{i2}^{-1} \\ \vdots \\ \Delta_{iK}^{-1} \end{bmatrix} \quad (3.2)$$

7. Steps 4 to 6 are repeated **grad_rep** times, using *independent* Δ_i draws, and an average gradient vector for θ^i is computed.
8. Through the application of Equation 2.38, an updated solution point θ^{i+1} is obtained.

9. Iteration or termination of the algorithm, depending on the convergence. Convergence is declared when θ^i and the corresponding function value $z(\theta^i)$ stabilize across several iterations.

SPSA provides a huge saving of computational time due to its constant number of perturbations for the gradient approximation. SPSA performs as well as FDSA (Figure 3.1). With no or little noise, FDSA is expected to follow the true descent to the optimal. SPSA may have approximated gradients that differ from the true gradients, but they are almost unbiased.

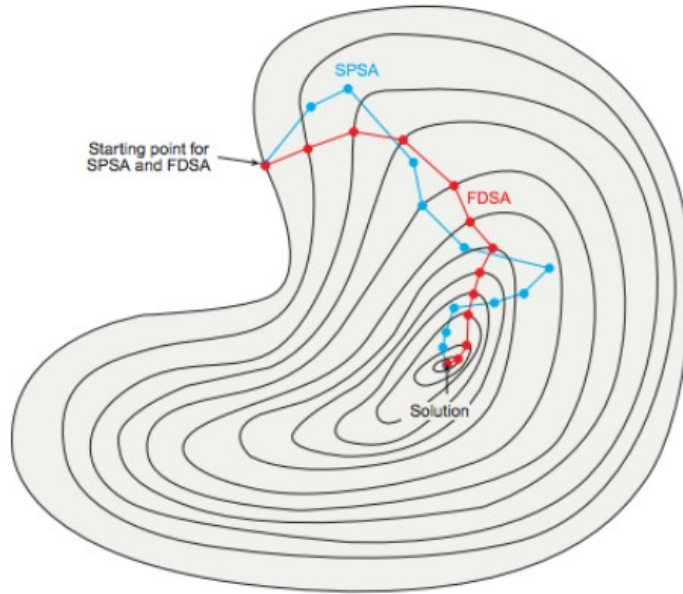


Figure 3.1: Relative paths of SPSA and FDSA in a 2 parameter problem

SPSA and its variations have been already applied in the field of DTA calibration. Dynamic emission models were calibrated using SPSA by *Huang* (2010). The researcher used a microscopic traffic simulator and the aggregate estimation ARTEMIS as a standard reference. A demonstration of SPSA's performance was also presented by Paz et al. (2012). They calibrated simultaneously all the parameters that COR-SIM models use.

3.3 Choice of non-negative coefficients

The performance of SPSA depends on the choice of the gain sequences to a considerable extent. There are great chances that the method will not converge to the optimal solution, if a wrong combination of system parameters has been assigned. In general, SPSA can easily be entrapped in a local minimum, and not depart from it in order to approach the optimal solution.

Spall (1998a) highlights some details, that someone should take into account for the efficient implementation of SPSA. Firstly, for poor quality measurements of $L(\theta)$, it is necessary to choose a smaller α and c , than in a low-noise setting. Giving to parameter α a value less than “1”, usually yields better finite-sample performance through maintaining a larger step size. However, *Balakrishna* (2006) noted, that if a too large value is chosen, then the SPSA may overlook a nearby solution and venture too far away. The parameter c should be set at a level approximately equal to the standard deviation of the measurement noise in $y(\theta)$ in order to keep the p elements of $\hat{g}_k(\hat{\theta}_k)$ from getting excessively large in magnitude. A large c may lead parameters components to their bounds really fast, thus rendering the gradient approximations invalid (*Balakrishna*, 2006). It is also noted in *Spall* (1998a), that the values of α and A can be chosen together to ensure that the algorithm will perform effectively.

3.4 Objective Function

Calibration involves the use of an objective function to determine the model cost (model-data errors). It is the function of differences between observations and simulations. Through an optimization procedure it is minimized or maximized, depending on the requirements of the specific application (*Legates and McCabe*, 1999). Some of the objective functions that were used for model calibration in hydrology are the mean square error, the absolute mean/maximum error, residual bias and the Nash

objective function (*Boyle et al.*, 2000; *Diskin and Simon*, 1977; *Gupta et al.*, 1998; *Servat and Dezetter*, 1991; *Yapo et al.*, 1998).

According to *Wu and Liu* (2014), the sum of square errors (SSR) is the most commonly used objective function for a variety of optimization processes even in recent years. It emphasizes the extreme values of a set of observation data and neglects the low values during model calibration, because squaring calculation usually means a relatively larger weight for peak or higher values. The correlation-based measures characterized by “square error” such as square correlation coefficient (r^2) and Nash-Sutcliffe Efficiency (NSE) are oversensitive to extreme values (outliers) and insensitive to additive and proportional differences between observations and simulations.

3.5 Validation process

Different guidelines, protocols and papers provide recommendations for calibration method validation, including the linearity assessment. Validation of the model is an evaluation of the performance of the model calibration obtained through a comparison of the results from the calibrated model and the observations/ measurements in real traffic conditions (*Schneeberger*, 2002) or “verification that model behavior accurately represents the real world system modeled” (*Daamen et al.*, 2014). The majority of them describe the Ordinary Least Squares Method (OLSM) as the statistical method to be used. Fitting a calibration function by OLSM requires several assumptions related to the residuals (normality, homoscedasticity and independency) and to the model. The linear first-order model for OLSM is described in detail in *de Souza and Junqueira* (2005).

The OLSM has the inconvenience of being very sensitive to the presence of outliers and/or high-leverage points. If only a few calibration points are available, a plot of the residuals would reveal a trend, if any is present.

The validation of the procedure is achieved through the following tests for residuals

assumptions (*de Souza and Junqueira, 2005*):

- Normality (Ryan-Joiner test)
- Homoscedasticity (Brown-Forsythe test)
- Independency (Durbin-Watson test)

The Root Mean Square Error (RMSE) and Root Mean Square Normalized error (RMSN) statistics are also some useful tools for the analysis of the performance of the various calibration estimators (*Balakrishna, 2006; Antoniou, 2004*). They measure the performance of the estimators in replicating the observed data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^S (y_i - \hat{y}_i)^2}{S}} \quad (3.3)$$

$$RMSN = \frac{\sqrt{S \sum_{i=1}^S (y_i - \hat{y}_i)^2}}{\sum_{i=1}^S y_i} \quad (3.4)$$

where y_i is the i^{th} “true” parameter value, and \hat{y}_i the estimated quantity.

CHAPTER IV

Case Studies

This chapter describes tree small case studies, which were implemented for the understanding of SPSA's algorithm behavior. The choice of non-negative coefficients is examined in detail. In each application, all results are presented and evaluated.

4.1 Objectives

The primary objectives of these case studies are to:

- operationalize SPSA's calibration procedure, as it was described in Chapter III
- evaluate SPSA's performance, depending on the choice of non-negative coefficients
- record potential problems / SPSA's failures
- suggest solutions to the problems that have been occurred

4.2 A simple static assignment problem

In the context of understanding how SPSA works, as well as how it is affected by the values of non-negative coefficients, the algorithm was applied to a problem of stochastic transportation network analysis. Based on *Ozguven and Ozbay* (2008)

research, a simple network with three links between one origin and one destination node was used (Figure 5.1). Each link has a predefined travel time function:

$$t(\theta_1) = 10(1 + 0.15(\frac{\theta_1}{200})^4) \quad (4.1)$$

$$t(\theta_2) = 20(1 + 0.15(\frac{\theta_2}{400})^4) \quad (4.2)$$

$$t(\theta_3) = 25(1 + 0.15(\frac{\theta_3}{300})^4) \quad (4.3)$$

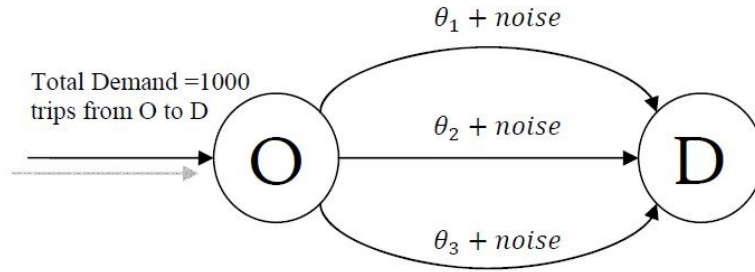


Figure 4.1: Network for Stochastic Approximation Analysis (*Ozguven and Ozbay, 2008*)

This simple network was chosen, because the deterministic optimal solution was known ($\theta^* = [358, 465, 177]$), considering as total demand 1000 trips from O to D. Knowing, therefore, the equilibrium values of the system, and in order to understand how fast the optimization algorithm could perform, a normally random noise was injected into the optimum solution. The amount of random perturbations is assumed to follow a uniform distribution with $\pm 30\%$ range from the true equilibrium values. The aim of the stochastic optimization algorithm is the minimization of the loss function, which in this specific implementation is represented by the equation:

$$y(\theta \pm c_k * \Delta_k) = t_i(\theta_{i,c}) - t_i(\theta \pm c_k * \Delta_k) \quad (4.4)$$

where t_i represents the travel time function, $\theta_{i,c}$ the already known optimal solution and $t_i(\theta \pm c_k * \Delta_k)$ the randomly perturbed measurements of $t_i(\bullet)$.

The root mean square normalized error (RMSN) was used to assess the performance of SPSA in replicating the initial correct data:

$$RMSN = \frac{\sqrt{S \sum_{i=1}^S (y_i - \hat{y}_i)^2}}{\sum_{i=1}^S y_i} \quad (4.5)$$

where y_i is the i^{th} observed measurement (in this case, the "true" parameter value), and \hat{y}_i the corresponding simulated (in this case estimated) quantity.

4.2.1 SPSA run following Spall's guidelines

Following the guidelines of *Ozguven and Ozbay* (2008), the performance is initially analyzed using the parameters $A = 20$, $\alpha = 0.602$, $c = 1$, $a = 0.027$ and $\gamma = 0.101$. The algorithm does not require complex and time-consuming calculations, so it was chosen to implement a total of 10,000 iterations, in order to depict SPSA's behavior. Figure 4.2 shows how fast the algorithm approximates the optimal solution. It can be easily observed that in the first two series of iterations, its speed is higher. However, it is not actually successful. More iterations reduce the distance between the noisy and the equilibrium values of the system as Table 4.1 shows, but SPSA's speed is extremely low.

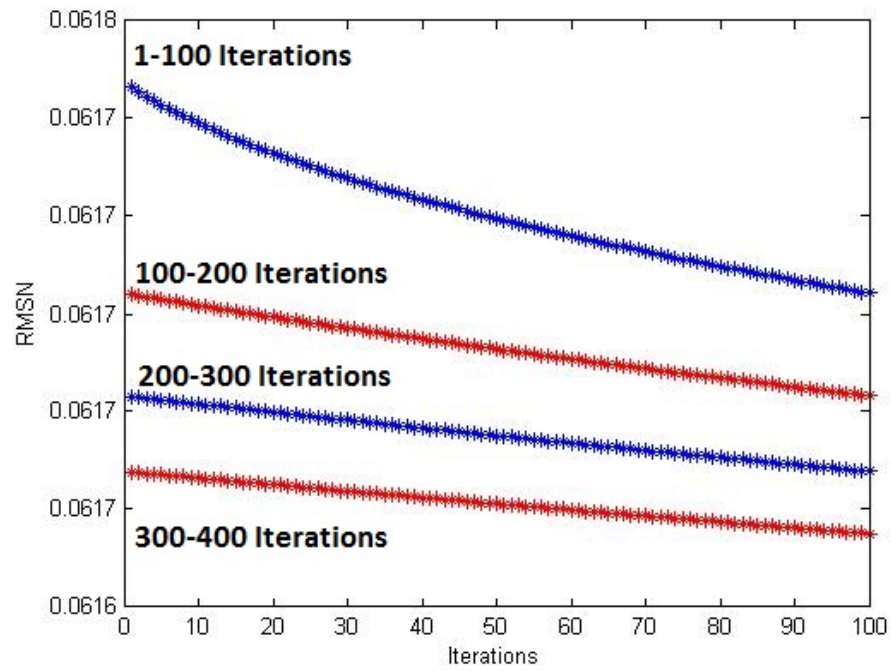


Figure 4.2: RMSN values using the guidelines of *Ozguven and Ozbay* (2008)

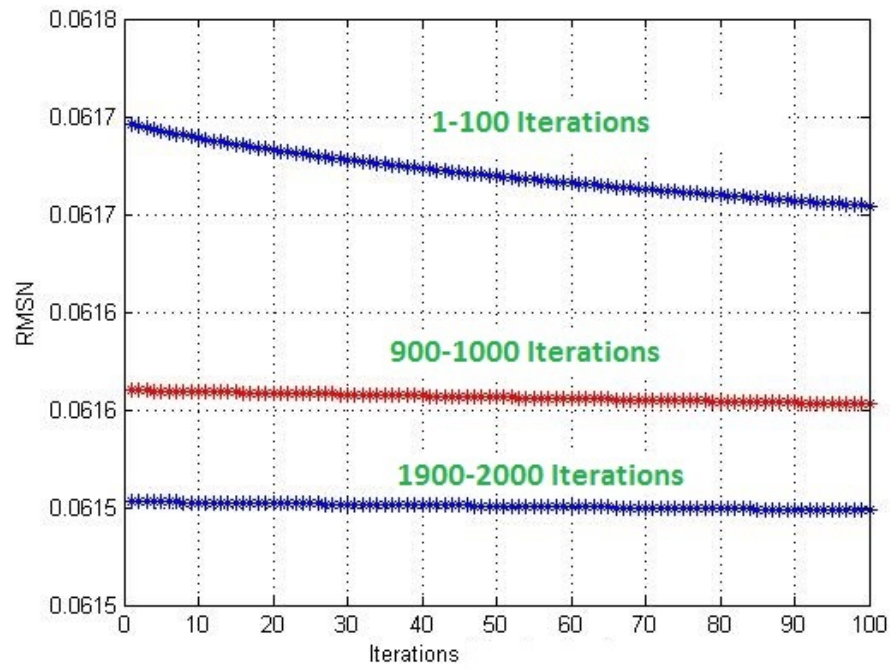


Figure 4.3: RMSN values using the guidelines of *Ozguven and Ozbay* (2008)

no.Iterations	RMSN	Improvement (%)
1	0.06175	-
100	0.06170	0.07
200	0.06168	0.10
300	0.06167	0.13
400	0.06166	0.15
1000	0.06160	0.23
2000	0.06155	0.32
4000	0.06148	0.44
10000	0.06135	0.65
100000	0.06071	1.67

Table 4.1: Summary of first implementations

In the context of reducing the errors of SPSA's algorithm, the optimisation problem was a bit changed. More specifically, the parameters of θ were reduced to two, replacing the third parameter with the equation:

$$\theta_3 = 1000 - \theta_1 - \theta_2 \quad (4.6)$$

Under the new circumstances, running again the algorithm seems to give much better results (see Figure 4.4). The algorithm approximates faster the optimum solution. In 1000 iterations, the algorithm gives results that the previous format had not presented even at 10000 iterations. This improvement appears to be due to the fact that the true value of the third parameter is considerably smaller than the other two, therefore, the algorithm is not able to perturb correctly all at once.

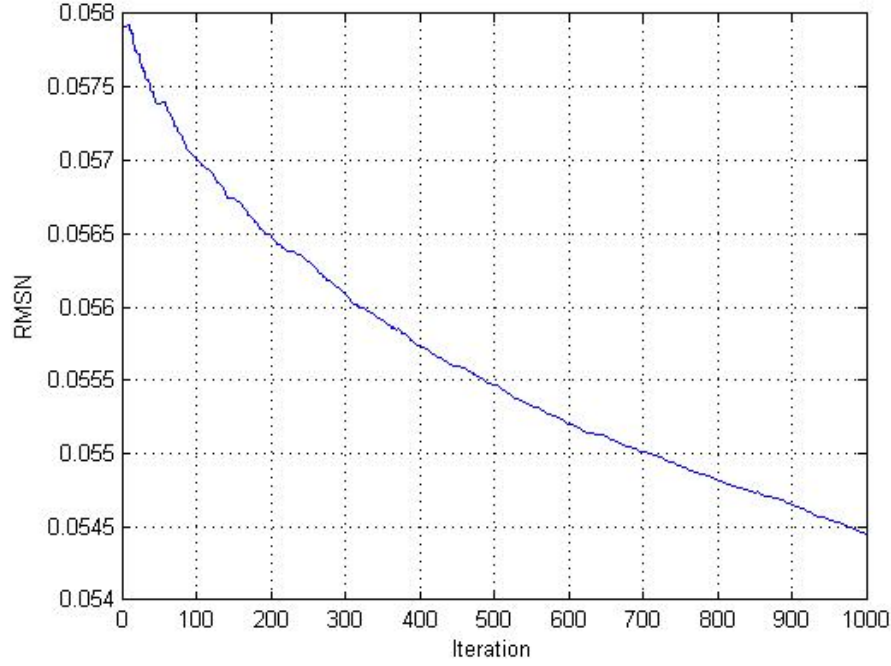


Figure 4.4: New RMSN values using the guidelines of *Ozguven and Ozbay* (2008)

The initial RMSN of the previous execution was small, therefore some small changes in SPSA's speed may not be depicted. For more distinct differences between the solutions that the SPSA gives, the initial values were changed to $\theta^* = [200, 300]$.

4.2.2 Sensitivity analysis of parameter c

As it is already mentioned, the parameter c plays a significant role in the speed of approaching the optimal solution. The SPSA algorithm were executed changing its value (specifically from $c=0.01$ to $c=10$ with a step=0.5). The number of iterations was reduced to 1000, because it is generally accepted that the SPSA algorithm is supposed to be able to approach the optimum solution in less than 100 iterations. Every 1000 iterations the c parameter changes and the algorithm starts over from the initial values of θ . The most characteristic results are presented in Figure 4.5, where

it was attempted to illustrate the curves that correspond to the 6 different values of the parameter one next to the other. The curves look quite similar; therefore, it seems that the parameter in this application does not optimize the result.

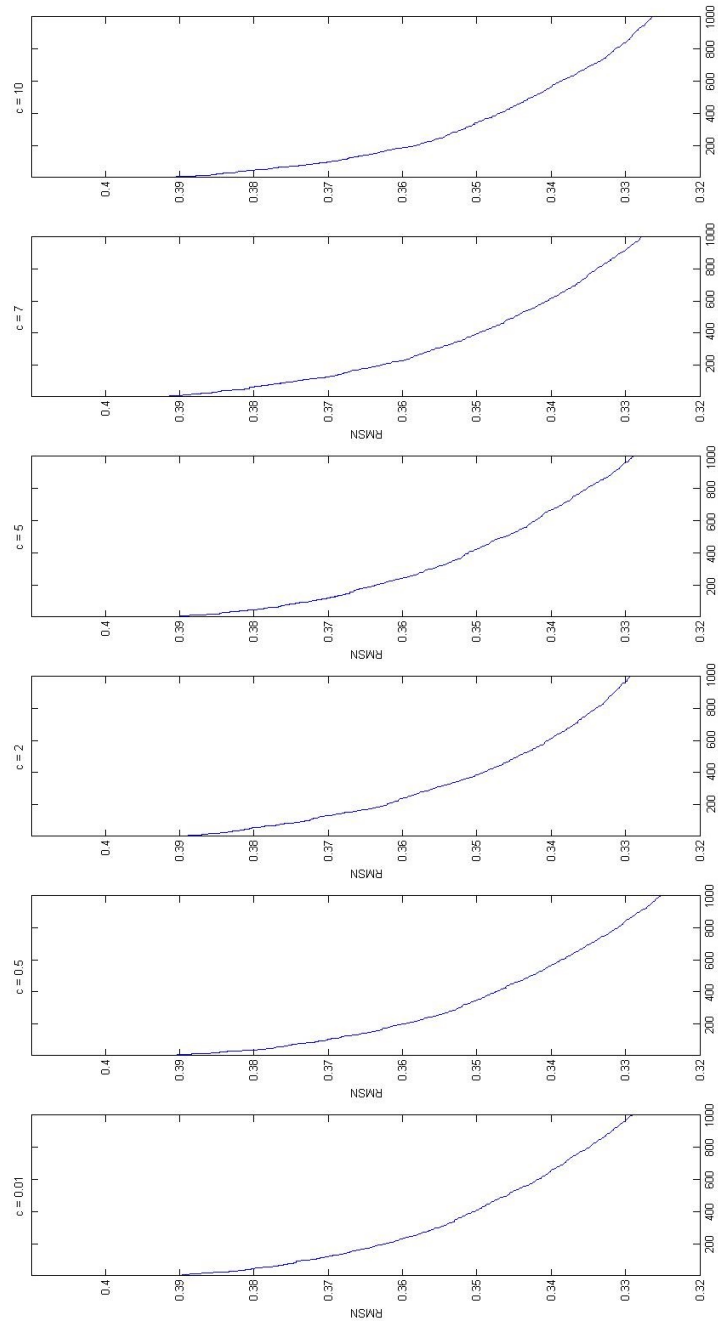


Figure 4.5: RMSN values using different values of c

4.2.3 Sensitivity analysis of parameter a

Advancing the process of finding the best values of SPSA's implementation, the same procedure was repeated, testing this time the parameter a. The range of values was selected to be among [0.01 - 10.0]. The results are presented in Figure 4.7. From this new diagram, it is evident, how much the performance of the algorithm can be affected by the input values. For the same number of iterations ($n=1000$), the RMSN can decrease either fast or slow. Setting "a" between 8.00 and 15.00 could give the best possible performance. The algorithm needs less than 40 iterations to reach $\text{RMSN}=0.01$. Table 4.2 presents the range of improvement which could be accomplished, as well as the iterations that SPSA needs for a final $\text{RMSN}=0.01$. It is noticed that for a value greater than 15, the results start to get worse.

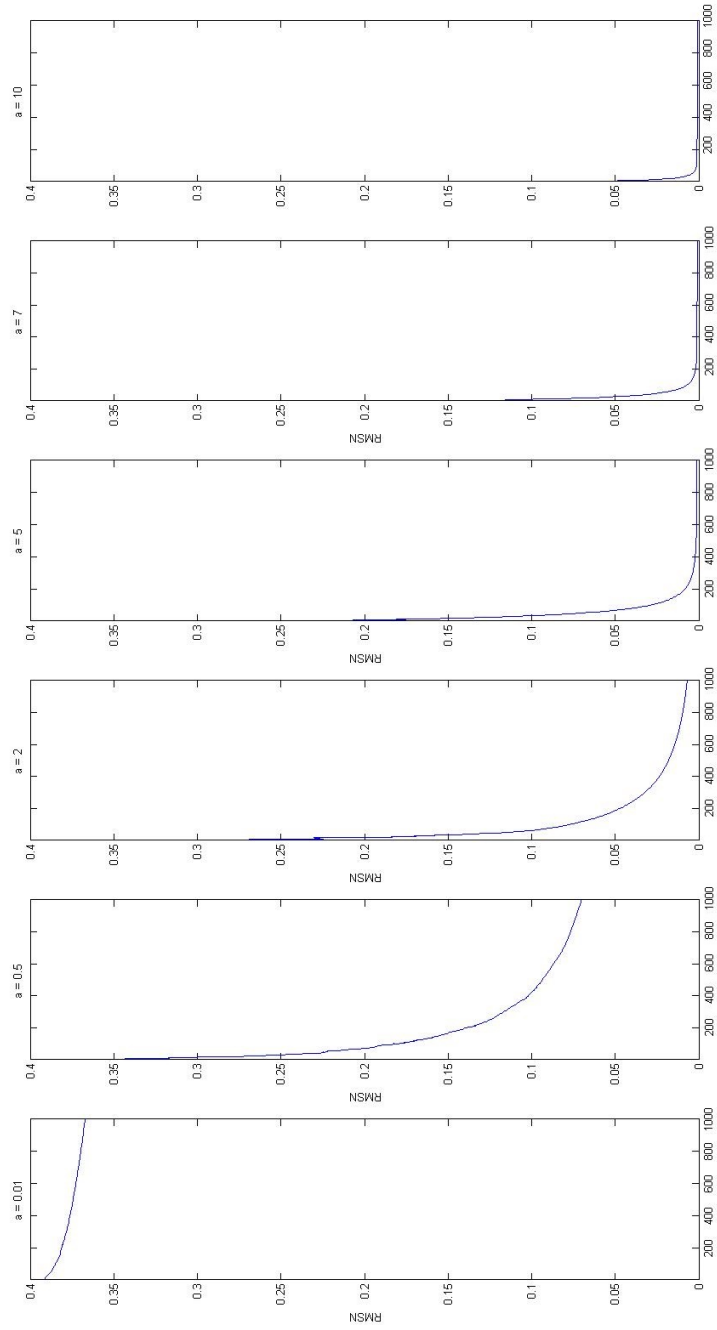


Figure 4.6: RMSN values using different values of a

Value of a	RMSN	No. Iterations until RMSN=0.01
0.01	0,3664	More than 1000
1.00	0.0334	More than 1000
2.00	0.0056	953
3.00	0.0019	384
4.00	0.0012	224
5.00	0.0010	100
6.00	0.0010	78
7.00	0.0010	86
8.00	0.0010	35
9.00	0.0010	32
10.00	0.0010	28
15.00	0.0010	9
20.00	0.0010	21
30.00	0.41	More than 1000
50.00	0.44	More than 1000

Table 4.2: Summary of first implementations

The first experiment was repeated, giving this time to the parameter “a” the value of 10. The results are shown in Figure 4.7. Once again, the curves look similar and the parameter c does not seem to affect SPSA’s convergence to smaller RMSN.

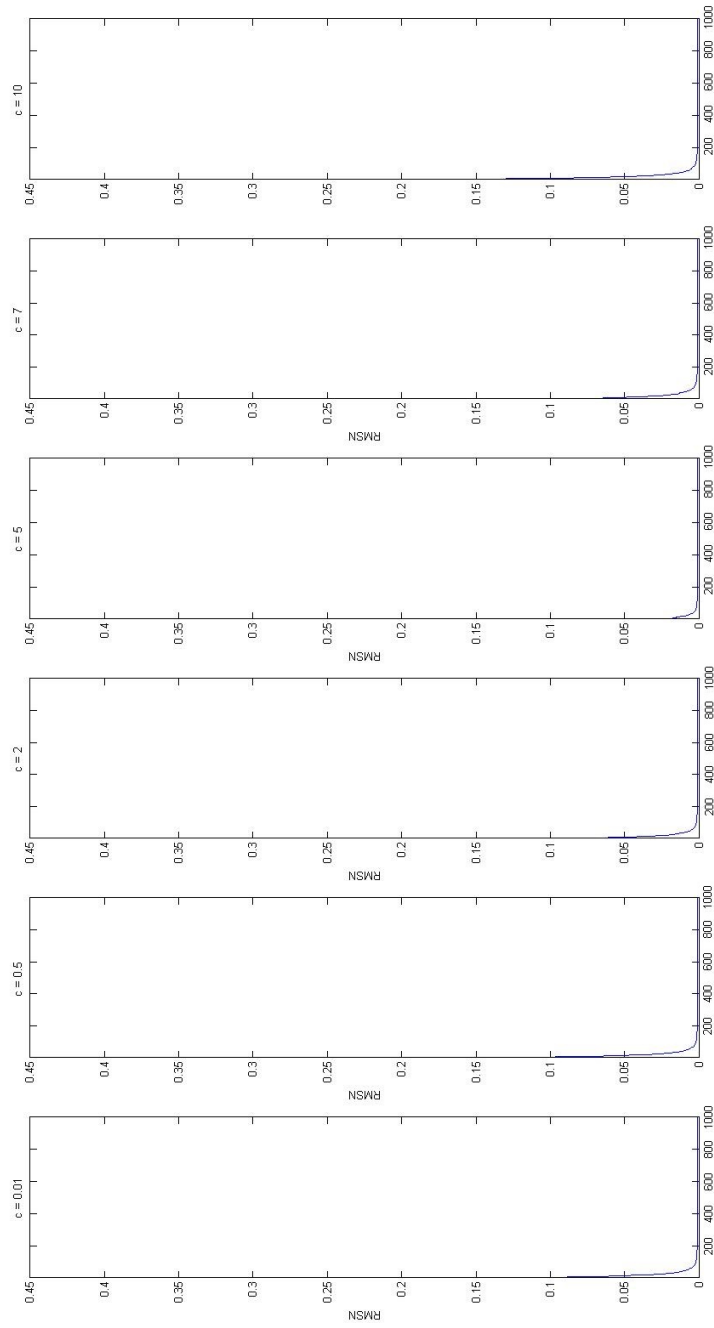


Figure 4.7: RMSN values using different values of c and $a=10$

Sometimes, the increase of iterations leads to the increase of the final error of the algorithm. If the only stopping criterion is the maximum allowed number of iterations, then the implementation may fail. It is important to use convergence criteria which will check the relative difference between the consecutive iterates obtained. That is:

$$\left| \hat{\theta}_{k+1} - \hat{\theta}_k \right| < \epsilon \quad (4.7)$$

where ϵ is a small predefined number.

4.2.4 Sensitivity analysis of parameter A

For different values of A, $a=10$, $c=1$ (keeping the guidelines of *Ozguven and Ozbay* (2008)) and for 1000 iterations, the SPSA algorithm presents the results shown in Figure 4.8 and in Figure 4.9. It seems that larger values of A delay the optimal solution approximation. This result also supports the previous observations on the variable a . Smaller A leads to SPSA's failure, because $\alpha^i = a/(A + k + i)^\alpha$ becomes really high for the necessary perturbation, and the algorithm cannot return close to the optimal solution. From the Table 4.4 it is summarized that the value $A=10$ could be chosen as the optimal.

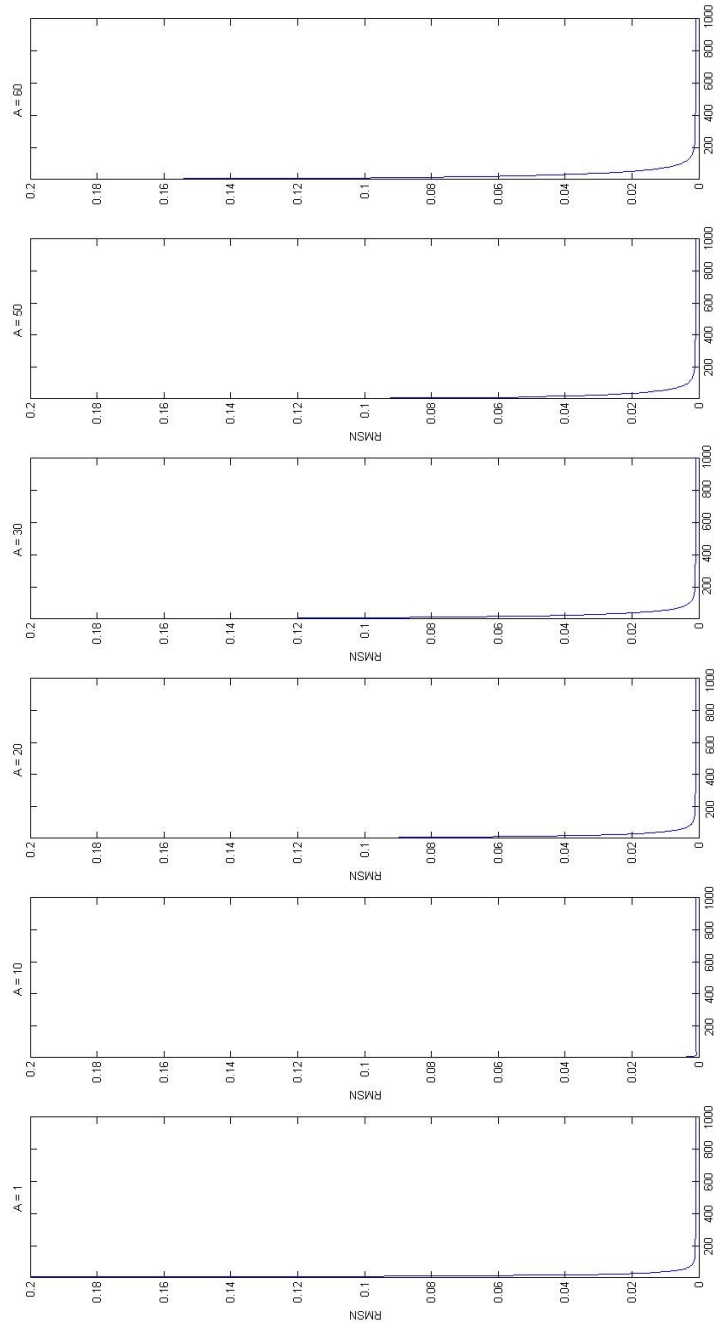


Figure 4.8: RMSN for different values of A , $c=1$ and $a=10$

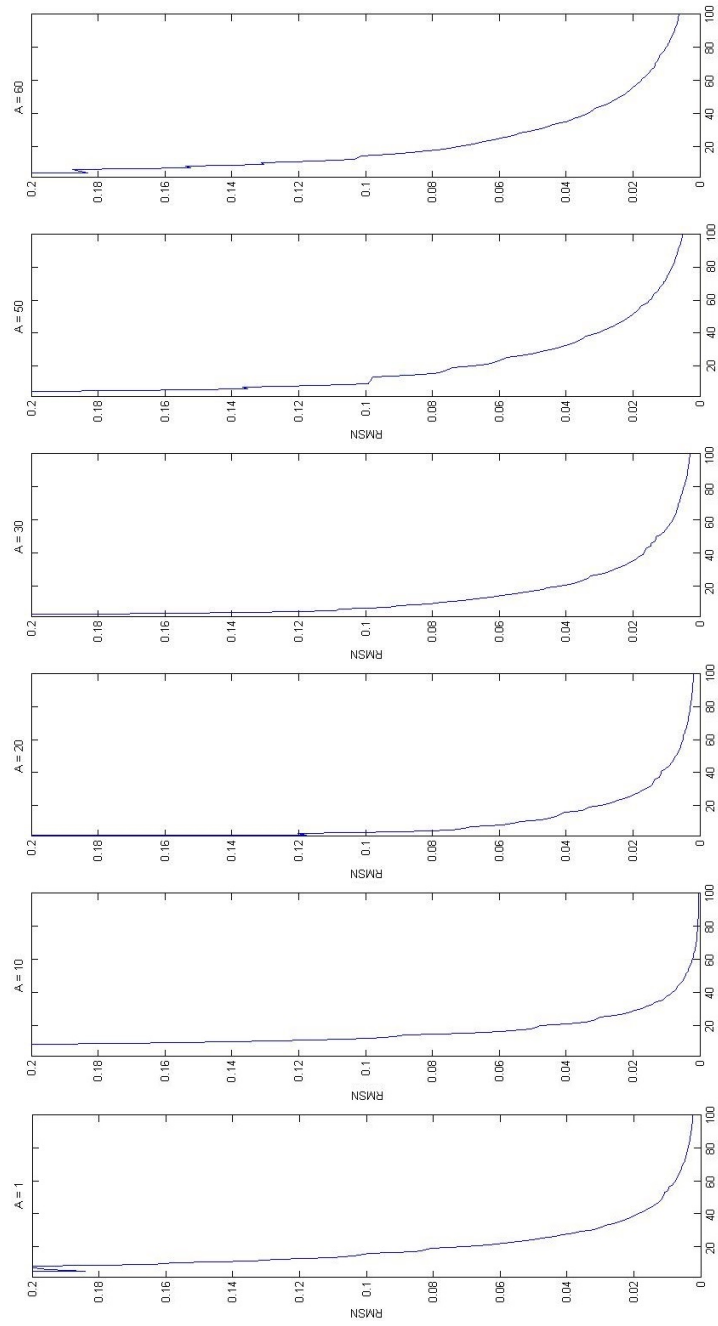


Figure 4.9: RMSN for different values of A , $c=1$ and $a=10$

Value of A	Final RMSN	No. Iterations until RMSN=0.01
1	0.001	40
3	0.001	28
5	0.001	28
8	0.001	37
10	0.001	3
20	0.001	44
30	0.001	57
40	0.001	85
50	0.001	57
60	0.001	76

Table 4.3: Summary of implementations (for different A)

4.2.5 Sensitivity analysis of parameter gamma

The parameter gamma appear to also affect significantly the final RMSN value (see Figure 4.10). Table 4.4 presents in detail all sets of SPSA's implementations. Values close to 0.01 and 0.07 tend to increase algorithm's speed. Therefore, gamma = 0.01 could be chosen for the following SPSA implementations.

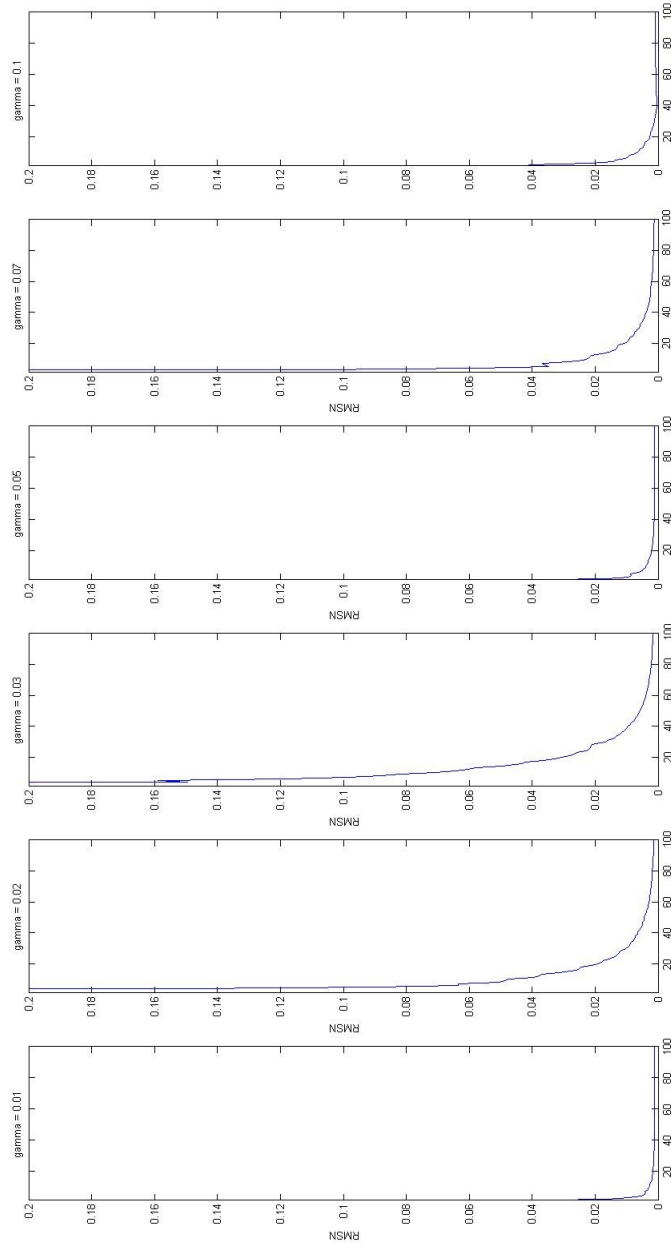


Figure 4.10: RMSN for different values of gamma, $A=10$, $c=1$ and $a=10$

Value of gamma	Final RMSN	No. Iterations until RMSN=0.01
0.001	0.001	23
0.01	0.001	5
0.02	0.002	40
0.03	0.001	30
0.04	0.001	24
0.05	0.001	14
0.06	0.001	19
0.07	0.001	3
0.08	0.001	16
0.09	0.001	29
0.1	0.002	42

Table 4.4: Summary of implementations (for different gamma)

4.2.6 Sensitivity analysis of parameter alpha

Finally, parameter alpha's behavior is studied. After a few more iterations, it is concluded that some values of alpha tend to decrease SPSA's speed to the optimal solution (see Figure 4.11). More specifically, values closer to 0.1 or 1 tend to find it approach with difficulty the optimal solution in the early iterations. Furthermore, it is observed that very small values of alpha, lead to SPSA's failure.

It is therefore clear that the algorithm can give completely different results, depending on the values given to the variables. Each one affects differently the final RMSN values, so it is important to know their behavior, before SPSA's implementation.

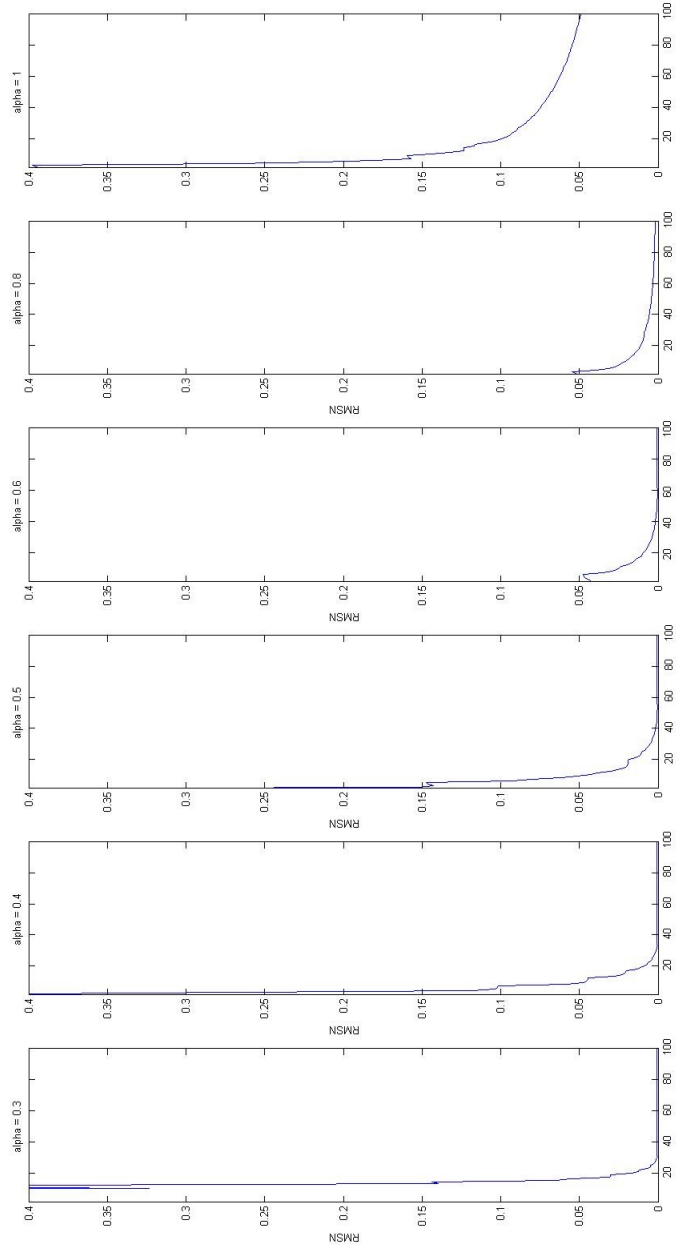


Figure 4.11: RMSN for different values of α , $\gamma=0.01$, $A=10$, $c=1$ and $a=10$

A simultaneous sensitivity analysis of all three important parameters was also implemented, in order to see which combination of values tends to give the best results in 100 iterations. It would be interesting to see, if the simultaneous perturbations will yield the same conclusions about their values. The most significant results are summarized in Table 4.5.

a	A	alpha	gamma	No. Iterations until RMSN=0.01
13	35	0.3	0.07	17
17	45	0.4	0.01	10
9	15	0.3	0.01	15
13	15	0.4	0.09	17
9	25	0.3	0.03	20
7	25	0.3	0.05	3
19	45	0.4	0.09	12
9	15	0.5	0.01	3
15	45	0.3	0.09	15
11	35	0.3	0.05	16
19	45	0.4	0.07	30
9	35	0.3	0.07	3
11	35	0.4	0.01	5
13	15	0.4	0.09	17

Table 4.5: Experiments changing “a”, “A” “gamma” and “alpha” parameters

4.2.7 Final set of parameters

It is clear that the algorithm behaves differently when the parameters’ values change. It seems that, as long as the value of the parameter “a” remains close to “9”, the chances of the SPSA’s “success” increase. The parameter “gamma” should also remain close to 0.01, and alpha close to 0.3. Using the values a=9, A=10 gamma=0.01 and alpha=0.3 as a set of parameters that gives a really good RMSN, the final diagram is shown in Figure 4.12). After only 9 iterations, the SPSA gives RMSN less than 0.01. The new final set of parameters improves vastly the algorithm’s performance, something obvious when it is also compared with the research of *Ozguven and Ozbay* (2008).

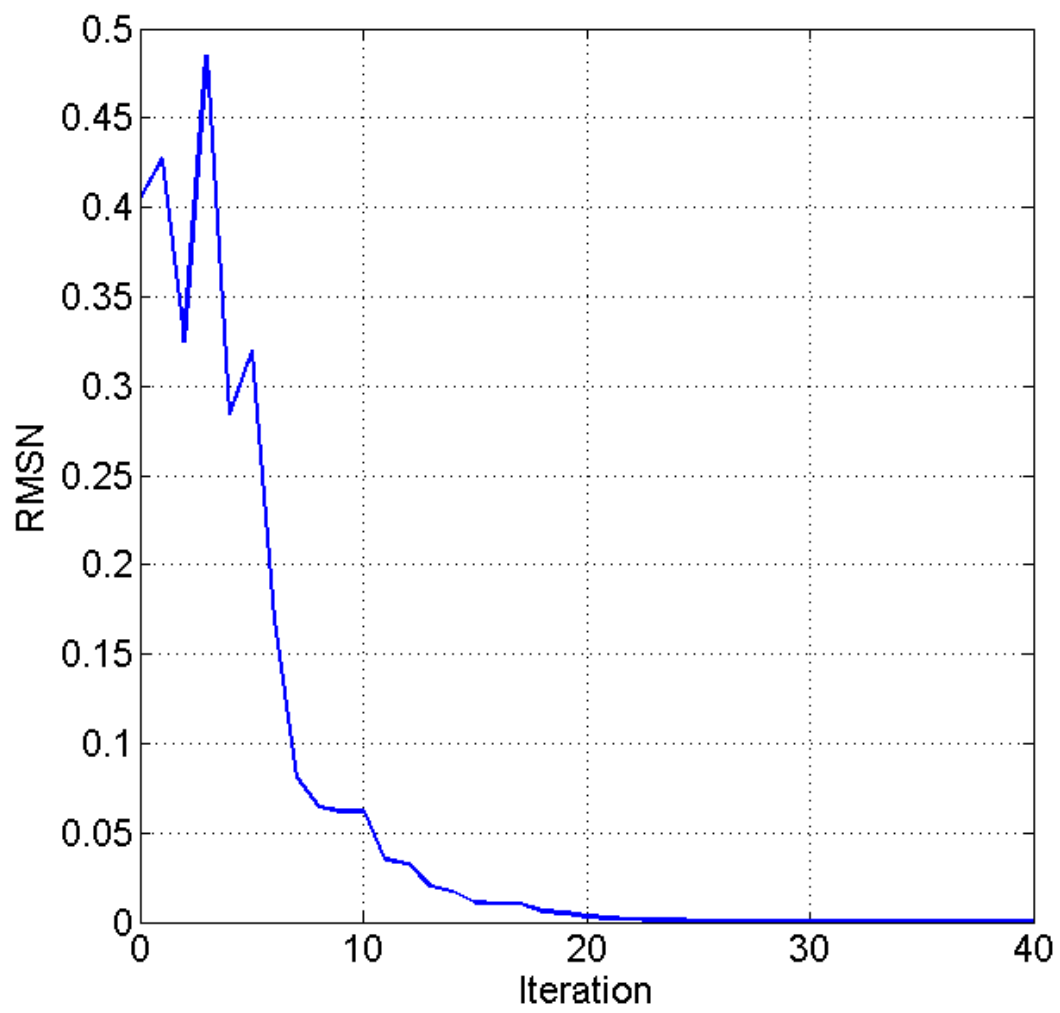


Figure 4.12: Final results of RMSN

4.3 Another experiment using a car-following model

A car following model can determine a vehicle's acceleration or deceleration rate as a function of the speed and relative position of the preceding vehicle. It is based on the idea that each driver controls a car under the stimuli from the preceding car, which can be expressed by the function of headway distance or the relative velocity of two successive cars. More details about car following models could be found in Section 2.3.

4.3.1 Transmodeler's Car-Following model overview

The car following acceleration, in the context of this application, is computed by the following formula:

$$A_i^\pm[t + \Delta t] = a^\pm \frac{V_i^{\beta^\pm}[t]}{D_{i,j-1}^{\gamma^\pm}[t]} (V_{j-1}[t] - V_j[t]) + \varepsilon_j^{CF} \quad (4.8)$$

where:

$A_i^\pm[t + \Delta t]$ = Acceleration rate of vehicle

$V_j[t]$ = Speed of the subject vehicle

$V_{j-1}[t]$ = Speed of the front vehicle

$D_{i,j-1}^{\gamma^\pm}[t]$ = Distance between the subject and front vehicles

$a^\pm, \beta^\pm, \gamma^\pm$ Model parameters

ε_j^{CF} Vehicle-specific error term for the car-following regime

The superscripts $^\pm$ indicate that the calculated acceleration could be positive (+, acceleration) or negative (-, deceleration). If the speed of the subject vehicle $V_j[t]$ is less the speed of the front vehicle $V_{j-1}[t]$ the acceleration rate will be positive (i.e. subject will accelerate). Otherwise, it will be negative (i.e. subject will decelerate).

The above car-following model is provided by Transmodeler, a versatile traffic simulator that simulates a wide variety of facility types, including mixed urban and freeway networks.

4.3.2 Model calibration using SPSA

For the calibration of the described car-following model, data from a series of experiments were used. They were conducted in the streets surrounding the city of Naples, Italy; they represent real traffic conditions in October 2002. All the necessary data were collected from 4 vehicles, which were moved in series under different traffic conditions. Noteworthy is that the vehicles were moving in streets with one lane in each direction. Therefore, the behavior of the drivers is not affected by lane changes. All vehicles were equipped with GPS receivers that tracked the location of each vehicle per 0.1 s. Specifically, they were equipped with dual-frequency devices GPS + GLONASS with horizontal accuracy $10 \text{ mm} + 1.0 \text{ ppm}$ and elevation accuracy $15 \text{ mm} + 1.0 \text{ ppm}$.

Due to environmental conditions, there were gaps in the above data, i.e. for some intervals of the experiment there were no recorded measurements. However, for the purpose of this research, it was preferred to cut the whole data package in smaller pieces, in order to have files with continuous actual measurements. The usage of some linear or polynomial interpolation method for the evaluation of missing measurements was not preferred. A more detailed description of the available data could be found in *Punzo et al.* (2005).

The data packages include location records of each vehicle (coordinates x , y , z). Using the above information the distances between vehicles, the distance travelled per 0.1s for each vehicle, and their respective speeds were calculated. More details concerning the necessary calculations and limitations could be found in *Papathanasopoulou* (2012).

The size and speed ranges of each data package are shown in Table 4.6 and in Figure 4.13.

a/a	No. Observations	Duration of measurements (s)
1695	1695	169.4
C621	621	62.0
A358	358	35.7
A172	172	17.1
C168	168	16.7
C171	171	17.0

Table 4.6: Characteristics of data packages

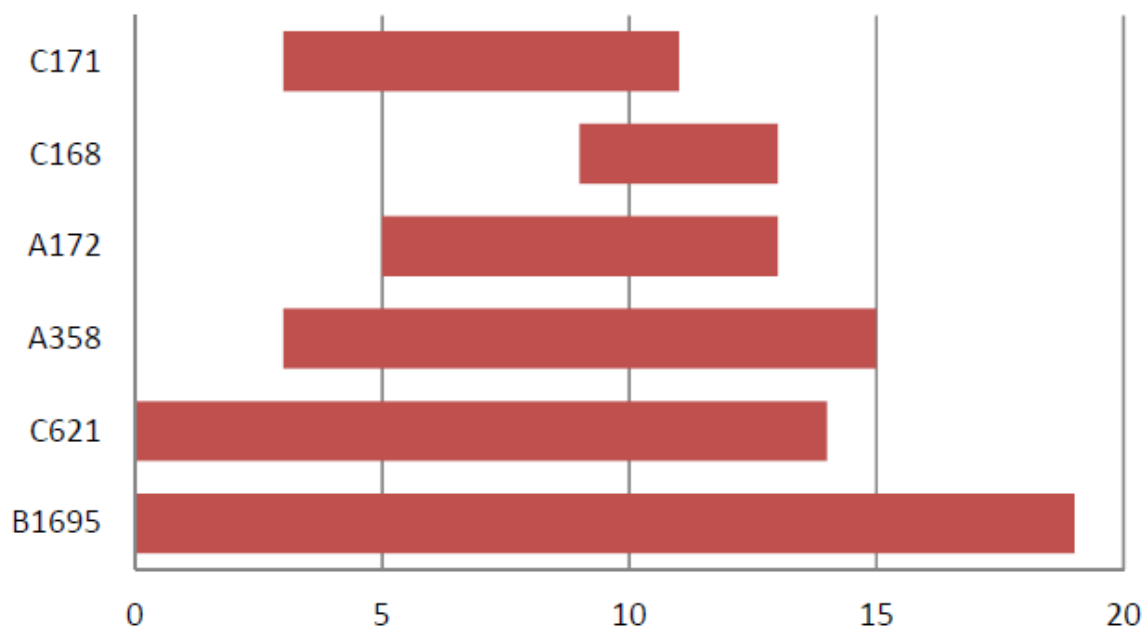


Figure 4.13: Range of speeds (m/s) (*Papathanasopoulou, 2012*)

For the experiment, the file 1695 was used, because it is the most extensive data file and it includes a wider range of velocities. With this choice, the creation of a more representative model could be accomplished. Following the guidelines of the previous experiment (see Section 4.2), SPSA's performance is initially analyzed using the basic parameter values, i.e. $A = 15$, $\alpha = 0.3$, $c = 1$, $a = 9$ and $\gamma = 0.01$. The algorithm does not require complex and time-consuming calculations, so it is chosen to implement a total of 1000 iterations, in order to illustrate SPSA's behavior.

As is known, there is no picture of the correct parameter values which will ultimately yield the correct acceleration results for the particular place and time. As a result, the car following model parameters **alpha**, **beta**, **gamma** took as initial values the default ones from the Transmodeler simulator (see Figure 4.14).

Acceleration - Car-Following			
Car Following Model Parameters			
	Alpha	Beta	Gamma
Accelerating	2.81	-1.67	-0.89
Decelerating	4.65	1.08	1.65

Figure 4.14: Initial values of car following model parameters

After several executions of SPSA, it was found that the SPSA algorithm is not able to find the optimal car-following model parameters for this particular set of measurements. The RMSN values increase exponentially. Probably, this is due to the fact that, the values of the vector *theta* are quite small, and the a_k and c_k coefficients of the algorithm influence more than necessary the requested vector.

The experiment was repeated, initializing the SPSA with smaller values. The basic parameters A , α , c , a and γ were divided by 100 in order to adjust their order of magnitude with the data of this application. The new results were again not satisfactory. The RMSN was still within an unacceptable range of values (greater than 1).

From several executions of SPSA algorithm, it was also noticed that the results could be completely different from time to time. This phenomenon occurs due to the stochasticity of the SPSA algorithm.

After several efforts to improve the existing results, it was decided to limit the vector parameters of θ from three to one. The new θ will include only the parameter **alpha**, which will once again be set at the default value from the Transmodeler simulator. The parameter **alpha** was chosen, because it is not an exponent, and it

could be managed more easily at the first steps of car-following model calibration.

Additionally, for the full understanding of SPSA's final results, the **alpha** values that fit the available observations were calculated ($RMSN = 0$). In Figure 4.15 the histogram of values' range is presented.

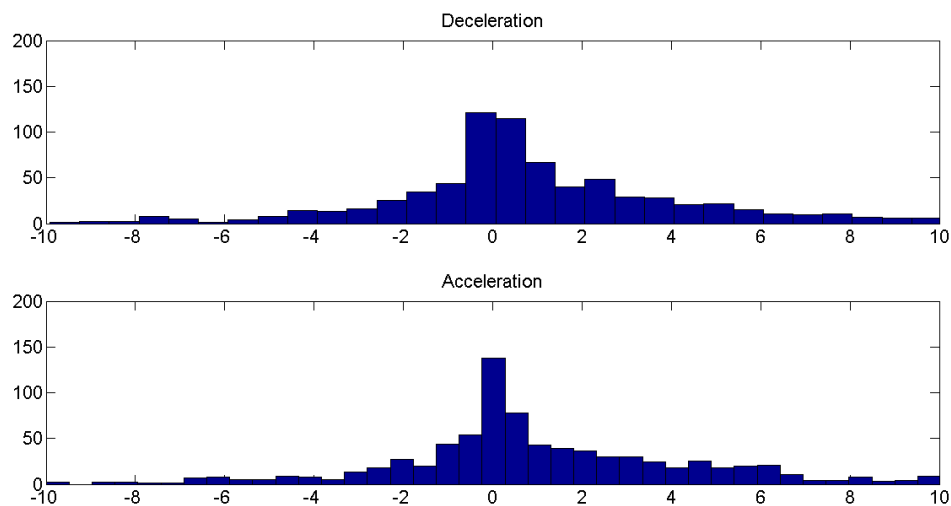


Figure 4.15: Fitted values of Car-Following model's parameter "alpha"

It is obvious that SPSA would never be able to converge to a certain value, as the histogram displays a wide range of fitted values per observation.

Therefore, it was chosen to implement SPSA per observation. The algorithm successfully managed to reach the optimal solution - the fitted value of **alpha**. In Diagram 4.16 the total number of iterations per acceleration value are presented. It is shown, that for positive acceleration values, the algorithm needs more time to reach acceptable values of RMSN. Diagram 4.17 shows how many sets of 500 Iterations the SPSA needs to terminate. The larger percentage of observations requires only 1 set.

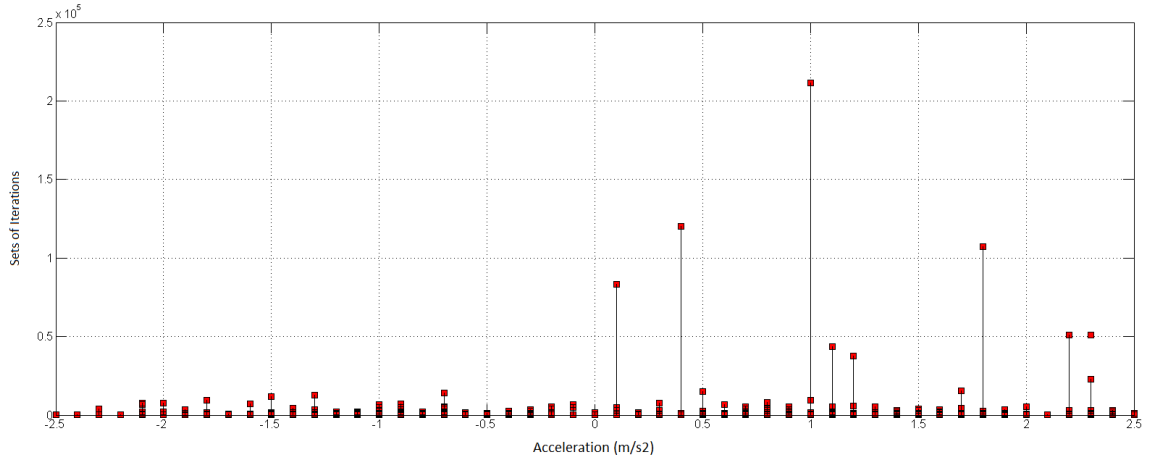


Figure 4.16: Necessary sets of iterations per acceleration value

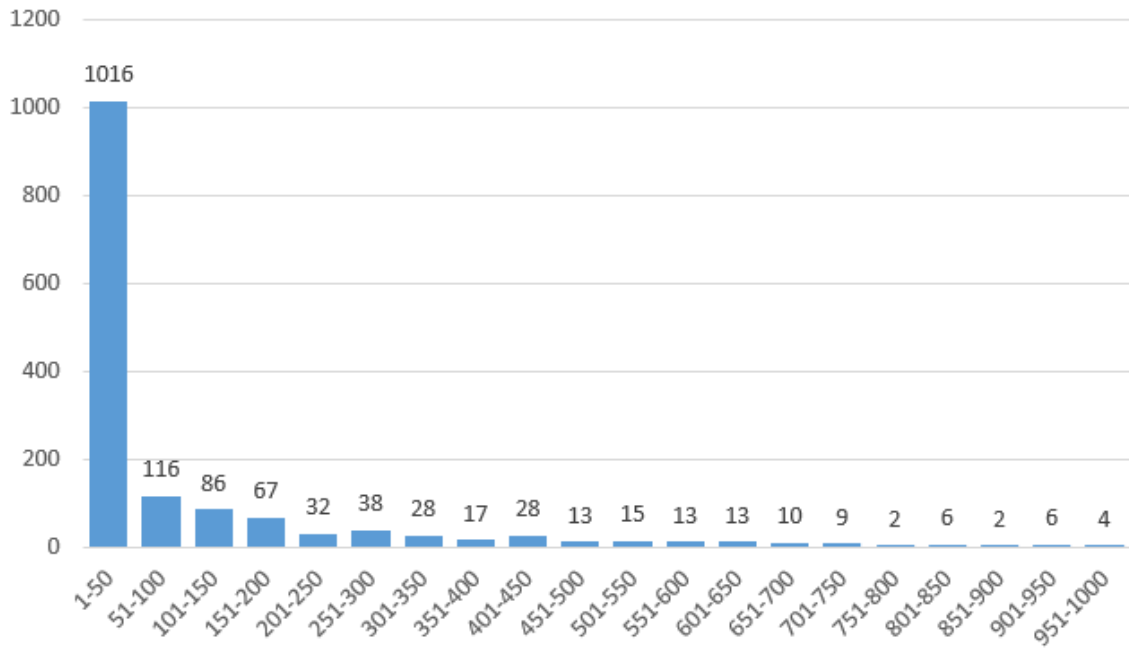


Figure 4.17: No. of necessary iterations sets for SPSA termination

The overall picture shows that the algorithm behaves correctly. The parameters seem to affect significantly the final result, therefore the correct choice of their values is important.

CHAPTER V

Towards complete DTA calibration

In this chapter, the on-line calibration approach using the DynaMIT DTA system is demonstrated. The experimental setup and design are described in detail.

5.1 Objectives

As it is already mentioned in Chapter II, Dynamic traffic assignment models try to represent traffic realism and human behavior. They depart from the standard static assignment assumptions to deal with time-varying flows. However, none of them can present a universal solution for general networks, because many ill-behaved system properties still exist.

Traffic conditions are estimated and subsequently predicted utilizing data from diverse resources. The required parameters for these procedures are calibrated off-line, generating in this way a library, from which the most appropriate can be selected and used. Off-line calibrations can be repeated several times, incorporating every time new archived surveillance information and recent operational experience.

On-line calibration is applied in order to steer the model parameters to values closer to the true ones. It is a systematic procedure whose purpose is to obtain those model parameter values that will minimize the discrepancy between the observed measurements and their simulated counterparts (when these parameters are used as

inputs to the models) (*Antoniou, 2004*).

The objective of this chapter is to demonstrate the on-line calibration approach using the DynaMIT DTA system. The experimental setup and design are described in detail.

5.2 Model and parameters

DynaMIT (Dynamic network assignment for the Management of Information to Travelers) is one of the DTA applications which has been implemented and tested in Traffic Management Centers (TMCs). It is a real-time computer system for traffic estimation, prediction and generation of traveler information and route guidance. It is aimed at supporting the operation of Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS).

DynaMIT combines a flexible microscopic demand simulator and a mesoscopic supply simulator to effectively capture complex demand and supply processes and their interactions. It estimates the current state of the network using both historical and real-time information, and generates prediction-based information for a given time horizon.

During the first phase, it provides estimations in terms of O-D flows, link flows, queues, speeds, densities and travel times. Two main models are used:

- A demand simulator that combines user behavior models and departure time choice with real-time OD estimation.
- A network state estimator that simulates driver decisions and collects information about the resulting traffic conditions

The above models interact with each other in order to provide demand and network state estimates that are congruent and utilize the most recent information available

from the surveillance system. Individual travel demand decisions, which in majority occur before the trip begins, include origin, destination, departure time and route. A disaggregate representation of demand that accounts for the individual's socioeconomic characteristics and access to information is used.

The prediction-based guidance phase consists of several interacting components:

- pre-trip demand simulation,
- OD flow prediction,
- network state prediction, and
- guidance generation to steer drivers towards optimal decisions.

DynaMIT leverages the best available knowledge of the current and anticipated network conditions, in order to provide the required information to the drivers. It has also the ability to trade-off level of detail (or resolution) and computational practicability, without compromising the integrity of its output.

Some of DynaMIT's important features are:

- Demand simulation using a micro-simulator. Analyzing the information provided by the ATIS, DynaMIT generates individual travelers and simulates their choices (whether to travel or not, and if yes, which would be their route choice and mode).
- Supply simulation using a mesoscopic traffic simulator. The development and dissipation of queues, spillbacks, and congestion in general are represented in a time-based environment. The level of detail is determined by the choice of time steps and the level of aggregation of vehicles into homogeneous packets.
- Simulation of demand-supply interactions.
- Adaptability to diverse ATIS requirements.

- Distinguish between informed and uninformed drivers.
- Treatment of real time scenarios such as incidents, special events, weather conditions, highway construction activities, fluctuations in demand, etc.
- Integration with the MITSIMLab microscopic traffic simulator. This cooperation allows offline evaluation and calibration

More detailed discussions about the features, framework and implementation of DynaMIT can be found at *Ben-Akiva et al.* (1997, 1998, 2011).

5.3 Experimental setup

The inputs of the on-line calibration module are:

A priori values of the model parameters

A priori information is available usually from the output of the off-line calibration implementation. More specifically:

- Time-dependent OD matrices;
- Segment capacities;
- Parameters of the speed-density relationship;
- and error covariance matrices and autoregressive fractions

could be available for the transition to the next phase.

Historical information of the transportation system

It is a set of information, which is not often modified, in relation to dynamic surveillance information. It reflects the foundation for the development of a trans-

portation model and it may include the network geometry, traffic control settings and layout of the surveillance system.

Surveillance data

It captures the prevailing traffic conditions. The accuracy and precision of this information is very important, because all the calibration procedure is based on the identification of the estimated (and predicted) conditions and these surveillance measurements.

5.3.1 Network description

For the on-line calibration procedure, the expressway network of Singapore will be used (Figure 5.1). The expressway system includes expressway links, on-ramps and off-ramps. There are in total 831 nodes connected by 1040 links. Each link is made up of several segments based on the geometry of the database.

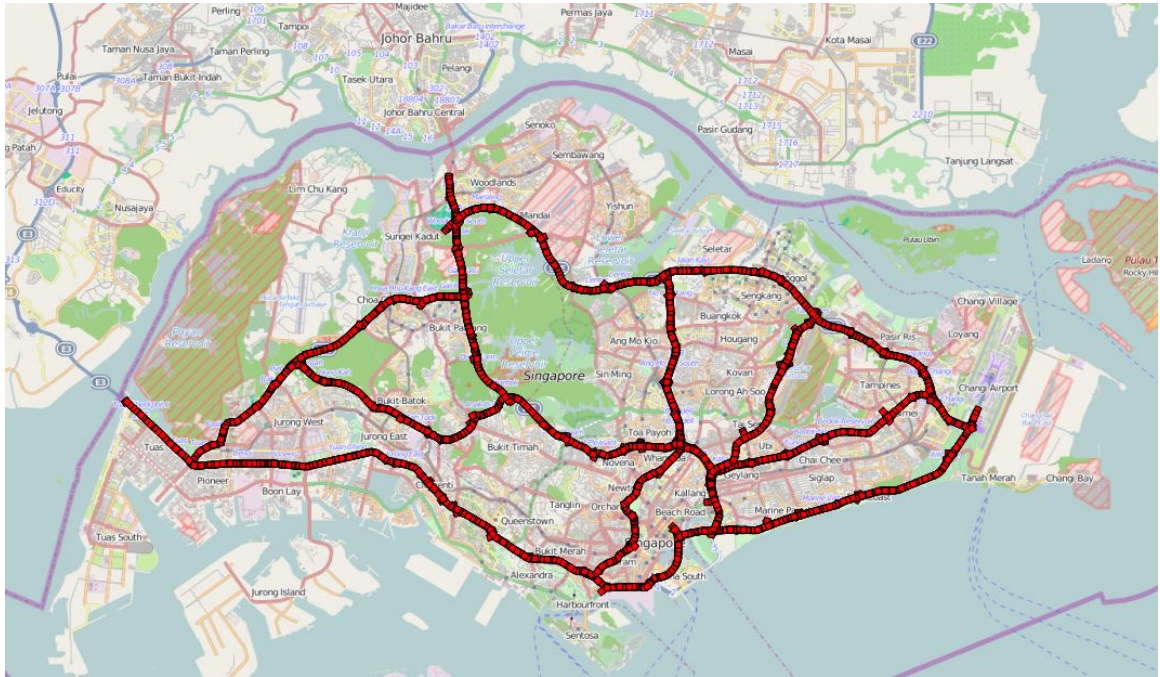


Figure 5.1: Singapore Expressway Network (OpenStreetMap)

5.3.2 Supply parameters' determination

The supply parameters were generated according to a script been written by SMART Future Urban Mobility IRG's researchers. More specifically:

- For short segments (i) the minimum speed of the segment is set to to it's free flow speed and (ii) the output capacity of the segment is set to $\text{JamDensity} * \text{Free-Flow Speed} * \text{No. of lanes}$
- For normal segments the minimum speed is calculated using speed density function with the density to be equal to jam density. If the resulting minimum speed is less than 10km/h, the minimum speed will be set to 10km/h.

The script presents the basic principles of the required parameters calculation. In certain expressways some values were changed again manually, as they have certain peculiarities, that a script can not capture.

5.3.3 Speed and flow data

Speed and flow data are provided by the Land Transport Authority of Singapore (LTA), in real time, for use in DynaMIT. The real time data feed are sent into the DynaMIT system every 5 minutes for on-line calibration purposes.

More specifically, flow data is provided by The EMAS system (Expressway Monitoring and Advisory System). All the necessary information is obtained from fixed cameras (in total 338 "sensors") which are mounted on street lamps at distances of approximately 500 to 1000 meters (Figure 5.2). The coordinates of the sensor locations were also provided. These sensor locations have been mapped to Aimsun segments. The final result was extracted and then imported in DynaMIT.

Speed data is provided by the LTA for each segment on the expressway. A total number of 3388 segments in the DynaMIT network are constantly updated. The provided data is derived from probe vehicles equipped with GPS. The exact method for



Figure 5.2: Example of cameras that measures traffic flow on Singapore expressways (photo taken by Stephen Robinson)

estimating the speed is proprietary and the details have not been published. However, it is known that taxis provide the bulk of the raw data used to make the calculations (Lu, 2014).

5.4 Data cleaning

The final achievable accuracy of the calibration process is highly dependable on the quality of the provided data. The calibrated parameters would wrongly fit to inconsistent data, and the real world traffic conditions will not be successfully predicted. Therefore, the evaluation of the quality of data is an important procedure before starting the off-line and on-line calibration process.

Lu (2014) implemented an initial calibration using 338 sensors. An inconsistency check gave also a comprehensive state of the network. It was noticed that there were

a lot of data inconsistencies introduced by incidents, as well as several malfunctioning sensors. Therefore, an additional check of network and sensors' status was necessary before the implementation and validation of new methodologies.

LTA provided SMART-FM IRG researchers with updated sensor data. A new off-line calibration of the expressway network gave really poor results (see Figure 5.3). LTA's data appears not compatible with the corresponding ones that DynaMIT provides.

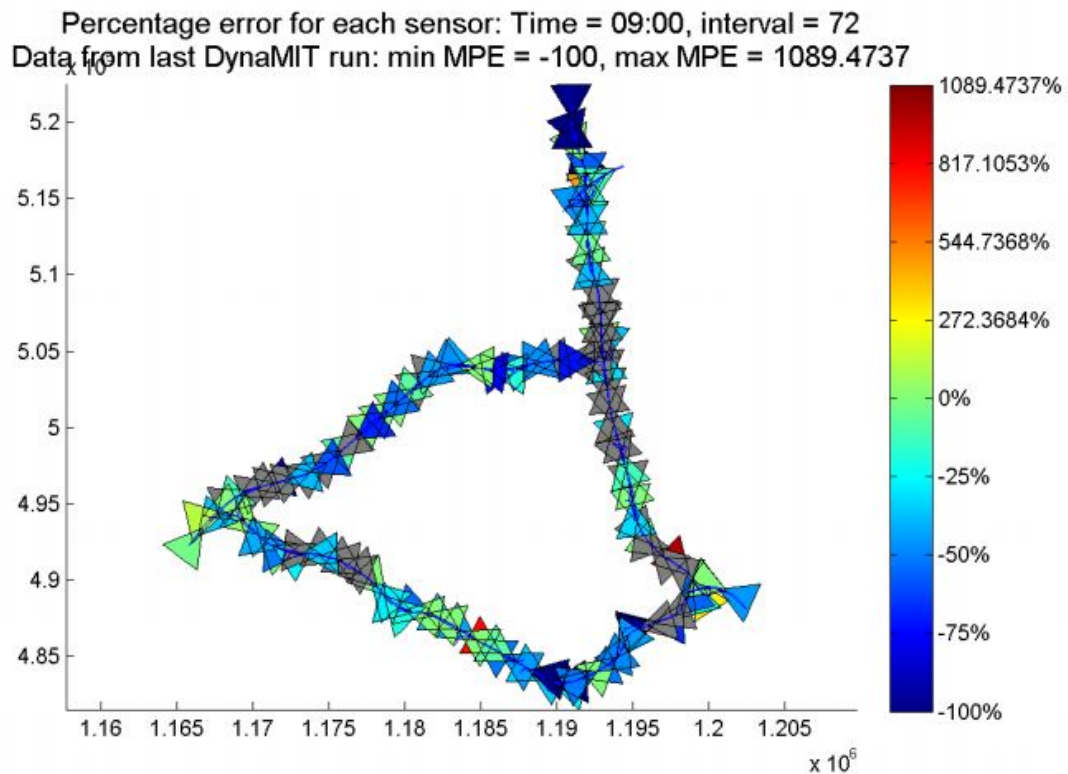


Figure 5.3: Map showing performance of each detector (Percentage Error)

It is therefore quite possible that there are errors:

- in LTA data
- in DynaMIT input files

Further checks were carried out, which were focused mainly on the structure of the

network and on the sensors location. Utilizing already used network files of Singapore network by Aimsun, a detailed inspection of the following was conducted:

- Check of segments and links associations
- Check of traffic stream on on-ramps and off-ramps
- Check of sensors coordinates using Google Street View
- Check of shortest paths

The last check could show if there is a miss-join of segments or lanes, that does not allow DynaMIT to compute correctly the shortest path.

5.5 On-Line Calibration in the state estimation process

Currently in DynaMIT there is code to partially perform on-line calibration of OD parameters. This is achieved using an assignment approach to estimate what the ODs should be using a Generalised Least Squared method.

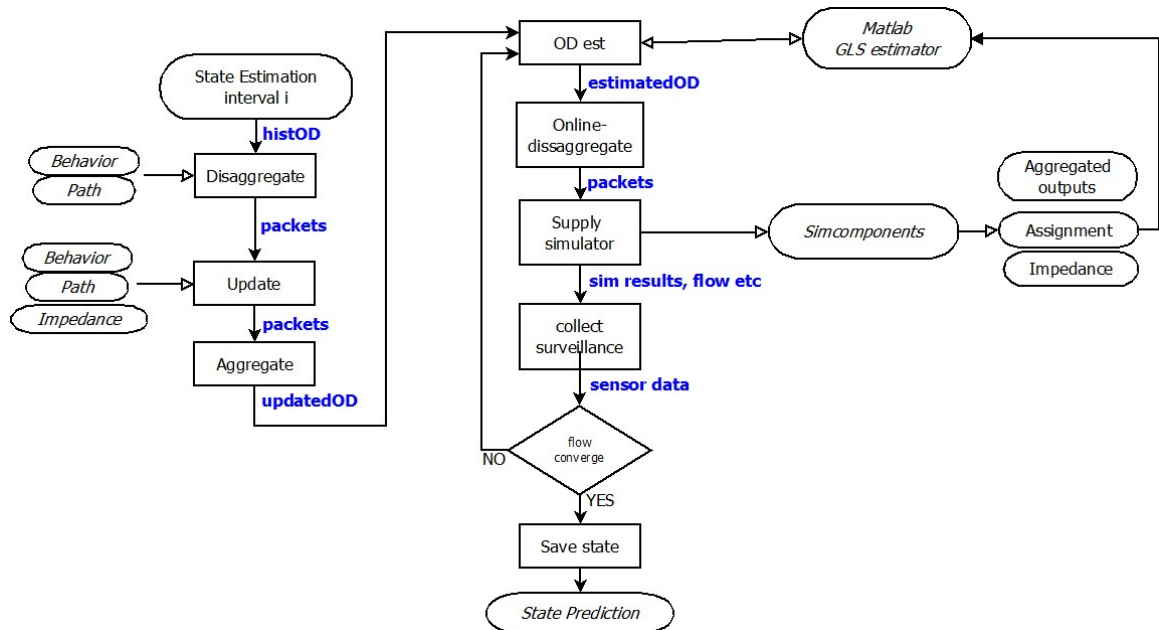


Figure 5.4: A GLS approach is used in the current DynaMIT-R to calibrate OD values

The estimation and prediction of OD flows is performed at the aggregate level: the total flow between each OD pair, across all individual drivers (*Balakrishna, 2006*). Sequential OD estimation for the departure intervals $h = 1, 2, \dots, T$ are iterated with error covariance estimation until convergence. The OD estimates from interval h form the a priori estimates for interval $h + 1$ for the sequential OD estimation.

The proposed process flow of the SP-EKF method is given in Figure 5.5 and described in greater detail in Table 5.5. The SPSA is present in the step of linearization, as it has already been mentioned in Section 2.2.6.

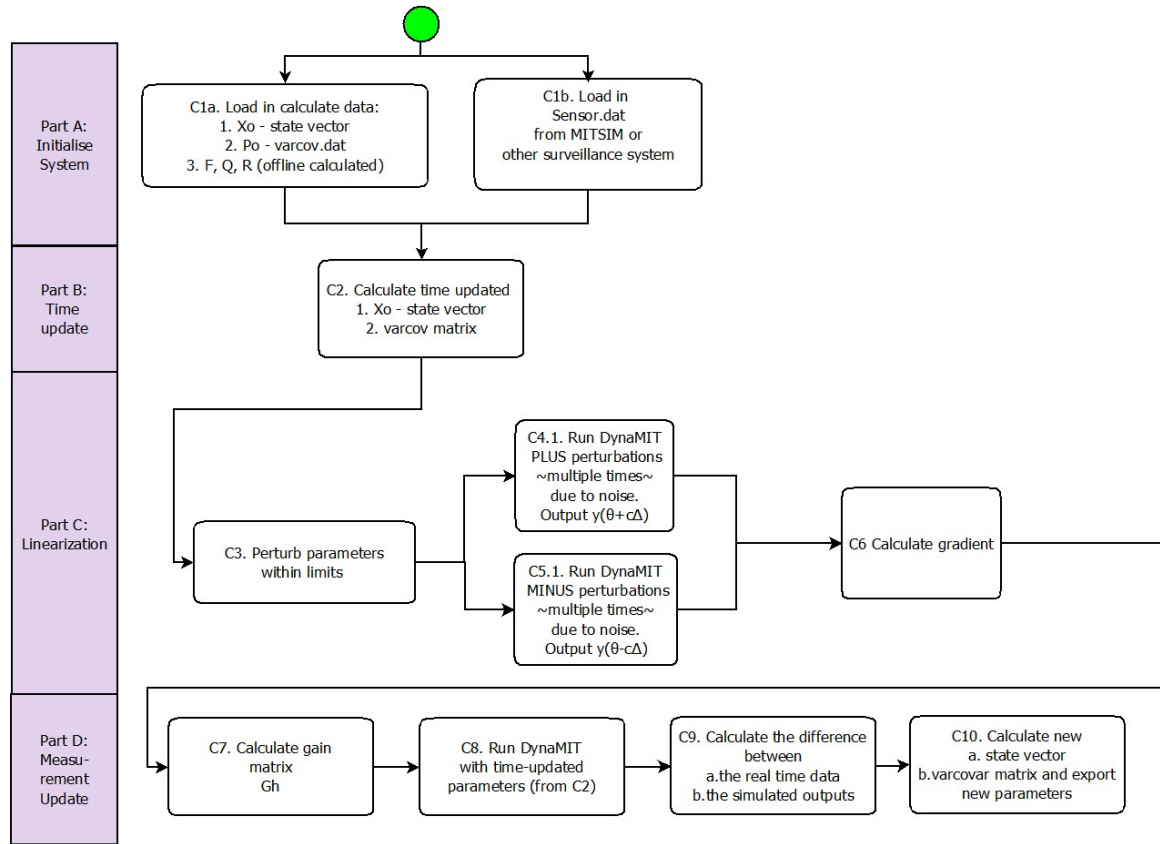


Figure 5.5: Process flow of the SP-EKF method

Task Name	Description	Inputs	Interval	File source
C1a.1 Insert state vector	For the first iteration of the on-line calibration process, initial values of the state vector are needed	ODs, capacities, speed-density relationship parameters etc. (state vector)	h-1	historical data or warm-up phase
C1a.2 Insert varcov.dat	Insert the initial values of Po - varcov.dat	varcov.dat	h-1	historical data
C1a.3 Insert F, Q, R matrices	During the time and measurement update calculations, autoregressive factors (matrix F) and error covariances (Q, R) are needed	F, Q, R (.txt)	-	offline calculated matrices (using historical data for each interval within one day)
C1b Insert necessary files from MITSIM	During the measurement update, the difference between the real-time data and the estimated data from DynaMIT must be calculated. Real-time data are represented by sensor counts	sensor.dat	h	MITSIM
C2.1 Calculate time updated state vector	According to the Equation 2.5 time updated state vector is calculated	state vector of interval h-1 (from C1a.1)	-	state vector of interval h

C2.2 Calculate time updated “varcov” matrix	Equation 2.6 - Time updated “varcov” matrix is calculated.	“varcov” matrix of interval $h-1$ (from C1a.1), Q (from C1a.3)	-	“varcov” matrix of interval h
C3 Perturb state vector within limits	1st step of SPSA implementation (Linearization phase)	state vector of interval h (from C2)	-	PLUS and MINUS perturbations
C4,5 Run DynaMIT	Run DynaMIT with PLUS and MINUS perturbations	from C3	-	RMSN (result from loss function y)
C6 Calculate gradient - $H(h)$	Following the final step of SPSA algorithm...	from C4,5	h	value of the gradient - $H(h)$
C7 Calculate Gain matrix	Equation 2.7	C2.2, C6, C1a.3	h	value of Gain matrix
C8 Run DynaMIT with the state vector of interval h	Use the output of step C2.1 and run DynaMIT	State vector	h	Simulated sensor counts
C9 Calculate (real data-simulated)	Calculation of an updated state vector according to the Equation 2.8	C8 data	h	-
C10	Calculation of the new state vector, and “varcov” matrix	C2.1, C6, C7, C9 data	h	final state vector, “varcov” matrix

Throughout the implementation of On-line calibration, the DynaMIT would be called at least 7 times. It is necessary to run DynaMIT more than tree times during the linearization step, in order to minimize the stochasticity of the results. The parallelization of certain applications would be useful, because the process requires quick calculations in real time.

CHAPTER VI

Conclusion

6.1 Summary

The availability of sufficiently accurate macroscopic and microscopic models is important for the design and the testing of modern freeway traffic control strategies. Their performance is largely independent of network's initial condition, data that could be incorporated in a model through different parameter values or small methodology changes.

The calibration framework is a need which is continuously highlighted through several researches. A well implemented calibration will lead to results close as possible to the observed ones in the field. The stage of calibration is very important in the final performance of the simulation model. Therefore, the main goal of this thesis was the development of an integrated calibration methodology using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm, that will contribute to the simulation model efficiency.

Sensitivity analysis was firstly implemented through a basic traffic simulation model. All the parameters of SPSA were analyzed, re-running each time the studied model. It was found that certain parameters affect significantly SPSA's converge, while some others not. More specifically, the parameter c appears not to improve the final result, while the parameters **alpha** **gamma** and **a** seem to contribute signifi-

cantly to the convergence of the algorithm.

Through a second experiment, using a Car-Following Model provided by the Transmodeler simulator, the above findings were confirmed. Even small changes in the values of important parameters could lead to the ultimate failure of the algorithm. Many iterations will once again allow the convergence of the algorithm to the optimal value, however the necessary time would be much higher, a conclusion not permitted to complex models and large data volume.

Traffic conditions are estimated and subsequently predicted utilizing data from diverse resources. The required parameters for these procedures are calibrated off-line, generating in this way a library, from which the most appropriate can be selected and used. Off-line calibrations can be repeated several times, incorporating every time new archived surveillance information and recent operational experience.

The second objective of this thesis was to demonstrate the on-line calibration approach using the DynaMIT DTA system. A case study with the entire expressway network in Singapore was presented. The calibration methodology was described in detail through a detailed table. It was scheduled to exploit real time field sensor counts and speeds data for the real time calibration. Unfortunately, the process has not reached the stage of implementation, because a data cleaning process was introduced and it is still in progress. Several checks showed, that Singapore network still needs a lot of improvements, as several segments and intersections are missing. Additionally, the data that Land Transport Authority send to the main server of DynaMIT seems to present some failures (inactivated sensors, coverage range determination weakness etc).

6.2 Future research directions

In the present thesis, the behavior of SPSA through two traffic simulation models was studied. For the formulation of integrated conclusions on SPSA's specifics,

further experimental analysis need to be demonstrated. The presented findings can not yet be generalized. It is important to further study the progress of the algorithm's calculations and to find solutions to still existing problems, such as the speed of convergence to the optimal solution.

The implementation of the on-line calibration approach, as it is previously analyzed step by step, should be also one of the future directions. The method has the issue of scalability due to the extremely heavy computational burdens, however the Simultaneous perturbation method should be able to accelerate significantly the estimation process.

It would be also interesting to see how the W-SPSA algorithm would perform. The parallelization of certain calculations can give very satisfactory results in an environment of rapid alternations and decisions. The on-line calibration demands quick solutions to any new formed questions.

The use of distributions of collected data (such as accelerations, using opportunistic sensors, such as smart-phone accelerometers) for calibration purposes could also be investigated. The extensive need for information nowadays taking advantage of pre-processed data, which through the distributions could include more levels of information.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Abdulhai, B., J.-B. Sheu, and W. Recker (1999), Simulation of its on the irvine fot area using” paramics 1.5” scalable microscopic traffic simulator: Phase i: Model calibration and validation, *California Partners for Advanced Transit and Highways (PATH)*.
- Abou-Rahme, N., S. Beale, B. Harbord, and E. Hardman (2000), Monitoring and modelling of controlled motorways.
- Adeli, H. (2001), Neural networks in civil engineering: 1989–2000, *Computer-Aided Civil and Infrastructure Engineering*, 16(2), 126–142.
- Al-Shihabi, T., and R. R. Maurant (2003), Toward more realistic driving behavior models for autonomous vehicles in driving simulators, *Transportation Research Record: Journal of the Transportation Research Board*, 1843(1), 41–49.
- Allström, A., A. M. Bayen, M. Fransson, D. Gundlegård, A. D. Patire, C. Rydergren, and M. Sandin (2014), Calibration framework based on bluetooth sensors for traffic state estimation using a velocity based cell transmission model, *Transportation Research Procedia*, 3, 972–981.
- Antoniou, C. (2004), On-line calibration for dynamic traffic assignment, Ph.D. thesis, Massachusetts Institute of Technology.
- Antoniou, C., M. Ben-Akiva, and H. N. Koutsopoulos (2007a), Nonlinear kalman filtering algorithms for on-line calibration of dynamic traffic assignment models, *Intelligent Transportation Systems, IEEE Transactions on*, 8(4), 661–670.
- Antoniou, C., H. N. Koutsopoulos, and G. Yannis (2007b), An efficient non-linear kalman filtering algorithm using simultaneous perturbation and applications in traffic estimation and prediction, in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 217–222, IEEE.
- Antoniou, C., R. Balakrishna, and H. N. Koutsopoulos (2011), A synthesis of emerging data collection technologies and their impact on traffic management applications, *European Transport Research Review*, 3(3), 139–148.
- Antoniou, C., A. Kondyli, G.-M. Lykogianni, and E. Gyftodimos (2013), Exploratory assessment of the Limiting Extended Kalman Filter properties, *Transport and Telecommunication*, 14(1), 1–12.

- Antoniou, C., V. Gikas, V. Papathanasopoulou, T. Mpimis, I. Markou, and H. Perakis (2014), Towards distribution-based calibration for traffic simulation, in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pp. 786–791, IEEE.
- Ashok, K. (1996), Estimation and prediction of time-dependent origin-destination flows, Ph.D. thesis, Massachusetts Institute of Technology.
- Ashok, K., and M. E. Ben-Akiva (1993), Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems, in *International Symposium on the Theory of Traffic Flow and Transportation (12th: 1993: Berkeley, Calif.)*. *Transportation and traffic theory*.
- Ashok, K., and M. E. Ben-Akiva (2000), Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows, *Transportation Science*, 34(1), 21–36.
- Balakrishna, R. (2006), Off-line calibration of dynamic traffic assignment models, Ph.D. thesis, Massachusetts Institute of Technology.
- Balakrishna, R., and H. N. Koutsopoulos (2008), Incorporating within-day transitions in simultaneous offline estimation of dynamic origin-destination flows without assignment matrices, *Transportation Research Record: Journal of the Transportation Research Board*, 2085(1), 31–38.
- Balakrishna, R., M. Ben-Akiva, and H. N. Koutsopoulos (2007), Offline calibration of dynamic traffic assignment: simultaneous demand-and-supply estimation, *Transportation Research Record: Journal of the Transportation Research Board*, 2003(1), 50–58.
- Barceló, J., and J. Casas (2005), Dynamic network simulation with aimsun, in *Simulation approaches in transportation analysis*, pp. 57–98, Springer.
- Ben-Akiva, M., E. Cascetta, and H. Gunn (1995), An on-line dynamic traffic prediction model for an inter-urban motorway network, in *Urban Traffic Networks*, pp. 83–122, Springer.
- Ben-Akiva, M., M. Bierlaire, J. Bottom, H. Koutsopoulos, and R. Mishalani (1997), Development of a route guidance generation system for real-time application, in *Proceedings of the IFAC Transportation Systems 97 Conference, Chania*, Citeseer.
- Ben-Akiva, M., M. Bierlaire, H. Koutsopoulos, and R. Mishalani (1998), Dynamit: a simulation-based system for traffic prediction, in *DACCORS Short Term Forecasting Workshop, The Netherlands*, Citeseer.
- Ben-Akiva, M., M. Bierlaire, H. N. Koutsopoulos, and R. Mishalani (2002), Real time simulation of traffic demand-supply interactions within dynamit, in *Transportation and network analysis: current trends*, pp. 19–36, Springer.

- Ben-Akiva, M., H. N. Koutsopoulos, T. Toledo, Q. Yang, C. F. Choudhury, C. Antoniou, and R. Balakrishna (2010), Traffic simulation with mitsimlab, in *Fundamentals of Traffic Simulation*, pp. 233–268, Springer.
- Ben-Akiva, M., H. N. Koutsopoulos, C. Antoniou, and R. Balakrishna (2011), Traffic simulation with dynamit, *Fundamentals of Traffic Simulation*, 145, 363.
- Beven, K., and A. Binley (1992), The future of distributed models: model calibration and uncertainty prediction, *Hydrological processes*, 6(3), 279–298.
- Bierlaire, M., and F. Crittin (2004), An efficient algorithm for real-time estimation and prediction of dynamic od tables, *Operations Research*, 52(1), 116–127.
- Bowman, J. L., and M. E. Ben-Akiva (2001), Activity-based disaggregate travel demand model system with activity schedules, *Transportation Research Part A: Policy and Practice*, 35(1), 1–28.
- Box, M. (1965), A new method of constrained optimization and a comparison with other methods, *The Computer Journal*, 8(1), 42–52.
- Boxill, S. A., and L. Yu (2000), An evaluation of traffic simulation models for supporting its.
- Boyle, D. P., H. V. Gupta, and S. Sorooshian (2000), Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods, *Water Resources Research*, 36(12), 3663–3674.
- Brackstone, M., and M. McDonald (1999), Car-following: a historical review, *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4), 181–196.
- Brauckhoff, D., K. Salamatian, and M. May (2009), Applying pca for traffic anomaly detection: Problems and solutions, in *INFOCOM 2009, IEEE*, pp. 2866–2870, IEEE.
- Brockfeld, E., R. D. Kühne, and P. Wagner (2004), Calibration and validation of microscopic traffic flow models, *Transportation Research Record: Journal of the Transportation Research Board*, 1876(1), 62–70.
- Cascetta, E. (1984), Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator, *Transportation Research Part B: Methodological*, 18(4), 289–299.
- Cascetta, E., and F. Russo (1997), Calibrating aggregate travel demand models with traffic counts: Estimators and statistical performance, *Transportation*, 24(3), 271–293.
- Cascetta, E., D. Inaudi, and G. Marquis (1993), Dynamic estimators of origin-destination matrices using traffic counts, *Transportation science*, 27(4), 363–373.

- Chen, C., H. Gong, C. Lawson, and E. Bialostozky (2010), Evaluating the feasibility of a passive travel survey collection in a complex urban environment: Lessons learned from the new york city case study, *Transportation Research Part A: Policy and Practice*, 44(10), 830–840.
- Chiu, Y.-C., J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks (2011), Dynamic traffic assignment - A Primer, *TRB*.
- Chui, C. K., and G. Chen (1999), Kalman filtering, *With real time applications*.
- Cipriani, E., M. Florian, M. Mahut, and M. Nigro (2011), A gradient approximation approach for adjusting temporal origin–destination matrices, *Transportation Research Part C: Emerging Technologies*, 19(2), 270–282.
- Daamen, W., C. Buisson, and S. P. Hoogendoorn (2014), *Traffic Simulation and Data: Validation Methods and Applications*, CRC Press.
- de Souza, S. V., and R. G. Junqueira (2005), A procedure to assess linearity by ordinary least squares method, *Analytica chimica acta*, 552(1), 25–35.
- Diskin, M., and E. Simon (1977), A procedure for the selection of objective functions for hydrologic simulation models, *Journal of Hydrology*, 34(1), 129–149.
- Dolšák, B., and M. Novak (2011), Intelligent decision support for structural design analysis, *Advanced Engineering Informatics*, 25(2), 330–340.
- Dunne, S., and B. Ghosh (2011), Regime-based short-term multivariate traffic condition forecasting algorithm, *Journal of Transportation Engineering*, 138(4), 455–466.
- Figueiredo, M., Á. Seco, and A. B. Silva (2014), Calibration of microsimulation models—the effect of calibration parameters errors in the models performance, *Transportation Research Procedia*, 3, 962–971.
- Fritzsche, H.-T. (1994), A model for traffic simulation.
- Ge, H.-x., X.-p. Meng, H.-b. Zhu, and Z.-P. Li (2014), Feedback control for car following model based on two-lane traffic flow, *Physica A: Statistical Mechanics and its Applications*, 408, 28–39.
- Geladi, P., and B. R. Kowalski (1986), Partial least-squares regression: a tutorial, *Analytica chimica acta*, 185, 1–17.
- Ghiasi, H., D. Pasini, and L. Lessard (2008), Constrained globalized neldermead method for simultaneous structural and manufacturing optimization of a composite bracket, *Journal of composite materials*, 42(7), 717–736.
- Gipps, P. G. (1981), A behavioural car-following model for computer simulation, *Transportation Research Part B: Methodological*, 15(2), 105–111.

- Grewal, M. S., and A. P. Andrews (2011), *Kalman filtering: theory and practice using MATLAB*, John Wiley & Sons.
- Guan, W., S. He, and J. Ma (2012), Review on traffic flow phenomena and theory, *Journal of Transportation Systems Engineering and Information Technology*, 12(3), 90–97.
- Gupta, H. V., S. Sorooshian, and P. O. Yapo (1998), Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information, *Water Resources Research*, 34(4), 751–763.
- Hazelton, M. L. (2000), Estimation of origin–destination matrices from link flows on uncongested networks, *Transportation Research Part B: Methodological*, 34(7), 549–566.
- Hoogendoorn, S. P., and P. H. Bovy (2001), State-of-the-art of vehicular traffic flow modelling, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4), 283–303.
- Huang, E. (2010), Algorithmic and implementation aspects of on-line calibration of dynamic traffic assignment, Ph.D. thesis, Massachusetts Institute of Technology.
- Janssen, P., and P. Heuberger (1995), Calibration of process-oriented models, *Ecological Modelling*, 83(1), 55–66.
- Jia, Y., J. Wu, and Y. Du (2014), An improved car-following model considering variable safety headway distance, *arXiv preprint arXiv:1407.3374*.
- Julier, S. J., and J. K. Uhlmann (1997), A new extension of the kalman filter to nonlinear systems, in *Int. symp. aerospace/defense sensing, simul. and controls*, vol. 3, pp. 3–2, Orlando, FL.
- Kalman, R. E. (1960), A new approach to linear filtering and prediction problems, *Journal of basic Engineering*, 82(1), 35–45.
- Kamga, C. N., K. C. Mouskos, and R. E. Paaswell (2011), A methodology to estimate travel time using dynamic traffic assignment (dta) under incident conditions, *Transportation Research Part C: Emerging Technologies*, 19(6), 1215–1224.
- Kiefer, J., J. Wolfowitz, et al. (1952), Stochastic estimation of the maximum of a regression function, *The Annals of Mathematical Statistics*, 23(3), 462–466.
- Kim, K., and L. Rilett (2004), A genetic algorithm based approach to traffic micro-simulation calibration using its data, in *83rd Annual Meeting of the Transportation Research Board, Washington, DC*.
- Konishi, K., H. Kokame, and K. Hirata (2000), Decentralized delayed-feedback control of an optimal velocity traffic model, *The European Physical Journal B-Condensed Matter and Complex Systems*, 15(4), 715–722.

- Koutsopoulos, H. N., and H. Farah (2012), Latent class model for car following behavior, *Transportation research part B: methodological*, 46(5), 563–578.
- Kundie, K. K. (2002), Calibration of mesoscopic traffic simulation models for dynamic traffic assignment, Ph.D. thesis, Massachusetts Institute of Technology.
- Lakhina, A., M. Crovella, and C. Diot (2004), Diagnosing network-wide traffic anomalies, in *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 219–230, ACM.
- Leclercq, L. (2005), Calibration of flow-density relationships on urban streets, *Transportation Research Record: Journal of the Transportation Research Board*, 1934(1), 226–234.
- Lee, D.-H., X. Yang, and P. Chandrasekar (2001), Parameter calibration for paramics using genetic algorithm, in *80th Annual Meeting of the Transportation Research Board, Washington, DC*.
- Legates, D. R., and G. J. McCabe (1999), Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation, *Water resources research*, 35(1), 233–241.
- Leutzbach, W. (1988), *Introduction to the theory of traffic flow*, vol. 47, Springer.
- Li, B. (2005), Bayesian inference for origin-destination matrices of transport networks using the em algorithm, *Technometrics*, 47(4).
- Liang-Qun, L., J. Hong-Bing, and L. Jun-Hui (2005), The iterated extended kalman particle filter, in *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, vol. 2, pp. 1213–1216, IEEE.
- Lighthill, M. J., and G. B. Whitham (1955), On kinematic waves. ii. a theory of traffic flow on long crowded roads, *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178), 317–345.
- Lin, P.-W., and G.-L. Chang (2007), A generalized model and solution algorithm for estimation of the dynamic freeway origin–destination matrix, *Transportation Research Part B: Methodological*, 41(5), 554–572.
- Liu, H., H. van Zuylen, H. van Lint, and M. Salomons (2006), Predicting urban arterial travel time with state-space neural networks and kalman filters, *Transportation Research Record: Journal of the Transportation Research Board*, 1968(1), 99–108.
- Liu, R. (2010), Traffic simulation with dracula, in *Fundamentals of Traffic Simulation*, pp. 295–322, Springer.
- Lu, L. (2014), W-spsa: An efficient stochastic approximation algorithm for the off-line calibration of dynamic traffic assignment models, Ph.D. thesis, Massachusetts Institute of Technology.

- Luersen, M., R. Le Riche, and F. Guyon (2004), A constrained, globalized, and bounded nelder–mead method for engineering optimization, *Structural and Multidisciplinary Optimization*, 27(1-2), 43–54.
- Ma, X., and H. N. Koutsopoulos (2008), A new online travel time estimation approach using distorted automatic vehicle identification data, in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 204–209, IEEE.
- Ma, X., Z. Huang, and H. Koutsopoulos (2014), Integrated traffic and emission simulation: a model calibration approach using aggregate information, *Environmental Modeling & Assessment*, pp. 1–12.
- Maher, M. (1983), Inferences on trip matrices from observations on link volumes: a bayesian statistical approach, *Transportation Research Part B: Methodological*, 17(6), 435–447.
- Mahmassani, H. S. (1993), Algorithm for dynamic route guidance in congested networks with multiple user information availability groups, in *International Symposium on Automotive Technology & Automation (26th). Dedicated conference on advanced transport telematics/intelligent vehicle highway systems*.
- Mahut, M., M. Florian, N. Tremblay, M. Campbell, D. Patman, and Z. K. McDaniel (2004), Calibration and application of a simulation-based dynamic traffic assignment model, *Transportation Research Record: Journal of the Transportation Research Board*, 1876(1), 101–111.
- Mahut, M., M. Florian, D. Florian, S. Velan, and N. Tremblay (2005), Equilibrium dynamic traffic assignment for large, congested networks, *INRO white paper*.
- Manne, R. (1987), Analysis of two partial-least-squares algorithms for multivariate calibration, *Chemometrics and Intelligent Laboratory Systems*, 2(1), 187–197.
- Marfia, G., M. Roccetti, and A. Amoroso (2013), A new traffic congestion prediction model for advanced traveler information and management systems, *Wireless Communications and Mobile Computing*, 13(3), 266–276.
- Maybeck, P. S. (1982), *Stochastic models, estimation, and control*, vol. 3, Academic press.
- Milkovits, M., E. Huang, C. Antoniou, M. Ben-Akiva, and J. A. Lopes (2010), Dynamit 2.0: The next generation real-time dynamic traffic assignment system, in *Advances in System Simulation (SIMUL), 2010 Second International Conference on*, pp. 45–51, IEEE.
- Muñoz, L., X. Sun, D. Sun, G. Gomes, and R. Horowitz (2004), Methodological calibration of the cell transmission model, in *American Control Conference, 2004. Proceedings of the 2004*, vol. 1, pp. 798–803, IEEE.

- Nelder, J. A., and R. Mead (1965), A simplex method for function minimization, *The computer journal*, 7(4), 308–313.
- Olstam, J. J., and A. Tapani (2004), Comparison of car-following models, *Tech. rep.*
- Ozguven, E. E., and K. Ozbay (2008), Performance evaluation of simultaneous perturbation stochastic approximation algorithm for solving stochastic transportation network analysis problems.
- Papathanasopoulou, V. (2012), Estimation of car-following models using machine learning techniques (in greek), Master’s thesis, National Technical University of Athens.
- Peeta, S., and A. K. Ziliaskopoulos (2001), Foundations of dynamic traffic assignment: The past, the present and the future, *Networks and Spatial Economics*, 1(3-4), 233–265.
- Peng, G.-h., and R.-j. Cheng (2013), A new car-following model with the consideration of anticipation optimal velocity, *Physica A: Statistical Mechanics and its Applications*, 392(17), 3563–3569.
- Punzo, V., D. J. Formisano, and V. Torrieri (2005), Part 1: Traffic flow theory and car following: Nonstationary kalman filter for estimation of accurate and consistent car-following data, *Transportation Research Record: Journal of the Transportation Research Board*, 1934(1), 1–12.
- Punzo, V., B. Ciuffo, and M. Montanino (2012), Can results of car-following model calibration based on trajectory data be trusted?, *Transportation Research Record: Journal of the Transportation Research Board*, 2315(1), 11–24.
- Radovic, Y., and A. Remouchamps (2002), Boss quattro: an open system for parametric design, *Structural and multidisciplinary optimization*, 23(2), 140–152.
- Rakha, H., and W. Wang (2009), Procedure for calibrating gipps car-following model, *Transportation Research Record: Journal of the Transportation Research Board*, 2124(1), 113–124.
- Rakha, H., Y. Gao, and M.-A. U. T. Center (2010), Calibration of steady-state car-following models using macroscopic loop detector data, *Tech. rep.*, Mid-Atlantic Universities Transportation Center.
- Ringberg, H., A. Soule, J. Rexford, and C. Diot (2007), Sensitivity of pca for traffic anomaly detection, in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, pp. 109–120, ACM.
- Robbins, H., and S. Monro (1951), A stochastic approximation method, *The annals of mathematical statistics*, pp. 400–407.

- Schneeberger, J. (2002), Microscopic simulation model calibration and validation: A case study of vissim for a coordinated actuated signal system.
- Servat, E., and A. Dezetter (1991), Selection of calibration objective functions in the context of rainfall-runoff modelling in a sudanese savannah area, *Hydrological Sciences Journal*, 36(4), 307–330.
- Sherali, H. D., and T. Park (2001), Estimation of dynamic origin–destination trip tables for a general network, *Transportation Research Part B: Methodological*, 35(3), 217–235.
- Sherali, H. D., A. Narayanan, and R. Sivanandan (2003), Estimation of origin–destination trip-tables based on a partial set of traffic link volumes, *Transportation Research Part B: Methodological*, 37(9), 815–836.
- Silcock, J. (1993), Sigsim version 1.0 users guide.
- Smith, L. I. (2002), A tutorial on principal components analysis, *Cornell University, USA*, 51, 52.
- Spall, J. C. (1998a), Implementation of the simultaneous perturbation algorithm for stochastic optimization, *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3), 817–823.
- Spall, J. C. (1998b), An overview of the simultaneous perturbation method for efficient optimization, *Johns Hopkins APL Technical Digest*, 19(4), 482–492.
- Spiliopoulou, A., M. Kontorinaki, M. Papageorgiou, and P. Kopelias (2014), Macroscopic traffic flow model validation at congested freeway off-ramp areas, *Transportation Research Part C: Emerging Technologies*, 41, 18–29.
- Toledo, T., and R. Beinhaker (2006), Evaluation of the potential benefits of advanced traveler information systems, *Journal of Intelligent Transportation Systems*, 10(4), 173–183.
- Toledo, T., H. N. Koutsopoulos, A. Davol, M. E. Ben-Akiva, W. Burghout, I. Andréasson, T. Johansson, and C. Lundin (2003), Calibration and validation of microscopic traffic simulation tools: Stockholm case study, *Transportation Research Record: Journal of the Transportation Research Board*, 1831(1), 65–75.
- Treiber, M., and A. Kesting (2013), Microscopic calibration and validation of car-following models—a systematic approach, *Procedia-Social and Behavioral Sciences*, 80, 922–939.
- Van Aerde, M. (1999), Integration: Users guide for model version 2.0, *Transportation Systems Group, Queens University and M. Van Aerde Associates, Ltd. Canada*.
- Van Aerde, M., and H. A. Rakha (1996), Travtek evaluation modeling study, *Tech. rep.*

- Vlahogianni, E., and M. Karlaftis (2011), Temporal aggregation in traffic data: implications for statistical characteristics and model choice, *Transportation Letters*, 3(1), 37–49.
- Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias (2005), Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach, *Transportation Research Part C: Emerging Technologies*, 13(3), 211–234.
- Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias (2008), Temporal evolution of short-term urban traffic flow: A nonlinear dynamics approach, *Computer-Aided Civil and Infrastructure Engineering*, 23(7), 536–548.
- Vlahogianni, E. I., M. G. Karlaftis, and J. C. Golias (2014), Short-term traffic forecasting: Where we are and where we were going, *Transportation Research Part C: Emerging Technologies*.
- Wan, E. A., and R. Van Der Merwe (2000), The unscented kalman filter for nonlinear estimation, in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158, IEEE.
- Wang, Y., M. Papageorgiou, and A. Messmer (2006), A real-time freeway network traffic surveillance tool, *Control Systems Technology, IEEE Transactions on*, 14(1), 18–32.
- Welch, G., and G. Bishop (1995), An introduction to the kalman filter.
- Wiedemann, R., and U. Reiter (1992), Microscopic traffic simulation: the simulation system mission, background and actual state, *Project ICARUS (V1052) Final Report. Brussels, CEC*, 2.
- Wilson, R. E. (2001), An analysis of gipps’s car-following model of highway traffic, *IMA journal of applied mathematics*, 66(5), 509–537.
- Wu, Y., and S. Liu (2014), A suggestion for computing objective function in model calibration, *Ecological Informatics*, 24, 107–111.
- Yang, Q., and H. N. Koutsopoulos (1996), A microscopic traffic simulator for evaluation of dynamic traffic management systems, *Transportation Research Part C: Emerging Technologies*, 4(3), 113–129.
- Yapo, P. O., H. V. Gupta, and S. Sorooshian (1998), Multi-objective global optimization for hydrologic models, *Journal of hydrology*, 204(1), 83–97.
- Yue, P., and L. Yu (2000), Travel demand forecasting models: a comparison of EMME/2 and QRS II using a real-world network, *Tech. rep.*, Citeseer.