

**ESTIMATION AND PREDICTION OF TRAVEL TIME FROM LOOP
DETECTOR DATA FOR INTELLIGENT TRANSPORTATION SYSTEMS
APPLICATIONS**

A Dissertation

by

LELITHA DEVI VANAJAKSHI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2004

Major Subject: Civil Engineering

**ESTIMATION AND PREDICTION OF TRAVEL TIME FROM LOOP
DETECTOR DATA FOR INTELLIGENT TRANSPORTATION SYSTEMS**

APPLICATIONS

A Dissertation

by

LELITHA DEVI VANAJAKSHI

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Laurence R. Rilett
(Chair of Committee)

Mark W. Burris
(Member)

Jyh-Charn (Steve) Liu
(Member)

Clifford H. Spiegelman
(Member)

Paul N. Roschke
(Interim Head of Department)

August 2004

Major Subject: Civil Engineering

ABSTRACT

Estimation and Prediction of Travel Time from Loop Detector Data for Intelligent Transportation Systems Applications. (August 2004)

Lelitha Devi Vanajakshi, B.Tech., University of Kerala, India;
M.Tech., University of Kerala, India

Chair of Advisory Committee: Dr. Laurence R. Rilett

With the advent of Advanced Traveler Information Systems (ATIS), short-term travel time prediction is becoming increasingly important. Travel time can be obtained directly from instrumented test vehicles, license plate matching, probe vehicles etc., or from indirect methods such as loop detectors. Because of their wide spread deployment, travel time estimation from loop detector data is one of the most widely used methods. However, the major criticism about loop detector data is the high probability of error due to the prevalence of equipment malfunctions. This dissertation presents methodologies for estimating and predicting travel time from the loop detector data after correcting for errors. The methodology is a multi-stage process, and includes the correction of data, estimation of travel time and prediction of travel time, and each stage involves the judicious use of suitable techniques. The various techniques selected for each of these stages are detailed below. The test sites are from the freeways in San Antonio, Texas, which are equipped with dual inductance loop detectors and AVI.

- Constrained non-linear optimization approach by Generalized Reduced Gradient (GRG) method for data reduction and quality control, which included a check for the accuracy of data from a series of detectors for conservation of vehicles, in addition to the commonly adopted checks.
- A theoretical model based on traffic flow theory for travel time estimation for both off-peak and peak traffic conditions using flow, occupancy and speed values obtained from detectors.
- Application of a recently developed technique called Support Vector Machines (SVM) for travel time prediction. An Artificial Neural Network (ANN) method is also developed for comparison.

Thus, a complete system for the estimation and prediction of travel time from loop detector data is detailed in this dissertation. Simulated data from CORSIM simulation software is used for the validation of the results.

DEDICATION

To my family and friends for their love and support...

ACKNOWLEDGEMENTS

First, I would like to express my sincere appreciation and thanks to my advisor and chair of my committee, Dr. Laurence Rilett, for his guidance and support during my study at Texas A&M University. It was personally a rewarding experience to work with Dr. Rilett.

I would also like to thank my committee members, Dr. Mark Burris, Dr. Steve Liu, and Dr. Cliff Spiegelman for their guidance, and for taking the time to work with me and to provide technical expertise to achieve my objectives. Special thanks to Dr. Messer and Dr. Nelson for their excellent teaching and for their help and support.

Special thanks to Mr. Sanju Nair for the technical help. I would also like to thank Dr. William Eisele, and Mr. Shawn Turner for their willingness to share their knowledge of the subject matter. I would like to thank Seung-Jun Kim, Dr. Montasir Abbas, and Anuj Sharma for their help in using CORSIM simulation software. And thanks to Jacqueline, Juliana, Malini, Ranhee, and Teresa, for the good times.

I would like to thank the research sponsor, the Texas Higher Education Board, as well as the TransLink® Research Program at the Texas Transportation Institute, Texas A&M University for providing the facilities and resources for my research.

I would like to thank the Texas Department of Transportation (TxDOT), San Antonio, for the data provided to accomplish this dissertation, especially Brian Fariello and Bill Jurczyn for their support on ITS data archiving issues.

I would like to thank my family for the support they provided throughout my studies which helped me to achieve my goals. Special thanks to my husband, Murali, for his encouragement and support, without which I would not have achieved this.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	xvi
CHAPTER	
I INTRODUCTION.....	1
1.1 Statement of the Problem.....	2
1.2 Research Objectives.....	4
1.3 Research Methodology	5
1.4 Contribution of the Research	9
1.5 Organization of the Dissertation	10
II LITERATURE REVIEW	12
2.1 Introduction.....	12
2.2 ILD Data and Data Errors	12
2.3 Travel Time Estimation From ILD Data for Freeways.....	18
2.4 Prediction of Travel Time on Freeways.....	22
2.5 Concluding Remarks.....	26
III DATA COLLECTION AND PRELIMINARY DATA ANALYSIS	28
3.1 Introduction.....	28
3.2 Field Data Sources	29
3.3 Field Data Collection	38
3.4 Test Bed	41
3.5 Preliminary Data Reduction.....	42
3.6 Simulated Data Using Corsim	49
3.7 Concluding Remarks.....	52
IV OPTIMIZATION FOR DATA DIAGNOSTICS.....	54
4.1 Introduction.....	54
4.2 Conservation of Vehicles	56
4.3 Generalized Reduced Gradient Optimization Procedure	60
4.4 Implementation	66

CHAPTER	Page
4.5 Results.....	72
4.6 Validation.....	95
4.7 Other Applications.....	107
4.8 Alternative Objective Functions and Constraints.....	112
4.9 Concluding Remarks.....	114
 V ESTIMATION OF TRAVEL TIME.....	 116
5.1 Introduction.....	116
5.2 Traffic Dynamics Model.....	118
5.3 Proposed Model for Travel Time Estimation.....	126
5.4 Results and Discussion.....	135
5.5 Concluding Remarks.....	165
 VI SHORT-TERM TRAVEL TIME PREDICTION.....	 166
6.1 Introduction.....	166
6.2 Models for Traffic Prediction.....	167
6.3 Model Parameters.....	184
6.4 Results.....	190
6.5 Concluding Remarks.....	211
 VII SUMMARY AND CONCLUSIONS.....	 214
7.1 Summary.....	214
7.2 Conclusions.....	218
7.3 Future Research.....	219
 REFERENCES.....	 221
APPENDIX A.....	241
APPENDIX B.....	242
APPENDIX C.....	247
APPENDIX D.....	249
VITA.....	288

LIST OF FIGURES

	Page
Fig. 2.1 Schematic diagram for illustrating extrapolation methods.....	19
Fig. 3.1 Actual loop in the field.....	30
Fig. 3.2 Schematic diagram of a single-loop detector in one lane of a roadway.....	31
Fig. 3.3 Schematic diagram of a dual-loop detector.....	31
Fig. 3.4 Raw ILD data.....	35
Fig. 3.5 AVI conceptual view.....	37
Fig. 3.6 Raw AVI data format.....	37
Fig. 3.7 AVI antennas.....	38
Fig. 3.8 Examples of TransGuide information systems.....	39
Fig. 3.9 (a) Map of the freeway system of San Antonio and (b) map of the test bed.....	40
Fig. 3.10 Schematic diagram of the test bed from I-35 N, San Antonio, Texas.....	43
Fig. 3.11 Occupancy distribution from I-35 site, location 2, on February 11, 2003.....	47
Fig. 3.12 Sample speed distribution from I-35 site, location 2, on February 11, 2003.....	47
Fig. 3.13 Sample volume distribution from I-35 site, location 2, on February 11, 2003.....	48
Fig. 3.14 Sample AVI travel time from I-35 site, on February 11, 2003.....	49
Fig. 3.15 Simulated data and the corresponding field data for February 11, 2003, for location 1.....	52
Fig. 4.1 Algorithm for the overall proposed method.....	55
Fig. 4.2 Illustration of the conservation of vehicles.....	56
Fig. 4.3 Algorithm for the GRG method.....	65
Fig. 4.4 Cumulative actual volumes for 24 hours at I-35 site on February 11, 2003.....	69

	Page
Fig. 4.5 Enlarged cumulative volumes for 1 hour at I-35 site on February 11, 2003	70
Fig. 4.6 Enlarged cumulative volumes for 1 hour at I-35 site on February 11, 2003	70
Fig. 4.7 Cumulative volumes after the optimization at I-35 site on February 11, 2003	73
Fig. 4.8 Enlarged cumulative volumes after optimization.....	74
Fig. 4.9 Enlarged cumulative volumes after optimization.....	74
Fig. 4.10 Cumulative actual volumes during 09:00:00-10:00:00 on February 10, 2003	75
Fig. 4.11 Cumulative optimized volumes during 09:00:00-10:00:00 on February 10, 2003	76
Fig. 4.12 Cumulative actual volumes during 05:00:00-06:00:00 on February 10, 2003	77
Fig. 4.13 Cumulative optimized volumes during 05:00:00-06:00:00 on February 10, 2003	77
Fig. 4.14 Cumulative actual volumes during 03:00:00-04:00:00 on February 10, 2003	78
Fig. 4.15 Cumulative optimized volumes during 03:00:00-04:00:00 on February 10, 2003	79
Fig. 4.16 Cumulative actual volumes during 05:00:00 to 06:00:00 on February 11, 2003	80
Fig. 4.17 Cumulative optimized volumes during 05:00:00 to 06:00:00 on February 11, 2003....	80
Fig. 4.18 Cumulative actual volumes during 11:30:00 – 12:30:00 on February 12, 2003	81
Fig. 4.19 Cumulative optimized volumes during 11:30:00 – 12:30:00 on February 12, 2003....	81
Fig. 4.20 Cumulative actual volumes during 10:00:00-11:00:00 on February 12, 2003	82
Fig. 4.21 Cumulative optimized volumes during 10:00:00-11:00:00 on February 12, 2003	83
Fig. 4.22 Cumulative actual volumes during 17:00:00 – 18:00:00 on February 13, 2003	84
Fig. 4.23 Cumulative optimized volumes during 17:00:00 – 18:00:00 on February 13, 2003....	84
Fig. 4.24 Cumulative actual volumes during 08:30:00 – 09:30:00 on February 14, 2003	85
Fig. 4.25 Cumulative optimized volumes during 08:30:00 – 09:30:00 on February 14, 2003....	86
Fig. 4.26 Cumulative actual volumes during 13:00:00 to 14:00:00 on February 14, 2003	87
Fig. 4.27 Cumulative optimized volumes during 13:00:00 to 14:00:00 on February 14, 2003....	87

	Page
Fig. 4.28 Cumulative actual volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405.....	90
Fig. 4.29 Cumulative optimized volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405.....	91
Fig. 4.30 Cumulative actual volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405.....	92
Fig. 4.31 Cumulative optimized volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405.....	93
Fig. 4.32 Validation of optimization performance using simulated data.....	97
Fig. 4.33 Validation of optimization performance using simulated cumulative data.....	98
Fig. 4.34 Performance of the optimization with varying amount of over counting of the detector.....	99
Fig. 4.35 Performance of the optimization with varying amount of under counting of the detector.....	99
Fig. 4.36 Performance of the optimization with varying amount of random error.....	100
Fig. 4.37 Comparison of the performance of optimization at Location 1	102
Fig. 4.38 Comparison of the performance of optimization at Location 2	103
Fig. 4.39 Comparison of the performance of optimization at Location 1	104
Fig. 4.40 Comparison of the performance of optimization at Location 2	105
Fig. 4.41 Imputation results with missing data in location 1 on February 10, 2003	109
Fig. 4.42 Imputation results with missing data in location 2 on February 10, 2003	109
Fig. 4.43 Imputation results with missing data in location 3 on February 10, 2003	110
Fig. 4.44 Imputation results with missing data in location 4 on February 10, 2003	110
Fig. 4.45 Imputation results with missing data in location 5 on February 10, 2003	111
Fig. 4.46 Variance from three consecutive locations on February 11, 2003	114
Fig. 5.1 Illustration of the conservation of vehicles	118

	Page
Fig. 5.2 Schematic representation of the total travel time during the interval (t_{n-1}, t_n) under normal-flows	122
Fig. 5.3 Schematic representation of the total travel time during the interval (t_{n-1}, t_n) under congested-flows.....	123
Fig. 5.4 Schematic diagram to illustrate the travel time calculation.....	129
Fig. 5.5 Travel time estimated by the N-D model using actual data for link 1 on February 11, 2003, for 24 hours.....	136
Fig. 5.6 Travel time estimated by the N-D model using actual data for link 2 on February 11, 2003, for 24 hours.....	136
Fig. 5.7 Travel time estimated by the N-D model using optimized data for link 1 on February 11, 2003, for 24 hours.....	137
Fig. 5.8 Travel time estimated by the N-D model using optimized data for link 2 on February 11, 2003, for 24 hours.....	138
Fig. 5.9 Estimated travel time on link 1 with density calculated from occupancy values on February 11, 2003, for 24 hours.....	139
Fig. 5.10 Estimated travel time on link 2 with density calculated from occupancy values on February 11, 2003, for 24 hours.....	139
Fig. 5.11 Volume distribution on February 11, 2003, for 24 hours at location 1	140
Fig. 5.12 Volume distribution on February 11, 2003, for 24 hours at location 2	140
Fig. 5.13 Volume distribution on February 11, 2003, for 24 hours at location 3	141
Fig. 5.14 Effect of combining extrapolation method on estimated travel time for low flow conditions on link 1 on February 11, 2003	141
Fig. 5.15 Effect of combining extrapolation method on estimated travel time for low flow conditions on the link 2 on February 11, 2003	142
Fig. 5.16 Comparison of N-D model and proposed model using optimized field data for February 11, 2003.....	143
Fig. 5.17 Variation in the performance of the N-D model and the developed model with varying values of $m(t_n)$ during transition from off-peak to peak condition	144
Fig. 5.18 Effect of optimization on the estimated travel time using simulated data	145

	Page
Fig. 5.19 Overall comparison of the proposed model with N-D model using simulated data....	146
Fig. 5.20 Estimated travel time with AVI for 24 hours on February 10, 2003 in link 1	147
Fig. 5.21 Estimated travel time with AVI for 24 hours on February 13, 2003 in link 1	148
Fig. 5.22 Estimated travel time with AVI for 24 hours on February 14, 2003 in link 1	148
Fig. 5.23 Estimated travel time with AVI for peak and transition periods (February 10, 2003) in link 1	149
Fig. 5.24 Estimated travel time with AVI for an off-peak period (February 10, 2003) in link 1	150
Fig. 5.25 Validation of the Travel time estimation model using simulation data for the off-peak condition.....	152
Fig. 5.26 Validation of the travel time estimation model using simulation data for peak condition	153
Fig. 5.27 Travel time estimated by different extrapolation methods for link 1 on February 11, 2003	155
Fig. 5.28 Travel time estimated by different extrapolation methods for link 2 on February 11, 2003	156
Fig. 5.29 Comparison of estimated travel time from extrapolation method, developed method, and AVI using field data on February 13, 2003	157
Fig. 5.30 Comparison of extrapolation and developed model results during afternoon.....	159
Fig. 5.31 Comparison of extrapolation and developed model results during the start of evening peak hours	159
Fig. 5.32 Comparison of extrapolation and developed model results during evening peak hours	160
Fig. 5.33 Comparison of extrapolation and developed model results during transition to evening off peak hours.....	161
Fig. 5.34 Comparison of extrapolation and developed model results with AVI values during off peak hours on February 10, 2003.....	162
Fig. 5.35 Comparison of extrapolation and developed model results with AVI values during peak and transition periods on February 10, 2003.....	162

	Page
Fig. 5.36 Comparison of the extrapolation method with the developed method using simulated data	163
Fig. 5.37 Relation between speed, occupancy, and travel time from February 10, 2003	164
Fig. 6. 1. Schematic diagram of a perceptron	170
Fig. 6. 2. Multi-layer perceptron.....	171
Fig. 6. 3. Model of a neuron	172
Fig. 6. 4. Separating hyperplanes	179
Fig. 6. 5. Support vectors with maximum margin boundary	179
Fig. 6. 6. The kernel method for classification	182
Fig. 6. 7. Log-sigmoid transfer function.....	187
Fig. 6. 8. Travel time distribution on link 1 on all 5 days	192
Fig. 6. 9. Travel time predicted by historic method for link 1 on February 10, 2003	193
Fig. 6. 10. Travel time predicted by real-time method for link 1 on February 10, 2003	194
Fig. 6. 11. Travel time predicted by ANN method for link 1 on February 10, 2003.....	195
Fig. 6. 12. Travel time predicted by SVM method for link 1 on February 10, 2003.....	195
Fig. 6. 13. Comparison of the predicted values with 1 day training data for link 1 on February 10, 2003	197
Fig. 6. 14. MAPE for prediction using 1 day's data for training.....	198
Fig. 6. 15. MAPE for prediction using 2 day's data for training.....	198
Fig. 6. 16. Travel time pattern of Monday, Tuesday, and Friday.....	199
Fig. 6. 17. MAPE for prediction using 3 day's data for training.....	200
Fig. 6. 18. MAPE for prediction using 4 day's data for training.....	201
Fig. 6. 19. Travel time distribution for link 2 from February 10 to February 14, 2003	203
Fig. 6. 20. MAPE for prediction using 1 day's data for training.....	204

	Page
Fig. 6. 21. MAPE for prediction using 2 day's data for training.....	205
Fig. 6. 22. MAPE for prediction using 3 day's data for training.....	205
Fig. 6. 23. MAPE for prediction using 4 day's data for training.....	206
Fig. 6. 24. Speed distribution at 159.998 for 1 week from August 4 to 8, 2003	208
Fig. 6. 25. Performance comparison using 1 day's data for training.....	209
Fig. 6. 26. Performance comparison with 2 day's training data	209
Fig. 6. 27. Performance comparison with 3 day's data for training	210
Fig. 6. 28. Performance comparison with 4 day's data for training	211

LIST OF TABLES

	Page
Table 2.1. Summary of Inductance Loop Detector Failure Mechanisms	13
Table 2.2. Typical Parameters and Threshold Values to Verify Reasonable Sensor Operation ..	15
Table 3.1 Screening Rules	45
Table 4.1 Data Details at the Study Sites Before Optimization	94
Table 4.2 Data Details at the Study sites After Optimization.....	94
Table 4.3 MAPE with Varying Amount of Over Counting Error in the Input Data	98
Table 4.4 Performance Measure at Each Site.....	106
Table 4.5 Imputation of Missing Data Using GRG.....	108
Table 5.1 Data Set to Illustrate the Travel Time Calculation on February 11, 2003.....	130

CHAPTER I

INTRODUCTION

Travel time is a fundamental measure in transportation engineering that can be understood and communicated by a wide variety of audience, including engineers, planners, administrators, and commuters. As a performance measure and decision-making variable, travel time is useful in many aspects of transportation planning, modeling, and decision-making applications. These applications include traffic and performance monitoring, congestion management, travel demand modeling and forecasting, traffic simulation, air quality analysis, evaluation of travel demand, and traffic operations strategies. Travel time information is becoming increasingly important for a variety of real-time transportation applications. These real-time applications include Advanced Traveler Information Systems (ATIS), Route Guidance Systems (RGS), etc., which are part of the Intelligent Transportation Systems (ITS). Thus, providing travelers with accurate and timely information to allow them to make decisions regarding route selection is one of the important applications that use travel time information in recent times.

The increasing reliance on travel time information indicates a need to measure travel time accurately and cost effectively. The traditional method for measuring travel time has been the “test vehicle method,” where travel time is collected manually or automatically using vehicles that are specifically dispatched to drive with the traffic stream for the specific purpose of data collection. Several other travel time measurement techniques have emerged in recent times with the advent of portable computers and fast communication systems. Essentially, the available travel time measurement techniques can be divided into two broad categories, namely, direct methods and indirect methods. In the case of direct methods, travel time is collected directly from the field. These methods include test vehicles, license plate matching, electronic Distance Measuring Instruments (DMI), video imaging, and probe vehicle techniques like Automatic Vehicle Identification (AVI) and Automatic Vehicle Location (AVL). In the indirect methods, travel time is estimated or calculated from other directly measured parameters like speed. Some examples of indirect sources of travel time are Inductance Loop Detectors (ILD), weigh-in-

This dissertation follows the style and format of the *ASCE Journal of Transportation Engineering*.

motion (WIM) stations, and aerial video (Travel Time Data Collection Handbook 1998; Turner 1996; Liu 2000).

While probe vehicle techniques like AVI and AVL have less error, they are more expensive and often require new types of sensors as well as public participation, and hence, they are not widely deployed (Turner 1996). Video imaging techniques lead to public disapproval because of privacy issues. Test vehicle techniques like DMI, even though cost effective, are limited to a few measurements per day per personnel and are only as accurate as the driver's judgment of traffic conditions. Moreover, the test vehicle method and license plate matching are time consuming, labor intensive, and expensive for collecting large amounts of data. Weigh-in-motion stations can collect data from only a selected group of vehicles, whereas aerial photography is not very popular due to the prohibitive cost involved in using it.

On the other hand, freeways in most metropolitan areas in North America are instrumented with loop detectors, which make them the best source of traffic data over a wide area. Hence, even though there are different techniques available for direct measurement of travel time, none of them are as popular and as inexpensive as loop detectors. Also, loop detectors provide an advantage when travel time data collection is required on a continuous basis over a long period of time. However, one of the major criticisms about loop detector data is the high probability of error due to the prevalence of equipment malfunctions. Thus, there is a need to check the accuracy of the data collected by the loop detectors before the data can be used for subsequent applications such as travel time estimation.

1.1 STATEMENT OF THE PROBLEM

Given the extensive use of travel time for numerous transportation applications, and the popularity of ILD data, there is a need to investigate methods to estimate travel time from this data. Also, drivers are interested in knowing how long it will take them to reach their destination, especially in urban freeway conditions, which makes prediction of travel time into future time steps important. However, loop detector data have inherent problems, such as detectors not responding at certain times, undercounting or overcounting the vehicles etc.

This leads to the first step of the present study, namely, checking the loop detector data for accuracy and finding a reliable method to correct any discrepancies. Even though there are many methods suggested in the literature for checking the loop detector data for accuracy at individual locations at a fixed time, there are not many tests available to check these data systematically over space and time. Thus, there is limited knowledge about the accuracy of loop detector data when observed as a series of detectors over a long interval of time, and how this data accuracy affects the accuracy of the final result. When loop detectors are investigated in series, a check for conservation of vehicles becomes necessary. Here, conservation of vehicles implies that the inflow of vehicles into any road section should equal the sum of the outflow of vehicles from that section and the number of vehicles within that section of road. Thus, the cumulative vehicle flow at the downstream location cannot exceed the cumulative flow at the upstream location for the same time period. In addition, the maximum difference between the upstream and downstream location cumulative flows cannot exceed the maximum number of vehicles that can be accommodated on the length of road between these two detector locations. In reality, even after correction at individual locations using standard error correction procedures, vehicle volumes obtained from adjacent loop detector locations may not comply with the conservation principle when the detectors are investigated as a series over a long interval of time. Thus, there is a need for a systematic method to check for the conservation of vehicles and to correct whenever the detector data violate the principle. The method selected should be able to preserve the integrity of the observed data as much as possible while correcting the data to follow the conservation of vehicles principle. Also, this method should be able to handle large amount of data in a systematic manner in a short computation time.

The challenge of providing travelers with accurate and timely travel time information for departure time decision, route selection decision etc., requires faster and more accurate methods of estimating and predicting travel time. There are different methods available to calculate travel time from loop detector data, the most popular among them being extrapolation of the point speed values. However, it is known that the accuracy of these speed-based methods reduces as the vehicle flow becomes larger. Other widely reported methods include statistical models and models based on the traffic flow theory, the majority of which are developed for either free-flow or congested-flow condition only. Thus, there is a need for an economically viable and accurate method of estimating travel time from loop detector data based on the theory of traffic flow that

can be used under varying traffic flow conditions. Also, there is a need for a faster and more accurate method of predicting the estimated travel time into future time steps to inform drivers in real-time about expected traffic conditions. This is important since the success of the ITS applications depends on the accuracy of this predicted travel time.

1.2 RESEARCH OBJECTIVES

This dissertation presents the development of a complete system for the estimation and prediction of travel time from dual-loop detector data. As discussed in earlier sections, there are errors in ILD data, which are unidentified when checked at individual locations. These errors become very significant when considering a series of loop detectors for a long period of time. The first objective of this dissertation work is to monitor a series of loop detectors for a long interval of time to determine whether the conservation of vehicles is satisfied. If there is a violation of the conservation of vehicles, a nonlinear, constrained optimization procedure will be used to correct the discrepancies. This dissertation work is one of the first attempts to monitor a series of detectors for conservation of vehicles and, when violated correct them systematically using an optimization technique. The validity of this technique is verified using simulated data generated using CORSIM simulation software. The detector network and the traffic volumes will be simulated based on the field data in order to mimic the actual field conditions.

The optimized data will be used as input for a theoretical model developed for the estimation of travel time. Travel time will also be estimated using the nonoptimized original data, to illustrate the improvement in the result due to optimization. The estimated travel time is compared with ground-truth data from AVI. Simulated data from CORSIM was also used to check the accuracy of the developed techniques. The estimated travel time will then be forecast to the future time step using Artificial Neural Network (ANN) and Support Vector Machine (SVM) techniques. This dissertation is one of the first studies to explore the use of SVM for the prediction of traffic variables. The accuracy of the predicted results will be verified using the test data used in the ANN and SVM techniques. Based on the details given above, the proposed research objectives can be summarized as follows:

- Development of a nonlinear, constrained optimization method to analyze and correct a series of loop detector data such that the principle of conservation of vehicles is satisfied at all times.
- Simulation of data from CORSIM simulation software to validate the optimization results.
- Development of an analytical model based on traffic flow theory to estimate travel time from loop detector data, taking into account varying traffic flow conditions.
- Validation of the results with field data collected using AVI and simulated data from CORSIM.
- Comparison of the estimated travel time results with results generated using some of the existing methods for travel time estimation.
- Development of a support vector machine model for the forecasting of travel time into future time steps.
- Development of an ANN model for the forecasting of travel time into future time steps.
- Comparison of the SVM and ANN model results to evaluate their performance in the travel time prediction application, with respect to real-time and historic methods.

1.3 RESEARCH METHODOLOGY

1.3.1 Literature Review

A comprehensive review of the literature was conducted covering many aspects of loop detector data collection and the procedures for travel time estimation and prediction. Literature specific to data collection using loop detectors, accuracy of loop detector data, methods for correcting the loop detector data, conservation of vehicles, and methods for travel time estimation and prediction from loop detector data were researched. Also, literature on optimization, ANN, and SVM applications were reviewed.

1.3.2 Study Design and Data Collection

Study design included the selection of the study corridors for data collection. The data were collected from the TransGuide Project area in San Antonio (TransGuide Technical Paper 2002).

Loop detector and AVI data were needed for the present study and, hence, corridors equipped with both loops and AVI were selected. Even though previous studies (Turner et al. 2000; Gold et al. 2001; Eisele 2001) have shown the occurrence of errors in the ILD data, it was reported that the frequency and the amount of missing data are lower at the TransGuide project area in San Antonio, when compared with other freeways in North America that are equipped with loop detectors. The study design task included selection of corridors equipped with both AVI and loop detectors to match the requirements of the present study. The test beds for this dissertation were selected from the I-35 freeway on the northeast side of San Antonio, since that was the only section equipped with both loop detectors and AVI. The selected section is approximately 2 miles in length and included five consecutive detectors in series. The main lane loops were placed approximately 0.5 miles apart. Because the study involves input-output analysis, data from the on-ramp and off-ramp detectors placed between the main lane detectors were also collected. The data were analyzed for continuous 24-hour periods for 5 consecutive days starting from February 10, 2003, Monday to February 14, 2003, Friday. AVI data were also collected from the same section.

Simulated data were generated using the simulation software CORSIM for testing the accuracy of the techniques employed in the work. A traffic network similar to the field test bed was coded in CORSIM, and detectors were placed approximately 0.5 miles apart. Traffic volumes similar to the field volume were generated in CORSIM. The output obtained from CORSIM was used to check the validity of the optimization technique and travel time estimation procedure.

1.3.3 Preliminary Analysis of the Data

After the collection of the field data, extensive data reduction and quality control were required to identify and correct any discrepancies. Quality control and analysis included checking for missing data and threshold checking of the speed, volume, and occupancy observations, individually as well as in combination at individual locations. The polling cycle of the TransGuide detector system during the data collection was 20 seconds, but the cycle occasionally skipped to 60 or 90 seconds, leading to missing data. The threshold value test compared speed, volume, and occupancy in each individual record of the data set with predefined threshold values.

1.3.4 Optimization

After the preliminary checks and corrections, the data were checked as a series for conservation of vehicles. In the present study data from a series of five detectors collected over 24-hour periods were analyzed in order to check whether the data obeyed the vehicle conservation principle. A methodology based on the Generalized Reduced Gradient (GRG) optimization was developed to automatically remove discrepancies from the data when the conservation principle was violated. GRG is a nonlinear optimization technique that can take into account nonlinear constraints. In addition to data correction, this method was useful in identifying the malfunctioning detector locations among all the detector stations analyzed and ranking them based on their performance. This was carried out by checking the data from the individual detector stations and determining which ones get changed the most after optimization. The optimization technique developed can also be used for imputing missing data if a detector location stops recording data for a short time due to malfunctioning. The imputed values will be based on the data from the upstream and downstream detector stations. The usefulness of the developed procedure was first checked using simulated data obtained from the CORSIM simulation software by introducing synthetic errors in the simulated data and then optimizing the data to check for the robustness of the procedure.

1.3.5 Estimation of Travel Time

A travel time estimation procedure was developed which can be used for both peak, off-peak, and transition period traffic flow conditions. The methodology proposed is based on a theoretical model suggested by Nam and Drew (1999) for the estimation of travel time from flow data obtained from loop detectors. The model by Nam and Drew is a dynamic traffic flow model based on the characteristics of the stochastic vehicle counting process and the principle of conservation of vehicles. The model estimates speed and travel time as a function of time directly from flow measurements. This dissertation incorporated several modifications to this theoretical model such that the model can be used for long analysis intervals and for varying traffic flow conditions. The proposed model is based on the traffic flow theory and uses flow, occupancy, and speed data from the detectors as input. An average travel time for all the vehicles that travel during a particular time interval between two selected locations was

calculated as output. The validity of the model was first checked using simulated data obtained from the CORSIM simulation software. Validation of the results while using field data was carried out using AVI travel time data. The performance of the proposed method was also compared with the performance of the extrapolation method, which is the most popular travel time estimation method currently used in the field.

1.3.6 Prediction of Travel Time

There are different techniques that are used for the prediction of traffic parameters including travel time. ANN is one of the most popular methods. ANN is a nonlinear dynamic model that can recognize patterns and can be used for representing complex nonlinear relationships. ANN is mainly used in short-term forecasting because of its ability to take into account spatial and temporal travel time information simultaneously. In the present study, application of a recently developed pattern classification and regression model called support vector regression (SVR) is studied. In SVR the basic idea is to map the nonlinear data into a high-dimensional feature space via a nonlinear mapping and perform linear regression in this space. Thus, linear regression in a high-dimensional (feature) space corresponds to nonlinear regression in the low-dimensional input space. In this dissertation the performance of SVR is compared with the performance of the ANN model suggested in previous studies for the travel time prediction. Also, the results are compared with real-time and historic method results.

1.3.7 Statistical Analysis and Comparison of Results

Statistical analysis was carried out to check the validity of the results. A performance measure of Mean Absolute Percentage Error (MAPE) was used to check the validity of the results. The validity checks were carried out independently at each stage of this dissertation. At the end of the first stage, the validity of the optimization technique was checked using CORSIM data. Detector data were generated in CORSIM, and errors were artificially introduced. The optimization procedure was carried out on this modified simulated data set and was compared to the actual simulated data. In the second stage in which travel time was estimated, validation of the results were carried out with both field data and simulated data. In the case of field data, the results obtained from the model were compared with AVI data and performance measures were calculated. In the case of simulated data, the estimated travel time was compared with the real

travel time calculated from simulation. The estimated travel time was then forecasted into future time steps in the final stage. The accuracy of the prediction methods was checked using the testing data in both ANN and SVM methods.

1.4 CONTRIBUTION OF THE RESEARCH

In North America, loop detectors have become the single largest source of real-time traffic data. However, the reliability of ILD data is questionable for nontraditional uses due to the large amount of discrepancies it contain. For traditional uses such as incident detection, these errors may not pose a big problem. However, for other applications such as travel time estimation or origin-destination estimation from ILD data, these errors may become more important. For such applications, the need for the development of methods to check the loop detector data for discrepancies and diagnosing them in a quick and efficient way cannot be overemphasized. The present study is a step in this direction in which a nonlinear, constrained optimization procedure for removing discrepancies in the detector data is suggested. This optimization method will also be useful to identify detector stations that are malfunctioning, as well as to impute the missing data when one of the detectors is not working for a short period of time. Also, the benefits of the loop detector data cannot be fully utilized without the ability to anticipate traffic conditions, which demand the development of a more accurate method for estimating and predicting traffic conditions. One such important parameter is travel time. The present study proposes a model based on traffic flow theory for the estimation of travel time. The significant feature of this model is its ability to account for the varying traffic flow conditions. An artificial neural network model and support vector regression model are developed for forecasting the parameters into future time steps. Finally, the accuracy of the predicted results is checked with respect to the field data as well as with the results from the existing field methods.

In summary, the contributions of this dissertation can be listed as follows:

- A system level method for the analysis of ILD data for data quality control.
- An efficient and automated optimization method to correct discrepancies in loop detector data, spatially and temporally, when the conservation of vehicles principle is violated.

- A new method to impute the missing data when a full detector location data are missing for a short interval of time, such as 15-30 minutes, and to pinpoint the detectors with the highest rate of malfunctioning.
- A method to prioritize the detector locations for maintenance.
- An analytical model for estimating travel time from loop detector data, which can be used under varying traffic flow conditions.
- An artificial neural network model and support vector regression model for predicting travel time into the future time steps.
- A study on the increase in the accuracy of the estimated travel time with the modifications carried out in this dissertation.
- A study on the applicability of the SVM technique in the prediction of traffic parameters.
- A comprehensive model for the prediction of travel time using ILD data.

1.5 ORGANIZATION OF THE DISSERTATION

This dissertation is organized into seven chapters. Chapter I is an introduction to the research and discusses the background of the problem, statement of the problem, research objectives, research methodology, contributions of the research, and the organization of the dissertation. Chapter II presents a literature review on loop detector, detector data, associated errors, methods for estimation of travel time from loop detector data, and prediction of travel time. Chapter III presents the details of the study corridor and the data collection procedures along with standard data reduction techniques adopted in this dissertation. Chapter IV describes the development of the proposed optimization model for the data analysis. Field data before and after optimization are graphically depicted and a check for the usefulness of the optimization procedure is carried out using simulated data. Chapter V details the existing methods for the estimation of travel time. A detailed description of the model suggested by Nam and Drew (1999), which is the basis for the model developed in this work, is also included in Chapter V. Then the development of the proposed travel time estimation model is described with results and validation. Chapter VI discusses the prediction methodologies adopted in this dissertation. The basics of the ANN and SVM techniques are briefly introduced, and the development of the ANN and SVM prediction procedures are explained next. The results obtained from ANN and SVM are

compared, and performance measures are calculated for each method. Chapter VII provides conclusions and recommendations based on the research. Suggestions for further research are also included in this chapter. The references are followed by a glossary of frequently used terms and acronyms used. The appendices also include MATLAB and C programs developed for the optimization procedure, estimation model, SVM and ANN techniques, and the C-programs developed for the extraction of travel time from CORSIM output.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter reviews the literature related to the various aspects of ILD data collection and its application in travel time estimation and prediction. The chapter starts with a summary of the literature on loop detector data accuracy and the different methods available for screening the data. It is followed by a review of some of the important investigations on data diagnostics. The methods available in the literature for the estimation of travel time from detector data are reviewed next. The final section of this chapter discusses the different methods reported for predicting travel time.

2.2 ILD DATA AND DATA ERRORS

National Electrical Manufacturers Association (NEMA) standards define a vehicle detector system as "... a system for indicating the presence or passage of vehicles." These systems provide input for traffic-actuated control, traffic responsive control, freeway surveillance, and data collection systems (NEMA 1983). Detectors have been used for highway traffic counts, surveillance, and control for the last 50 years (Labell et al. 1989). The three main types of vehicle detectors used in current practice are inductance loop detectors, magnetic detectors, and magnetometers. Of these, the most widely used is the inductance loop detector system (Raj and Rathi 1994; Traffic Detector Handbook 1991). Singleton and Ward (1977) reported a survey of available vehicle detectors and a comparison of their detection characteristics, physical installation parameters, operational characteristics, and relative costs.

The data supplied by conventional inductance loop detectors include vehicle presence, vehicle count, and occupancy. Although loops cannot directly measure speed, speed can be estimated by using a two-loop speed trap (dual-loop detectors) or a single-loop detector and an algorithm whose inputs are effective loop length, average vehicle length, time over the detector, and the number of vehicles counted (Klein 2001). Loop data are typically relayed to a centralized

Traffic Management Center (TMC) for analysis. Although the loops are read many times per second, the data are usually accumulated and amplified at the pull box and then reported to the TMC at intervals of 20 to 30 seconds.

Some of the limitations of ILD are well documented in the literature. For instance, Middleton et al. (1999) evaluated ILD performance and emphasized the need for proper saw cutting and careful installation to ensure proper use, the need for extensive maintenance and calibration, and the need for traffic control when repairs are needed. The quality of the data recorded by the ILD is affected by any malfunctions arising from problems like improper installation, wire failure, inadequate loop sealants, etc. Quality control procedures become difficult for loop detector data because (1) the large volume of data makes it difficult to detect errors using traditional manual techniques and (2) the data collection is continuous, which makes equipment errors and malfunctions more likely than that experienced in data collection techniques that are performed occasionally. Table 2.1 (Klein 2001) below shows a summary of the common failure reasons reported.

Table 2.1. Summary of Inductance Loop Detector Failure Mechanisms

State	Major failure
Alaska	No loop failures reported
California	Improper sealing and foreign material in saw slot
Idaho	Improper sealing
Montana	Improper sealing
Nevada	Improper sealing and pavement deterioration
Oregon	Improper sealing
Utah	Improper sealing and pavement deterioration
Washington	Improper sealing and foreign material in saw slot

Since the introduction of electronic surveillance in the roadways in the 1960s, procedures that examine the detector output as well as the collected data for errors have continued to evolve. Some of the earlier studies conducted on loop detector data errors, its causes and effects include

Dudek et al. (1974), Courage et al. (1976), Pinnell-Anderson-Wilshire and Associates (1976), Bikowitz and Ross (1985), Chen and May (1987), Gibson et al. (1998). Bender and Nihan (1988) performed a literature review on state-of-the-art inductance loop detector failure identification and on the methods to identify inaccurate data resulting from loop detector malfunctions. Berka and Lall (1998), while discussing video-based surveillance, reported that the reliability of loop detectors is low. These studies confirmed the common concerns associated with the accuracy of loop detector data and illustrated the importance and need for understanding the amount of error in the data and diagnosing these errors before using the data for specific applications.

Jacobson et al. (1990) divided loop detector data errors and screening tests into two main categories: microscopic and macroscopic. Microscopic level screening tests occur at the microprocessor level, where detector pulses are scanned and checked for errors. For example, pulses or gaps in actuation less than some predefined interval may be ignored. These checks are performed in the field. Macroscopic screening tests typically occur at the central computer center after the data have been aggregated over time.

Studies on loop detector data errors at the microscopic level usually require reprogramming and/or modification of the detector device and depend on the type of loop detector. One approach is to check the on-time of the detectors, against either an average value or a predefined constant (Chen and May 1987; Coifman 1999; Nihan et al. 1990). May et al. (2003) reported the use of three microscopic tests: detection of vehicle presence for less than 1/15 second, an absence of vehicle lasting less than 1/15 second, and more than two valid pulses in a second. Nihan and Wong (1995) developed an error-screening algorithm based on the density to occupancy ratio and the volume to capacity ratio. Ametha (2001) developed an algorithm to detect errors at the microscopic level, based on an average vehicle length, and comparing this value with a calibrated threshold range. Zhang et al. (2003) proposed a detector event data collection system that can sample loop actuations with a sampling rate of 60 Hz or higher, thus enabling real-time data collection at the microscopic level. However, macroscopic approaches are more commonly adopted since they are independent of the sensor type and are carried out at the data processing level (Peeta and Anastassopoulos, 2002).

Common macroscopic studies compare volumes, occupancies, or speeds with specific threshold values. Usually, a single parameter, such as speed, volume, or occupancy, is compared to predefined upper and lower system threshold values. For example, typical maximum threshold values are 3000 vehicles/hour for volume, 80 mph for speed, and 90% for occupancy (Jacobson et al. 1990; Park et al. 2003; Turochy and Smith 2000; Eisele 2001). Payne et al. (1976) extended the malfunction detection algorithms from individual locations to comparison checks between the neighboring sensors. They reported a comparison of occupancy values at adjacent locations to check for intermittent malfunctions, which often remain unidentified if considered as individual locations. Bellamy (1979) reported a study on the undercounting of vehicles with single-loop detector data and suggested a theoretical method for calculating the probability of a vehicle not being detected. Table 2.2 (Klein 2001) gives the details of typical parameters and threshold values used to verify reasonable sensor operation.

Table 2.2. Typical Parameters and Threshold Values to Verify Reasonable Sensor Operation

Parameter	Typical threshold value
Maximum flow rate	2400 vehicles/hour
Minimum flow rate	0 vehicles/hour
Maximum occupancy	50%
Minimum occupancy	10%
Maximum speed	80 miles/hour
Minimum flow rate to conduct high-speed check	10 vehicles/hour
Maximum number of actuation periods	5 cycles
Maximum number of constant call periods	5 cycles
Maximum number of presence periods	5 cycles

The main disadvantage of single-parameter threshold tests, which take into account only one parameter at a time, is that they assume that the acceptable range for a parameter is independent of the values of the other parameters. Because combinations of parameters are not tested, single-parameter threshold tests cannot identify unreasonable combinations. Typically, the combinations of parameter tests take advantage of the relationships among the three parameters,

namely, mean speed, volume, and occupancy (Jacobson et al. 1990; Cleghorn et al. 1991; Payne and Thompson 1997; Turochy and Smith 2000; Turner et al. 2000; Coifman and Dhoorjaty 2002). All of these approaches involve either setting limits for acceptable values of one parameter within a given range of another parameter or a combination test such as zero occupancy with nonzero volume or zero volume with nonzero speed, etc. Other approaches include the application of a Fourier-based fault-tolerant framework for detecting and correcting data errors, multivariate screening methods based on the variants of the Mahalanobis distance, etc. (Gupta 1999; Peeta and Anastassopoulos 2002; Park et al. 2003).

The detector malfunctions discussed above deal with discrepant data from ILD. Another malfunction could be related to the nonresponse of the detectors for a short period of time due to various reasons, which results in a loss of data for that time period. Turner et al. (1999, 2000), in their study on the quality of the archived ITS data, indicated that missing data values (nonresponsive) are a common occurrence. They reported that on average, more than 20% of the volume and the speed data from the TransGuide traffic monitoring system were not available for numerous reasons, ranging from equipment failure to communication disruption to software failure. Gold et al. (2001) described various methods for imputing nonresponse in traffic volumes occurring in intervals of less than 5 minutes using factor-up and straight-line interpolation methods as well as polynomial and kernel regression. Bellemans et al. (2000) constructed a function that approximated the available data points and used that for calculating the missing values. Smith and Conklin (2002) used time-of-day historical average lane distribution patterns at a particular location coupled with current available detector data, to estimate missing detector data. Other reported techniques for the imputation of data include time series models, artificial neural network techniques, and regression analysis (Sharma et al. 2003; Chen et al. 2003). Van Lint et al. (2003) reported the use of an exponential smoothing and spatial interpolation method for the imputation of missing data. Smith et al. (2003) reported a comparison of several heuristic and statistical imputation techniques.

All the above correction procedures are applied for a single location and therefore cannot account for systematic problems over a series of detectors. For instance, if the total number of vehicles counted by two consecutive detector locations is observed over a period of time, the difference in the cumulative counts should not exceed the number of vehicles that can be

accommodated in that length of the road under the jam density condition. This follows directly from the application of the principle of conservation of vehicles. However, it can be shown that this constraint can be violated if some of the detectors are under- or over-counting vehicles. For many traffic applications, such as incident detection, this might not be an issue. However, for other applications that rely on accurate system counts, such as origin-destination (OD) estimation, travel time estimation, etc. this can be problematic. While most of the existing error detection and diagnostic tests do take into account possible malfunctions of the loop detector by observing the data at a specific point, the problems related to balancing consecutive detector data for vehicles being under- or over-counted has not been well addressed. The lack of interest in this area may be due to the requirements of most of the currently practiced traffic flow models that rely on data generated at a particular station point rather than from a series of station points at the same time.

Given the considerable reach of ILD data in ITS applications, very few investigations have been carried out to analyze the detector data as a series and check whether the conservation of vehicles is followed (Zuylen and Brantson 1982; Petty 1995; Cassidy 1998; Windover and Cassidy 2001; Zhao et al. 1998; Nam and Drew 1996, 1999; Kikuchi et al. 1999, 2000; Kikuchi 2000; and Wall and Dailey 2003). While all the above studies acknowledged the fact that conservation is violated, few of them (Zuylen and Brantson 1982; Petty 1995; Nam and Drew 1996, 1999; Kikuchi et al. 1999, 2000; Wall and Dailey 2003) discuss methods for correcting the data in such situations. Zuylen and Brantson (1982) developed a methodology that relied on an assumption about the statistical distribution of the data to eliminate the discrepancy in the data. Petty (1995), Nam and Drew (1996, 1999), and Wall and Dailey (2003) used a simple adjustment factor for correcting the discrepancy. Kikuchi et al. (1999) and Kikuchi (2000) studied an arterial signalized network and proposed methodologies to adjust the observed values using the concept of fuzzy optimization. Kikuchi et al. (2000) discussed different methods to correct the discrepancy, including a fuzzy optimization technique, and they demonstrated this approach on single hourly flow from an arterial network with 13 intersections. The investigation by Wall and Dailey (2003) required one properly calibrated reference detector that can be assumed to be correct in order to calculate the correction factor. However, it is almost impossible to detect the malfunctioning detector or properly working detector from the loop

detector data. The only way to determine this is to go to the field and measure the data, which is cost prohibitive.

From the above detailed review of literature, it is clear that there are only very few investigations related to checking loop detectors at a system level for errors due to violation of conservation of vehicles. As the aim of this dissertation is to develop methodologies for estimating and forecasting travel time from ILD data, there is a need to consider series of detectors. Thus, the accuracy of the data with respect to neighboring detectors needs to be considered. Hence, a procedure will be developed to investigate a series of detectors and a systematic method will be developed and implemented if the data violate the conservation of vehicles. The methodology developed will be applied in such a manner that the integrity of the original data will be least changed. Once the data are checked for violation of the conservation of vehicles principle and corrected, they will be used for estimation of travel time. In the following section, literature related to the existing methodologies for the estimation of travel time from ILD data is reviewed.

2.3 TRAVEL TIME ESTIMATION FROM ILD DATA FOR FREEWAYS

The different methods reported for the estimation of travel time from ILD can be broadly divided into three main categories, namely, extrapolation methods, statistical methods, and methods based on traffic flow theory.

2.3.1 Extrapolation Methods

Travel Time Data Collection Handbook (1998) reports the extrapolation technique as the simplest and most widely accepted method for the estimation of travel time from inductance loop detector data. The extrapolation method is based on the assumption that speed can be assumed to be constant for the small distance between the measurement points, usually the distance between the two detector stations (approximately 0.5 miles). Since the distance between the two detectors is known a priori, the travel time is calculated as the distance divided by the speed (Sisiopiku et al. 1994; Ferrier 1999; Quiroga 2000; Lindveld and Thijs 1999; Dhulipala 2002; Cortes et al. 2002; Van Lint and van der Zijpp 2003; Lindveld *et al.* 2000; Dailey 1997). Thus,

the extrapolation methods assume that the point estimate of speed is representative of the average speed between the adjacent loop detectors.

The three different extrapolation approaches normally adopted at present are explained below with the help of Figure 2.1.

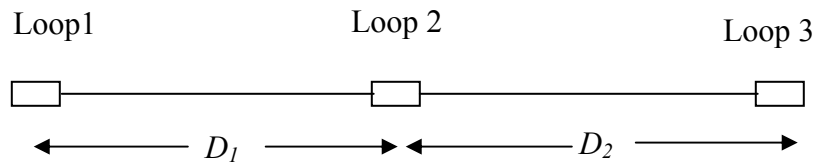


Fig. 2.1 Schematic diagram for illustrating extrapolation methods

2.3.1.1 Half-Distance Approach

The half distance method assumes that the speed measured by a specific set of dual-loop detectors is applicable to half the distance on both sides. Thus, the travel time between detectors 1 and 2 is defined as

$$T_{1-2} = \frac{1}{2} \left(\frac{D_1}{v_1} + \frac{D_1}{v_2} \right), \quad (2.1)$$

where,

v_1 and v_2 = average speed measured at loops 1 and 2, respectively, for the time interval,
and T_{1-2} = travel time from detector 1 to detector 2.

2.3.1.2 Average Speed Approach

In this method the average speed of the vehicles traveling in segment 1 to 2 is assumed to be the average of the speeds measured by detector stations 1 and 2. Hence, the travel time between loop 1 and loop 2 is given as

$$T_{1-2} = \frac{D_1}{(v_1 + v_2)/2}, \quad (2.2)$$

where,

v_1 and v_2 = average speeds at loop 1 and loop 2 respectively.

2.3.1.3 Minimum Speed Approach

In this method, the minimum of the speeds reported by the two detectors is assumed to be the speed of the vehicles traveling in the segment from detector 1 to 2. Thus, the travel time between loop 1 and loop 2 is

$$T_{1-2} = \frac{D_1}{v_{\min}}, \quad (2.3)$$

where,

v_{\min} = minimum speed among v_1 and v_2 .

The main disadvantage of the extrapolation methods is reported as the decreasing performance with increasing flow conditions (Ferrier 1999; Lindveld and Thijs 1999; Lindveld et al. 2000). Quiroga (2000) indicated a significant discrepancy between travel time displayed on the dynamic message signs calculated based on the extrapolation method and the actual travel time during peak periods. While the discrepancy was very small during off-peak periods, it was significant during the peak periods, suggesting the need for a more accurate algorithm during the peak periods. This is due to the fact that the extrapolation method is applicable only when the variation in the traffic condition is lower such as in off-peak hours. Also, the assumption of constant speed between the detection points holds true only at low to moderate volume

conditions, where the variability in the flow is lower (Coifman 2002; Oh et al. 2003; Eisele 2001). At high volume conditions the variation of speed can be such that the assumption of a constant speed no longer holds true even for a small section of road. Thus, the error in the travel time results calculated using extrapolation methods tends to increase during congested periods due to their failure to capture the congestion occurring between the detector stations.

2.3.2 Statistical Methods

Statistical methods apply different statistical methods such as time series analysis, the Kalman filtering method, etc., for estimation of travel time from ILD data. Dailey (1993) demonstrated the viability of using cross-correlation techniques with inductance loop data to measure the propagation time of traffic. Van Arem et al. (1997) proposed a linear input-output autoregressive moving average (ARMA) model to estimate travel time from loop detector data. Other major methods include the use of signature matching techniques (Coifman 1998; Coifman and Cassidy 2002; Kuhne et al. 1997; Sun et al. 1998, 1999, 2003; Abdulhai and Tabib 2003; Pfannerstill 1989; Takahashi et al. 1995; May et al. 2003), ANN (Blue et al. 1994), and fuzzy logic and neural networks (Palacharla and Nelson 1999) for the estimation of travel time from loop detector data.

2.3.3 Theoretical Models

Theoretical models are developed for the estimation of travel time from loop detector data based on traffic flow theory. The advantage of these models is that since they are based on the traffic flow theory, they can capture the dynamic characteristics of traffic. Most of these models apply the conservation of vehicles principle and compare the inflow of a section during previous time period with its outflow during the current time period (Bovy and Thijs 2000). Nam and Drew (1996, 1998, 1999) presented a macroscopic model for estimating freeway travel time in real-time directly from flow measurements based on the area between the cumulative volume curves from loop detectors at either end of the link. Petty et al. (1998) suggested a model for estimating travel time directly from flow and occupancy data, based on the assumption that the vehicles that arrive at an upstream point during a given interval of time have a common

probability distribution of travel times to a downstream point. Hoogendoorn (2000) reported a model that filtered the data into different classes like cars, trucks, etc., and class-specific travel times were calculated using the extended Kalman filtering technique. The study reported fairly reasonable results during free-flow and near free-flow conditions; however, the accuracy decreased as congestion started. Coifman (2002) utilized the linear approximation of the flow-density relationship to estimate travel time from dual-loop detector data. The results were reported as satisfactory except at the transition periods from congested to uncongested and vice versa. Oh et al. (2003) discussed the estimation of travel time based on fluid model relations using the true density in a section calculated from the cumulative counts at two detector points.

Most of the theoretical studies used for travel time estimation from loop detector data give satisfactory results for specific conditions only. For instance, some of the models perform well for normal-flow conditions only (Nam and Drew 1996, Hoogendoorn 2000, Oh et al. 2003), while some other models are applicable for congested traffic flow conditions only (Nam and Drew 1998). Thus, there is a need for a comprehensive model that can be used for estimating travel time under varying traffic flow conditions. The proposed study suggests such a model, which can be used for both normal and congested traffic flow conditions. The basis for this work is the dynamic traffic flow model developed by Nam and Drew (1999). A detailed discussion of this model will be given in Chapter V.

2.4 PREDICTION OF TRAVEL TIME ON FREEWAYS

The effectiveness of the ATIS and the degree to which drivers are willing to follow the suggested routes depends on the accuracy and timeliness of travel time information provided. Real-time information on current traffic conditions can be useful to drivers in making their route decisions, but traffic conditions can fluctuate over short time periods. There may be a substantial difference between the current link travel time and the travel time on the link when traversed after a short time. Hence, accurate predictions are more beneficial than current travel time information since conditions may change significantly before travelers complete their journey. There are several methods available for the prediction of travel time, and they can be broadly classified into historic profile approaches, regression analysis, time series analysis, and

ANN models. An overview of literature in these commonly used methods for predicting travel time is discussed below.

2.4.1 Historic and Real-Time Approaches

The historic profile approach is based on the assumption that a historic profile can be developed for travel time that can represent the average traffic characteristics over days that have a similar profile. Thus, a historical average is used for predicting the future values. An important component of the historic approach is the classification of days into day types with similar profiles. The travel time is defined by:

$$T(t, \Delta) = \bar{T}(t), \quad (2.4)$$

where,

$$\begin{aligned} \bar{T}(t) &= \text{average of past travel time at time } t, \text{ and} \\ T(t, \Delta) &= \text{travel time at time step } \Delta. \end{aligned}$$

This approach is relatively easy to implement and it is fast in terms of computation speed. Also, this method can be valuable in the development of prediction models since they explain a substantial amount of the variation in traffic over time periods and days (Shbaklo et al. 1992). However, for the same reason, the value of static prediction is limited because of its implicit assumption that the projection ratio remains constant. Commuters in general have an idea about the average travel time under usual traffic conditions and will be interested in conditions under not-so-common conditions, that is, when average values are not representative of the current or future traffic conditions. Thus, the historic method performs reasonably well under normal conditions; however, it can misrepresent the conditions when the traffic is abnormal.

In the real-time approach, it is assumed that the travel time from the data available at the instant when prediction is performed represents the future condition. Thus, travel time at one time step ahead is given by

$$T(t, \Delta) = T(t), \quad (2.5)$$

and travel time at two time steps ahead is given by

$$T(t, 2\Delta) = T(t), \quad (2.6)$$

where,

Δ = the time step.

This method can perform reasonably well for the prediction of travel time for the next few time steps under traffic flow conditions without much variation. Hoffman and Janko (1990) and Kaysi et al. (1993) discussed the use of historical data profiles as part of the development of a prediction algorithm for the Leit und Information system Berlin (LISB) system in West Berlin. Their approach involved the creation of a historical data profile to develop a prediction algorithm. The Advanced Driver and Vehicle Advisory Navigation ConcEpt (ADVANCE) project in the Chicago metropolitan area used the historic and real-time data for their travel time prediction model (Thakuriah et al. 1992; Tarko and Roupail 1993; Boyce et al. 1993). Seki (1995) used historical data after correcting them by type of day for prediction of travel time. Manfredi et al. (1998) developed a prediction system as part of the Development and Application of Coordinated Control of Corridors (DACCORD) project, where the prediction was based on historical and current data. The Urban Traffic Control System (UTCS) (Kreer 1975; Stephanedes et al. 1981), as well as in several traveler information systems in Europe including AUTOGUIDE (Jeffrey et al. 1987) also used historic and real-time methods.

2.4.2 Regression Analysis Method

Most of the conventional forecasting techniques may be classified as regression based. This method necessitates the identification of relevant variables that are strongly correlated to travel time, such as flow, occupancy, travel time in the neighboring links, etc. Kwon et al. (2000) presented an approach to predict travel time using linear regression with a stepwise variable selection method using flow and occupancy data from loop detectors and historical travel time information. Application of a linear regression model for short-term travel time prediction was also discussed by Rice and van Zwet (2002) and Zhang and Rice (2003). They proposed a

method to predict freeway travel time using a linear model in which the coefficients vary as smooth function of the departure time.

2.4.3 Time Series Models

The existing link travel time forecasting models also include time series and Kalman filtering models. The time series method of travel time forecasting involves the examination of historical data, extracting essential data characteristics, and effectively projecting these characteristics into the future. This technique predicts the travel time in a future time step from the travel time at a current time step and previous time steps. The Box and Jenkins technique is a widely used approach, and the most widely employed method under this category is the AutoRegressive Integrated Moving Average Model (ARIMA) method (Sen et al. 1991; Anderson et al. 1994). Oda (1990) adopted an auto regressive model for the prediction of travel time. Iwasaki and Shirao (1996) discussed a short-term prediction scheme of travel time on a long section of a motorway using an auto regressive method. The parameters of the prediction model were identified adapting an extended Kalman filtering method. Angelo et al. (1998), Al-Deek (1998), and Ishak and Al-Deek (2002) implemented models using nonlinear time series with multifractal analysis for the prediction of travel time. Saito and Watanabe (1995) developed a system for predicting the travel time for 60 minutes ahead using an auto regression model based on the change in traffic conditions for the previous 30 minutes. Another important technique widely used is the Kalman filtering technique (Yasui et al. 1995; Chen and Chien 2001; Chien and Kuchipudi 2002, 2003; Chien et al. 2003; Nanthawichit et al. 2003).

2.4.4 Neural Network Models

The ANN model, with its learning capabilities, is reported to be suitable for solving complex problems like travel time prediction. Different traffic variables are used as input to the ANN model in the travel time prediction problems. Some of the reported researches about the application of ANN for the prediction of freeway travel time are discussed below.

Cherrett et al. (1996) reported the use of a feed-forward ANN model for the prediction of link journey time. Ohba et al. (1997) proposed a travel time prediction model using a mixed structure type neural network system. Park and Rilett (1998, 1999), Park et al. (1999), Rilett and Park (2001), and Kisgyorgy and Rilett (2002), suggested different modifications to the basic ANN model to take into account the nonlinear nature of travel time data for travel time prediction. The modifications included feed-forward neural network, clustering techniques combined with ANN, or modular neural network (MNN), and ANN that incorporated expanded input nodes, or spectral basis neural network (SNN). Matsui and Fujita (1998) used a neural network-driven fuzzy reasoning for the prediction of travel time. You and Kim (2000) developed a hybrid travel time forecasting model based on nonparametric regression and Geographic Information System (GIS) technology. Zhu (2000) tested the use of feed-forward neural network for the prediction of travel time. Innamma (2001) studied the predictability of travel time based on online travel time measurements with video using neural networks. Huisken and van Berkum (2002) discussed a method to predict travel time within a short-term range with ANN models using flow and speed values. Van Lint et al. (2002) investigated travel time prediction using state space neural networks. Dharia and Adeli (2003) reported the use of counter-propagation neural networks for the forecasting of freeway link travel time.

2.5 CONCLUDING REMARKS

Some of the significant investigations on ILD data, the errors associated with them, the available methodologies for correcting these errors, travel time estimation, and prediction using ILD data were reviewed in this chapter. From the discussion in this chapter, it was clear that the available diagnostics for the ILD data mainly aim at errors at individual detector locations. There is a lack of systematic analysis of detectors in a series and the errors associated with them. The few studies that analyzed the detectors as a series and checked for conservation of vehicles did not suggest a systematic procedure to correct the data when the conservation of vehicles is violated.

The chapter also reviewed the literature on the estimation and prediction of travel time from ILD data. One observation that can be made from these studies was that different techniques perform well for varying traffic flow conditions. One of the important objectives in this dissertation is to

develop a procedure that will estimate travel time for varying traffic flow conditions. Also, for the prediction of travel time to future time steps, ANN is the most popular technique suggested in literature. In this dissertation, a new technique called support vector regression will also be investigated and its performance will be compared with the predictions of ANN. Details of this will be elaborated in Chapter VI. The following chapter will describe the study corridors, data collection, and data quality control. The details of the analysis will be discussed in the following chapters.

CHAPTER III

DATA COLLECTION AND PRELIMINARY DATA ANALYSIS

3.1 INTRODUCTION

The main objective of this dissertation is to develop techniques/methodologies for the estimation and prediction of travel time from ILD data. Thus, the main data of interest in this dissertation are ILD data collected from the field. As discussed in Chapter II, one of the main concerns about the use of ILD data is the low quality. This chapter will discuss the details of the ILD, ILD data, different preliminary quality control and error correction procedures employed, etc. For the validation of the results, 'ground-truth' data collected using AVI are used. Details related to AVI data collection and their quality controls are also detailed in this chapter.

One disadvantage of using the AVI data for validation purposes in the present study is the difference in the sample sizes of ILD and AVI. ILD data include all the vehicles that cross the two points of interest. Hence, the average travel time calculated will be based on the complete population of vehicles. In the case of AVI, only few participating vehicles will be identified, and hence, the sample size is smaller. Also, the AVI data and ILD data may not match exactly, spatially or temporally. Thus, the comparison between the AVI travel time and the travel time calculated from ILD data cannot shed much light on the accuracy of travel time calculated from ILD data. Hence, this dissertation used simulated data using the CORSIM simulation software, in addition to the use of filed data, for the validation of the models. The details of the simulation software used, the actual process of simulation, and the details of the simulated data were also explained in this chapter. Thus, this chapter will describe the different data sources, and the test corridors, the characteristics of the data, as well as the preliminary data reduction and quality control.

3.2 FIELD DATA SOURCES

3.2.1 Inductance Loop Detector

Detectors have been used for highway traffic counts, surveillance, and control for the last 50 years (Labell et al. 1989). The initial developments in the detection technology were based on sound, pressure on the road surface, weight of vehicles, etc. Over the years different types of detectors such as acoustic detectors (based on sound), optical detectors (based on opacity), magnetic detectors (based on geomagnetism), infrared, ultrasonic, radar, and microwave detectors (based on reflection of radiation), inductance loop detectors (based on electromagnetic induction), seismic, and inertia-switch detectors (based on vibration), etc., have been developed. The three main types of vehicle detectors used in current practice are inductance loop detectors, magnetic detectors, and magnetometers. Out of these, the most popular is the inductance loop detector system (Traffic Detector Handbook 1991).

The principal components of an inductance loop detector system include one or more turns of insulated loop wire wound in a shallow slot sawed in the pavement, a lead-in cable that runs from the curbside pull box to the intersection controller cabinet, and a detector electronics unit housed in the intersection controller cabinet. The wire loop is an inductive element in an oscillatory circuit and is energized by the electronics unit at frequencies that range from 10 to 200 kHz. When a vehicle passes over the loop or stops within the loop, it decreases the inductance of the loop. This decrease in inductance then actuates the detector electronics output relay, sending a pulse to the controller unit, signifying that it has detected the passage or presence of a vehicle (Traffic Detector Handbook 1991). When a vehicle enters the detection zone, the sensor is activated and remains so until the vehicle leaves the detection zone. The 'on' time, referred to as the vehicle occupancy time, requires the vehicle to travel a distance equivalent to its length plus the length of the detection zone. The vehicle occupancy time is a function of vehicle speed, vehicle length, and detection zone length. Controllers measure the time of the transition from 'off' to 'on' and back to 'off' of the pulses. Traffic flow parameters such as flow and occupancy are then calculated from these data (May 1990).

The two main types of inductance loop detectors in use are single-loop detectors and dual-loop detectors. In the case of single-loop detectors, a single-loop of wire is used at each detector location, whereas in the case of dual-loop detectors, two such loops are placed a small distance apart at each detector location. Figure 3.1 shows a picture of a single-loop detector placed in field. Figures 3.2 and 3.3 show schematic representations of single-loop and dual-loop detectors.



Fig. 3.1 Actual loop in the field

(Source: <http://transops.tamu.edu/content/sensors.cfm>)

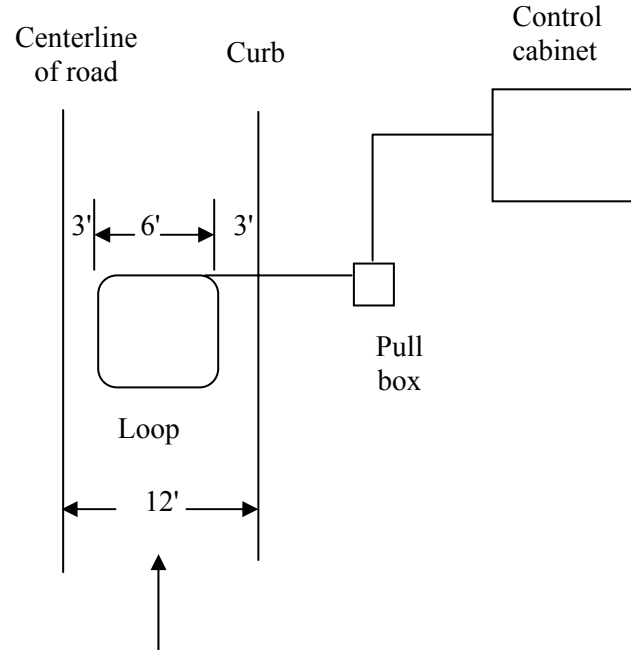


Fig. 3.2 Schematic diagram of a single-loop detector in one lane of a roadway

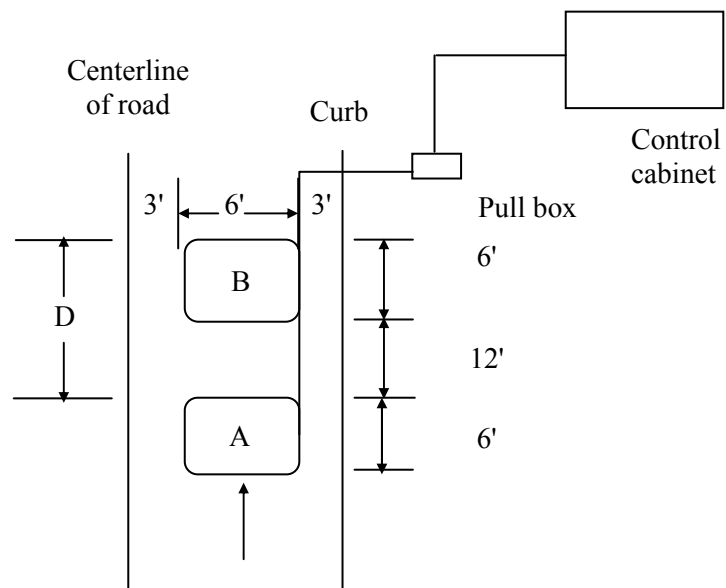


Fig. 3.3 Schematic diagram of a dual-loop detector

The data supplied by the conventional single inductance loop detectors are vehicle passage, presence, count, and occupancy. The single-loops cannot measure speed directly, but is estimated based on an algorithm whose inputs are effective loop length, average vehicle length, time over the detector, and the number of vehicles counted (May 1990). The following equations are used to calculate the parameters such as flow, density, and speed from the single-loop detector data.

$$q = \frac{1}{\sum_{n=1}^N \frac{(t_{on})_{n+1} - (t_{on})_n}{N-1}}, \quad (3.1)$$

$$(t_{occ})_n = (t_{off})_n - (t_{on})_n, \quad (3.2)$$

$$v_n = \frac{L_n + L_d}{(t_{occ})_n}, \quad (3.3)$$

where,

- q = flow (vehicles per second),
- N = total number of vehicles,
- $(t_{occ})_n$ = individual occupancy time (seconds),
- $(t_{on})_n$ = instant of time the vehicle n is detected (seconds),
- $(t_{off})_n$ = instant of time the vehicle is exited (seconds),
- v_n = vehicle speed (feet per second),
- L_n = vehicle length (feet), and
- L_d = detection zone length (feet).

In the case of dual-loop detectors, flow and occupancy are reported when the vehicle crosses the first loop of the dual trap. Speed calculations are made when the vehicle passes the second loop, based on the known distance between the two loops and the time taken to travel from the first loop to the second loop (Texas Department of Transportation (TxDOT) 2000; Sreedevi and Black 2001).

Thus, flow and occupancy are calculated in an identical manner to the single-loop detector, as given in equation 3.1 and 3.2. The speed is calculated as follows:

$$v_n = \frac{D}{[(t_{on})_n]_B - [(t_{on})_n]_A}, \quad (3.4)$$

where,

A = first loop in the dual-loop detector,

B = second loop in the dual-loop detector, and

D = distance from the upstream edge of detection zone A to the upstream edge of detection zone B (feet).

A local control unit (LCU) accumulates speed, occupancy, and volume from the detector channels, keeps a moving average of these measurements, and sends the data to traffic management center (TMC) at intervals of 20 to 30 seconds for analysis with a computer-based algorithm (TransGuide Technical Brochure 2000). From this, the flow rate, percent occupancy and average speed for that particular time interval are calculated. Percent occupancy is a surrogate for density and is obtained by determining the percent of time a detector is occupied and is calculated as follows (May 1990):

$$O = \frac{\sum_{n=1}^N (t_{occ})_n}{\Delta t} \times 100, \quad (3.5)$$

where,

O = percent occupancy time,

$(t_{occ})_n$ = individual occupancy time (seconds),

N = number of vehicles detected, and

Δt = selected time period (seconds).

Density is calculated from the percent occupancy as

$$k = \frac{52.8}{\overline{L}_v + L_d} O, \quad (3.6)$$

where,

k = density (vehicles per lane-mile), and

\overline{L}_v = average vehicle length (feet).

The San Antonio corridors are equipped with dual inductance loop detectors at approximately 0.5-mile spacing. The loop detectors used at the I-35 site are 6 feet by 6 feet (1.83 m by 1.83 m) buried 1 inch (2.54 cm) below the road surface, and are centered in each lane. The two loops in the dual-loop detectors are installed 12 feet (3.66 m) apart longitudinally and are made up of differing number of turns to minimize cross talk. The loop detector signals are sent to LCUs, where the data are analyzed to determine volume, occupancy, and speed for that 20-second interval. The LCU also continually checks the loops for “long” periods of continuous presence or complete lack of presence, which may indicate loop detector problems. Speed values are reported when vehicles pass the second loop, and volume and occupancy data are reported from the first loop detector (TransGuide Technical Brochure 2000).

The format of the raw data collected from the field is shown in Figure 3.4. The first and second columns pertain to the date and time, respectively. The third column shows the detector number, which includes details such as to whether the detector is on an exit ramp (EX), entry ramp (EN), or on the main lane (L). The lanes are numbered in increasing order from the median to the curb as L1, L2, L3, etc. The interstate name and mile marker are also provided in the detector number. The speed, volume, and occupancy values are indicated in the fourth, fifth, and sixth columns, respectively, for every 20-second period.

02/10/2003	00:00:28	EX1-0035N-166.829	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:28	EX2-0035N-166.836	Speed=60	Vol=000	Occ=000
02/10/2003	00:00:28	L2-0035S-166.833	Speed=87	Vol=000	Occ=000
02/10/2003	00:00:28	L3-0035N-166.833	Speed=61	Vol=002	Occ=005
02/10/2003	00:00:28	L3-0035S-166.833	Speed=70	Vol=002	Occ=002
02/10/2003	00:00:29	EX1-0035N-168.108	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:29	EX1-0035S-167.857	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:29	L1-0035N-167.942	Speed=71	Vol=001	Occ=001
02/10/2003	00:00:29	L2-0035N-167.942	Speed=76	Vol=001	Occ=001
02/10/2003	00:00:29	L3-0035N-167.942	Speed=77	Vol=001	Occ=001
02/10/2003	00:00:29	L4-0035N-167.942	Speed=64	Vol=001	Occ=001
02/10/2003	00:00:30	EN1-0035N-169.306	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:30	EX1-0035S-169.286	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:31	EN1-0035N-170.580	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:31	EX1-0035N-170.148	Speed=-1	Vol=002	Occ=002
02/10/2003	00:00:31	EX1-0035N-170.578	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:31	EX1-0035S-170.378	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:31	L2-0035N-170.425	Speed=59	Vol=000	Occ=000
02/10/2003	00:00:31	L2-0035S-170.425	Speed=62	Vol=003	Occ=003
02/10/2003	00:00:31	L3-0035N-170.425	Speed=63	Vol=002	Occ=002
02/10/2003	00:00:31	L4-0035N-170.425	Speed=62	Vol=002	Occ=002
02/10/2003	00:00:32	EN1-0035S-170.917	Speed=-1	Vol=000	Occ=000
02/10/2003	00:00:32	EN1-0035S-170.929	Speed=-1	Vol=000	Occ=000

Fig. 3.4 Raw ILD data

The ILD data from TransGuide area used in this dissertation were archived based on the server which processed the data. There were 5 servers reporting the data for the selected days and each of them were reporting data from approximately 100 detector locations. This came to around 30MB files from each of the servers for each of the days. For a selected location, the size of the data files was approximately 600 KB per day.

3.2.2 Automatic Vehicle Identification (AVI)

Automatic Vehicle Identification refers to technology used to identify a particular vehicle when it passes a particular point. Automatic vehicle monitoring or AVM involves the tracking of vehicles at all times. Early development of AVI occurred in the United States (Hauslen 1977; Roth 1977; Fenton 1980) beginning with an optical scanning system in the 1960s in the railroad industry to automatically identify the rolling stock. Since then there have been enormous

advances in this area for different applications varying from toll collection systems to advanced traveler information system (Scott 1992).

The AVI system needs AVI readers, vehicles that have AVI tags (probe vehicles), and a central computer system, as shown in Figure 3.5. Tags, also known as transponders, are electronically encoded with unique identification (ID) numbers. Roadside antennas are located on roadside or overhead structures or as a part of an electronic toll collection booth. The antennas emit radio frequency signals within a capture range across one or more freeway lanes. When a probe vehicle enters the antenna's capture range, the radio signal is reflected off the electronic transponders. The reflected signal is slightly modified by the tag's unique ID number. The captured ID number is sent to a roadside reader unit via coaxial cable and is assigned a time and date stamp and antenna ID stamp. These bundled data are then transmitted to a central computer facility via telephone line, where they are processed and stored. Unique probe vehicle ID numbers are tracked along the freeway system, and the travel time of the probe vehicles is calculated as the difference between the time stamps at sequential antenna locations (Traffic Detector Handbook 1991).

AVI systems have the ability to continuously collect large amounts of data with minimal human resource requirements. The data collection process is mainly constrained by sample size (Traffic Detector Handbook 1991). Figure 3.6 shows a sample set of raw AVI data that are collected by the reader. The first column is the AVI reader number. The second column is the anonymous tag ID of the vehicles. The third column gives the time followed by the date.

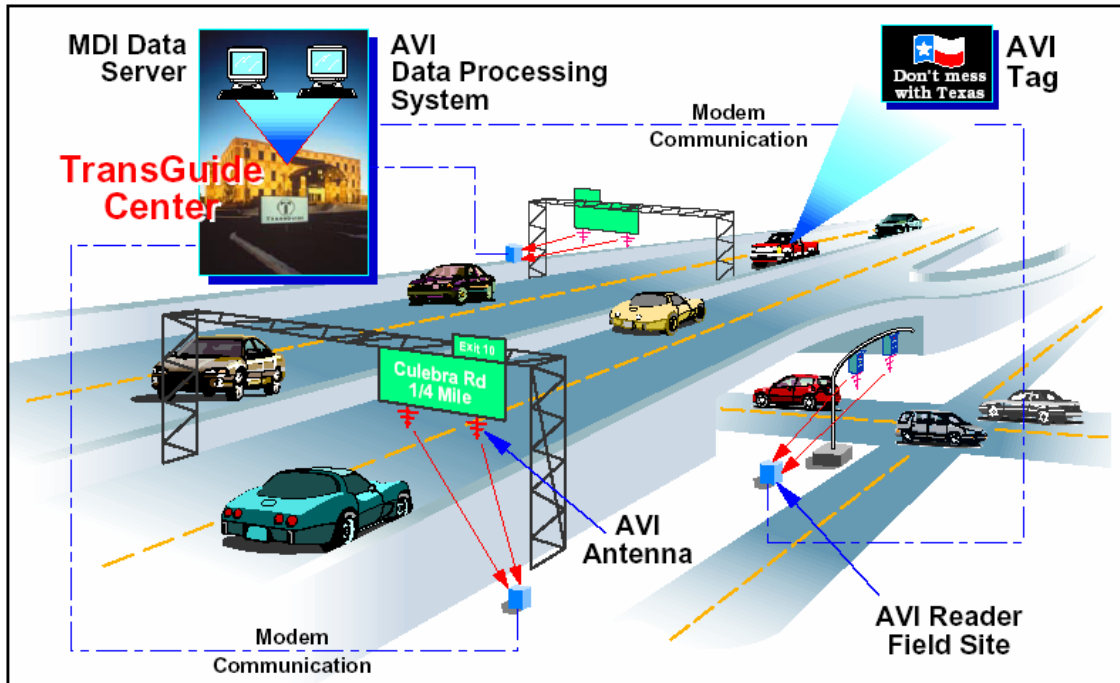


Fig. 3.5 AVI conceptual view

(Source: <http://www.TransGuide.dot.state.tx.us/>)

```

142 HCTR0092677553...!H$           &00:58:21.63 02/11/03%16-0-06-0
142 OTA.00095021C0...^D$         &00:58:29.44 02/11/03%16-0-12-0
145 ARFWP10647.....              &00:57:30.68 02/11/03%1B-0-03-0
145 ARFWD3018.....              &00:57:30.88 02/11/03%1B-0-03-0
145 ARFWP14316.....              &00:57:30.94 02/11/03%1B-0-01-0
145 DDS0112                       &00:58:08.38 02/11/03%1D-1-04-1
145 DDS0223                       &00:58:08.73 02/11/03%1D-1-01-1
144 ARFWP14872.....              &00:59:08.08 02/11/03%19-1-01-1
145 DNT.004672118B...^?D$      &00:59:06.92 02/11/03%1D-1-08-1
144 ARFWP9898.....              &00:59:08.41 02/11/03%19-1-01-1
144 ARFWP11606.....              &00:59:34.27 02/11/03%19-0-01-0
144 ARFWD248.....                &00:59:34.59 02/11/03%19-0-02-0
144 ARFWP2143.....              &00:59:34.68 02/11/03%19-0-01-0
145 OTA.00074756F8...^D$       &00:59:47.38 02/11/03%1B-0-04-0
144 OTA.00794375F2...^D$       &00:59:44.61 02/11/03%19-1-01-1
137 OTA.00625142E0...^D$       &01:00:05.42 02/11/03%2D-0-04-0
141 OTA.005238872C...^D$       &01:00:09.00 02/11/03%32-0-0B-0
142 ARFWD3624.....              &01:00:18.19 02/11/03%16-0-07-0

```

Fig. 3.6 Raw AVI data format

The AVI data from the TRANSGUIDE site are archived in three different categories every day. These are tag archive, link archive, and site archive. Each of these files was of the size 1.5MB/day. Tag archive is the one that contains all the vehicle information as shown in Figure 3.6. One day's tag archive file will have data from all the 15 AVI stations for 24 hours. There will be around 20,000 data points for each day from the 15 AVI stations. Figure 3.7 shows the picture of an AVI antenna in the field.



Fig. 3.7 AVI antennas

(Source: http://www.houstontranstar.org/about_transtar/docs/2003_fact_sheet_2.pdf)

3.3 FIELD DATA COLLECTION

Data for the present study were collected from the TransGuide web site, where the data were archived for research purposes. TransGuide, Transportation Guidance System, is San Antonio's advanced traffic management system (ATMS). TransGuide was designed to provide information to motorists about traffic conditions such as accidents, congestion, and construction. With the

use of inductance loop detectors, color video cameras, AVI, variable message signs (VMS), lane control signals (LCS), traffic signals, etc., TransGuide can detect travel time and respond rapidly to accidents and emergencies (Texas Department of Transportation (TxDOT) 2003). The specifically stated system goals are to detect incidents within 2 minutes, change all affected traffic control devices within 15 seconds from alarm verification, allow San Antonio police to dispatch appropriate response, assure system reliability and expandability, and support future Advanced Traffic Management System (ATMS) and ITS activities (TransGuide Technical Brochure 2003). Figure 3.8 shows two examples of the information provided to travelers through the TransGuide system.



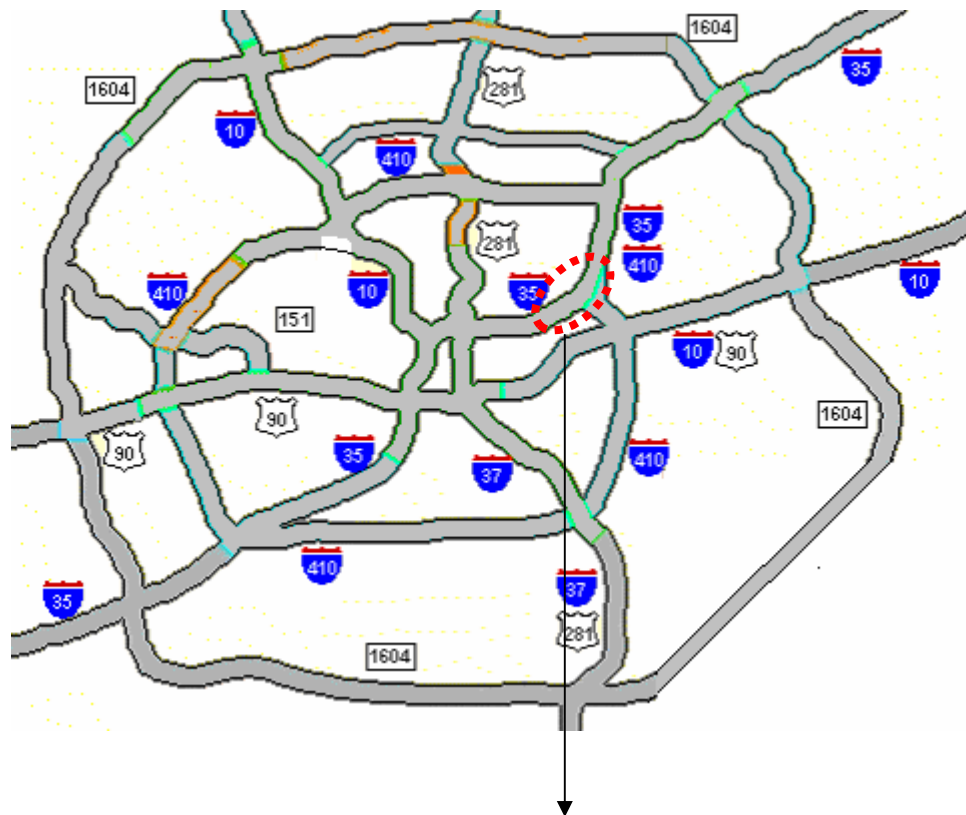
Fig. 3.8 Examples of TransGuide information systems

(Source: http://www.TransGuide.dot.state.tx.us/docs/atms_info.html)

The first phase of the San Antonio TransGuide system became operational on July 26, 1995 and included 26 miles of downtown freeway. This phase of the TransGuide system includes variable message signs, lane control signals, loop detectors, video surveillance cameras, and a communication network covering the 26-instrumented miles. Now operational on 77 miles, the system will eventually cover about 200 miles of freeway. The section on I-35 between New Braunfels Avenue and Walzem Road went online in March 2000, which is the selected test bed

for the present study (TexHwyMan 2003). Figure 3.9a shows the San Antonio freeway system indicating the location of the test bed selected for the present study. Figure 3.9b shows the enlarged map of the selected test bed, and is detailed in the following section.

(a)



(b)

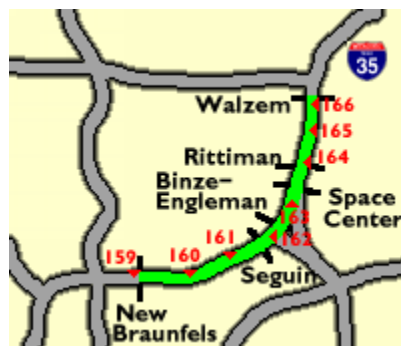


Fig. 3.9 a) Map of the freeway system of San Antonio and b) map of the test bed
(Source: <http://www.TransGuide.dot.state.tx.us>)

3.4 TEST BED

The I-35 section was selected based on the availability of the loops and AVI in the same location. The selected test bed is a three-lane road with on ramps and off ramps in between the detectors as shown in Figure 3.10a. The data were analyzed for continuous 24-hour periods for 5 consecutive days starting from February 10, 2003, Monday to February 14, 2003, Friday. For the study period the data were reported in 20-second intervals. Thus for a 24-hour period, around 4000 records were available for each of the detectors. A series of five detectors from stations 159.500 to 161.405 including all the ramps in between was analyzed. AVI data were also collected from the same section.

The present problem of estimation and prediction of travel time from ILD data using the suggested model necessitated aggregating the data from all three lanes and analyzing them as a single lane. The data were aggregated because the travel time estimation model suggested in the present study is mainly based on the conservation of vehicles principle (see Chapter V for more details). Although lane-by-lane data were available from the loop detectors, no lane changing data were available. The constraint related to conservation of vehicles cannot be imposed on individual lanes due to the lack of lane changing data. The vehicles entering the section of road under consideration can change lanes before exiting the section. Hence, in addition to lane-by-lane data, one also needs details related to the number of vehicles that changed lane from and/or to the lane under consideration. The data used in this dissertation are from ILD, and it is impossible to get the lane changing details using this technology. Hence, lane changing is not taken into account while developing the model for the estimation of travel time.

In case the lane changing data are available from a different data source, such as video data, the models used for the estimation of travel time need to be modified accordingly to incorporate lane changing into account. In the present form, the models do not consider the inflow and outflow from adjacent lanes by lane changing. Hence, the data from the detectors in different lanes at each of the detector stations were aggregated together and assumed as a single lane in this dissertation.

Since this dissertation investigated a series of detectors and analyzed the total inflow and outflow at each entry-exit pair, data from the ramps in between the entry-exit pair were also needed. The entrance and exit ramp data were added to the appropriate main lane data. This is required because the present study uses an input-output analysis, and the ramp data are also part of the input or output. Also, the volume coming from ramps becomes part of the vehicles in the road section under consideration and the travel time is affected due to this incoming volume from ramps. Figure 3.10b shows the accumulation process and the resulting five consecutive detector locations in the present study.

3.5 PRELIMINARY DATA REDUCTION

The traffic data obtained from loop detectors are used for different applications such as graphical displays, traffic forecasting programs, and incident detection algorithms. Ensuring the accuracy of traffic data prior to their use is of utmost importance for the proper functioning of incident detection algorithms and other condition monitoring applications. Techniques to screen such data and to remove suspect data have evolved during the last few years and are detailed in Chapter II.

In the present study, the initial data screening and quality control of detector data were carried out based on suggestions in previous literature. The methods selected for the preliminary data screening in the present study are discussed in the sections below.

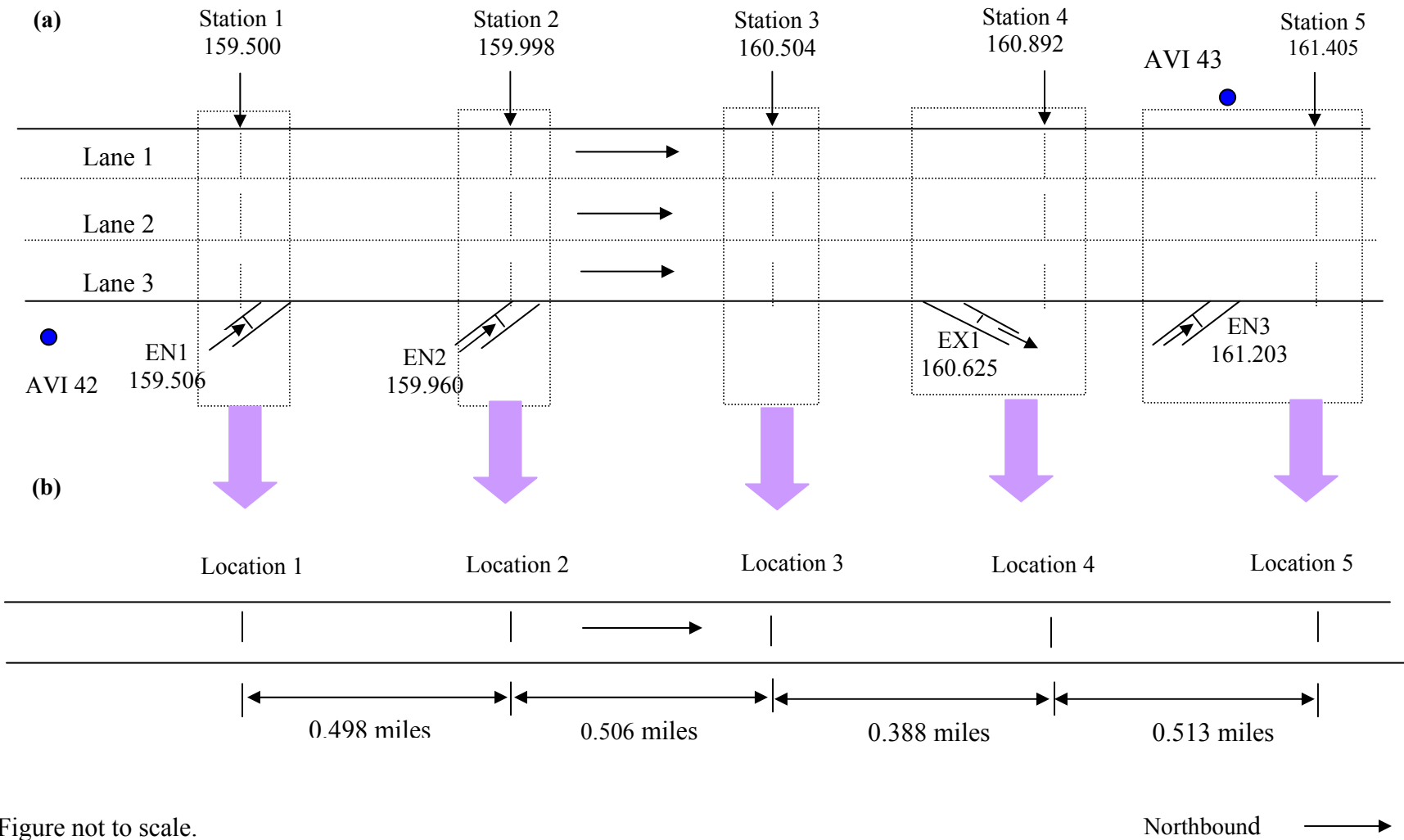


Figure not to scale.

Fig. 3.10 Schematic diagram of the test bed from I-35 N, San Antonio, Texas

3.5.1 Detector Data

There are five servers in the TransGuide center that are dedicated to data storage and data processing. The data from the sites are sent to any one of the five servers available. To extract data from a particular detector, the first step is to find out which server processed the selected detector number. Once that is known, the entire data set is searched and the data corresponding to the specific detector number and for a specific lane are extracted to a new file. Thus, for a three-lane roadway, there will be three files corresponding to each of the lanes for one detector number. MATLAB programs were developed to extract these data and are shown in Appendix D.

As a first step, extensive quality control and data reduction were performed. The data were cleaned of unreasonable values of speed, volume, and occupancy, both individually and in combination. Also, the polling cycle of the data during the data collection period was 20 seconds, but the cycle occasionally skipped to larger intervals. Preliminary data reduction and quality control were performed when the data had any of the above errors, and they are discussed in the following sections.

3.5.1.1 Test for Individual Threshold Values

The threshold value test examined speed, volume, and occupancy in each individual record of the data set. If the observed value was outside the feasible region, that particular value was discarded and was assumed to be equal to the average of the previous time step and next time step values. A maximum threshold value of 3000 vehicles per hour per lane was used as the volume threshold. This value was based on previous studies (Jacobson et al. 1990; Turochy 2000; Turner et al. 2000; Park et al. 2003; Eisele 2001). For speed, a threshold value of 100 mph (160 kmph) was used, and occupancy values exceeding 90% were discarded. Table 3.1 shows the screening rules incorporated in the MATLAB code to identify erroneous data. These rules were established in previous works (Turner et al. 2000; Park et al. 2003; Eisele 2001; Brydia et al. 1998). Rules one, two, and three are the thresholds set for individual parameters.

3.5.1.2 Test for Combinations of Parameters

All combinations of one of the three parameters, speed, volume, or occupancy, being zero, with the other two being non-zero were examined. Similarly, combinations with one being non-zero, with other two being zero were also checked. When such unreasonable combinations were found, the zero values were replaced with the average of the previous time step and next time step values.

Rule four in Table 3.1 represents the condition of all traffic parameters being zero. This occurs when vehicles are either stopped over the loop detectors or when there are no vehicles present in that time step. This happens mostly due to vehicles not being present during off-peak traffic conditions in early mornings. These data are removed from the data set so that they will not affect the average speeds when taking the average of the 2-minute intervals. Rule five identifies observations when the speed, volume, and occupancy are in the acceptable and expected ranges for a 20-second period. The remaining rules are used to identify suspicious combinations of speed, volume, and occupancy and their cause is unknown. The unreasonable observations in this category were replaced with an average of the previous and next values.

Table 3.1 Screening Rules

SCREENING RULES	
Individual tests	
1) $q > 17$	Error
2) $v > 100$	Error
3) $o > 90$	Error
Combination tests	
4) $v = 0, q = 0, o = 0$	Discard
5) $v = 0 - 100, q = 0 - 17, o = 0 - 90$	Accept
6) $v = 0, q = 0, o > 0$	Error
7) $v = 0, q > 0, o > 0$	Error
8) $v = 0, q > 0, o = 0$	Error
9) $v > 0, q = 0, o = 0$	Error
10) $v > 0, q > 0, o = 0$	Error
11) $v > 0, q = 0, o > 0$	Error

q = volume per 20 second, v = speed in mph, and o = percent occupancy.

3.5.1.3 Missing Data

Gold et al. (2001) reported that when the polling cycle is less than 2 minutes, the current observation contains the sum of the traffic characteristics between the previous and current observations. This means the volume indicated in the current observation is the sum of the volume since the previous observation and the speed is the average speed since the previous observation. Therefore, the current speed can be used for the speed of the previous observation and half of the volume of the current observation can be used for the volume in the previous step.

The polling cycle of the San Antonio data for the selected locations during the data collection dates was 20 seconds, but it was observed that the cycle occasionally skipped to 60 or 120 seconds. When this happened one of the following two things have occurred. Either the first interval was skipped and all the data were recorded in the next interval, or the first interval data were missed altogether. The decision to use specific values was made based on the magnitude of the values reported in the interval after the missing interval in comparison to the neighboring values. In the case of aggregated interval, the data were split into 20-second intervals, whereas in the case of missing intervals, the data were imputed with the average of the previous and next interval data.

Programs were developed in MATLAB to carry out the threshold checking, combination checks, and imputation. After these corrections, the data were aggregated into 2-minute intervals. Thus, an original file with data for a 24-hour time period having around 4300 records will be reduced to 720 records after aggregation. The 2-minute data from different lanes of the same detector station were added together and assumed as a single detector location as explained earlier. Subsequently, the entry ramp and exit ramp volume data were added to the appropriate main lane detectors. Sample distribution of occupancy, speed, and volume as a function of time, after all the quality control and data reduction are carried out, are shown in Figures 3.11, 3.12, and 3.13 for February 11, 2003 for location 2.

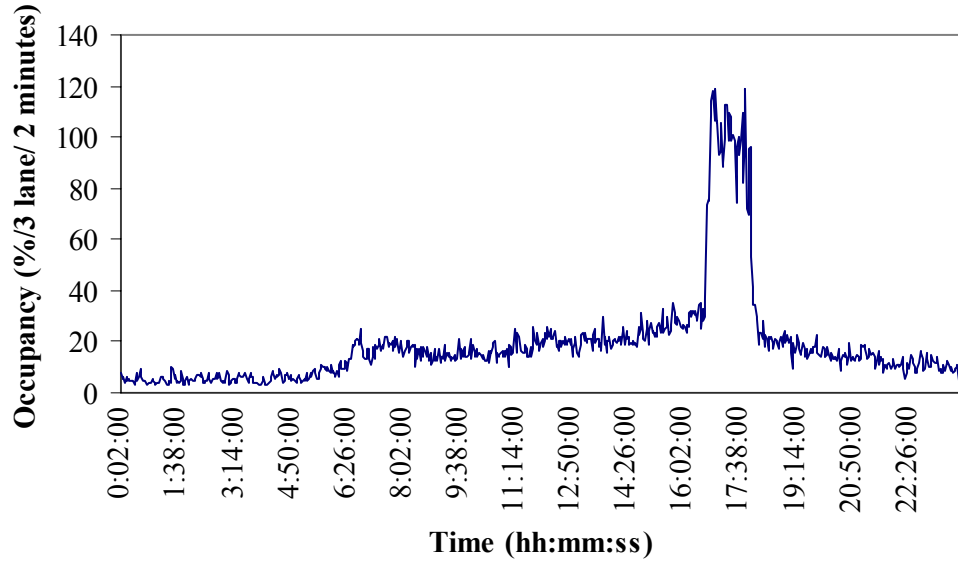


Fig. 3.11 Occupancy distribution from I-35 site, location 2, on February 11, 2003

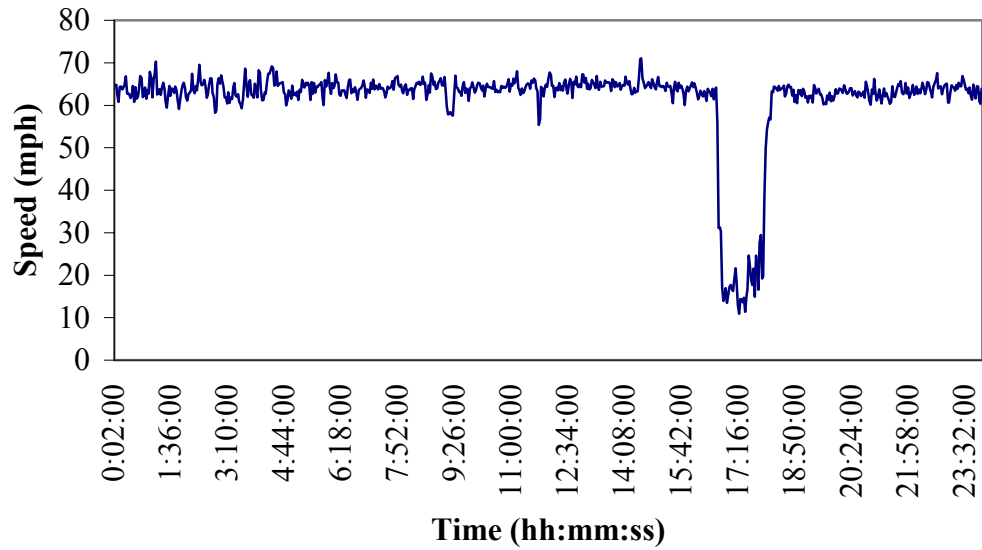


Fig. 3.12 Sample speed distribution from I-35 site, location 2, on February 11, 2003

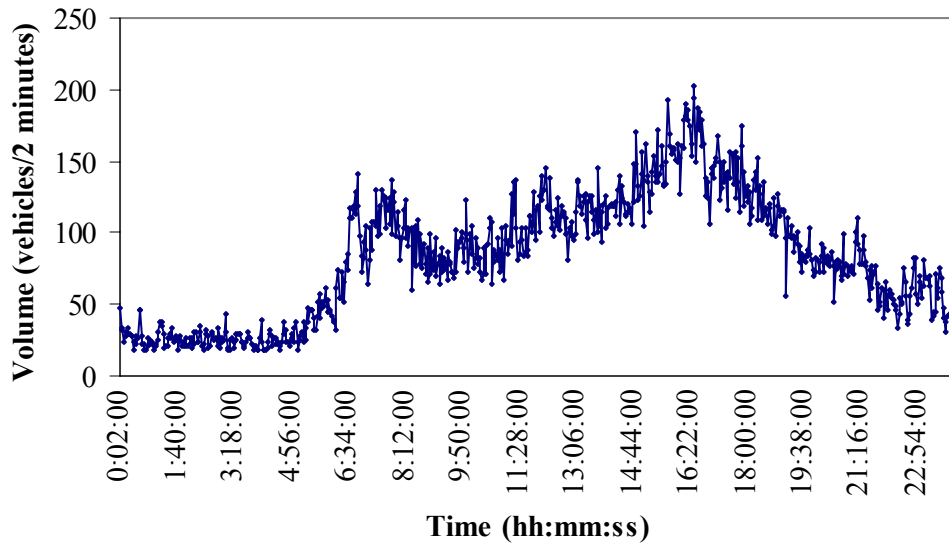


Fig. 3.13 Sample volume distribution from I-35 site, location 2, on February 11, 2003

3.5.2 AVI Data

The AVI data collected from the field for a selected day were first sorted based on the vehicle identification number. These data were then sorted based on the AVI reader number and the time stamp. Thus, the time a selected vehicle crosses different AVI antennas are grouped together. MATLAB programs were developed to carry out this sorting and are shown in Appendix D. After sorting the data, the data quality was checked before carrying out the travel time calculation. For the AVI data, the quality control mainly included the removal of outliers. The primary source of these outliers is motorists that are read at the starting station of the corridor, exit the freeway, and then reenter the freeway later. This provides large outlier readings of travel time. In the present case, threshold values were determined based on the length of the section and the minimum and maximum reasonable travel time for that distance. Also observations with magnitude more than four times the mean of the previous 10 observations were considered as outliers. However, none of the data considered in this dissertation showed the presence of outliers. Once the outliers were removed, the link travel time was calculated by matching unique tag reads recorded by the AVI readers at the start and the end of the defined AVI links. The travel time was averaged for all the vehicles during the selected

study interval, 2 minutes. Figure 3.14 shows the travel time obtained from AVI on February 11, 2003.

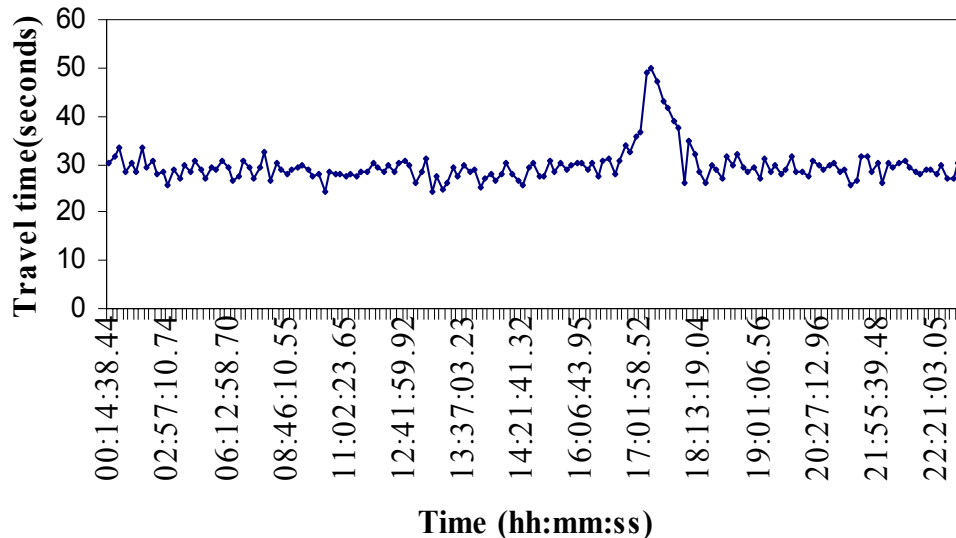


Fig. 3.14 Sample AVI travel time from I-35 site, on February 11, 2003

3.6 SIMULATED DATA USING CORSIM

Simulated data were generated using the simulation software CORSIM (CORridor SIMulation), for testing the accuracy of the methods developed and techniques employed in the work. CORSIM is one of the most widely used microscopic traffic simulation models in the United States. CORSIM was developed by the Federal Highways Administration (FHWA) and includes two separate simulation models, NETSIM (NETwork SIMulation) and FRESIM (FREeway SIMulation). NETSIM is a traffic simulation model that describes in detail the operational performance of vehicles traveling in an urban traffic network. FRESIM represents the simulation of freeway traffic. The stochastic and dynamic nature of the model allows accurate representation of actual traffic conditions (CORSIM User's Guide 2001). CORSIM simulates traffic utilizing the car-following model. The basic idea of car-following models is that the

response of the following vehicle's driver is dependent on the movement of the vehicle immediately preceding it (May 1990). Car-following models are composed of equations that give the acceleration of the following vehicle with respect to the behavior of the lead vehicle. Thus, CORSIM simulates vehicles by maintaining space headway between simulated vehicles. CORSIM can be used to model an existing field network and collect the flow, speed, occupancy, or travel time data similar to that collected from the field. This simulated data can be used for validating traffic models when there is a lack of field data.

CORSIM is designed primarily to represent the spatial interactions of drivers on a continuous, rather than a discrete basis for analysis of freeway and arterial networks (Rilett et al. 2000). CORSIM is a stochastic model, applying a time step simulation to describe traffic operations, randomly assigning drivers and vehicles to the decision-making process. It applies time step simulation, where one time step represents one second. Each vehicle is modeled as a distinct object that is moved every second, while each variable control device in the network is also updated every second for drivers to react. The input requirement includes network details and traffic details. The network is made up of links and nodes, and the traffic demand is input as volume in vehicles per hour. The output provides details such as travel time, delay, queues, and environmental measures. Surveillance statistics like vehicle counts, percentage occupancy, and average speed values can be obtained by choosing the detector option (CORSIM User's Guide 2001).

The traffic simulation for the present work used the FRESIM subcomponent. A traffic network similar to the field test bed was created in CORSIM, and detectors were placed 0.5 miles apart. Traffic volumes from the field were given as input to CORSIM at 30-minute intervals. A corridor with seven links was generated for the present analysis. Detectors were placed in each link to collect the flow, speed, and occupancy rate. The default parameters in CORSIM were used since they gave acceptable results without much error, as shown below. Varying flow rates were input to the simulation, based on field data in order to have simulated flow variations comparable to the field data variations. A 15-minute initialization time was given for the system to reach equilibrium. The inputs are given by modifying the corresponding record types. The direct output from CORSIM will not contain travel time details. Hence, the binary time step file (.tsd) that describes the state of each individual vehicle within the simulation model at each 1-

second time step in the simulation is used for the estimation of travel time. These data are stored for each link and time step within the model and are specially designed to provide quick access to data within each individual time step data (.tsd) file. A conversion program written in C++ was used to convert the binary time step data file to an ASCII file that could be utilized to analyze the output results. The conversion program extracts vehicle-specific data at 1-second time increments between specific nodes of the corridor, including node number, time step (in 1-second increments), global vehicle identification number, vehicle fleet, vehicle type, vehicle length, vehicle acceleration, and vehicle speed. Because the data included 1-second time increments, the majority of vehicles on the link were included in multiple time steps as they traversed the network. Hence, each vehicle's entry and exit time were determined and the travel time was then calculated as the difference between the entry and exit time. Programs were developed in C programming language to carry out these operations. The obtained travel time value for each of the individual vehicles was then averaged for 2-minute intervals and was used for the validation.

The detector output is given in the .OUT file. Every 20 seconds data were extracted to be comparable to the field scenario. Programs were developed in PERL to extract the speed, flow, and occupancy values from the output file. The output obtained from CORSIM was used for checking the validity of the optimization technique and travel time estimation procedure as described in Chapters IV and V. The developed CORSIM network is given in Appendix C, and the programs developed for extracting the data are given in Appendix D.

The simulated volumes were compared with the corresponding actual values to check how the integrity of the original data is maintained. Simulated data and the corresponding field data for February 11, 2003, are shown in Figure 3.15 for illustration. Mean absolute percentage error (MAPE), as defined below, is calculated for each set of data to determine the change in the data from the actual values.

$$\text{MAPE} = \frac{\sum \frac{|actual - estimated|}{actual}}{\text{Number of observations}} \times 100, \quad (3.7)$$

The MAPE value came to be 14%, showing that the simulated data represent the actual data reasonably well.

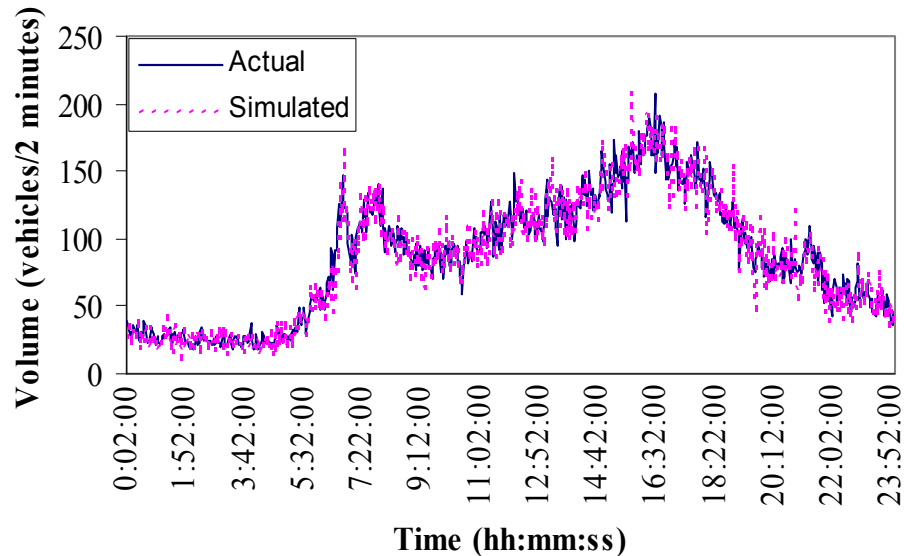


Fig. 3.15 Simulated data and the corresponding field data for February 11, 2003, for location 1

3.7 CONCLUDING REMARKS

This chapter described the details of the study corridors used, the data used in the analysis, and the preliminary data reduction. The data were collected from the archived collection of the TransGuide system in San Antonio. The study sites were selected from the I-35 N freeway in San Antonio, since it was equipped with both loop detectors and AVI. Details about the working of AVI and loops were described briefly before presenting the details of the data used. The details of the test bed and the collected data were given next. The preliminary data quality checks and the corrections carried out were detailed subsequently. These data quality control procedures were based on previous investigations. Data were also simulated using CORSIM simulation software, and the details were given in the last section. A network similar to the field network was generated in CORSIM, and the data were used for checking the validity of the

models developed.

All of the preliminary data quality control techniques discussed in this chapter are useful to correct data collected at a single location, and therefore cannot account for systematic problems over a series of detectors. For an application such as travel time estimation, there is a need to consider a series of detectors. In such cases when detectors are analyzed as a series, more discrepancies are identified in the data, even after applying the screening methods at individual locations. For instance, if the total number of vehicles counted by two consecutive detector locations is observed over a period of time, the difference in the cumulative counts should not exceed the number of vehicles that can be accommodated in that length of the road under the jam density condition. However, this constraint will be violated if some of the detectors are under- or overcounting vehicles. For many traffic applications such as incident detection, this might not be an issue. However, for other applications that rely on accurate system counts, such as origin-destination (OD) estimation and certain travel time estimation techniques, this can be a problem. While most of the existing error detection and diagnostic tests do take into account possible malfunctions of the loop detector by looking at the data at a specific point, the problems related to balancing consecutive detector data for vehicles being under- or overcounted has not been well addressed. This lack of interest in this area may be due to current applications being based on data generated at a particular station point rather than series of station points at the same time. In other words, since the error does not adversely affect the result of the applications, they are typically ignored. However, if the loop detector data are to be successfully used for new applications, these issues of system data quality will need to be addressed.

Thus, most of the existing error detection and diagnostic tests do take into account possible malfunctions of the loop detector by looking at the data at a specific point. However, the problems related to balancing consecutive detector data for vehicles being under- or overcounted have not been well addressed. This analysis of the detector data as a series, the problems related to balancing consecutive detector data, and the correcting methodology suggested forms the crux of the next chapter.

CHAPTER IV

OPTIMIZATION FOR DATA DIAGNOSTICS[†]

4.1 INTRODUCTION

Chapter III discussed the preliminary data quality control carried out on detector data at individual locations. It was concluded that while substantial failures in loop detector data are easily identified using the current technologies, more subtle failures such as biases in volume counts may go unidentified and, hence, there is a need to analyze the data at a system level. For example, in an application such as estimating travel time between two detector stations, where the data from the neighboring detectors need to be compared, there is a need to check the conservation of vehicles principle. Conservation of flow is one of the basic traffic principles that any volume data as a series must meet. In this dissertation the conservation of vehicles is checked by comparing the cumulative flow curves from consecutive detector stations. As discussed in Chapter II, very few studies have been reported that systematically analyzed a series of detector locations over a long interval of time to check whether the collected data follow the conservation of vehicles. Most of those studies, when faced with a violation of conservation of vehicles, suggested applying simple adjustment factors to rectify the problem, rather than applying any systematic methodology.

In this dissertation a correction procedure based on nonlinear optimization is used for identifying and correcting the data when the conservation of vehicles principle is violated. The generalized reduced gradient method is chosen, where the objective function and constraints are selected such that the conservation of vehicles principle is followed with least change to the original data. Figure 4.1 shows the general flow chart for the overall data reduction process. Note that the proposed optimization technique can also be readily adapted for other applications. Two such applications, namely, to impute missing data, and to locate the worst performing detector station among a series of detectors are also illustrated in this chapter.

[†] Part of this chapter is reprinted with permission from “Loop detector data diagnostics based on conservation of vehicle principle” by Vanajakshi, L. and Rilett, L. R., Accepted for publication in *Transp. Res. Rec.*, TRB, National Research Council, Washington, D. C.

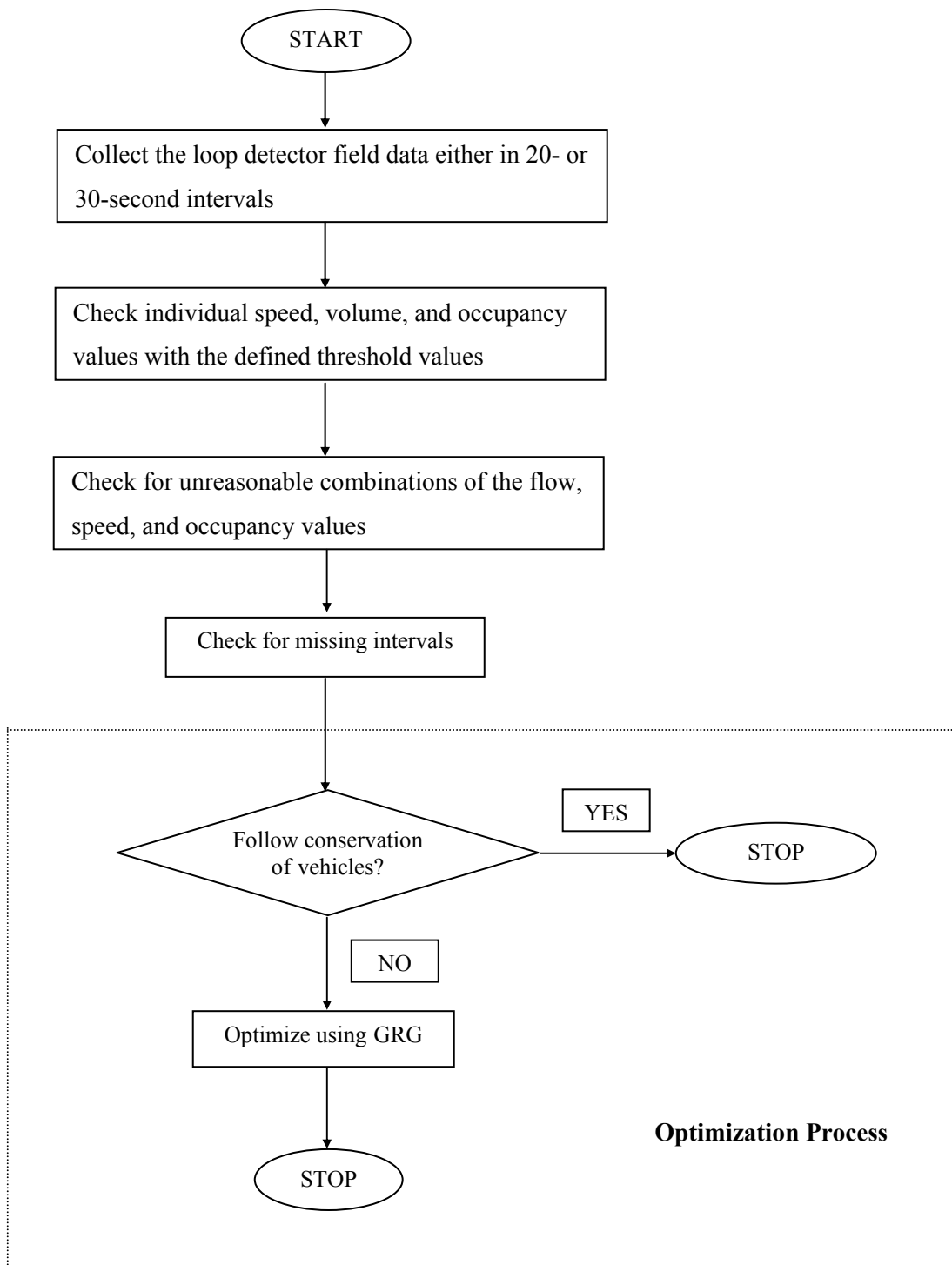


Fig. 4.1 Algorithm for the overall proposed method

The following section details the conservation principle in vehicular traffic and the related literature. The GRG method is detailed next, followed by the actual implementation of the procedure for an example problem. Next, the validation of the optimization procedure is illustrated using simulated data. The applicability of the method for different conditions and other applications is detailed in the last section.

4.2 CONSERVATION OF VEHICLES

The concept of conservation of vehicles (Lighthill and Whitham 1955; Richards 1956) states that the difference between the number of vehicles entering and leaving a link during a specific time interval corresponds to the change in the number of vehicles traveling on the link. The simplest and the most general way in which this can be stated is that vehicles cannot be created or lost along the road (Daganzo 1997).

This concept is further explained using a one-lane road with two detectors located at each end, as shown in Figure 4.2. The number of vehicle arrivals and departures are measured continuously and aggregated regularly at the upstream location x_1 and downstream location x_2 , respectively.

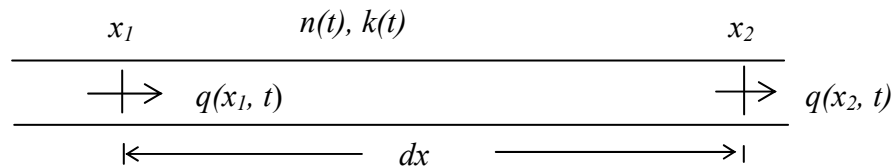


Fig. 4.2 Illustration of the conservation of vehicles

Referring to Figure 4.2, let $q(x_1, t)$ denote the flow measured at location x_1 at time t , and let $q(x_2, t)$ denote the flow measured at location x_2 at the same time t . Let $n(t)$ be the number of vehicles traveling over the link distance dx between the detector stations x_1 and x_2 at time t and $k(t)$ the corresponding density of vehicles.

Under the principle of conservation of vehicles, the number of vehicles on the length of road dx

between upstream location x_1 and downstream location x_2 , in an interval of time dt must equal the difference between the number of vehicles entering the section at x_1 and the number of vehicles leaving the section at x_2 , which is equal to $x_1 + dx$, in that time interval. If the number of vehicles on the length dx at time t is $k dx$ and the number of vehicles entering in time dt at x is expressed as $q dt$, then the conservation equation is as shown below (Drew 1968).

$$k dx - \left(k - \frac{\partial k}{\partial t} dt \right) dx = q dt - \left(q + \frac{\partial q}{\partial x} dx \right) dt, \quad (4.1)$$

where,

q = flow (vehicles per hour),

k = density (vehicles per mile),

dx = length of road (miles), and

dt = time interval (hours).

Based on the fact that $q = ku$, where u is the space mean speed in vehicles per unit time, the following simplified form for the above equation may be derived (Drew 1968; Kuhne and Michalopoulos 1968).

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0. \quad (4.2)$$

Let $Q(x_1, t_n)$ and $Q(x_2, t_n)$ be the cumulative number of vehicles entering and exiting the link, respectively, from time t_1 to t_n , which can be expressed as

$$Q(x_1, t_n) = \Delta t \sum_{i=1}^n q(x_1, t_i) \text{ and} \quad (4.3)$$

$$Q(x_2, t_n) = \Delta t \sum_{i=1}^n q(x_2, t_i). \quad (4.4)$$

Under ideal conditions, the cumulative volume at an upstream location should be more than or equal to the cumulative volume at the downstream location at any instant of time. Based on Figure 4.2 this can be expressed as:

$$Q(x_1, t_n) \geq Q(x_2, t_n). \quad (4.5)$$

The equality condition in equation 4.5 holds for the case when all the vehicles that entered the section had exited by the end of the time interval.

Also, the maximum difference between the upstream and downstream location cumulative flows cannot exceed the maximum number of vehicles that can be accommodated between these two locations at jam density as expressed in Equation 4.6.

$$Q(x_1, t_n) - Q(x_2, t_n) \leq n_{jam}, \quad (4.6)$$

where,

n_{jam} = maximum number of vehicles that can be accommodated between locations x_1 and x_2 at jam density.

Thus, if there are no systematic errors present in the data, the difference in the total number of vehicles counted by the two consecutive detectors should equal the number of vehicles between the two detector locations as shown in equation 4.7.

$$Q(x_1, t_n) - Q(x_2, t_n) = n(t_n). \quad (4.7)$$

Based on the cumulative flows recorded at x_1 and x_2 , there can be two scenarios in which the conservation of vehicles principle can be violated. In the first case when $Q(x_2, t_n)$ becomes more than $Q(x_1, t_n)$, 'extra' vehicles are said to be 'created.' In the second case when $Q(x_1, t_n)$ is more than $Q(x_2, t_n)$ and the difference is larger than the maximum number of vehicles that can be accommodated in the road length under consideration, vehicles are said to be 'lost.' Both of these conditions violate the conservation of vehicles principle. These differences can be due to errors of the detectors at the upstream location, the downstream location, or both.

Reported studies that checked the conservation of vehicles are limited and include Zuylen and Brantson (1982), Petty (1995), Zhao et al. (1998), Cassidy (1998), Nam and Drew (1996, 1999), Kikuchi et al. (1999, 2000), Kikuchi (2000), Windover and Cassidy (2001), and Wall and Dailey (2003). While all of the above studies acknowledged the fact that conservation is

violated, few of them (Zuylen and Brantson 1982; Petty 1995; Nam and Drew 1996, 1999; Kikuchi et al. 1999, 2000; Wall and Dailey 2003) discussed methods of correcting this problem. Zuylen and Brantson (1982) developed a methodology that relied on an assumption about the statistical distribution of the data to eliminate the discrepancy in the data. The algorithm is developed assuming a Poisson distribution or a normal distribution. Petty (1995), in a report on the development of a program for freeway service patrol, discussed how to correct the loop detector count data, based on the conservation of vehicles principle. The correction procedure suggested was to use compensation factors, which are computed as a fraction of the flow from the detector under consideration to the neighboring main lane flow. Nam and Drew (1996, 1999) and Wall and Dailey (2003) used a simple adjustment factor for correcting the discrepancy. Nam and Drew calculated adjustment factors as the ratio of inflow to outflow for every 30-minute and adjusted the flow at the downstream point accordingly to balance the flow. The investigation by Wall and Dailey (2003) required one properly calibrated reference detector that can be assumed to be correct in order to calculate the correction factor. Kikuchi et al. (1999) studied an arterial signalized network and proposed methodologies to adjust the observed values using the concept of fuzzy optimization. Kikuchi et al. (2000) reported six different methods that can be used to adjust traffic volume data so that they will follow vehicle conservation and thus be useful for the subsequent analysis steps. They concluded that there is no single unique method that can be used under different situations. The data they used for analysis were from a small arterial network, and a single hourly inflow and outflow at each signal was compared.

From the above discussion, it can be seen that there is no systematic studies reported for correcting continuous data collected from freeways when the data violate the conservation of vehicles. In the case of freeway data, the suggested methods are limited to the use of simple correction factors. This may work in situations where the analysis is for a small section of roadway or for a short duration of time where the resulting discrepancy is small in magnitude. For example, in the study reported by Nam and Drew (1999), the analysis was for a 4-hour period and the magnitude of the error was 200 vehicles over the total period. However, for most real-life traffic applications, the number of locations as well as the duration of study will be large. Hence, the amount of discrepancy may become large as well. In such cases, a systematic method is needed for diagnosing the data. The nature of this problem can be summarized as follows:

Given: a set of vehicle volumes from consecutive locations.

Objective: adjust the volumes such that the values are consistent with respect to conservation of vehicles.

Ideally, a method to solve the problem that finds a consistent set of adjusted values for a given set of observed values meeting the following requirements is needed:

1. To ensure conservation of flow at any point at any time,
2. To handle situations in which some data are missing or questionable,
3. To preserve the integrity of the observed data as much as possible, and
4. To handle a large amount of data (for example, continuous 24-hour data per day) in a systematic manner in a short computation time.

In this dissertation, an optimization approach that can meet the above requirements is selected to balance the loop detector data. The details of the selected method and its implementation are discussed in the following section. This dissertation represents the first application of this kind of an optimization technique for quality control of ILD data collected from freeways.

4.3 GENERALIZED REDUCED GRADIENT OPTIMIZATION PROCEDURE

Of the different methods currently used in engineering optimization fields, the most popular are the methods based on linearization of the problem because they are easy to solve (Gabriele and Beltracchi 1987). Such methods include successive linear programming, methods of feasible directions, and the generalized reduced gradient method (Gabriele and Beltracchi 1987). Each of these methods is based on linearizing the objective function and constraints at some stage of problem solving to determine a direction of search. This direction is then searched for the local improvement in the objective function, while at the same time avoiding severe violation of the constraints (Gabriele and Beltracchi 1987).

GRG is one of the most popular techniques among the above and has a reputation for its robustness and efficiency (Venkataraman 2001; Eiselt et al. 1987). The GRG method is an

extension of the Wolfe reduced gradient method (Wolfe 1963, 1967), which solves problems with linear constraints and a nonlinear objective function (Abadie and Carpenter 1969; Himmelblau 1972). The extension in the GRG algorithm from the Wolfe algorithm is to take into account nonlinear constraints also. The general steps involved in a GRG optimization are as follows (Abadie 1970; Gabriele and Ragsdell 1977):

1. Partition the variables into dependent and independent categories, based on the number of equality constraints involved,
2. Compute the reduced gradient,
3. Determine the direction of progression of the independent variables and modify them, and
4. Modify the dependent variables in order to verify the constraints.

The GRG algorithm solves the original problem by solving a sequence of reduced problems. The reduced problems are solved by a gradient method (Lasdon et al. 1978). The general form of a GRG problem will be as follows:

$$\text{Minimize } F(\mathbf{X}), \quad (4.8)$$

$$\text{Subject to } g_j(\mathbf{X}) < 0, \text{ which will be converted to } g_j(\mathbf{X}) + X_{j+n} = 0, \quad (4.9)$$

$$\text{or } g_j(\mathbf{X}) > 0, \text{ which will be converted to } g_j(\mathbf{X}) - X_{j+n} = 0, \quad (4.10)$$

$$h_k(\mathbf{X}) = 0, \quad (4.11)$$

where,

\mathbf{X} = column vector of design variables,

$F(X)$ = objective function,

g = inequality constraints,

h = equality constraints,

X_{j+n} = slack/surplus variables,

j = number of inequality constraints (1, m),

k = number of equality constraints (1, l), and

n = number of original variables.

To start with, the slack/surplus variables (X_{j+n}) are included in the original set of design variables, thus having $n+m$ total variables. The \mathbf{X} vector now includes the original variables as well as the slack/surplus variables. The variables are then partitioned in to $(n-l)$ independent/decision/basic variables (\mathbf{Z}) and $(m+l)$ dependent/state/nonbasic variables (\mathbf{Y}). Now, with these variables and with all equality constraints, the original optimization task can be stated as:

$$\text{Minimize } F(\mathbf{X}) = F(\mathbf{Z}, \mathbf{Y})^T, \quad (4.12)$$

$$\text{Subject to } h_j(\mathbf{X}) = 0, \quad j = 1, m+l \quad (4.13)$$

Now, differentiating the above objective and constraint functions yields,

$$dF(\mathbf{X}) = \nabla_Z^T \mathbf{F}(\mathbf{X}) \cdot d\mathbf{Z} + \nabla_Y^T \mathbf{F}(\mathbf{X}) \cdot d\mathbf{Y}, \quad (4.14)$$

$$dh_j(\mathbf{X}) = \nabla_Z \mathbf{h}_j(\mathbf{X}) \cdot d\mathbf{Z} + \nabla_Y \mathbf{h}_j(\mathbf{X}) \cdot d\mathbf{Y}, \quad j = 1, m+l \quad (4.15)$$

where, subscripts Z and Y correspond to the gradient with respect to the dependent and independent variables, respectively.

Now, equation 4.15 can be rewritten as follows:

$$d\mathbf{h}(\mathbf{X}) = \begin{bmatrix} \nabla_Z^T \mathbf{h}_1(\mathbf{X}) \\ \nabla_Z^T \mathbf{h}_2(\mathbf{X}) \\ \nabla_Z^T \mathbf{h}_3(\mathbf{X}) \\ \vdots \\ \nabla_Z^T \mathbf{h}_{m+l}(\mathbf{X}) \end{bmatrix} d\mathbf{Z} + \begin{bmatrix} \nabla_Y^T \mathbf{h}_1(\mathbf{X}) \\ \nabla_Y^T \mathbf{h}_2(\mathbf{X}) \\ \nabla_Y^T \mathbf{h}_3(\mathbf{X}) \\ \vdots \\ \nabla_Y^T \mathbf{h}_{m+l}(\mathbf{X}) \end{bmatrix} d\mathbf{Y}, \quad (4.16)$$

or

$$d\mathbf{h}(\mathbf{X}) = \mathbf{A} d\mathbf{Z} + \mathbf{B} d\mathbf{Y}, \quad (4.17)$$

where, \mathbf{A} is an $(m+l) \times (n-l)$ matrix and \mathbf{B} is an $(m+l) \times (m+l)$ matrix, since there are $(n-l)$ \mathbf{Z} variables and $(m+l)$ \mathbf{Y} variables. One restriction here is that the \mathbf{B} matrix should not be singular

(i.e. the inverse of the matrix should exist). If it becomes singular, the selection of the dependent and independent variable need to be changed such that \mathbf{B} will not be singular.

For any change in the decision variables, the equality constraints must remain satisfied for feasibility. It follows that $dh_j(\mathbf{X}) = 0$, for $j = 1, m+l$ in equation 4.15 for any change in the independent variable $d\mathbf{Z}$. Since $d\mathbf{h}(\mathbf{X}) = 0$, equation 4.17 can be solved for the corresponding change $d\mathbf{Y}$ in the dependent variables in order to maintain feasibility.

$$d\mathbf{Y} = -\mathbf{B}^{-1}\mathbf{A} d\mathbf{Z}. \quad (4.18)$$

Substituting Equation 4.18 into Equation 4.14 and rearranging, one gets the following expression,

$$d\mathbf{F} = \left\{ \nabla_{\mathbf{Z}}^T \mathbf{F}(\mathbf{X}) - \nabla_{\mathbf{Y}}^T \mathbf{F}(\mathbf{X}) \left[\mathbf{B}^{-1} \mathbf{A} \right] \right\} \cdot d\mathbf{Z}. \quad (4.19)$$

The generalized reduced gradient \mathbf{G}_R is defined by $\frac{dF(\mathbf{X})}{dZ}$ and can be represented as:

$$\mathbf{G}_R = \frac{dF(\mathbf{X})}{dZ} = \nabla_{\mathbf{Z}} \mathbf{F}(\mathbf{X}) - \left[\mathbf{B}^{-1} \mathbf{A} \right]^T \nabla_{\mathbf{Y}} \mathbf{F}(\mathbf{X}). \quad (4.20)$$

The generalized reduced gradient can now be used to determine the search direction \mathbf{S} in the decision variables as:

$$\mathbf{S} = -\mathbf{G}_R. \quad (4.21)$$

Then a one-dimensional search is performed with respect to the independent variable. For a selected step size, searching in the search direction, the dependent vector is updated using Newton's method for solving simultaneous nonlinear equations for $d\mathbf{Y}$. Having found the minimum in the search direction, the process is repeated until convergence is achieved (Vanderplaats 1984). In this case the convergence criterion was when all the constraints reach a value of 1×10^{-4} . The search direction is found such that any active constraints remain precisely active for some small move in that direction. If a move results in an active constraint being violated, Newton's method is used to return to the constraint boundary. More details of the GRG method as well as the available software for this method can be found in Gabriele and Ragsdell (1980), Lasden and Warren (1978), and Abadie (1978). Figure 4.3 shows the steps of the GRG algorithm discussed in this section for a one-dimensional search as a flowchart (Vanderplaats, 1984).

For the present problem of adjusting the detector data for violation of the conservation of vehicles principle, the optimization problem can be formulated for a series of I detectors in sequence as:

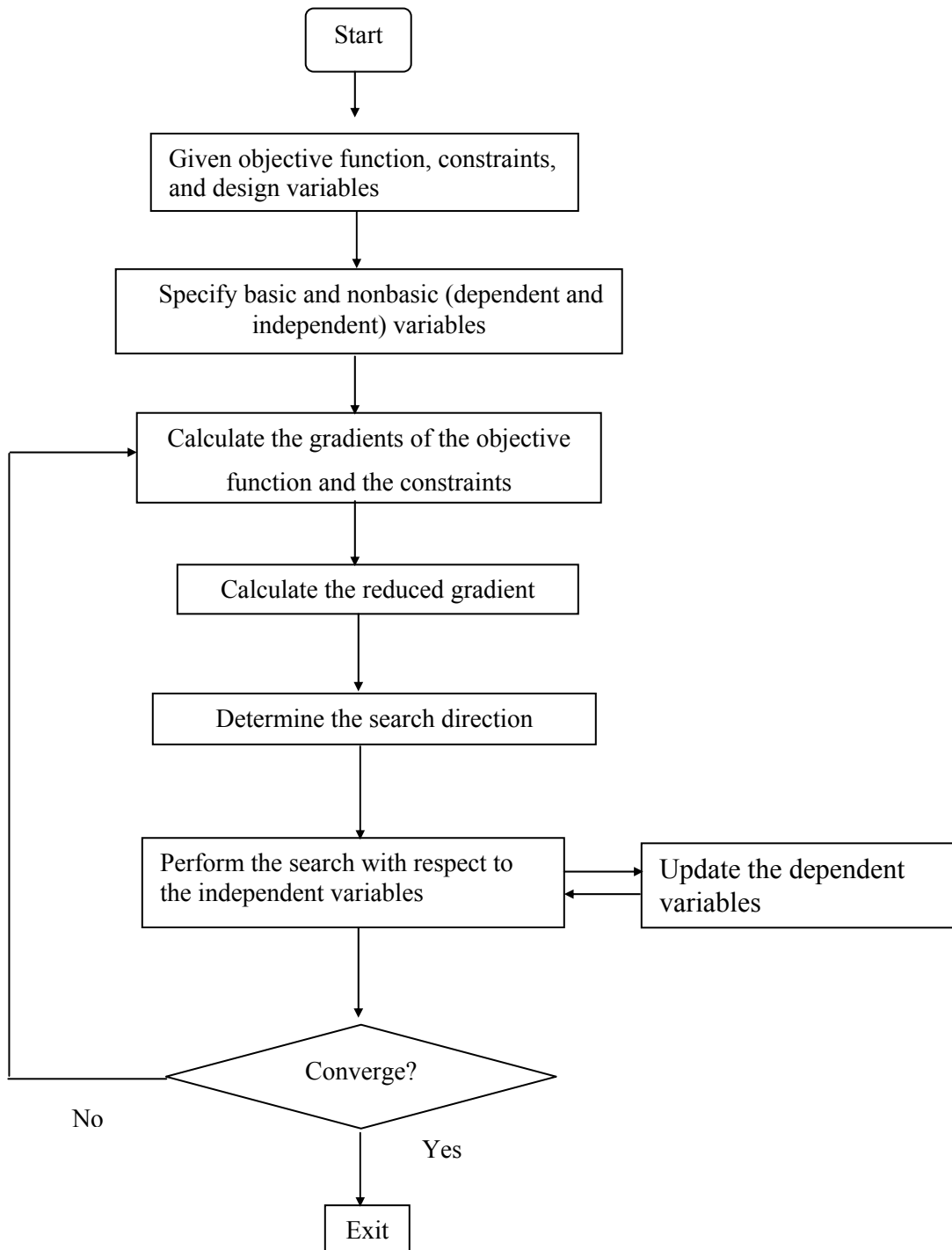


Fig. 4.3 Algorithm for the GRG method

$$\min \sum_{i=1}^{I-1} \left(Q_t^{(i)} - Q_t^{(i+1)} \right)^2, \quad (4.22)$$

Where,

t = time,

i = detector number,

$Q^{(i)}$ = cumulative number of vehicles at detector i , and

I = total number of locations,

subject to the constraints,

$$\left(Q_t^{(i)} - Q_t^{(i+1)} \right) \geq 0, \quad \forall i = 1, m-1 \quad (4.23)$$

$$\left(Q_t^{(i)} - Q_t^{(i+1)} \right) \leq z, \quad \forall i = 1, m-1 \text{ and} \quad (4.24)$$

$$\left(Q_t^{(i)} - Q_{t-1}^{(i)} \right) \geq 0, \quad \forall i = 1, m \quad (4.25)$$

where z is the maximum number of vehicles that can be accommodated between the two locations.

The constraints in this case are selected based on the restrictions discussed earlier. The first constraint, shown in equation 4.23, is based on the condition that the cumulative flow at the first detector location should be greater than or equal to the cumulative flow at location 2, which in turn should be greater than or equal to the cumulative flow at location 3, at all times. The second constraint, shown in equation 4.24, stipulates that the maximum difference cannot exceed the maximum number of vehicles that can be accommodated in that road length at jam density conditions. The constraint shown in equation 4.25 is that the value at a particular time step cannot be less than the value for the previous time step, since the variables used are cumulative values.

4.4 IMPLEMENTATION

To illustrate the procedure, a corridor consisting of three detectors is considered for analysis. The detector locations on San Antonio I-35 freeways were spaced approximately 0.5 miles apart, making the corridor length approximately 1.5 miles. The three consecutive detector locations

selected were detector numbers 159.500, 159.998, and 160.504 as shown in Figure 3.10. The analysis was carried out for a period of 24 hours for all 5-days under consideration.

As said earlier, the objective function was to adjust the observed volumes to meet the conservation of vehicles constraint. The objective function and associated constraints given in equations 4.22 – 4.25 are modified to suit the three detector series as given below:

$$\min \left[\left(Q_t^{(1)} - Q_t^{(2)} \right)^2 + \left(Q_t^{(2)} - Q_t^{(3)} \right)^2 \right], \quad (4.26)$$

where,

t = current time, and

Q = cumulative number of vehicles at each detector,

subject to the constraints,

$$\left(Q_t^{(1)} - Q_t^{(2)} \right) \geq 0, \text{ and } \left(Q_t^{(2)} - Q_t^{(3)} \right) \geq 0, \quad (4.27)$$

$$\left(Q_t^{(1)} - Q_t^{(2)} \right) \leq z, \text{ and } \left(Q_t^{(2)} - Q_t^{(3)} \right) \leq z; z = 500, \text{ and} \quad (4.28)$$

$$Q_t^{(1)} - Q_{t-1}^{(1)} \geq 0, Q_t^{(2)} - Q_{t-1}^{(2)} \geq 0, \text{ and } Q_t^{(3)} - Q_{t-1}^{(3)} \geq 0. \quad (4.29)$$

The z value is calculated based on the known distance between the two consecutive detectors and an assumed average vehicle length. Given that the length of road is 0.5 miles (805 m), the maximum number of vehicles at jam density, assuming 25 feet (7.7 m) as the average vehicle length, is 105 vehicles per lane between the two detector locations. Hence, the maximum difference in the cumulative volumes between each pair cannot theoretically exceed 315 for the three lanes, cumulatively. A z value of 500 was used in this dissertation as the maximum number of vehicles that can be accommodated in the study length.

The cumulative volumes for the selected three consecutive detector locations for all 5 days were studied first. One sample plot for a 24-hour period on February 11, 2003, is shown in Figure 4.4.

As discussed earlier, if conservation of vehicles is followed, the cumulative volume at location 1 should always be greater than or equal to that at location 2, which in turn should be greater than or equal to location 3 values at all time intervals. Also, the maximum difference should be less

than the maximum number of vehicles that can be accommodated. Contrary to this, in the I-35 cumulative volume plot in Figure 4.4, it may be seen that the location 2 volume is consistently lower than that of location 3. This is shown enlarged in Figure 4.5 for a 1-hour period from 8:00:00 to 9:00:00. Also, the cumulative flow at location 3 became larger than both locations 1 and 2 at certain points, as shown enlarged in Figure 4.6 from 18:00:00 to 19:00:00. These are clearly violations of the conservation of vehicles principle, and show the necessity to check for the systematic errors even after standard error checking has been carried out.

It is clear from the results that some or all of the detectors under consideration are malfunctioning. There are two different ways of approaching this problem further. The first case is where the specific detectors that are malfunctioning are determined by collecting the corresponding ground truth data for each of the detectors involved. In this case one can find out the exact detectors that are malfunctioning by comparing with the ground truth data and the corrections can be applied to these detectors alone. In the second case, the ground truth data may not be available and the corrections have to be applied based on some assumptions as to which are the detectors that need to be corrected. For example, the error in the data from the three locations shown in Figure 4.4 can be from any of the 11 detectors involved. The only way to pinpoint the malfunctioning detector(s) is by a manual data collection to be carried out at each of the 11 detector points and compare with the corresponding detector data. However, most of the studies that use detector data do not collect the ground truth data. One reason for this may be that the manual data collection can be very expensive, especially if the analysis is for a long period of time over a long stretch of roadway. Moreover, most of the research studies using detector data use archived loop detector data for model development, calibration, and validation. The availability of ground-truth data for the archived data is very low as in the present dissertation.

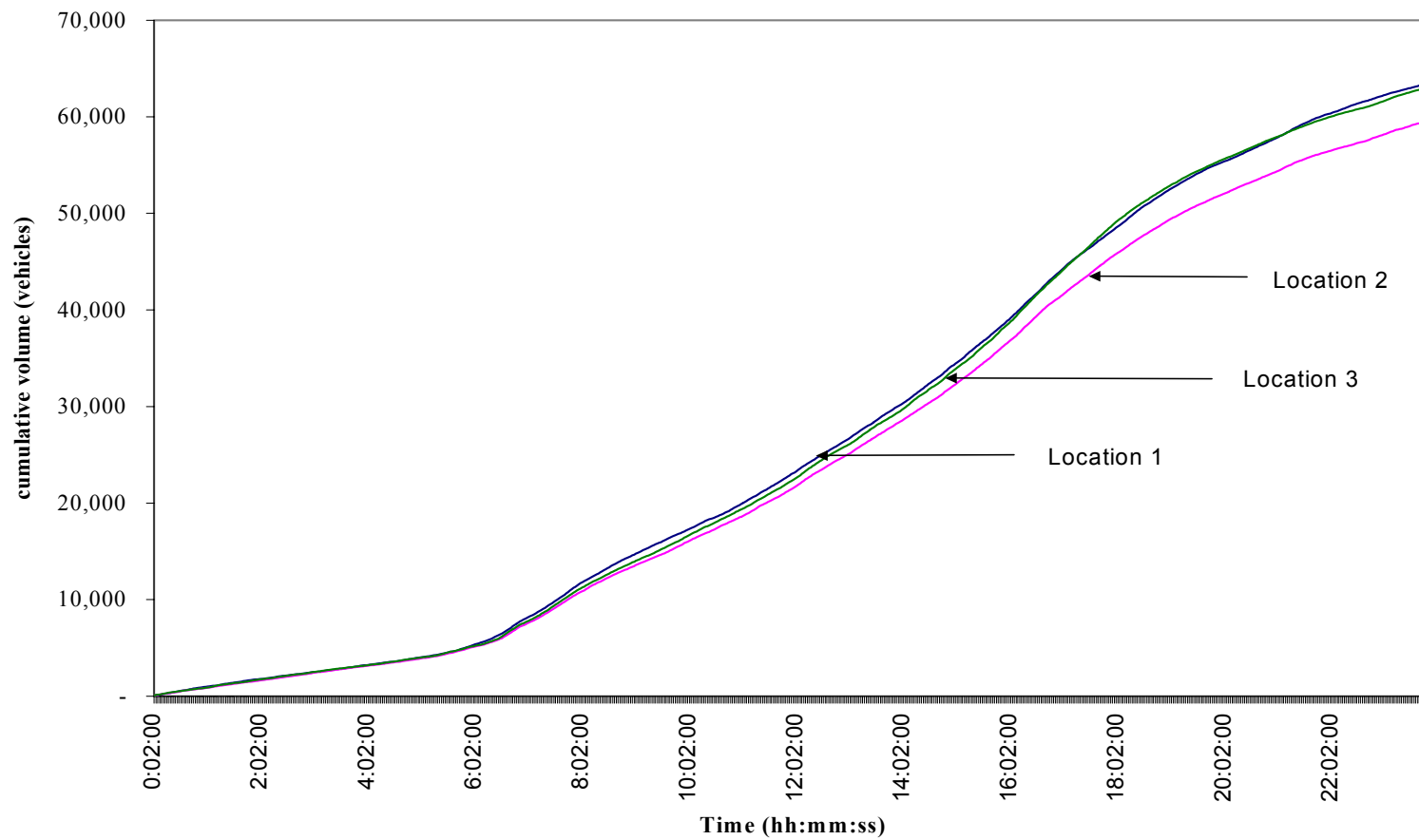


Fig. 4.4 Cumulative actual volumes for 24 hours at I-35 site on February 11, 2003

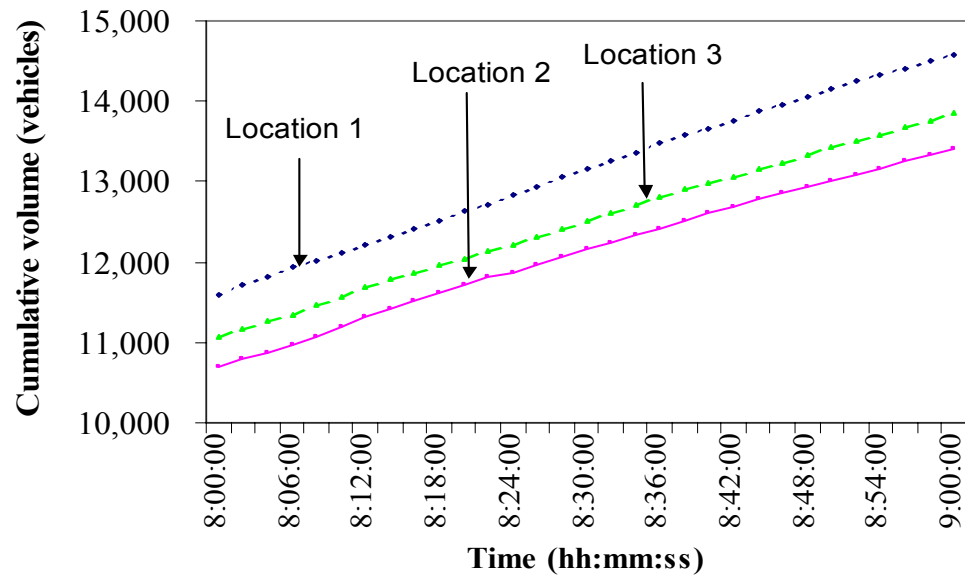


Fig. 4.5 Enlarged cumulative volumes for 1 hour at I-35 site on February 11, 2003

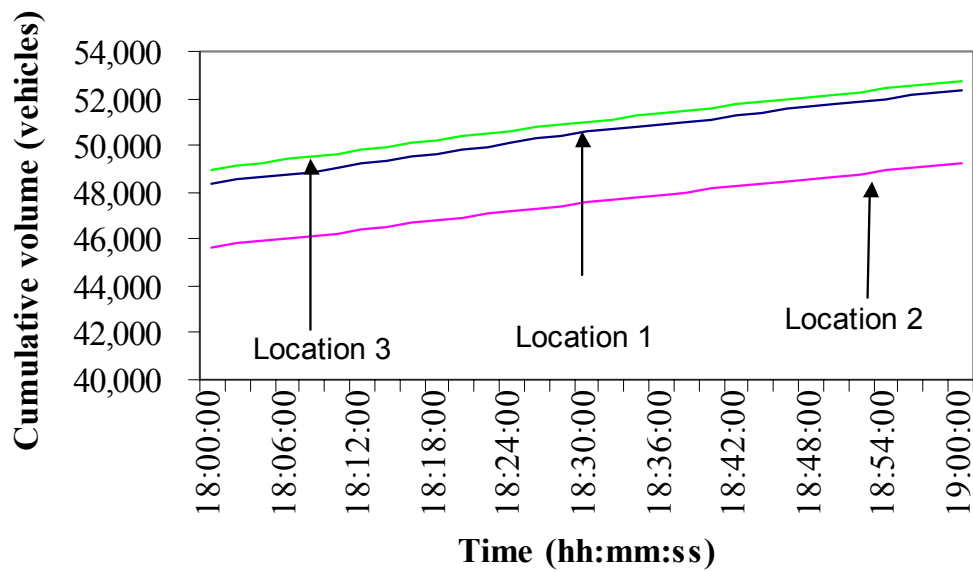


Fig. 4.6 Enlarged cumulative volumes for 1 hour at I-35 site on February 11, 2003

In this dissertation, an assumption was made that any of the detectors could be malfunctioning and hence the optimization is done equally to all the detectors under study. However, as discussed already, if the data for cross-checking are available and the malfunctioning detector(s) is(are) exactly known, the present method can still be applied specifying the particular location data to be optimized, instead of optimizing the data from all the locations.

To summarize, it was found that the field data, even after the preliminary data quality control, violated the conservation of vehicles principle. An assumption of equal error from all the detectors involved is made due to lack of specific data on which detector(s) are malfunctioning. Hence, the algorithm developed for removing the type of discrepancies in the data set should (1) make the cumulative flow at successive detector points smaller than the previous point and (2) keep the difference between the cumulative flows in the successive points not more than the maximum number of vehicles that can be accommodated within that length of road. The GRG algorithm with the objective function and constraints as given in equations 4.26-4.29 was implemented using MATLAB.

As discussed in the GRG theory, the first step in the implementation was to convert the inequality constraints to equality constraints. Thus, equations 4.27 – 4.29 need be converted to equality constraints by adding suitable variables as shown below:

$$(Q_t^{(1)} - Q_t^{(2)}) - v_1 = 0, \text{ and } (Q_t^{(2)} - Q_t^{(3)}) - v_2 = 0, \quad (4.30)$$

$$(Q_t^{(1)} - Q_t^{(2)}) + v_3 = z, \text{ and } (Q_t^{(2)} - Q_t^{(3)}) + v_4 = z, \quad (4.31)$$

$$Q_t^{(1)} - Q_{t-1}^{(1)} - v_5 = 0, \quad Q_t^{(2)} - Q_{t-1}^{(2)} - v_6 = 0, \text{ and } Q_t^{(3)} - Q_{t-1}^{(3)} - v_7 = 0. \quad (4.32)$$

As can be seen from equation 4.30-4.32 there are six original variables ($n = 6$) and are $Q_t^{(1)}$, $Q_t^{(2)}$, $Q_t^{(3)}$, $Q_{t-1}^{(1)}$, $Q_{t-1}^{(2)}$, and $Q_{t-1}^{(3)}$. Also, there are seven slack/surplus variables ($m = 7$), which are denoted as v_1 through v_7 inclusive. Thus, the total number of variables is 13, and they need to be partitioned into independent and dependent variables. As discussed earlier, this partition should be based on the condition of obtaining a nonsingular **B** matrix in equation 4.17. In this example, the cumulative volume values (Q_i) were selected as the independent variables (**Z**) and

the slack/surplus variables (v_i) are selected as the dependent variables (\mathbf{Y}). The original number of equality constraints is zero ($l = 0$). This makes the size of the \mathbf{A} and \mathbf{B} matrix (7×6) and (7×7), respectively. The reduced gradient and search direction were calculated as per equation 4.20 and 4.21. Then, the linear search is carried out until convergence is reached. In this dissertation, the convergence criterion was when all the constraints reach a value of 1×10^{-4} . For 24-hour data, the computational time varied from 12 to 24 hours on a Windows XP machine (P-4 processor, 2.4 GHz processing speed, 1-GB RAM) running MATLAB release 13.

4.5 RESULTS

The cumulative volumes after the optimization for the same site as shown in Figure 4.4 is given in Figure 4.7. From this figure it can be seen that now the flow values follow the conservation of vehicles for the entire 24-hour period without violating any constraints. Figures 4.8 and 4.9 show enlarged cumulative flow values corresponding to Figures 4.5 and 4.6. The condition shown in Figure 4.6 is one of the worst-case scenarios, where the cumulative flow at location 3 is greater than the cumulative flows at locations 1 and 2. Under this situation, the optimization brings the cumulative flows at all the three locations to equal values, meeting the minimum requirement, as shown in Figure 4.9.

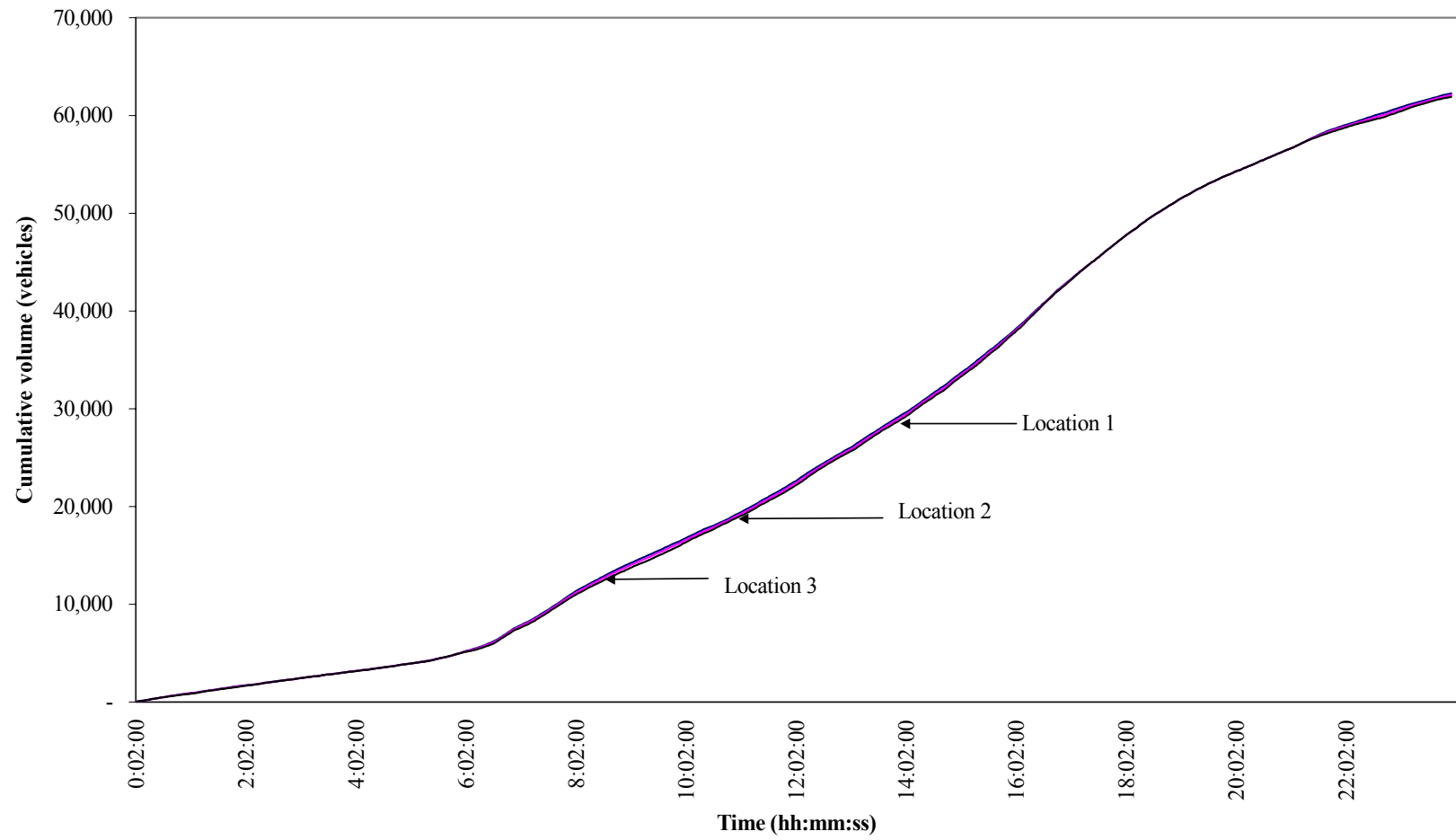


Fig. 4.7 Cumulative volumes after the optimization at I-35 site on February 11, 2003

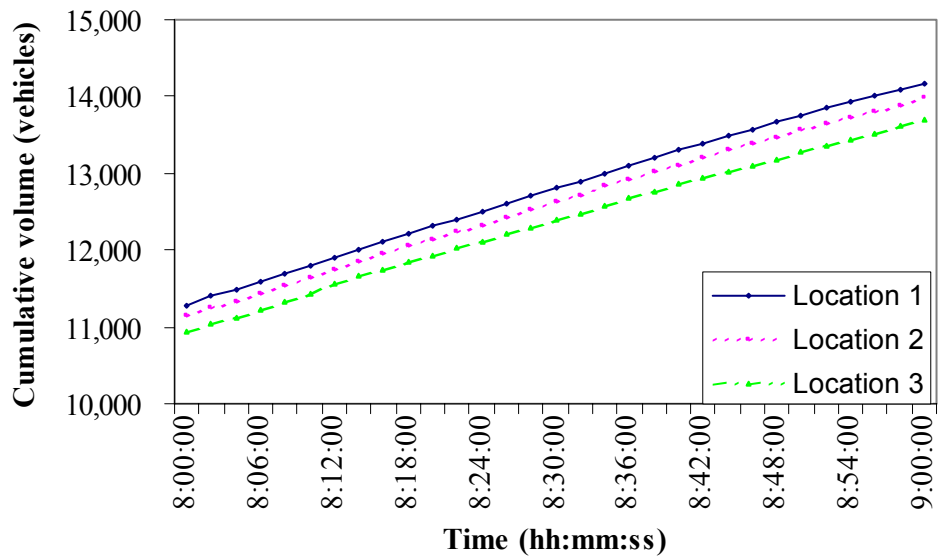


Fig. 4.8 Enlarged cumulative volumes after optimization

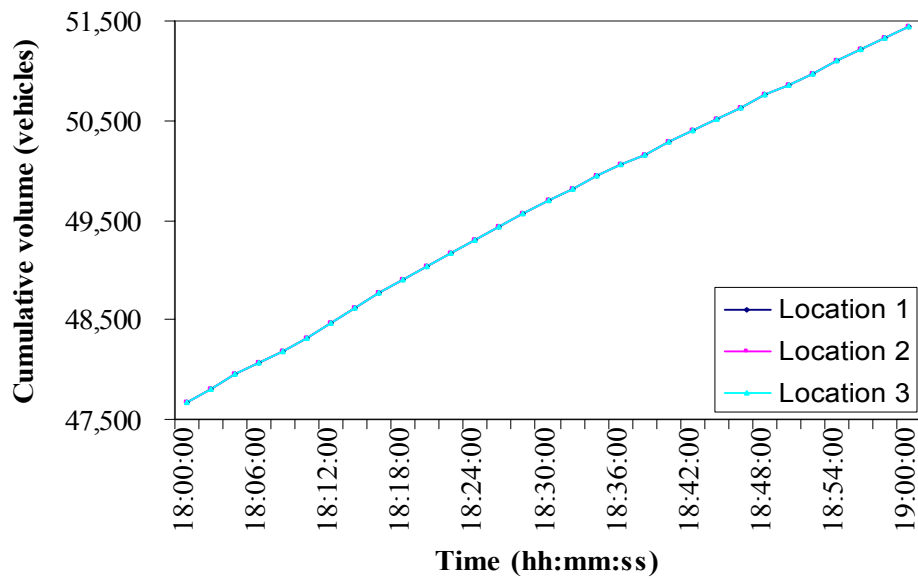


Fig. 4.9 Enlarged cumulative volumes after optimization

Similar results were obtained for other dates also. Sample graphs from February 10 to February 14, 2003, for locations 1, 2, and 3 (detector numbers 159.500 to 160.504 as shown in Figure 3.10) for different 1-hour durations are shown below to illustrate the performance of the optimization procedure under varying traffic flow conditions. Also, these figures compare the performance of the optimization under different problem scenarios.

Figure 4.10 shows the cumulative flows from locations 1, 2, and 3 from 09:00:00 to 10:00:00 on February 10, 2003. Here, the location 2 cumulative flow is more than the cumulative flow from location 1 throughout the 1-hour period. The same cumulative flows after optimization is shown in Figure 4.11. The cumulative flows after optimization follow all the constraints, thus following the conservation of vehicles principle.

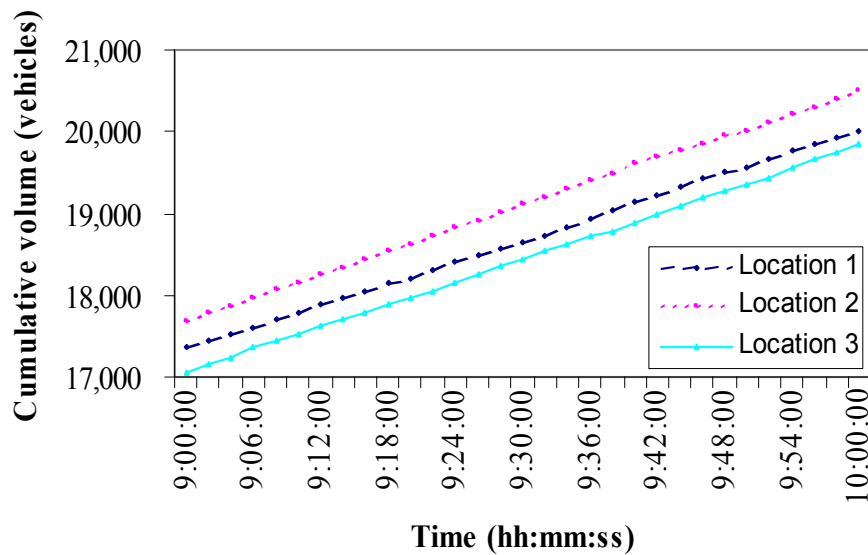


Fig. 4.10 Cumulative actual volumes during 09:00:00-10:00:00 on February 10, 2003

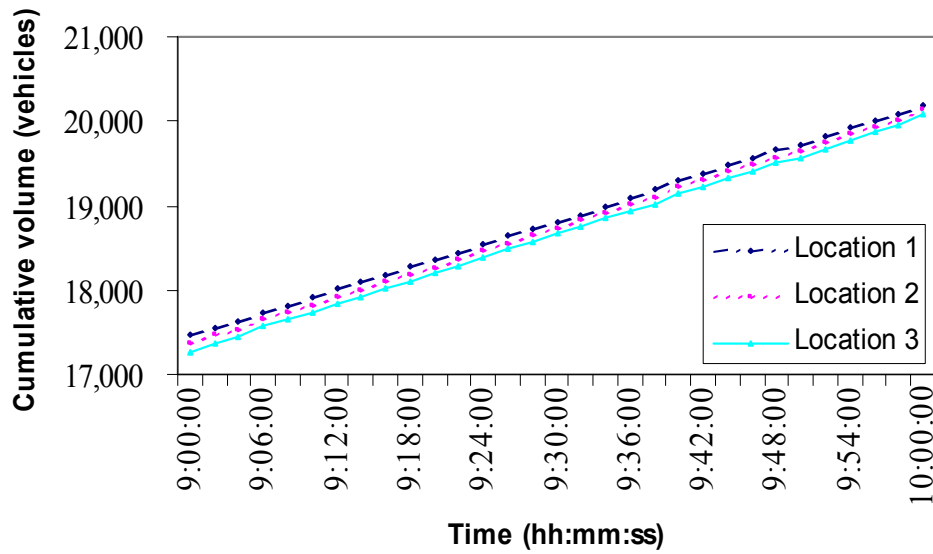


Fig. 4.11 Cumulative optimized volumes during 09:00:00-10:00:00 on February 10, 2003

Figure 4.12 shows the cumulative flows obtained on the same date from 05:00:00 to 06:00:00. In this case the location 3 cumulative flow is found to be more than the location 2 cumulative flow. Figure 4.13 shows the same cumulative flow curves after optimization and it can be seen that the optimization was able to correct the data in this scenario also.

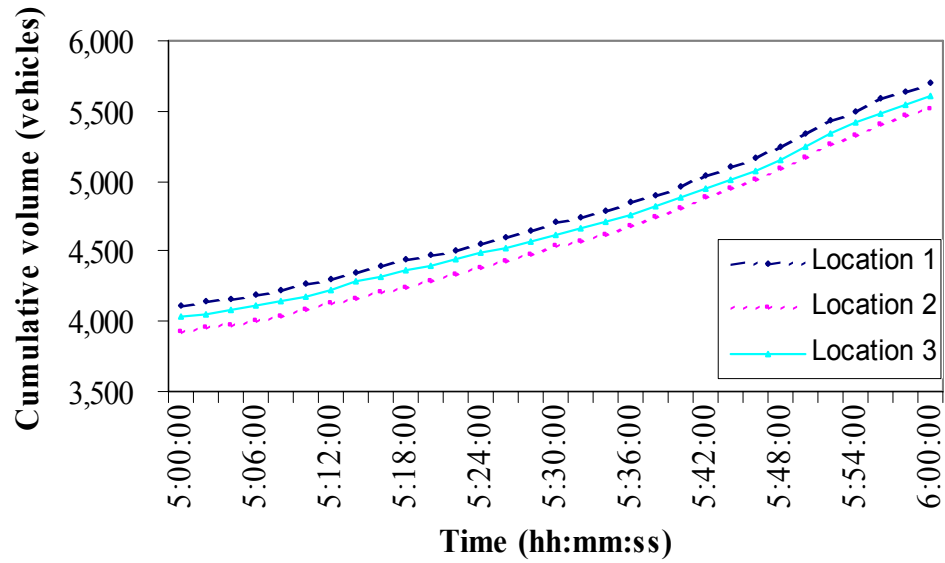


Fig. 4.12 Cumulative actual volumes during 05:00:00-06:00:00 on February 10, 2003

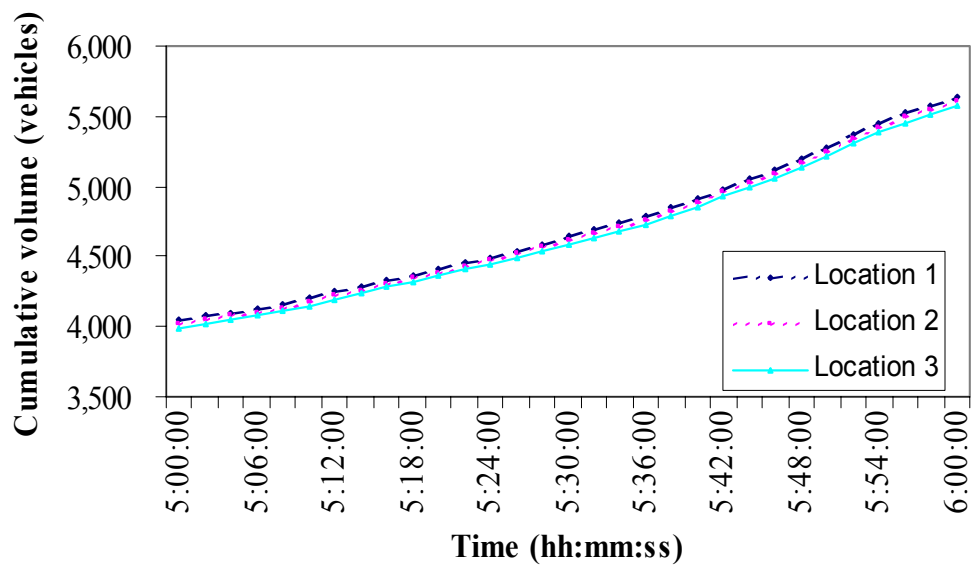


Fig. 4.13 Cumulative optimized volumes during 05:00:00-06:00:00 on February 10, 2003

Figure 4.14 shows the cumulative flows obtained February 10, 2003, from 03:00:00 to 04:00:00. In this case the location 3 cumulative flow is found to be more than location 1 and location 2 cumulative flows. However, cumulative flows at 1 and 2 were in the correct order, with location 1 cumulative flow more than location 2.

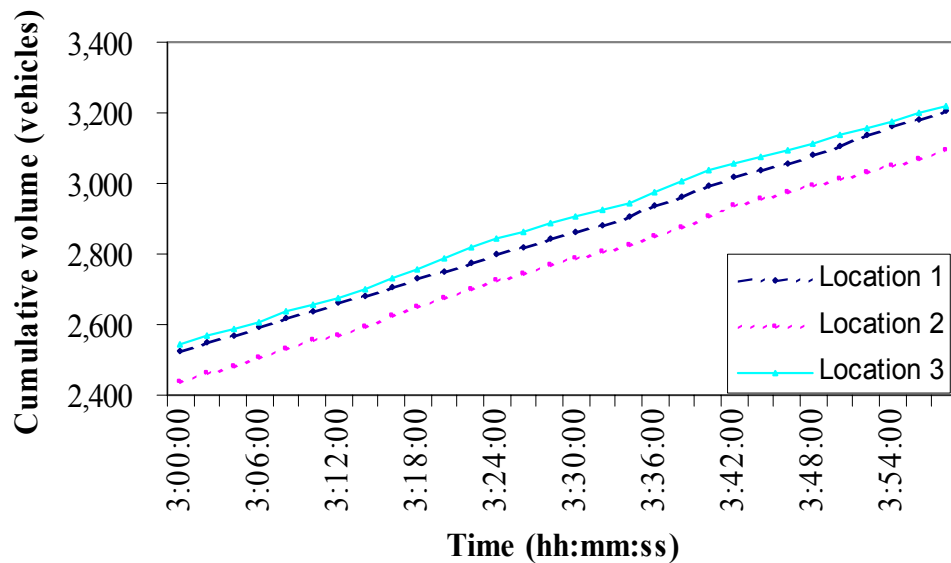


Fig. 4.14 Cumulative actual volumes during 03:00:00-04:00:00 on February 10, 2003

These cumulative flows after the optimization are shown in Figure 4.15. This is one of the worst-case scenarios, where the cumulative flow at location 3 being greater than the cumulative flows at locations 1 and 2. Under this situation, the optimization brings the cumulative flows at all the three locations to equal values, meeting the minimum requirement.

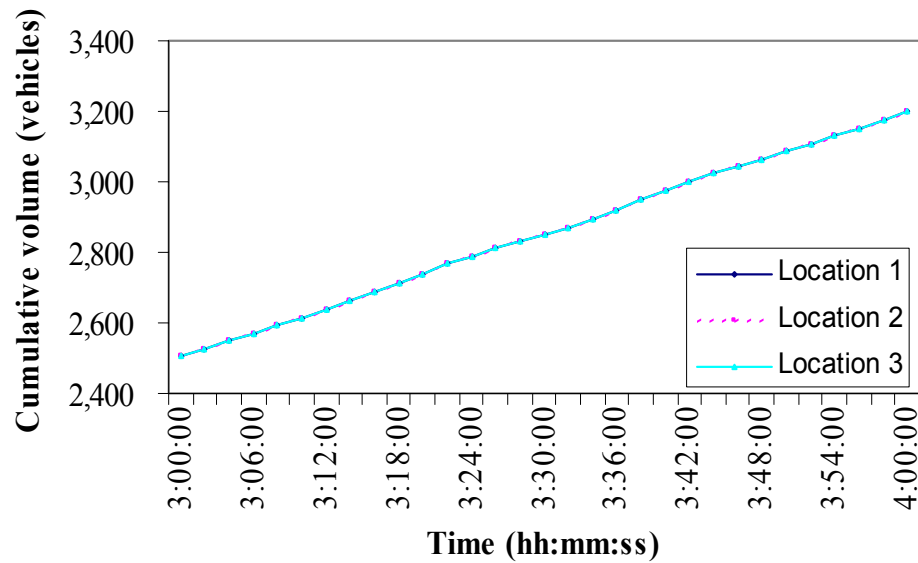


Fig. 4.15 Cumulative optimized volumes during 03:00:00-04:00:00 on February 10, 2003

Figure 4.16 shows the cumulative flows from February 11, 2003, from 05:00:00 to 06:00:00. In this case also, the location 3 cumulative flow is more than location 1 and location 2 cumulative flows. However, in this case, cumulative flow at location 1 and 2 were also violating the constraints, with location 1 cumulative flow less than that at location 2.

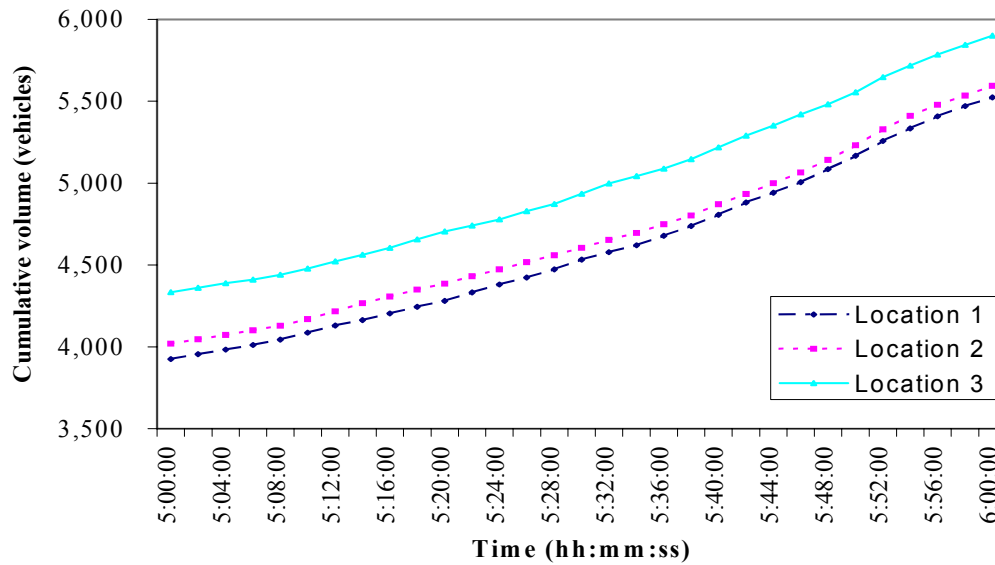


Fig. 4.16 Cumulative actual volumes during 05:00:00 to 06:00:00 on February 11, 2003

This data after the optimization is shown in Figure 4.17. It can be seen that in this scenario also, the optimization corrected the cumulative flows to meet the minimum requirements.

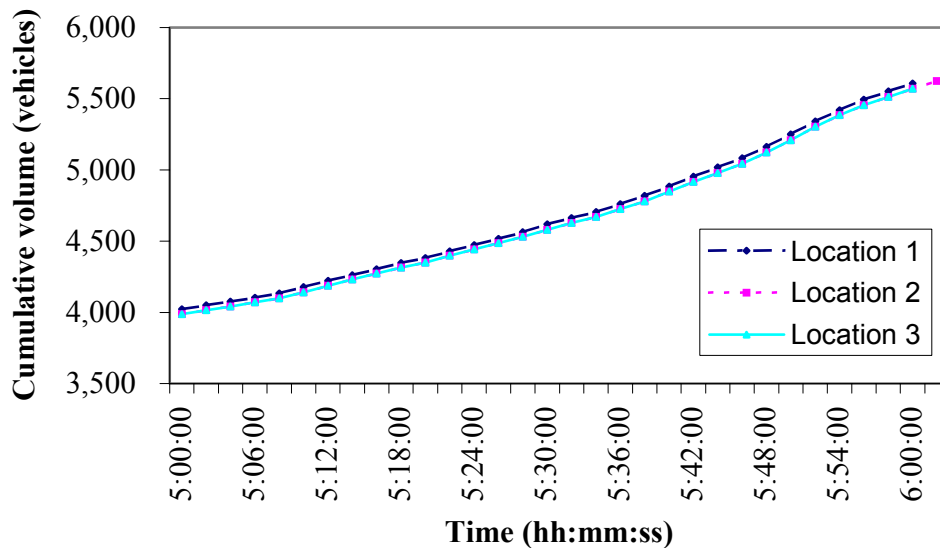


Fig. 4.17 Cumulative optimized volumes during 05:00:00 to 06:00:00 on February 11, 2003

Figure 4.18 shows the cumulative flows from February 12, 2003 from 11:30:00 to 12:30:00. Here the cumulative flow at location 3 was shown to be more than location 1 but less than that at location 2 and the corresponding values after optimization are shown in Figure 4.19.

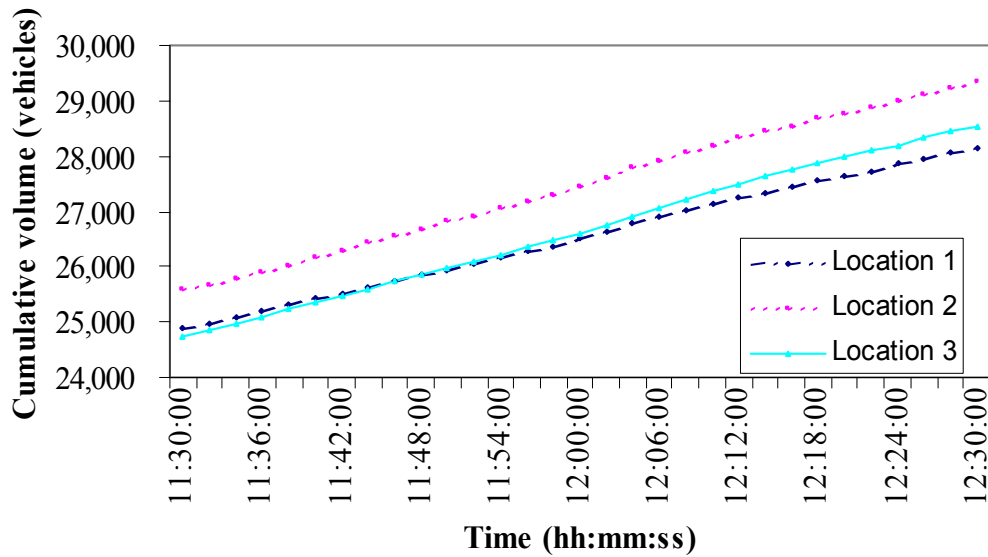


Fig. 4.18 Cumulative actual volumes during 11:30:00 – 12:30:00 on February 12, 2003

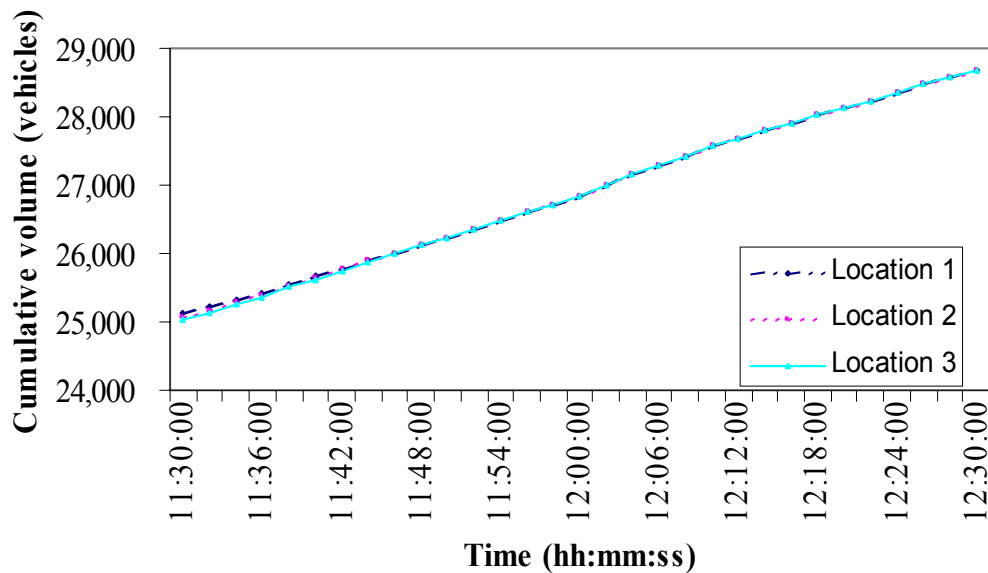


Fig. 4.19 Cumulative optimized volumes during 11:30:00 – 12:30:00 on February 12, 2003

On February 12, 2003, 10:00:00 to 11:00:00 cumulative volumes were plotted and are shown in Figure 4.20. This scenario showed cumulative flow at location 2 exceeding cumulative flow at location 1.

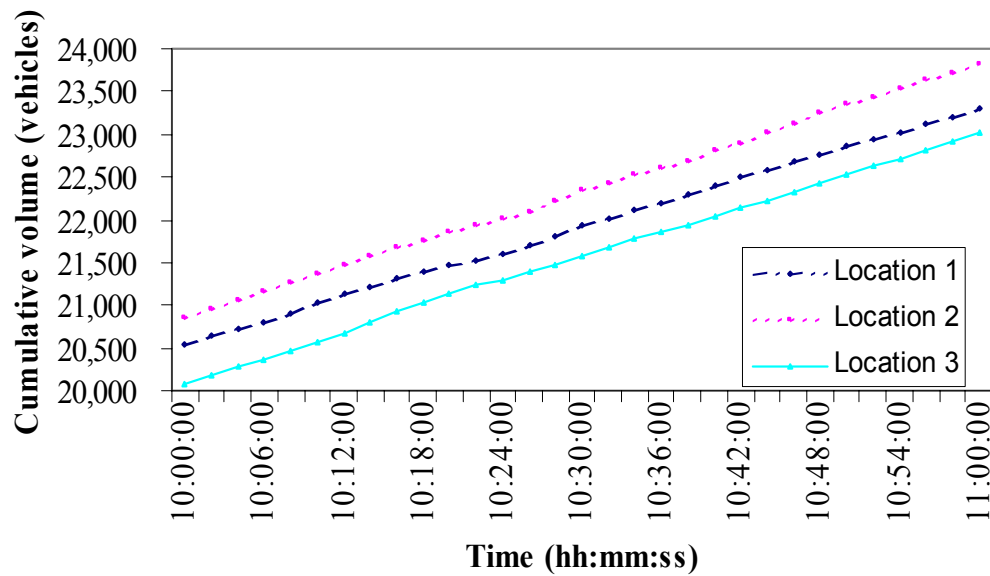


Fig. 4.20 Cumulative actual volumes during 10:00:00-11:00:00 on February 12, 2003

The values after optimization are shown in Figure 4.21. Again, the optimization was able to bring the cumulative volumes to satisfy the constraints.

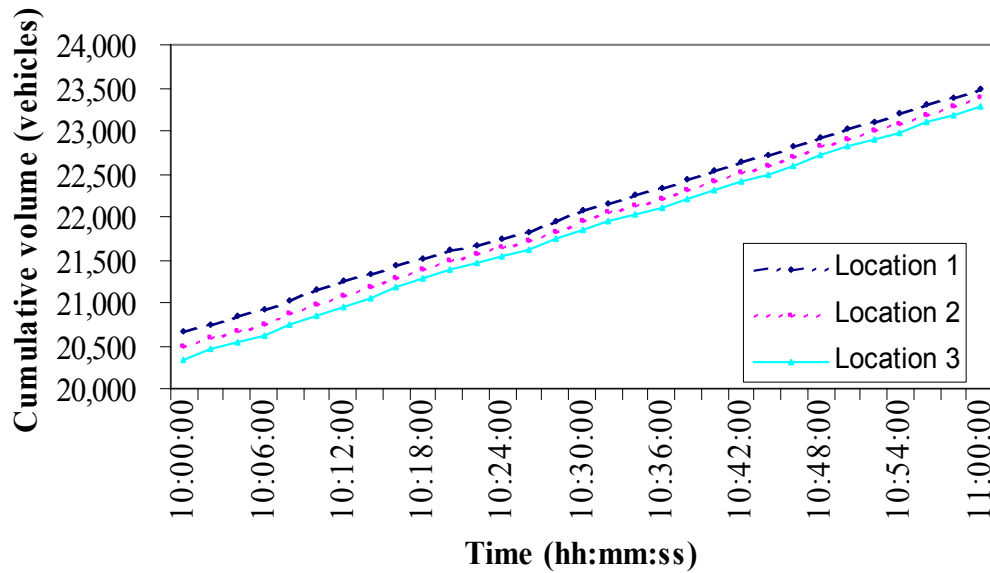


Fig. 4.21 Cumulative optimized volumes during 10:00:00-11:00:00 on February 12, 2003

On February 13, 2003, cumulative volumes from 17:00:00 to 18:00:00 were plotted and are shown in Figure 4.22. This scenario showed cumulative flow at location 2 exceeding cumulative flow at location 1, with the differences between each of the cumulative flows being very high. The corresponding flows after optimization is shown in Figure 4.23.

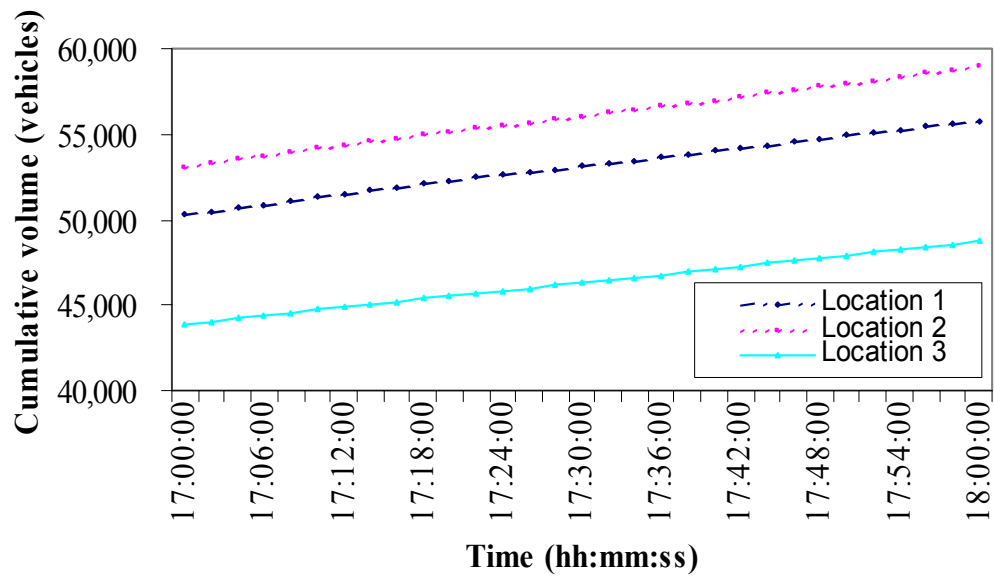


Fig. 4.22 Cumulative actual volumes during 17:00:00 – 18:00:00 on February 13, 2003

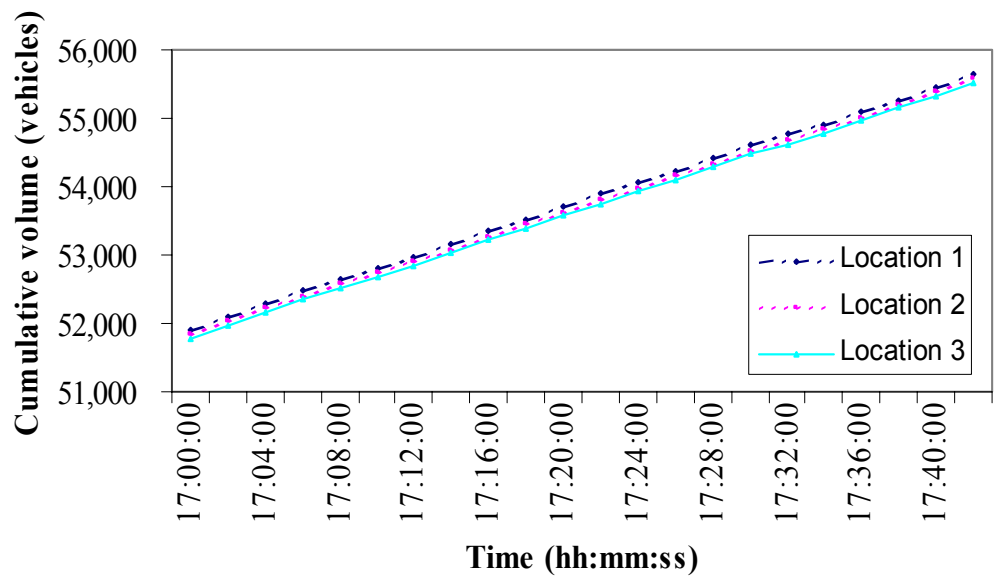


Fig. 4.23 Cumulative optimized volumes during 17:00:00 – 18:00:00 on February 13, 2003

Analysis of February 14, 2003, data from 08:30:00 to 09:30:00 is shown in Figure 4.24. In this case the cumulative volumes were in the correct order. However, the difference exceeded the maximum allowed difference.

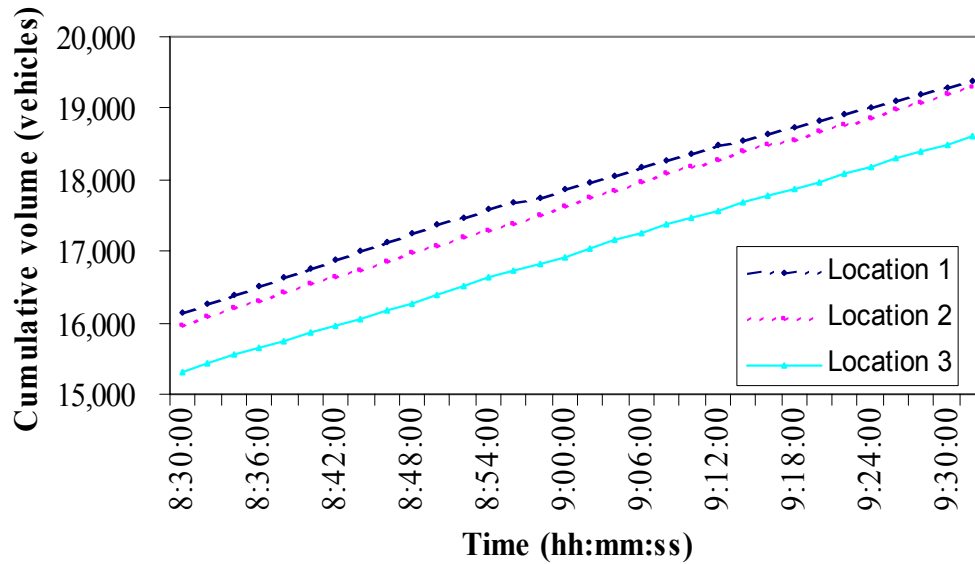


Fig. 4.24 Cumulative actual volumes during 08:30:00 – 09:30:00 on February 14, 2003

The corresponding volumes after optimization are shown in figure 4.25.

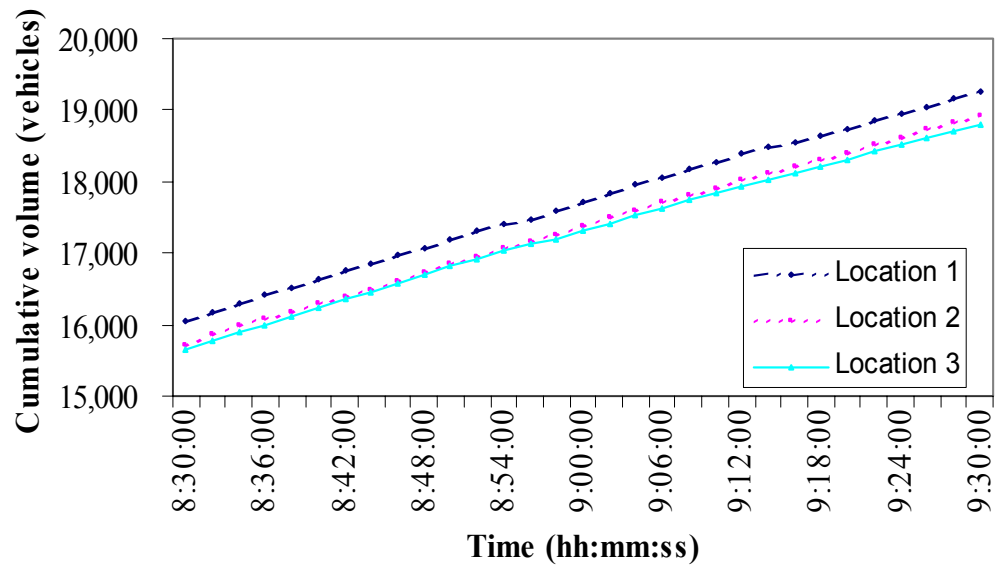


Fig. 4.25 Cumulative optimized volumes during 08:30:00 – 09:30:00 on February 14, 2003

Figure 4.26 shows 13:00:00 to 14:00:00 data from the same day and it showed the cumulative flow at location 2 exceeding that at location 1. The corresponding cumulative curve after optimization is shown in Figure 4.27 with all the values following the constraints.

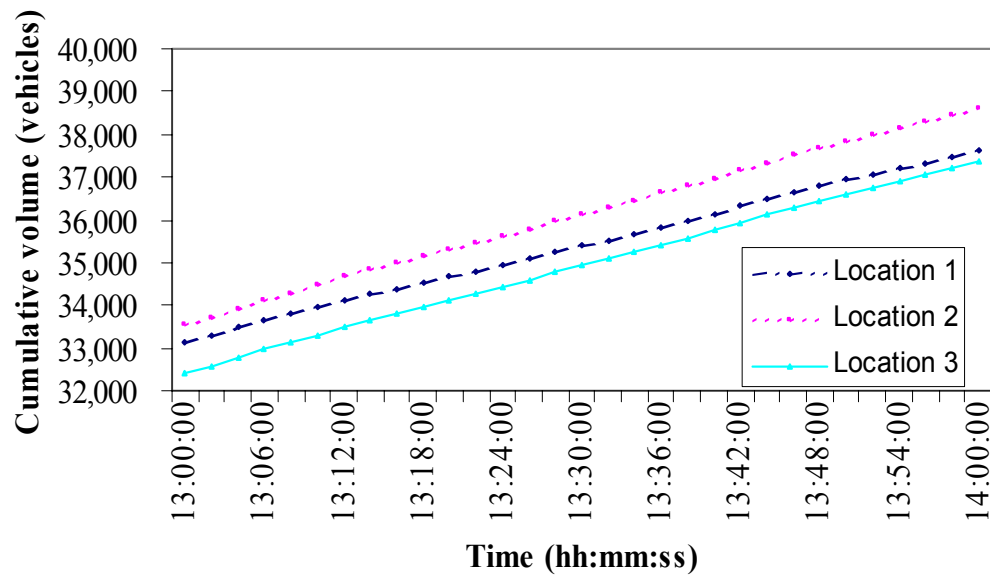


Fig. 4.26 Cumulative actual volumes during 13:00:00 to 14:00:00 on February 14, 2003

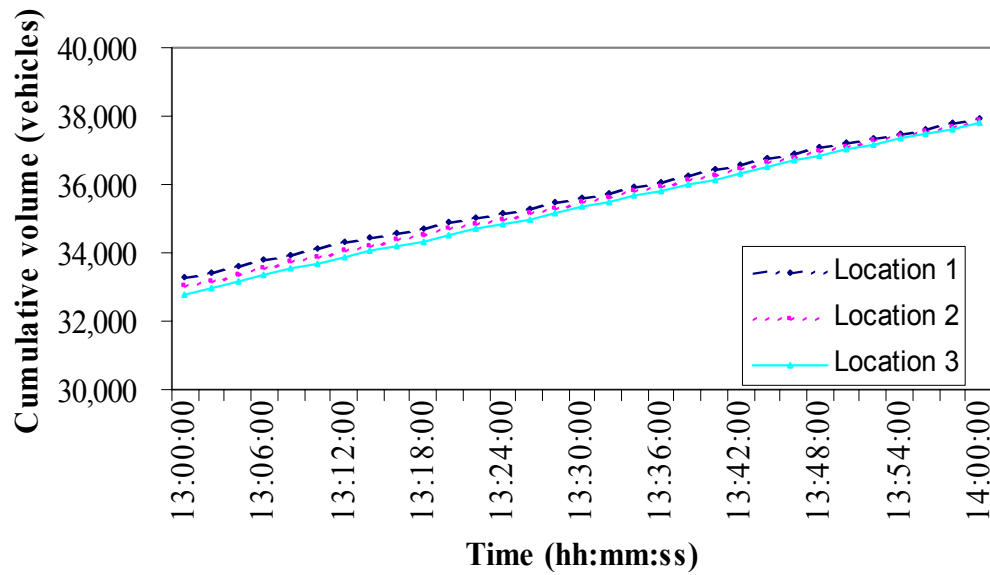


Fig. 4.27 Cumulative optimized volumes during 13:00:00 to 14:00:00 on February 14, 2003

Overall, it was found that the data from all the 5 days under consideration was violating the conservation of vehicles principles at different time periods and that the developed optimization technique was able to correct the values under all the different problem scenarios. Thus, the optimization proves to be a good technique for correcting the flow data when it violates the conservation of vehicles principle under different scenarios that can happen in the field data.

As is obvious from the objective function and the constraints, the algorithm can be extended to any number of detector series. In this dissertation, the analysis is extended to a five-detector series for illustration. The objective function and the constraints in the case of a five-detector series are as given below:

$$\min \left[\left(Q_t^{(1)} - Q_t^{(2)} \right)^2 + \left(Q_t^{(2)} - Q_t^{(3)} \right)^2 + \left(Q_t^{(3)} - Q_t^{(4)} \right)^2 + \left(Q_t^{(4)} - Q_t^{(5)} \right)^2 \right], \quad (4.33)$$

Where,

t = current time, and

Q = cumulative number of vehicles at each detector,

subject to the constraints,

$$\left(Q_t^{(1)} - Q_t^{(2)} \right) \geq 0, \left(Q_t^{(2)} - Q_t^{(3)} \right) \geq 0, \left(Q_t^{(3)} - Q_t^{(4)} \right) \geq 0, \text{ and } \left(Q_t^{(4)} - Q_t^{(5)} \right) \geq 0, \quad (4.34)$$

$$\left(Q_t^{(1)} - Q_t^{(2)} \right) \leq z, \left(Q_t^{(2)} - Q_t^{(3)} \right) \leq z, \left(Q_t^{(3)} - Q_t^{(4)} \right) \leq z, \text{ and } \left(Q_t^{(4)} - Q_t^{(5)} \right) \leq z, \quad (4.35)$$

$$Q_t^{(1)} - Q_{t-1}^{(1)} \geq 0, Q_t^{(2)} - Q_{t-1}^{(2)} \geq 0, Q_t^{(3)} - Q_{t-1}^{(3)} \geq 0, Q_t^{(4)} - Q_{t-1}^{(4)} \geq 0, \text{ and} \quad (4.36)$$

$$Q_t^{(5)} - Q_{t-1}^{(5)} \geq 0.$$

Here, the number of original variables is 10, and the number of slack/surplus variables is 13, making a total of 23 variables. The size of the resulting **A** and **B** matrix will be (13×10) and (13×13) , respectively.

It was found that the complexity of the problem as well as the computational time increases with increase in the number of variables. This can be seen from the increase in the size of **A** matrix from (7×6) to (13×10) and **B** matrix from (7×7) to (13×13) , when the analysis was changed

from 3-detector series to 5-detector series. This increase in size of the matrices will lead to more computational time for the matrix operations involved such as matrix inversion, multiplication, etc. For example, the optimization of a one-day data (24-hours data having 720 data records) iterates the matrix manipulations approximately 1600 times for each record (i.e. a total of $720 \times 1600 = 1152000$ times). For a 3-detector data series, the matrix sizes were (7×6) and (7×7) and for a five day data, these matrix operations have to be performed on (13×10) and (13×13) matrices. This lead to high computation times and, hence, in this dissertation only one sample run was conducted for a five-detector series, just to illustrate the performance of the optimization method for longer sections.

Sample results from February 10, 2003, for a series of five detectors are shown in Figures 4.28 and 4.29. This included all the five detector locations shown in Figure 3.10. Figure 4.28 shows the cumulative flows before optimization for a 1-hour period from 06:30:00 to 07:30:00. It can be seen that the conservation of vehicles is violated, with location 4 cumulative volume greater than the cumulative values at locations 1, 2, or 3. Figure 4.29 shows the cumulative volumes after optimization for the same time period. Similar figures for another 1-hour period from 08:00:00 – 09:00:00 on February 10, 2003, are shown in Figures 4.30 and 4.31. In Figure 4.30 it can be seen that the cumulative flow at location 2 is more than that at location 1. Figure 4.31 shows the corresponding cumulative volumes after optimization. Thus, the optimization procedure proves to be useful for optimizing longer sections with more detectors.

Tables 4.1 and 4.2 show the minimum and maximum values in the original and optimized 24-hour data for all 5 days. From the results it can be seen that the optimization method was able to correct data under varying traffic flow conditions for longer sections also.

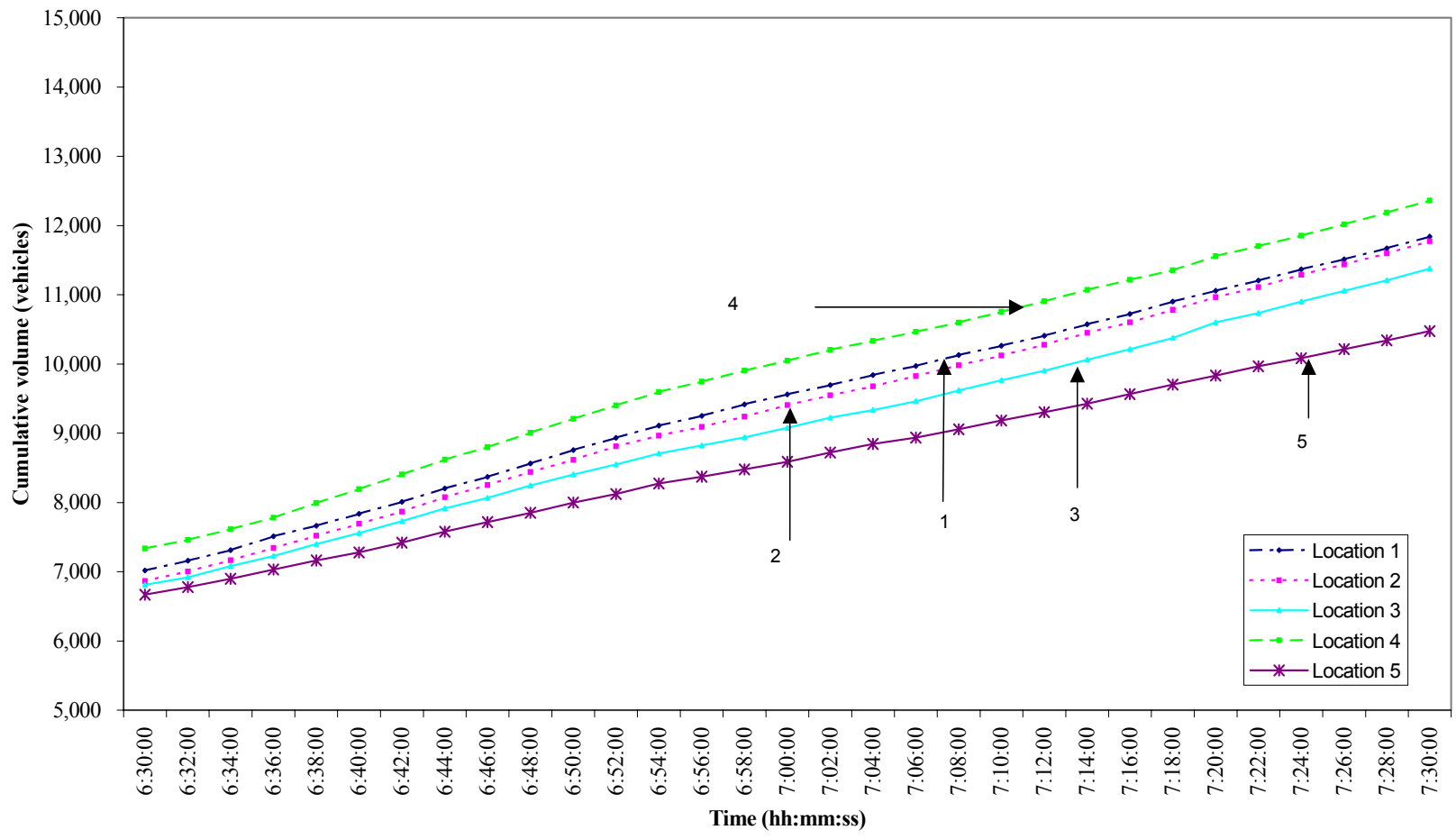


Fig. 4.28 Cumulative actual volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405

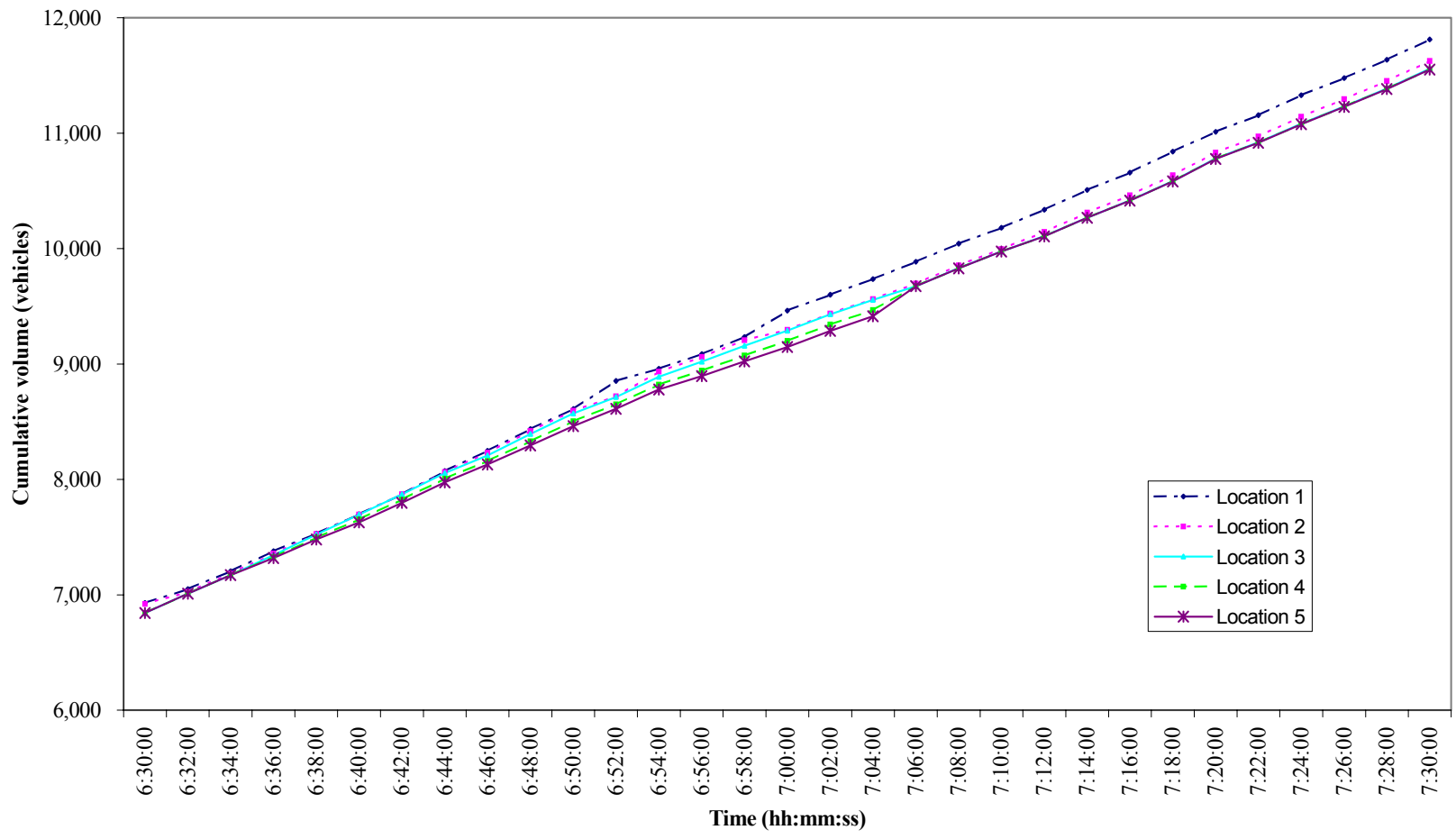


Fig. 4.29 Cumulative optimized volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405

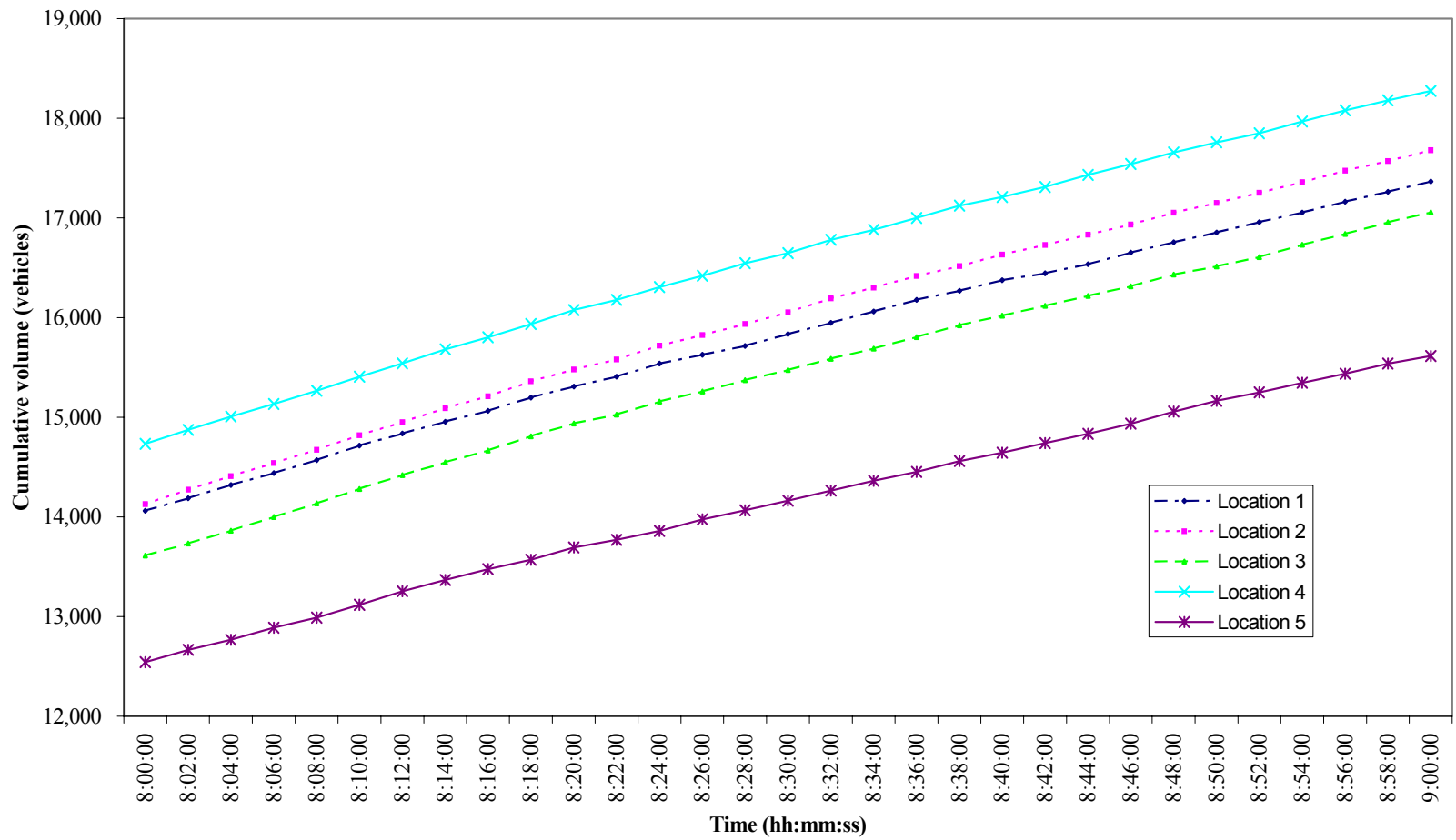


Fig. 4.30 Cumulative actual volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405

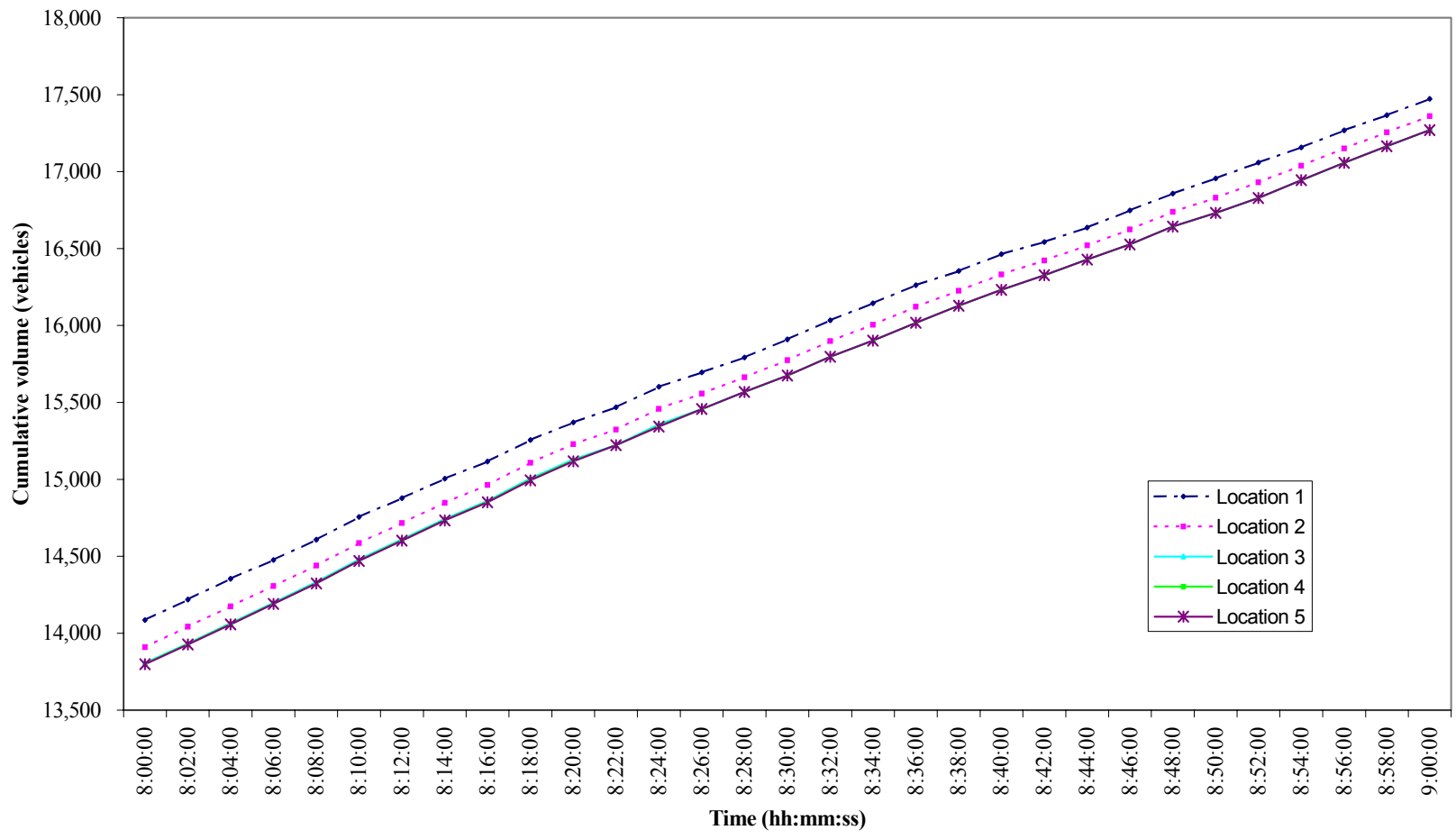


Fig. 4.31 Cumulative optimized volumes for 1 hour on February 10, 2003, for five consecutive detector stations from 159.500 to 161.405

Table 4.1 Data Details at the Study Sites Before Optimization

Date	Number of vehicles in link 1 before optimization		Number of vehicles in link 2 before optimization		Number of vehicles in link 3 before optimization		Number of vehicles in link 4 before optimization	
	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum
February 10, 2003	-4922	195	-136	1747	-2934	40	7	11167
February 11, 2003	-5090	126	-128	2340	-3232	-3	2	12273
February 12, 2003	-4387	9	-53	1868	-3038	-1	-4249	1421
February 13, 2003	-5015	79	-149	2001	-3757	-15	1	12833
February 14, 2003	-3564	303	-4	2201	-3287	2	12	12744

Table 4.2 Data Details at the Study Sites After Optimization

Date	Number of vehicles in link 1 after optimization		Number of vehicles in link 2 after optimization		Number of vehicles in link 3 after optimization		Number of vehicles in link 4 after optimization	
	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum
February 10, 2003	0	216	0	110	0	89	0	54
February 11, 2003	0	109	0	69	0	105	0	59
February 12, 2003	0	183	0	136	0	70	0	66
February 13, 2003	0	186	0	124	0	86	0	49
February 14, 2003	0	498	0	220	0	71	0	39

4.6 VALIDATION

Validation of the optimization procedure can be carried out either using field ground truth data or using simulated data. However, as discussed earlier, the present study was using archived data and the corresponding ground truth flow data were not available. Hence, simulated data generated using CORSIM simulation software was used for validation purposes. The use of simulated data also has the advantage that there can be more control over the data, and it will be easier to carry out sensitivity analysis for varying amounts of errors.

A traffic network similar to the field test bed was created in CORSIM, and detectors were placed at 0.5-mile distances as discussed in Chapter III. Data were generated for 5 hours from 06:00:00 to 10:00:00. Traffic volumes from the field were used as input to CORSIM at 30-minute intervals. Ten different flow rates were input for the 5-hour study, in order to have simulated flow variations comparable to the field data variations. Detectors were placed in each link to collect the flow, speed, and occupancy rate. The output data were extracted from the simulation at 1-minute intervals as detailed in Chapter III.

In the field, different types of malfunctions occur to loop detectors, most of which may be identified by analyzing the detector data at individual locations. However, analyzing the data at individual locations may not identify systematic errors, such as detectors continuously undercounting or overcounting the vehicles. This kind of constant bias in the data is one of the main reasons for the violation of conservation of vehicles in the data. These are the errors that are to be identified and corrected by the present optimization procedure. Hence, such errors were introduced in the simulation data and the performance of optimization was studied. A sensitivity analysis was carried out to find out the performance of optimization under varying types and magnitudes of errors. The purpose of this sensitivity analysis was to find out the range up to which the optimization procedure can be applied as an acceptable procedure for diagnosing the data.

Three consecutive detectors were selected from the simulation and a four-hour data were used for the sensitivity analysis. First the effect of constant undercounting or overcounting of the detectors was studied. The detector data without any error, if given as

input to the optimization will not be optimized as it satisfies all of the minimum requirements, thus giving an MAPE of zero. The error was first introduced in the data as a constant 10% over counting at detector location 2 by adding 10% of actual data to the observations as given below:

$$q_{new}^t = q_{old}^t(1 + \beta), \quad (4.37)$$

where,

q_{new}^t = data after introducing error,

q_{old}^t = actual data, and

β = bias introduced.

The optimization procedure was carried out on the data with the introduced error. The simulated data, the data with errors, and the data after optimization were compared. Figure 4.32 shows the plot of the actual data, the data after introducing the error, and the data after optimization, for the detector for which the error was introduced. It can be seen that the optimization was able to correct the error in the data in this case with a minimal change to the original data. Figure 4.33 shows the effect of this optimization on the corresponding cumulative flow data of the same detector.

The optimized volumes are compared with the corresponding actual values obtained from the simulation to check whether the integrity of the original data is maintained after optimization. MAPE, as defined in Chapter III, is used as a performance measure. MAPE for the data with errors and for the optimized data were calculated with respect to the true simulated values. The magnitude of the MAPE values reduced from 10% to 4.57% after optimization.

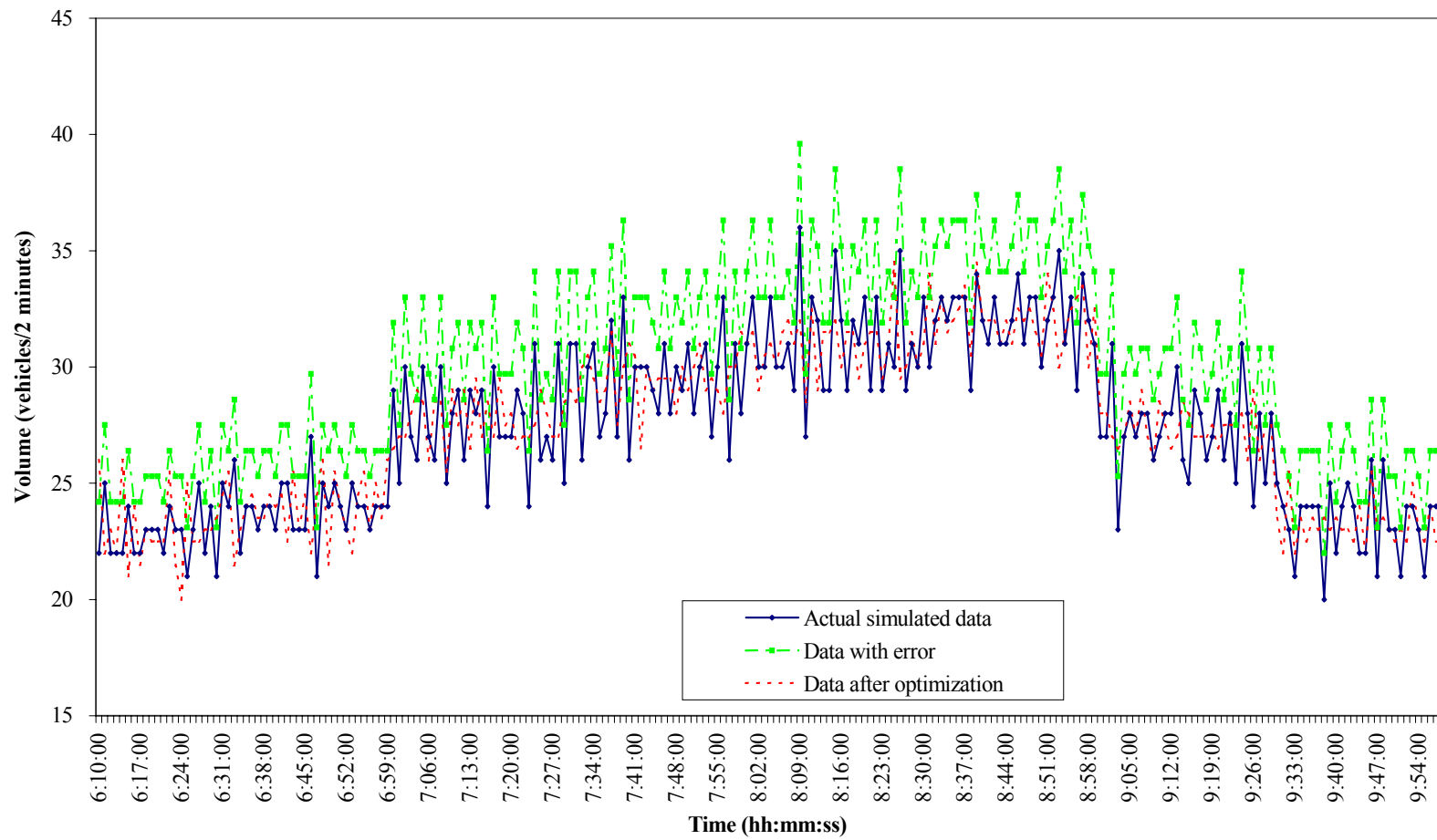


Fig. 4.32 Validation of optimization performance using simulated data

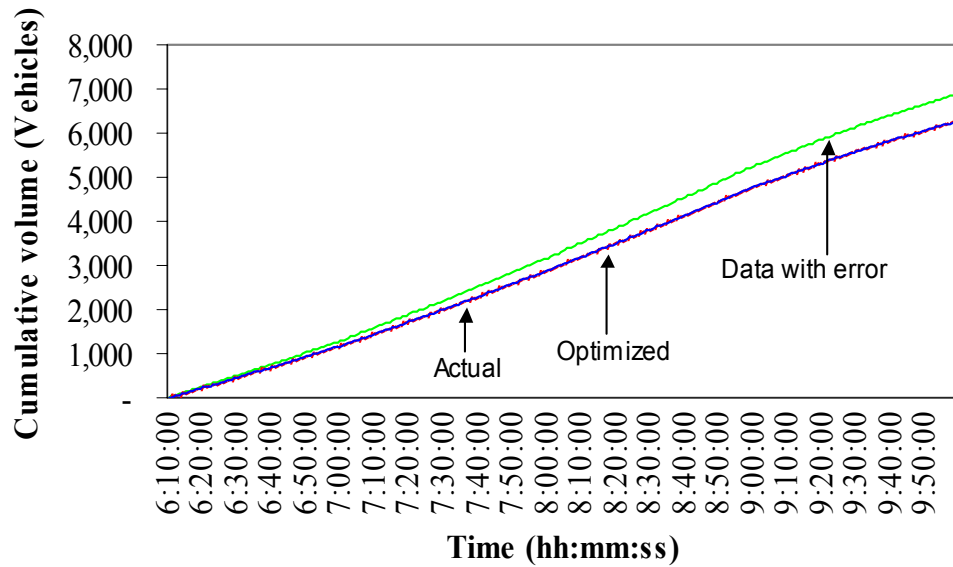


Fig. 4.33 Validation of optimization performance using simulated cumulative data

This analysis was repeated with varying error values. The error was varied from 1% to 150% and the results are shown in Table 4.3 and are plotted in figure 4. 34. It can be seen that as the error in the input data is increasing, the MAPE between the optimized value and the actual value also increasing.

Table 4.3 MAPE with Varying Amount of Over Counting Error in the Input Data

Error%	MAPE
0	0
1	1.563299
10	4.579592
20	7.947224
30	11.25403
40	14.61498
50	17.97029
70	24.67011
100	34.70231
150	50.94552

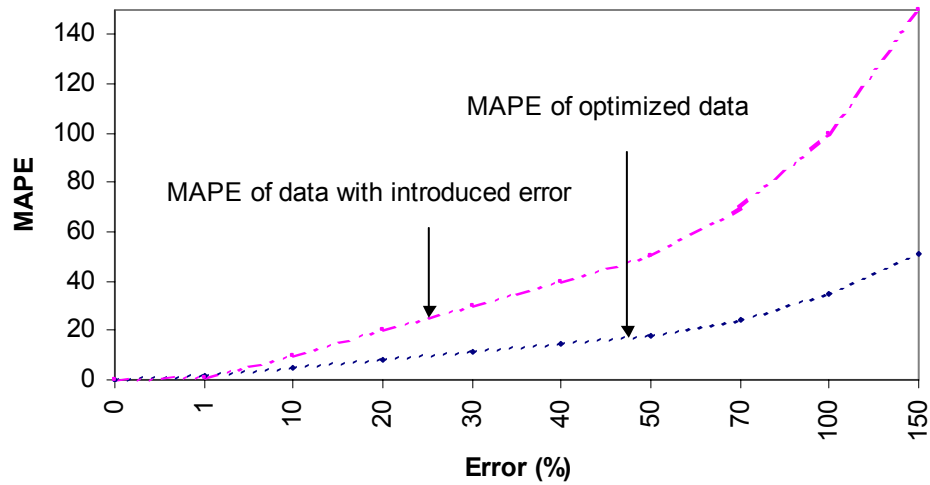


Fig. 4.34 Performance of the optimization with varying amount of over counting of the detector

In a similar way, the effect of undercounting on the optimization performance was carried out. The results obtained are shown in Figure 4.35.

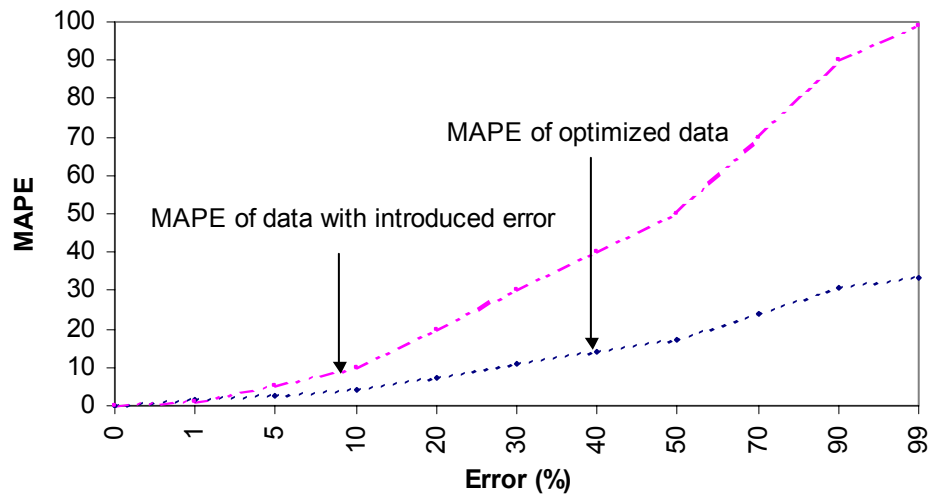


Fig. 4.35 Performance of the optimization with varying amount of under counting of the detector

The effect of having random error at a detector location is also studied. A normal distribution was assumed for the error distribution with the following density function.

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad (4.38)$$

where,

x = value for which normal distribution is needed,

μ = mean of the distribution, and

σ = standard deviation of the distribution.

The standard deviation was varied from 0 to 50 in this dissertation. The MAPE in each of the cases was calculated and is plotted in Figure 4.36.

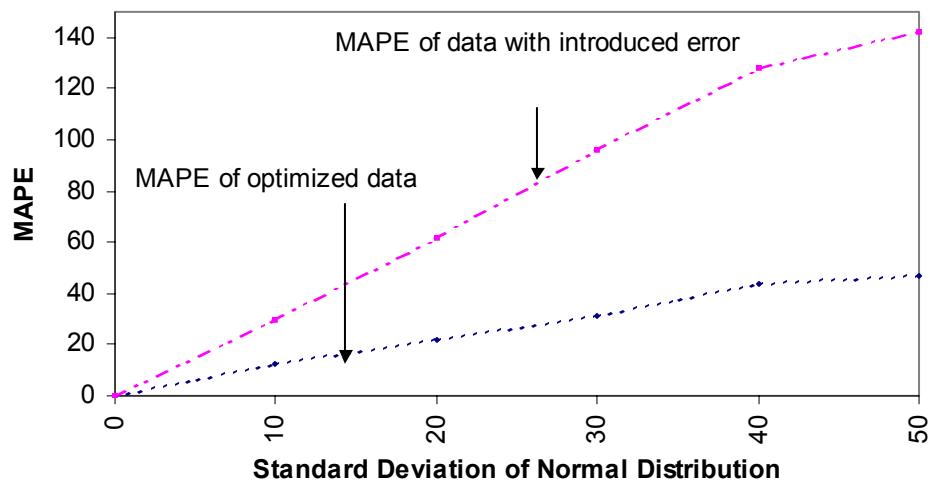


Fig. 4.36 Performance of the optimization with varying amount of random error

It can be seen that the optimization was able to give acceptable results, assuming a 40% MAPE as the maximum acceptable error, up to 100% over counting or under counting. In the case of random errors, the optimization gave acceptable results up to a standard deviation of 40. Thus, it can be seen that the optimization procedure was able to perform well with constant errors as well

as random errors.

Also, the performance of the optimization was checked under situations where 2 out of the 3 detectors are malfunctioning. This was carried out under two different scenarios. The first scenario considered two detectors having constant bias in the data and the second scenario considered one detector having a random error and the other having a constant bias.

One sample run was carried out for each of the two scenarios discussed above. The results of the first scenario with $\beta_1 = -10\%$ and $\beta_2 = 20\%$ at location 1 and 2 are shown in Figures 4.37 and 4.38 respectively. Figure 4.37 shows the actual volume, volume with introduced error and the volumes after the optimization for location 1. Similar figure for location 2 is shown in Figure 4.38. It can be seen that even with two out of the three detectors having error, the optimization was able to reduce the error in the data. The MAPE value reduced from 10% to 5% at location 1 and from 20% to 6% at location 2.

In the second scenario, optimization was carried out with a constant error of 10% at the first location and a random error following normal distributions with a standard deviation of 10 at the second location. The results obtained are given in Figures 4.39 and 4.40

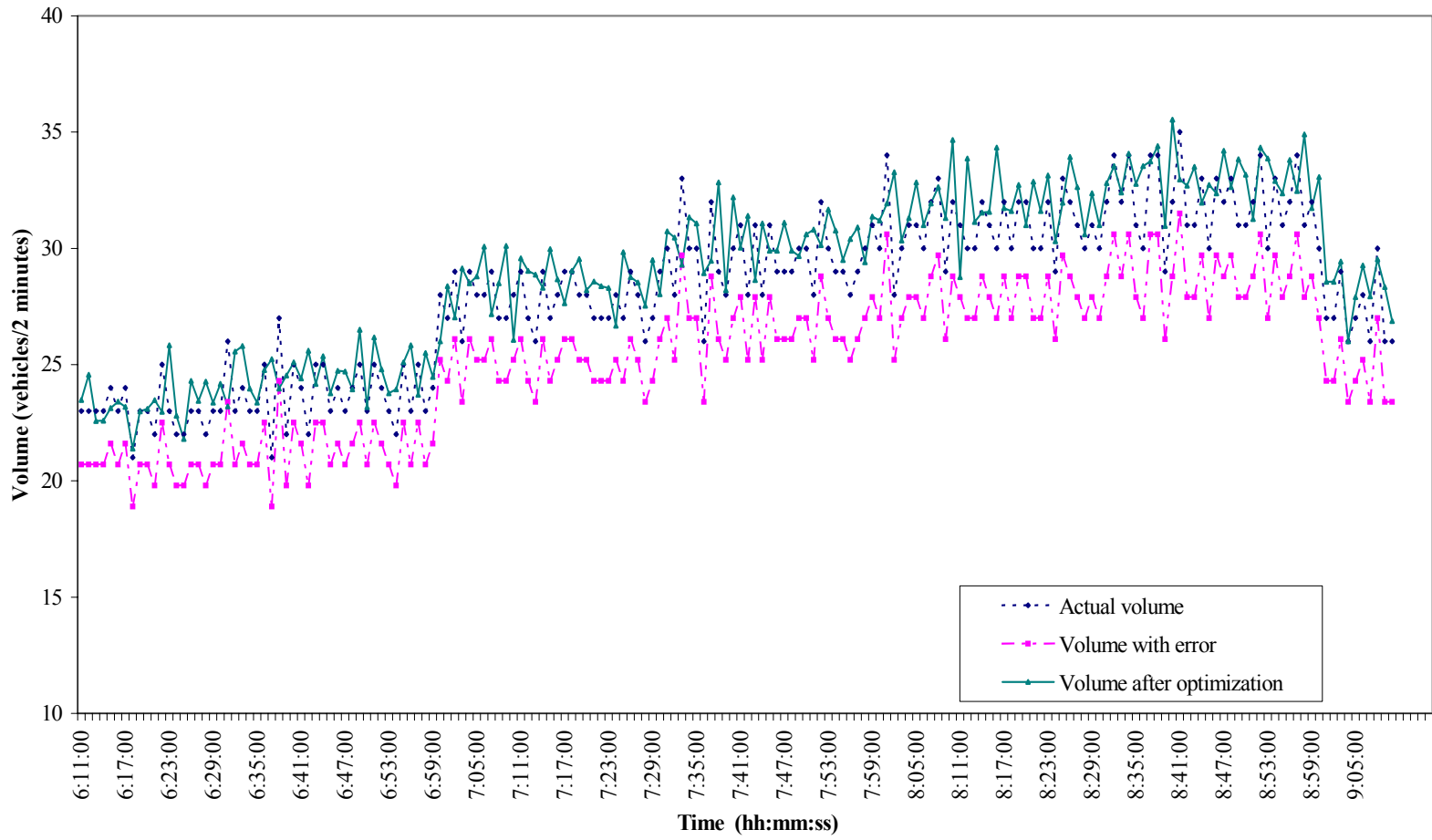


Fig. 4.37 Comparison of the performance of optimization at Location 1

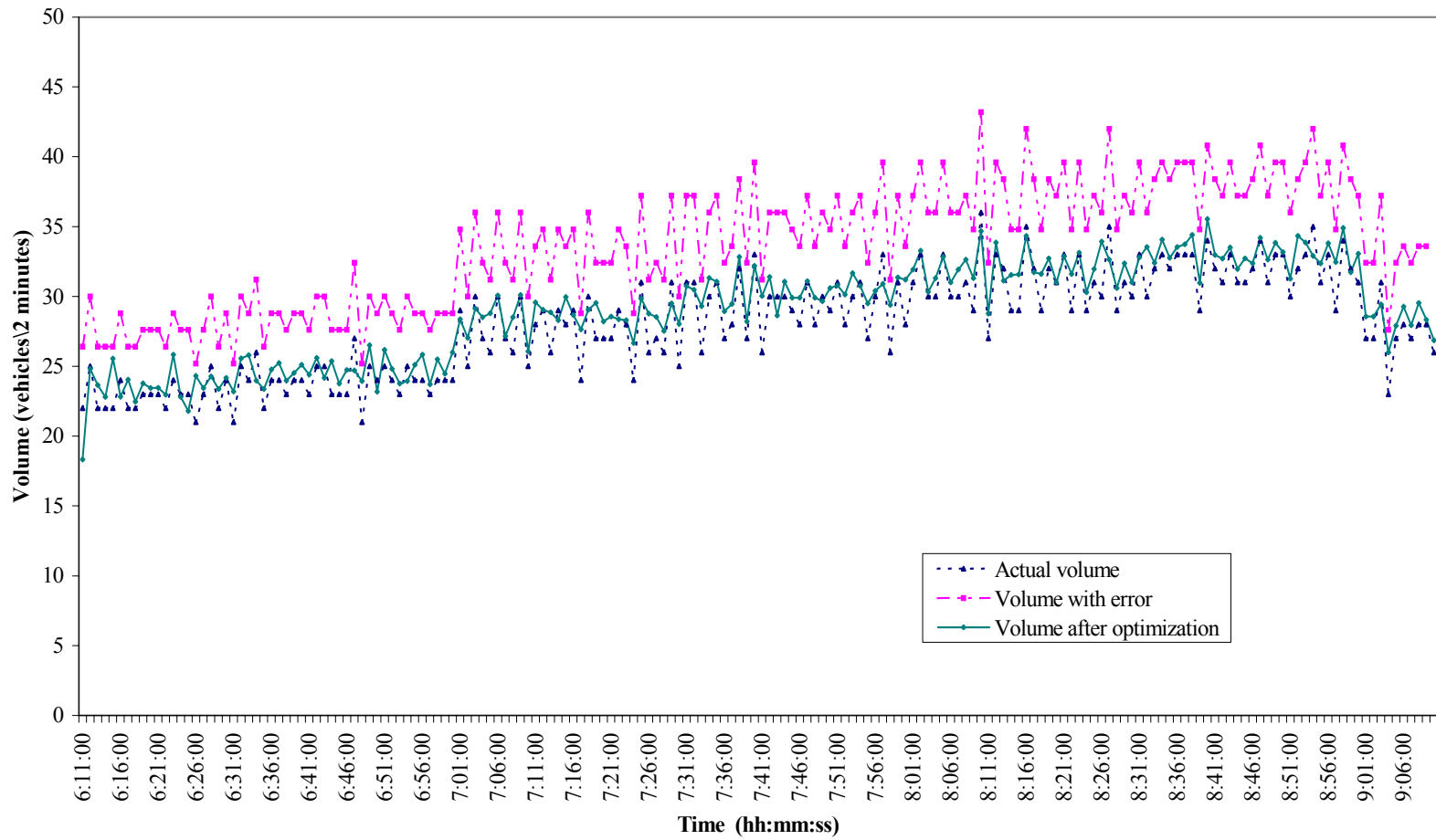


Fig. 4.38 Comparison of the performance of optimization at Location 2

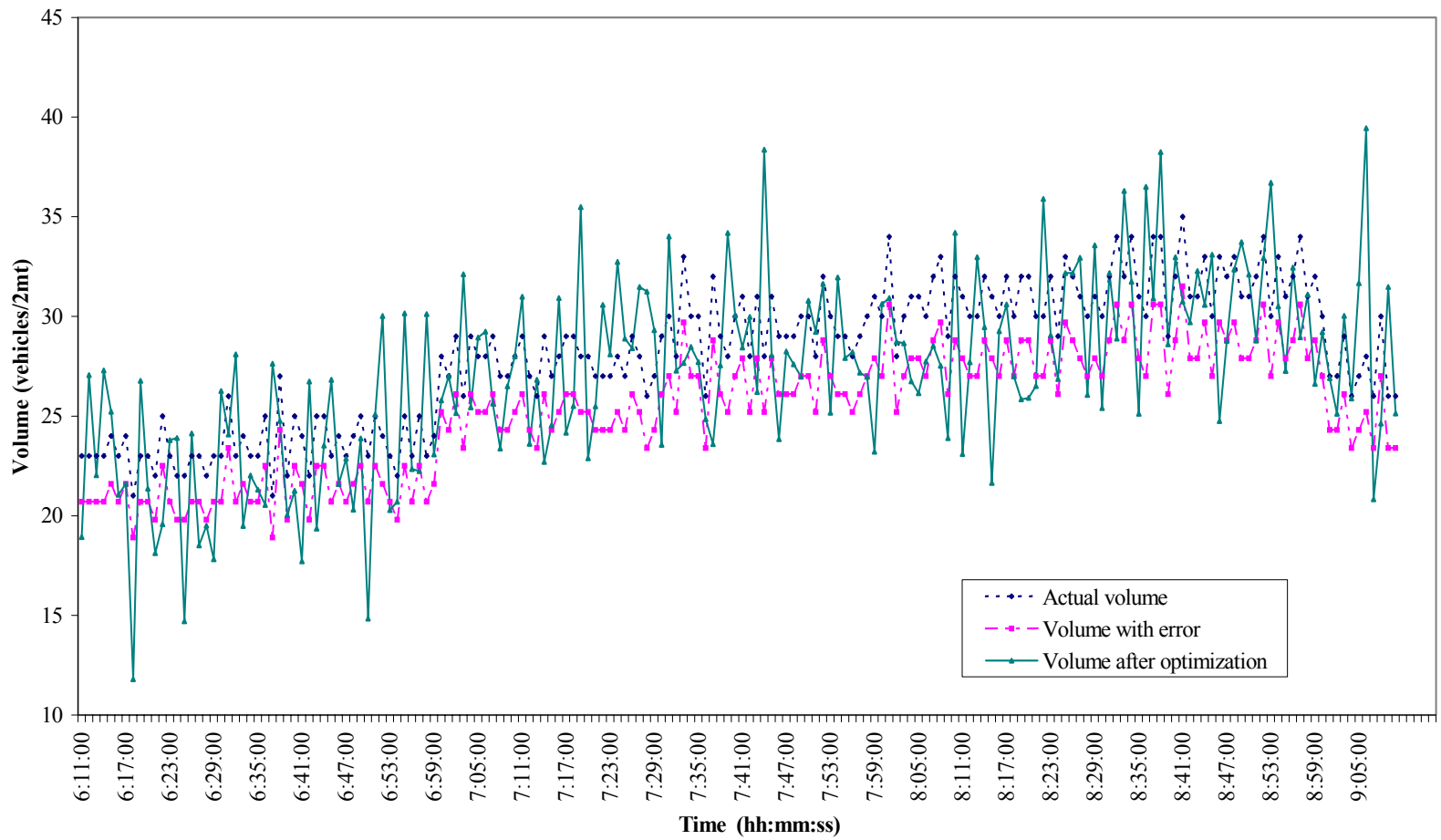


Fig. 4.39 Comparison of the performance of optimization at Location 1

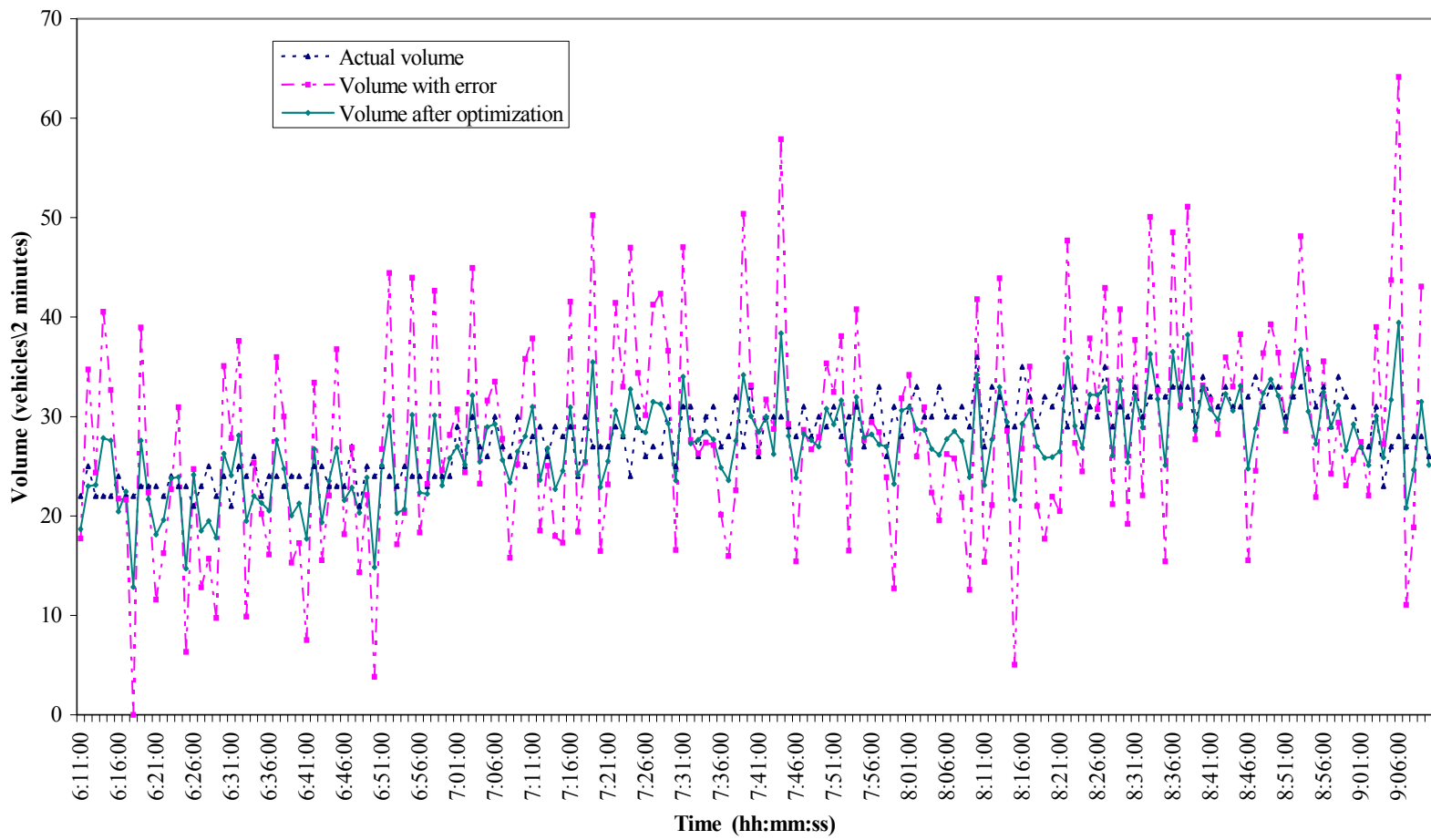


Fig. 4.40 Comparison of the performance of optimization at Location 2

It can be seen that with this combination of errors, the optimized values at location 1 have more variation than it had before the optimization. This was due to the fact that it has to take into account the wide variation at location 2 also into account as shown in Figure 4.40. The MAPE at the first location increased slightly from 10% to 12%, while the second location MAPE was reduced from 29% to 12%.

For the optimization of the field data, a comparison was carried out between the optimized volumes and the corresponding field values. Even though it is known that the field data have discrepancies, this was carried out to check how much the integrity of the original data is maintained after the optimization.

The results are tabulated in Table 4.4 for all the selected locations and all the selected days for the whole 24-hour period. The small magnitude of the MAPE values around 10% as shown in Table 4.4 (a value of 40% or more is considered to be large in practical applications (ezforecaster 2003)) show that the GRG method is able to perform the optimization without changing the original data's integrity. Thus, it can be observed that the optimization procedure follows all the requirements, namely, the data follow conservation of vehicles at all points at all time, handle a large amount of data, and preserve the integrity of the observed data as much as possible.

Table 4.4 Performance Measure at Each Site

Date	Location 1 MAPE (%)	Location 2 MAPE (%)	Location 3 MAPE (%)	Location 4 MAPE (%)	Location 5 MAPE (%)
February 10, 2003	7.56	6.19	5.88	8.17	14.86
February 11, 2003	8.43	6.63	7.06	8.71	15.76
February 12, 2003	7.67	6.32	6.73	8.06	10.19
February 13, 2003	8.32	6.64	6.43	8.65	11.30
February 14, 2003	5.99	5.56	5.85	8.01	14.56

4.7 OTHER APPLICATIONS

In addition to removing the discrepancy in the available data, the proposed optimization technique can also be used for imputing missing data if any of the detector locations under consideration miss recording data for some period of time. As discussed in Chapter II, missing data values (nonresponse) is a common occurrence with ITS data, and different imputation methods were reported. The detectors in general report data at 20- or 30-second intervals. However, sometimes the intervals get skipped and the data are reported at a larger interval, which can range from 1-minute to 10-minutes. Some of the reasons for this can be detector malfunctions, communication disruption, or software failure.

In this dissertation, the efficacy of the proposed optimization approach for imputation is illustrated in the following manner. Separate sample data sets with missing values were generated for locations 1, 2, 3, 4, and 5 of I-35 for February 10, 2003, by replacing the data with zeros for an interval of 15 minutes. The optimization program was run for these data with missing numbers. Based on the objective function and the constraints specified for the optimization, the missing numbers will be imputed depending on the values at the other locations for that time step, as well as the optimized number for the same location in the previous time step. Table 4.5 shows a sample set of data to illustrate the imputation for a 12-minute interval at location 2 on February 10, 2003. The values given are the time interval in the first column, followed by the field values obtained for each of the time intervals. The third column shows the optimized values corresponding to the field values. The fourth column shows the data introduced as zero to represent the missing numbers, and the fifth column gives the corresponding imputation results. The MAPE between the actual and the imputed values is calculated and is shown in the last column. It can be seen that the optimization procedure was able to impute the missing data reasonably well.

Table 4.5 Imputation of Missing Data Using GRG

Time	Actual	Optimized	Missing	Imputed	MAPE %
0:30	29	29.93	0	20.26	30
0:32	28	29.18	0	19.98	28
0:34	23	23.93	0	16.19	29
0:36	21	22.74	0	15.59	25
0:38	26	24.99	0	16.46	36
0:40	21	26.93	0	20.09	4.4
0:42	22	23.73	0	16.33	25

Figures 4.41 to 4.45 show the results obtained for missing data at locations 1, 2, 3, 4, and 5 respectively, after optimization for the five-detector series. For comparison, the result from optimization for the corresponding data without any missing values also is plotted in the corresponding figures. The original field data and the data with the missing numbers are also plotted in each of the graphs. It can be seen that with the missing values, the optimization retained the trend in the data and the numbers are able to follow the original optimized results. The MAPE values were calculated by comparing the optimization results with missing values with the corresponding actual values for the missing period of 12 minutes with 8 observations. The MAPE values obtained are also shown in the corresponding figures.

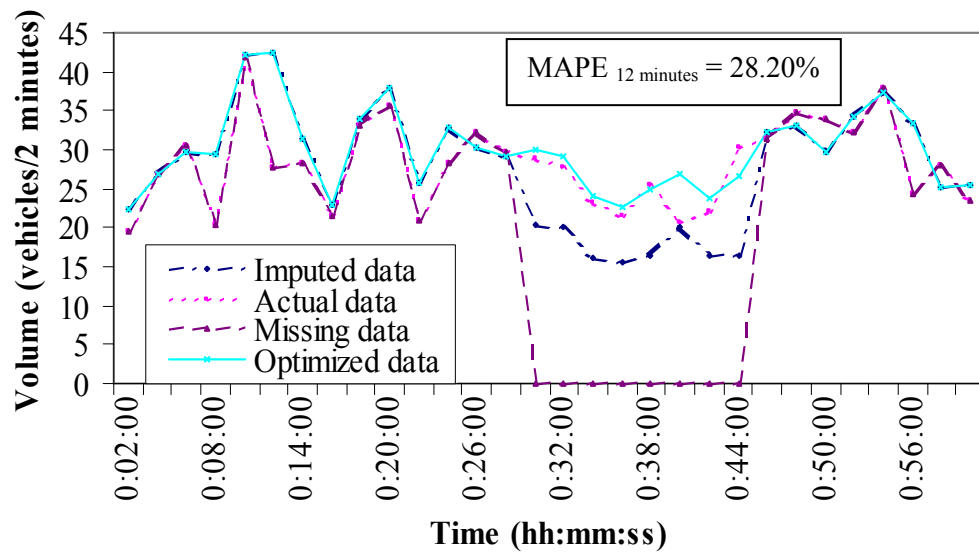


Fig. 4.41 Imputation results with missing data in location 1 on February 10, 2003

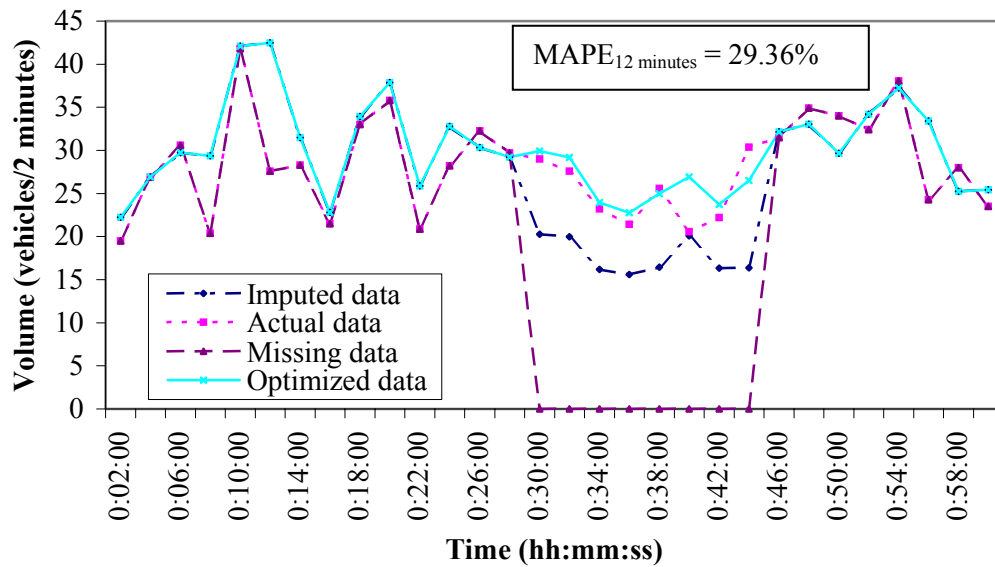


Fig. 4.42 Imputation results with missing data in location 2 on February 10, 2003

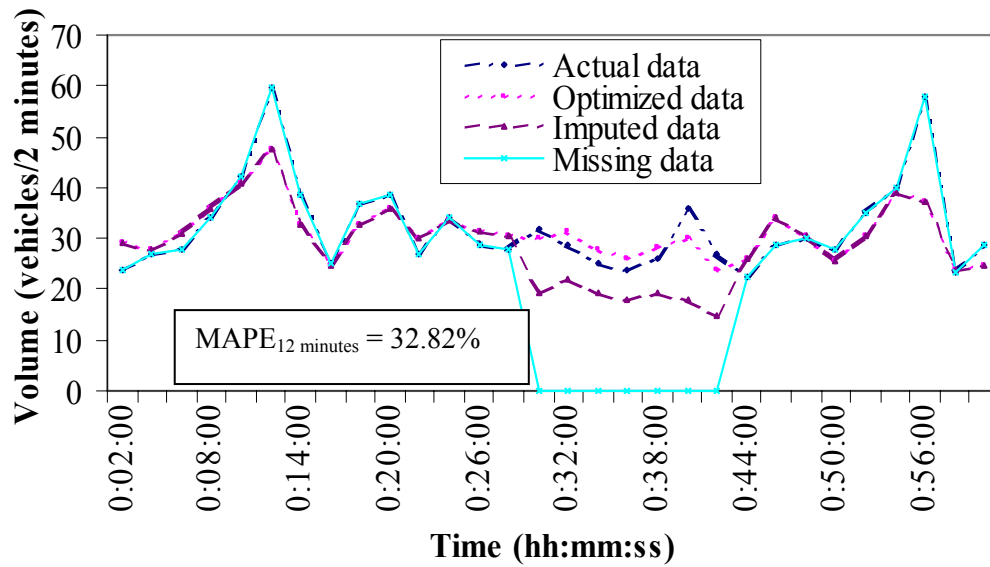


Fig. 4.43 Imputation results with missing data in location 3 on February 10, 2003

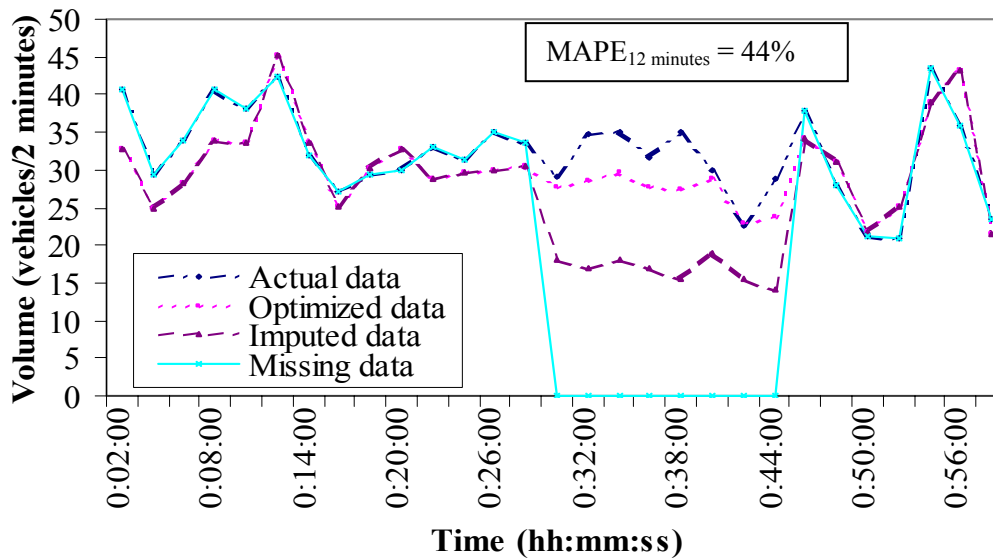


Fig. 4.44 Imputation results with missing data in location 4 on February 10, 2003

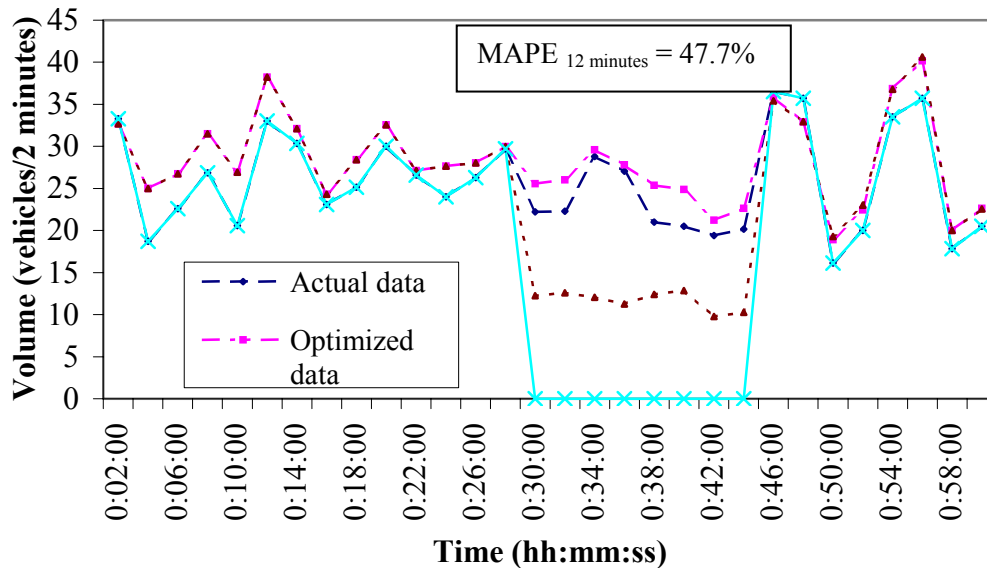


Fig. 4.45 Imputation results with missing data in location 5 on February 10, 2003

Also, this method can be used for finding the worst-performing detector stations based on the amount of error at each location. This information can be used for prioritizing the detectors for maintenance. This can be accomplished by comparing the MAPE values. For example, it can be seen in Table 4.4 that the MAPE for location 5 is higher than all the other locations, which is an indication that the detectors at location 5 is performing poorer than those at all the other locations. This will be useful to decide that the detectors at location 5 need priority in maintenance. However, with the present method of analysis, where all the detectors at a location are added together and assumed as a single detector, it will not be possible to identify which specific detector in that location is malfunctioning. To identify the specific malfunctioning detector within the identified location, the analysis should be carried out at a lane-by-lane level. The issues related to this kind of analysis of the detectors for each lane separately is discussed in detail in Chapter III.

4.8 ALTERNATIVE OBJECTIVE FUNCTIONS AND CONSTRAINTS

As discussed already, the constraints in the present optimization are selected based on restrictions, such as the cumulative flow at each detector location should be greater than or equal to the cumulative flow at the succeeding detector at all times. Another constraint is that the maximum difference between the cumulative flows should not exceed the maximum number of vehicles that can be accommodated in that road length at jam density conditions. These constraints are based on the worst and best scenarios. However, one could also choose different constraints which will change the computation time and the accuracy of the results.

The objective function used in the present strategy can also be modified to make the optimization procedure to incorporate more features of traffic flow. For example, the use of a weighted objective function can be one of the possibilities. One way to carry out this will be to assign weight to the variables in the objective function based on the standard deviation or variance at each of the locations (Taylor et al. 1969). However, to take variance into account in the present study, one of the following assumptions must be made:

- a) Variance is the same for a small interval of time, say 10 minutes (constant temporal flow), and/or
- b) Variance is the same for the consecutive locations (constant spatial flow).

The first assumption will assume a constant variation in traffic flow over time, while the second one makes an assumption that the flow is uniform in nature. Also, there is a need to know the relationship between the variance in the data and the error due to malfunctioning of the ILD. However, no literature was found on the relationship between the variance of traffic flow and the accuracy of the data recorded by the detectors. Thus, another assumption about how much variance is due to error and how much is due to natural variation in the traffic flow needs to be made.

In case the assumption is made that more variance at one location means more error at that point, one can assign more weight to the cumulative flow at that point based on the weight calculation given below (Miller and Miller 1993).

$$w_i = \frac{s_i^{-2}}{\sum s_i^{-2}}, \quad (4.39)$$

where,

s_i^2 = variance, and

n = number of observations.

Then, this weight can be assigned to the variables in the original objective function of the optimization given in equation 4.26 based on the assumptions made. For example, if the assumption is that variance should be same for the interval of time under consideration, the objective function can be:

$$\min \left[(Q_t^{(1)} w^{(1)} - Q_t^{(2)} w^{(2)})^2 + (Q_t^{(2)} w^{(2)} - Q_t^{(3)} w^{(3)})^2 \right], \quad (4.40)$$

where,

$w^{(1)}$ = weight based on the variance at location 1 as given in Equation 4.39.

If the assumption is that the variance is same for the consecutive locations, the objective function can be:

$$\min \left[(Q_t^{(1)} - Q_t^{(2)})^2 w^{1-2} + (Q_t^{(2)} - Q_t^{(3)})^2 w^{2-3} \right], \quad (4.41)$$

where,

w^{l-2} = weight based on the difference in variance between locations 1 and 2.

However, the accuracy of the above assumptions are not clear. As discussed already, there is a need to find out the relation between error in the detector data and variance of the data. This will

give an idea of how much of variance is due to error and how much is due to natural variation in the traffic flow. For example, Smith et al. (2003) argued that reducing the natural variance in the traffic data is an undesirable approach. Thus, the only reasonable inference one can make based on the variance at consecutive locations is that if there is a large change in variance at one location compared to the neighboring locations that may indicate a malfunction of the detector at that location. In the present study, the variance in the data at the consecutive locations was compared. Figure 4.46 show the plot of the variance in the data obtained from the consecutive locations. It can be seen that the variances did not have much variation between the locations. Hence, minimization of variance is not taken in the objective function or constraints in the present study. However, if it is known that the variation at a location is due to error in data collection, incorporating the variance in the objective function may lead to better result.

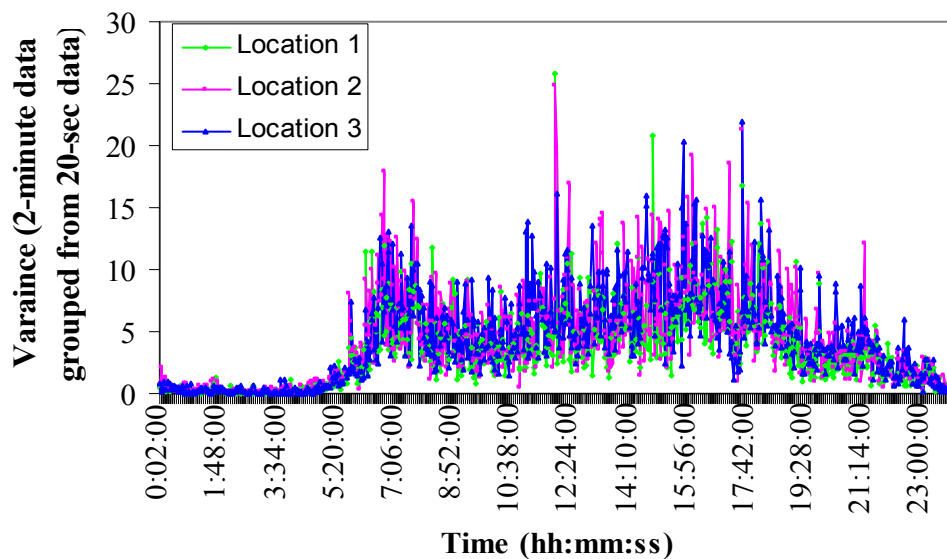


Fig. 4.46 Variance from three consecutive locations on February 11, 2003

4.9 CONCLUDING REMARKS

In this chapter, the loop detector data initially screened and corrected for common discrepancies were considered for further analysis. The data were analyzed as a series, rather than as individual

locations, and it was found that the conservation of vehicles principle was violated in one of the two following ways: in the first case there were a larger number of vehicles exiting than entering the test section, while in the second, the cumulative volume entering became unreasonably higher than the cumulative volume exiting. The cumulative volume curves of the data after the usual error corrections clearly showed that this approach of observing the detectors as a series could identify discrepancies that were unidentified with the commonly adopted error-checking procedures at individual locations. An optimization algorithm to adjust the volume data so that they will satisfy the conservation of vehicles was proposed. The objective of the method was to minimize the difference between the entry-exit observations using a GRG optimization. The data obtained after the optimization were consistent with the conservation of vehicles without violating any constraints. This method of correcting the loop detector data is more useful and convenient than the application of volume adjustment factors when dealing with large amounts of data for a longer duration and having large discrepancies. Also, the optimization technique proved to be very useful for imputing missing data as well as for prioritizing the detector stations for maintenance. This dissertation represents the first application of this kind of an optimization technique for quality control of the ILD data. The optimized data will be used in the estimation of travel time and will be discussed in the next chapter. The influence of this optimization on the final estimated travel time will also be discussed in that chapter.

CHAPTER V

ESTIMATION OF TRAVEL TIME

5.1 INTRODUCTION

Travel time, or the time required to traverse a roadway between any two points of interest, is a fundamental measure in transportation. Engineers and planners have used travel time and delay studies since 1920s to evaluate transportation facilities and plan improvements (Travel time data collection handbook 1998). In recent times with the increasing interest in Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS), providing travelers with accurate and timely travel time information has gained paramount importance.

Travel time can be measured directly using probe vehicles/test vehicles, license plate matching, electronic distance measuring instruments, Automatic Vehicle Identification (AVI), Automatic Vehicle Location (AVL), and video imaging, or it can be estimated from indirect sources like Inductance Loop Detectors (ILD), weigh-in-motion stations, or aerial video. While techniques like AVI and probe vehicles have less error, they are more expensive and often require new types of sensors as well as public participation; hence they are not widely deployed in urban areas (Turner 1996). Other methods, such as the test vehicle method, are time consuming, labor intensive, and expensive for collecting large amounts of data. On the other hand, most of the metropolitan areas in North America have their freeway network instrumented with ILD, which makes them the best source of traffic data over a wide area for a long period of time. Hence, at present ILDs are the most cost effective and popular way of obtaining travel time information for ATIS applications.

As discussed in earlier chapters, ILDs can be either single-loop or dual-loop. The data supplied by single-loop detectors include volume and occupancy. An algorithm is then used for estimating the speed using inputs such as effective loop length, average vehicle length, time over the detector, and the number of vehicles counted (Klein 2001). In the case of dual-loop detectors, the speed value will be automatically calculated based on the known distance between the two loops and the time a vehicle takes to cross the two loops. However, neither of these ILDs can collect travel time data directly, and so travel time has to be estimated from the available ILD data such

as flow, speed, or occupancy. Also, the data obtained from the ILDs are not for individual vehicles, but an aggregated value for all the vehicles traveling in the interval in which the data is reported. Thus, the travel time estimation should be based on the aggregate/average values reported by the ILDs for the small aggregation intervals, usually 20 or 30 seconds.

Accurate estimation of travel time from loop detector data is a difficult task due to the fact that the detector data is a point measurement, whereas travel time is a dynamic parameter averaged over distance. Thus, the travel time estimated based on spot speeds tends to underestimate section travel times due to the failure to capture traffic congestion occurring between the detector stations. For example, the most popular method adopted in the field today for the estimation of travel time from ILD data is based on the extrapolation of the point speed values. However, it is known that the accuracy of speed-based methods declines as the flow becomes larger because this method cannot take into account the variation in flow between the two measurement points. Oh et al. (2003) reported that the travel time estimated from single or dual-loop detector speed values would be correct only under the assumption that the traffic condition in the section is either homogenous or a linear combination of the two points. However, this assumption is not valid under congested traffic conditions. Thus, the travel time estimated tends to be biased under congested traffic conditions. Other estimation methods include statistical and traffic flow theory based models, the majority of which are developed for either the free-flow condition or the congested-flow condition (Nam and Drew 1996, 1998; Hoogendoorn 2000; Oh et al. 2003). Thus, most of these models were not developed taking into account the varying traffic flow conditions during the transition period from peak to off-peak or off-peak to peak conditions. Some attempts have been made in the past to estimate travel time using re-identification of vehicles at the second location (Coifman 1998; Coifman and Cassidy 2002; Sun et al. 1998, 1999). However, these methods require the use of sophisticated equipments and/or programs, which are not typically available to most traffic management centers.

The present study proposes a travel time estimation procedure using ILD data. The methodology proposed is based on a theoretical model suggested by Nam and Drew (1999) for the estimation of travel time from ILD flow data. Several modifications to this theoretical model are proposed in this dissertation. The details of the model by Nam and Drew (1999) are discussed in the next section followed by the proposed changes in the model. In the results section, a comparison is

carried out between the results obtained from the Nam and Drew model and the proposed model. Also the travel time estimated using the proposed method is compared with the results from the extrapolation method as well as with the direct travel time measured using AVI. In order to perform a more comprehensive analysis, the modifications are validated using simulated data from CORSIM simulation software.

5.2 TRAFFIC DYNAMICS MODEL

The traffic dynamics model (which will be called the ‘N-D model’ henceforth) for estimating freeway travel time from ILD flow measurements suggested by Nam and Drew (1995, 1996, 1998, 1999) is based on the characteristics of the stochastic vehicle counting process and the principle of conservation of vehicles. An inductive modeling approach was adopted in their study along with geometric interpretations of cumulative arrival-departure diagrams. The link travel time was calculated as the area between the cumulative volume curves from loop detectors at either end of the link. Instead of the usual approach of generalizing point measurements over a link, this work showed a judicious application of traffic flow theory to yield better travel time estimates from point data. Exponential averaging was used to increase the stability of the time series estimation of travel time.

The method can be explained using a one-lane road with two detectors located at each end, as shown in Figure 5.1. The number of vehicle arrivals and departures are measured continuously at the upstream location x_1 and downstream location x_2 .

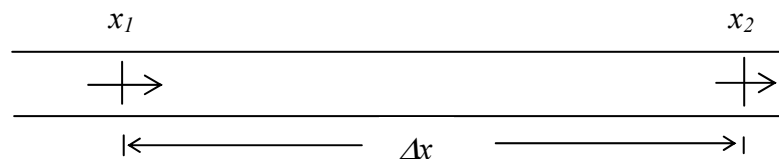


Fig. 5.1 Illustration of the conservation of vehicles

Referring to Figure 5.1, let $q(x_1, t)$ denote the flow per unit time measured at location x_1 at time t , and let $q(x_2, t)$ denote the flow measured at location x_2 at the same time t . The flows are regularly aggregated at Δt intervals for each detector. Thus the total number of vehicles entering and exiting the link during Δt respectively are

$$q(x_1, t + \Delta t)\Delta t, q(x_2, t + \Delta t)\Delta t. \quad (5.1)$$

Under the principle of conservation of vehicles, the difference between the above two quantities equals the change in the density, $k(t)$, over the link distance Δx . The equation of conservation of vehicles then becomes

$$\left[q(x_1, t + \Delta t) - q(x_2, t + \Delta t) \right] \Delta t = [k(t + \Delta t) - k(t)] \Delta x. \quad (5.2)$$

Rearranging the terms in the above equation, the conservation equation was written by Nam and Drew in the following form:

$$\frac{q(x_1, t) - q(x_2, t)}{\Delta x} + \frac{q(x_1, t + \Delta t) - q(x_1, t)}{\Delta x} - \frac{q(x_2, t + \Delta t) - q(x_2, t)}{\Delta x} = \frac{k(t + \Delta t) - k(t)}{\Delta t}. \quad (5.3)$$

Let $Q(x_1, t_n)$ and $Q(x_2, t_n)$ be the cumulative number of vehicles entering and exiting the link respectively, which can be expressed as

$$Q(x_1, t_n) = \Delta t \sum_{i=1}^n q(x_1, t_i), \text{ and} \quad (5.4)$$

$$Q(x_2, t_n) = \Delta t \sum_{i=1}^n q(x_2, t_i). \quad (5.5)$$

The initial conditions were

$$Q(x_1, t_0) = 0, \quad Q(x_2, t_0) = -n(t_0) \leq 0, \quad (5.6)$$

where,

$n(t_0)$ = number of vehicles traveling on the link at time t_0 .

The relationship between the link distance Δx and the data aggregation interval Δt is maintained as

$$\frac{\Delta x}{v_f} < \Delta t \leq 5 \text{ min} , \quad (5.7)$$

where, v_f = the free-flow speed on the link.

According to the characteristics of the stochastic vehicle counting process, the variables $Q(x_1, t_n)$ and $Q(x_2, t_n)$ are nonnegative and nondecreasing, and this leads to equation 5.8 and 5.9.

$$Q(x_1, t_n) - Q(x_1, t_{n-1}) = q(x_1, t_n) \Delta t \geq 0, \quad (5.8)$$

$$Q(x_2, t_n) - Q(x_2, t_{n-1}) = q(x_2, t_n) \Delta t \geq 0. \quad (5.9)$$

Also, the cumulative number of vehicles leaving downstream cannot exceed those arriving at upstream (based on the conservation of vehicles principle). Therefore,

$$Q(x_1, t_n) \geq Q(x_2, t_n). \quad (5.10)$$

The equality condition in equation 5.10 holds when there are no arrivals and subsequently the link is empty for the time interval Δt .

Let $n(t)$ be the number of vehicles traveling over the link distance Δx between the detector stations x_1 and x_2 at time t_n and is given as

$$n(t_n) = Q(x_1, t_n) - Q(x_2, t_n). \quad (5.11)$$

Then, the density at time t_n , $k(t_n)$ is calculated as follows:

$$k(t_n) = \frac{n(t_n)}{\Delta x} = \frac{Q(x_1, t_n) - Q(x_2, t_n)}{\Delta x}. \quad (5.12)$$

The N-D study developed two separate models: one for normal-flow conditions and the other for congested-flow conditions. The distinction between normal and congested-flow was made based on the number of vehicles entering and exiting during the specific time interval. This variable $m(t_n)$ was defined as the number of vehicles that enter the link during the interval t_{n-1} to t_n and that exit the link during the same interval. Under the first-in first-out condition $m(t_n)$ is given as

$$m(t_n) = Q(x_2, t_n) - Q(x_1, t_{n-1}). \quad (5.13)$$

The variable $m(t_n)$ was considered as a dynamic link performance measure, and different equations for estimating the travel time were suggested depending on whether $m(t_n)$ is positive (normal-flow) or equal to or less than zero (congested-flow).

5.2.1 Case 1. Normal-flow Condition

Nam and Drew assumed that the traffic characteristics of vehicles traveling under normal-flow conditions are represented by the vehicles that enter the link during the interval t_{n-1} to t_n and that exit the link during the same interval. The total travel time of these vehicles is schematically shown in Figure 5.2 as the hatched area.

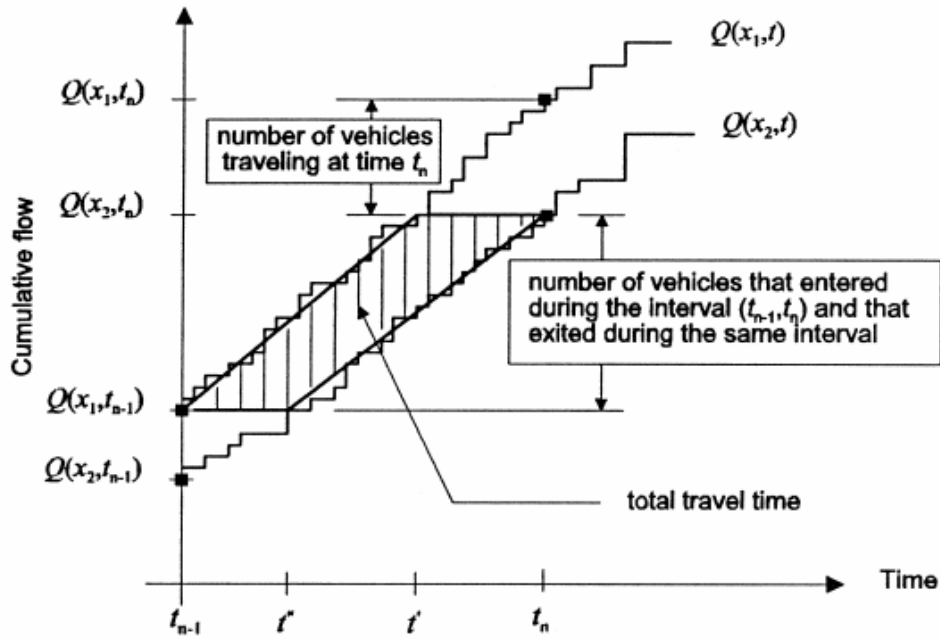


Fig. 5.2 Schematic representation of the total travel time during the interval (t_{n-1}, t_n) under normal-flows

(Source: Nam and Drew 1999)

Thus, analytically the total travel time of all the vehicles that entered and exited the link in that time period is equal to the shaded area and can be calculated as

$$T(t_n) = \frac{1}{2} [(t'' - t_{n-1}) + (t_n - t')] m(t_n), \quad (5.14)$$

where,

t' = time of entry into the link of the last vehicle that exits the link during the interval and

t'' = time of departure from the link of the first vehicle that enters the link during the interval.

After interpolating the values of t' and t'' , and substituting them in Equation 5.14, the travel time $T(t_n)$ was calculated for the vehicles that enter and exit during the same interval ($m(t_n)$) and is given in Equation 5.15:

$$T(t_n) = \frac{\Delta x \left[q(x_i, t_n)k(t_{n-1}) + q(x_{i+1}, t_n)k(t_n) \right]}{2 \left[q(x_i, t_n)q(x_{i+1}, t_n) \right]}, \tag{5.15}$$

where,

- Δx = distance between the detector locations (meters),
- $q(x_i, t_n)$ = flow at location i from t_{n-1} to t_n (vehicles per second), and
- $k(t_n)$ = density in the link between location i and $i+1$ at time t_n (vehicles per meter).

5.2.2 Case 2. Congested-flow Condition

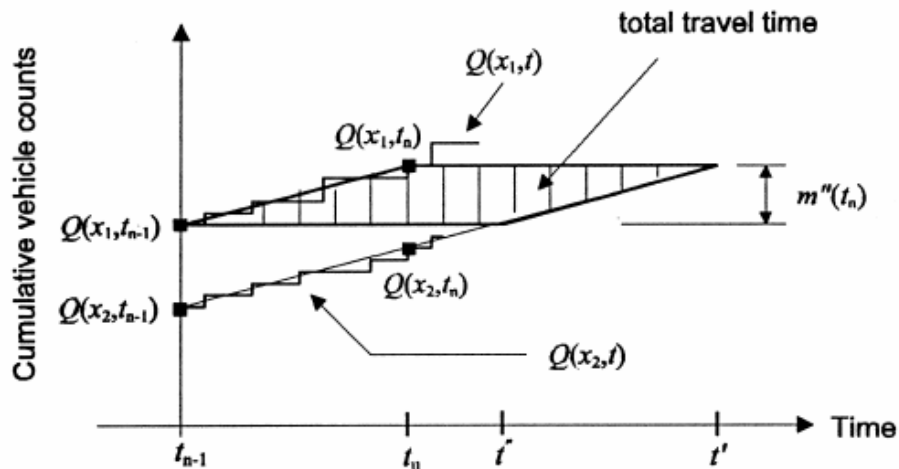


Fig. 5.3 Schematic representation of the total travel time during the interval (t_{n-1}, t_n) under congested-flows

(Source: Nam and Drew 1999)

Nam and Drew stipulated the traffic condition as congested when the value of the variable $m(t_n)$ is either zero or negative. Under such conditions, none of the vehicles that enter the link during the interval t_{n-1} to t_n exit the link during the same interval. Then, the travel time is calculated based on all the vehicles that enter during the interval under consideration, and the value corresponding to $m(t_n)$ for congested condition is calculated as

$$m''(t_n) = Q(x_1, t_n) - Q(x_1, t_{n-1}). \tag{5.16}$$

Thus, under the congested-flow conditions, the travel time is calculated as the shaded area in Figure 5.3, and this is equal to

$$\frac{1}{2} \left[(t'' - t_{n-1}) + (t' - t_n) \right] m''(t_n), \quad (5.17)$$

where,

t' = expected time of departure from the link of the last vehicle that enters the link during the interval (t_{n-1}, t_n) , and

t'' = expected time of departure from the link of the first vehicle that enters the link during the same time interval.

After interpolating the values of t' and t'' , and substituting them in Equation 5.17, the travel time $T(t_n)$ and is calculated as shown in Equation 5.18:

$$T(t_n) = \frac{\Delta x \left[k(t_{n-1}) + k(t_n) \right]}{2 q(x_{i+1}, t_n)}. \quad (5.18)$$

After the calculation of travel time, exponential averaging was applied to smooth the dynamic travel time estimates. This numerical technique favored the most recent estimate by assigning weight factors. Thus, the instantaneous travel time estimate at the next time interval (t_{n-1}, t_n) was calculated as

$$\bar{T}(t_n) = \bar{T}(t_{n-1}) + \alpha \left[T(t_n) - \bar{T}(t_{n-1}) \right] \leq T_f, \quad (5.19)$$

where,

α = exponential weighing factor $\frac{\Delta t}{T}$, Δt is the aggregation interval, and T is the

smoothing time interval,

$T(t_n)$ = instantaneous travel time estimates by time t_n ,

$\bar{T}(t_{n-1})$ = exponentially smoothed travel time estimates by time t_{n-1} , and

T_f = free-flow link travel time.

Nam and Drew (1999) validated the model using loop detector data during the morning peak hours (6:00 to 10:00 AM) on March 14, 1994, on a section of the Queen Elizabeth Way in Toronto, Canada. The ILD data were reported every 30 seconds and was accumulated to 2-minute intervals before analysis. Data from all the lanes for a given ILD were aggregated and were treated as single lane data. At the end of the 4-hour study period, the data showed a 3 percent difference in traffic counts, which indicated a violation of the conservation of vehicles principle. A volume adjustment factor was determined for each 30-minute period, and the measurements were adjusted to correct this discrepancy in the data. The models were then validated using the corrected data.

The main criticisms of the above model were the necessity to know the number of vehicles in the link at the start of data collection and the high sensitivity of the travel time estimates with respect to the errors in the measurements from the detectors (Son 1996; Petty et al. 1998; Dhulipala 2002; Oh et al. 2003). Another drawback of this method was related to the calculation of densities from cumulative flow measurements. Here, the accuracy of the estimation solely depends on the accuracy of the flow values. This would be an efficient method to find out the true density in a section, if the ILDs were working perfectly and an automatic initialization process could be employed frequently, on the order of every few minutes (May 1990). However, in reality the detectors may not be working perfectly. Many researchers, such as May (1990), have raised this concern pointing out that the calculated density from input-output counts getting frequently affected even with a low level of detector errors. Petty et al. (1998) expressed their concern in the following manner: "Loop detectors are notorious for over- and under-counting vehicles. Hence the cumulative flow lines that Drew and Nam were relying upon can systematically drift over time (indeed they might even cross)." Issues related to the use of cumulative flow curves for the calculation of density was raised by Oh et al. (2003): "A simple subtraction of cumulative arrivals at the two detectors would yield the number of vehicles in between the detectors and thus the density of the section between them. If such a density is known at any point in time, we can make a very good estimate of the true section travel time using simple fluid model relations for traffic. The reality however is different, in that the detectors in the field are not perfect and each detector has its own tendency to undercount vehicles and thus the cumulative arrival counts at two detectors (with their own "cumulative count drift") cannot be used to find the density at any time." Son (1996) pointed out another

drawback of this model as the calculation of travel time under normal-flow considering only the vehicles that entered and exited in the same time interval. To address all these concerns, a number of modifications are proposed in this dissertation and are detailed in the following section.

5.3 PROPOSED MODEL FOR TRAVEL TIME ESTIMATION

As discussed in the previous section, many previous studies have pointed out certain drawbacks in the N-D model, which will be addressed in this dissertation. Concerns related to the quality of the ILD flow data values and its effect on the calculated travel time is one important factor that has been raised by several researchers. For instance, there can be no violation of the conservation of vehicles principle as far as the N-D model is concerned. However, loop detectors are mechanical devices, and errors in their measurements are inevitable. Nam and Drew chose to use volume adjustment factors to correct the field data when there was a violation of the conservation of vehicles principle. This may be sufficient when the analysis is for small intervals of time over small stretches of roadway. However, when the analysis is for longer intervals of time, for instance, one full day, a systematic methodology to check and correct the data is required. In this dissertation, this concern was addressed by the use of an optimization procedure, the details of which were discussed in Chapter IV.

The necessity of knowing the number of vehicles in the link at the start of observation is another limitation pointed out in earlier criticisms of the N-D model. In this dissertation, 24-hour data starting at midnight 12 were analyzed. It was assumed that at midnight, zero vehicles are inside the link since the traffic flow is very low at that time. This assumption was made after analyzing the ILD data for different days.

As discussed previously, the N-D model consists of two separate models, one for normal-flow and the other for congested-flow. The classification of normal and congested-flow is based on whether any vehicle that entered at a specific time step was able to exit the link in the same time step. This classification does not consider the transition period between normal and congested-flow. For example, during the transition from normal to congested-flow, at some time steps the majority of the vehicles which enter the section may not exit in the same period due to the start

of congestion. However, because few vehicles out of the total vehicles that entered are able to exit in the same time step, the N-D model will consider the flow as normal-flow. Thus, the N-D model will calculate the travel time based on those few vehicles that entered and exited in the same period. It can be seen that the majority of vehicles that did not exit in the same time interval will be left out of the calculation of travel time. This will affect the calculated travel time value during the transition period. The fact that a portion of vehicles is not considered for calculating travel time under normal-flow conditions is another drawback of the N-D model (Son 1996). This is addressed in this dissertation by taking a weighted average of the travel time of vehicles based on normal-flow and congested-flow conditions at each time step during the transition period.

As detailed in the previous section, calculation of density from the cumulative flow was another criticism of the N-D model. In this dissertation, this drawback was taken into account by calculating density from the occupancy values reported by ILD. Also, previous literature showed that the travel time is almost independent of flow during very low flow conditions. This led to the final modification of calculating travel time from speed during very low traffic flow conditions. Each of the above mentioned modifications are explained in detail in the following sections.

5.3.1 Modification I. Optimization to Enforce Conservation of Vehicles Principle

The development of the N-D model was based on the equation of conservation of vehicles and the flow-density-speed relation. If the traffic flow follows the conservation of vehicles principle, the cumulative flow at the upstream point at any time should be greater than or equal to the cumulative flow at the downstream point. Also, the difference between the cumulative flow at the upstream and downstream locations cannot exceed the maximum number of vehicles that can be accommodated in the section between the two detectors. In the N-D model, the analysis was carried out for a 4-hour period, and the data showed a 3 percent error at the end of the study period. The authors suggested a volume adjustment factor to correct the error. This type of adjustment factor will be reasonable and practical for problems with a small number of observations. When applied to large amount of data over long periods of time, however, the difference between the observed and the adjusted values propagate and are very difficult to

control. For example, in the present study, the analysis was carried out for 24-hour time periods, and the magnitude of error was of the order of thousands of vehicles as shown in Table 4.1. In such cases, the use of an adjustment factor will not be sufficient to correct the systematic errors. Thus, when the analysis is for a long interval of time and/or for a long stretch of detectors, one needs a more systematic method of analysis and correction of data. In the present study a nonlinear optimization method, namely the Generalized Reduced Gradient method (GRG), is adopted to enforce the conservation of vehicles principle and has been discussed in detail in Chapter IV.

5.3.2 Modification II. Normal-flow Model

As discussed earlier, the study by Nam and Drew consisted of two separate models: one for normal-flow conditions and the other for congested-flow conditions. The distinction between normal and congested-flow was made based on the number of vehicles entering and exiting during a specific time interval, designated as $m(t_n)$ as in Equation 5.13. However, it can be seen that for calculating the travel time under normal-flow conditions, the N-D model considered only those vehicles that entered and exited the section in the same time period. However, in the case of congested-flow, travel time is averaged for all vehicles that entered in that time step. This is further explained using Figure 5.4. Under normal-flow conditions, such as between time step t_{n-1} and t_n in Figure 5.4, the N-D model considers only the area ACDE for the calculation of travel time. Thus, it can be seen that under normal-flow conditions, a portion of the vehicles will not be considered while calculating travel times (areas ABC and DEF). This becomes more serious as flow increases or decreases in the transition between off-peak to peak flow or peak to off-peak flow. At these times, a small percent of the total vehicles that entered may be exiting the section in the same interval whereas majority may be exiting in the next interval. For example, in Figure 5.4, during the time interval from t_n to t_{n+1} only the area FGHI will be considered, and the vehicles represented by areas EFG and HIJ will be ignored. Because a portion of the vehicles is able to exit in the same interval, $m(t_n)$ will be positive, and hence the chosen model would be the normal-flow model. Thus, the travel time is calculated based on the few vehicles that managed to exit in the same interval, ignoring the majority that did not exit in the same time interval. The above concept can be illustrated using the field data shown in Table 5.1. This specific data set pertains to link 1 of the I-35 section on February 11, 2003. In time interval 6, there were 71

vehicles inside the link, and only 13 of these were able to exit in the same time period. Because 13 of them were able to exit, the N-D model will consider it as a normal-flow condition and the travel time will be calculated based on those 13 vehicles. The information on the remaining 57 vehicles would be ignored.

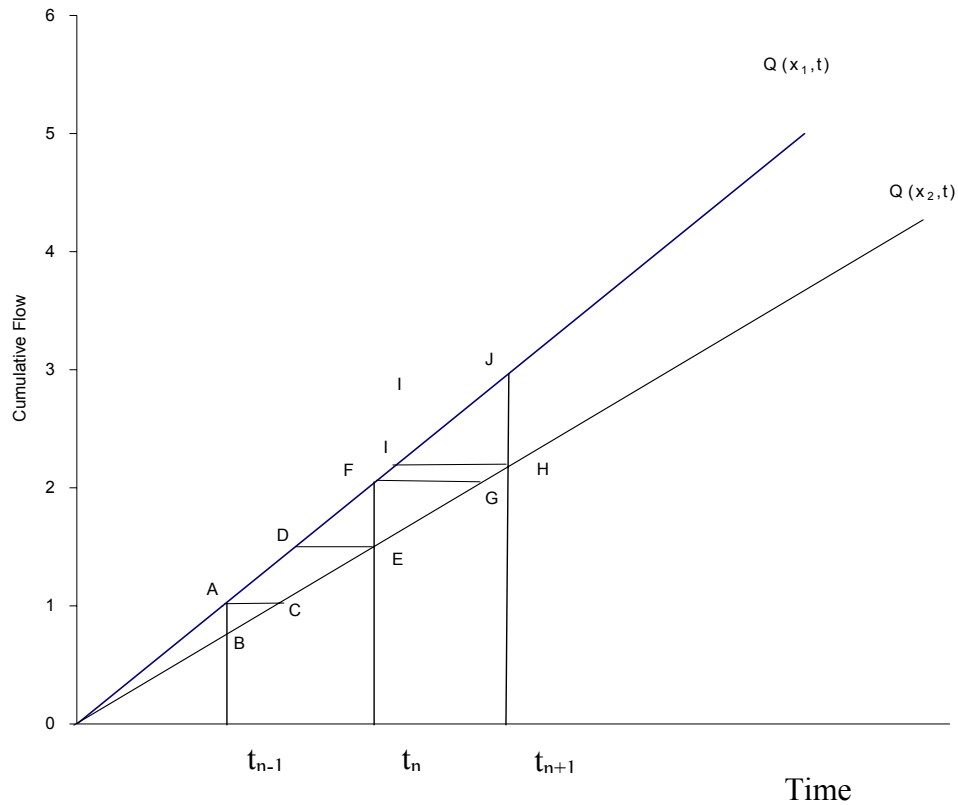


Fig. 5.4 Schematic diagram to illustrate the travel time calculation

Table 5.1 Data Set to Illustrate the Travel Time Calculation on February 11, 2003

Time interval	q_1	q_2	Q_1	Q_2	Number of vehicles in the link	m
1	135	131	53840	53767	73	61
2	121	124	53961	53891	70	51
3	104	110	54065	54001	64	40
4	125	123	54190	54124	66	59
5	104	104	54294	54228	66	38
6	84	79	54378	54307	71	13
7	100	101	54478	54408	70	30
8	99	97	54577	54505	72	27
9	112	120	54689	54625	64	48

The present study overcomes this disadvantage by modifying the normal-flow model. The modification is carried out by applying the normal-flow model to the vehicles that are able to enter and exit in the same interval, and the congested-flow model is applied to those vehicles that are not able to exit in the same interval. Thus, a weighted average of the travel time is calculated of the normal-flow vehicles and congested-flow vehicles traveling in the same time interval. This weighted average is based on the proportion of normal-flow vehicles and congested-flow vehicles within the same time step.

Let $m_p(t_n)$ be the ratio of normal-flow vehicles to the total inflow for that time period t_{n-1} to t_n which can be represented as:

$$m_p(t_n) = \frac{m(t_n)}{q(x_i, t_n)}. \quad (5.20)$$

Thus, the new equation for estimating travel time under normal-flow condition will be as shown in Equation 5.21.

$$T_i = m_p(t_n) \frac{\Delta x}{2} \left[\frac{q(x_1, t_n)k(t_{n-1}) + q(x_2, t_n)k(t_n)}{q(x_1, t_n)q(x_2, t_n)} \right] + (1 - m_p(t_n)) \frac{\Delta x}{2} \left[\frac{k(t_{n-1}) + k(t_n)}{q(x_2, t_n)} \right]. \quad (5.21)$$

All the variables in Equation 5.21 are the same as in Equations 5.15 and 5.18. It can be seen that this modification helps to model the transition flow in a more accurate way. For instance, when the flow condition is completely normal, the value of $m_p(t_n)$ will be 1, and hence the second term in the above equation will vanish. In the transition stage the second term will take into account the travel time of those vehicles that fall in the congested condition that were ignored in the N-D model.

This freeway travel time function given in equation 5.21 has two independent measures, $q(x_1, t_n)$ and $q(x_2, t_n)$. The relationship between the travel time and the flow rates can be found by differentiating this function with respect to the two flow variables. Re-writing equation 5.21 using Equation 5.3 the following equation is obtained:

$$T_i = \frac{m_p(t_n)}{2q_1q_2} [k(t_{n-1})\Delta x(q_1 + q_2) + \Delta t(q_1 - q_2)q_2] + \frac{1 - m_p(t_n)}{2q_2} [2k(t_{n-1})\Delta x + (q_1 - q_2)\Delta t]. \quad (5.22)$$

The final differential with respect to $q(x_1, t_n)$ is obtained as follows:

$$\frac{\partial T}{\partial q_1} = \frac{m_p(t_n)(q_2\Delta t - k(t_{n-1})\Delta x)}{2q_1^2} + \frac{(1 - m_p(t_n))\Delta t}{2q_2}, \quad (5.23)$$

where,

$$q_1 = q(x_1, t_n), \text{ and}$$

$$q_2 = q(x_2, t_n).$$

Due to the precondition of the normal-flow (Equation 5.13) that $Q(x_2, t_n) > Q(x_1, t_{n-1})$, the quantity $q_2 \Delta t$ will always be greater than $k(t_n - 1) \Delta x$. Also, both the numerator and denominator in the second term in Equation 5.23 are always positive, making Equation 5.23 always positive. This means that as the traffic demand given by $q(x_1, t_n)$ increases, the travel time also increases.

The final differential with respect to $q(x_2, t_n)$ is calculated and is shown in Equation 5.24.

$$\frac{\partial T}{\partial q_2} = - \left[\frac{m_p(t_n) [q_2^2 \Delta t + k(t_{n-1}) q_1 \Delta x]}{2q_1 q_2^2} \right] - \left[\frac{(1 - m_p(t_n)) [2k(t_{n-1}) \Delta x + q_1 \Delta t]}{2q_2^2} \right]. \quad (5.24)$$

It can be seen that both the numerator and the denominator for the first and second terms are positive, making Equation 5.24 always negative. This means that as the outflow quantity $q(x_2, t_n)$ increases, the travel time decreases. Thus, it can be seen that the new travel time function has a desirable relationship with both the flow variables under normal and congested conditions by increasing with increasing inflow and decreasing with increasing outflow.

As in the original model, exponential averaging was applied to smooth the dynamic travel time estimates. This smoothing gives stable estimates over time. An α value of 0.2 was adopted, thus smoothing the exponentially averaged estimates over the time interval $5 \Delta t$.

5.3.3 Modification III. Calculation of Density

The N-D model calculates the density from the cumulative flow values. Thus, the accuracy of the estimated travel time depends solely on the accuracy of the measured flow values. If the point detectors are working perfectly, this method is appropriate to calculate the true density in a section. However, in reality the detectors may not be working perfectly (Vanajakshi and Rilett 2004b; Turner et al. 2000; Chen and May 1987). Moreover, if there is a malfunction in the detectors, the flow data get more affected. This is because of the nature in which the detectors collect traffic data. The flow data from the detectors are reported as a *cumulative* number, whereas speed and occupancy are *averaged* data for the accumulation time interval (every 20- to

30-second interval). Hence the effect of a detector malfunction, like missing vehicles, will have less impact on speed and occupancy in comparison to flow data. In such cases, the calculation of density from the flow values may not yield the best results.

Even though in the present study the flow data are corrected for the discrepancy based on the conservation of vehicles constraint, there can still be more “unaccounted” errors in the data. Hence, in the present study the use of occupancy values for the calculation of density is suggested instead of using the flow values. The present study calculated the density from the ILD occupancy values using the following equation (May 1990).

$$k = 52.8 \times \frac{O}{(\overline{Lv} + L_d)}, \quad (5.25)$$

where,

k = density (vehicles per mile),

\overline{Lv} = average vehicle length (feet),

L_d = detection zone length (feet), and

O = percent occupancy.

Even though this method has the disadvantage of requiring an estimate of the average vehicle length, it was found that compared to the use of cumulative flow curves, this method gave more reasonable results. This particular fact will be illustrated in the subsequent results sections.

5.3.4 Modification IV. Use of Extrapolation Method for Low Volume Conditions

Many of the previous studies have reported that speed, and in turn travel time, is not dependent on the flow under low traffic flow conditions (Van Aerde and Yagar 1983; Persaud and Hurdle 1988; Sisiopiku et al 1994a; Faouzi and Lesort 1995; HCM 2000; Bovy and Thijs 2000; Coifman 2001). The Highway Capacity Manual (2000) remarks on this issue as follows: “All recent studies indicate that speed on freeways is insensitive to flow in the low to moderate range” and the low to moderate volume includes up to 1300 passenger cars per hour per lane (pcphpl) for a 70 mph freeway system. Sisiopiku et al. (1994a), in their study on the correlation

between travel time and detector data concluded that travel time is independent of both flow and occupancy under low traffic conditions.

Thus, the accuracy of the estimated travel time during low traffic conditions is questionable in the N-D model because the estimated travel time in that model is a function of the measured flow. This issue was not addressed in the original work because the data analyzed in their study was restricted to only the morning peak traffic flow. In this dissertation the analysis was carried out for continuous 24 hours, which included very low traffic flow conditions also.

On the other hand, as discussed earlier, methods based on speed values tend to have more bias in the resulting travel time during congested periods due to the failure to capture the variations occurring between the detector stations. However, under low traffic flow conditions they are more suitable than the methods based on flow. Hence, in the present study, the use of the extrapolation method is suggested for low flow conditions so that accuracy can be maintained consistently under all varying flow conditions. A cut-off value of 50 vehicles per 2-minute interval over all the three lanes added together is set for this data based on the HCM recommendation. Thus, when the flow is less than 50 vehicles per 2 minutes over the three lanes, the method based on speed values will be used, and the developed model based on flow values will be used otherwise. This can be algorithmically represented as follows:

if flow < 50 vehicles/2 minute/3 lane,
 then use extrapolation method (Equation 2.1, 2.2, or 2.3);
else
 use developed method (Equation 5.18 or 5.21). (5.26)

In summary, the major modifications to the N-D model in the application of travel time estimation can be summarized as follows:

1. The original N-D model is based on the premise that the loop detector data follow the conservation of vehicle principle at all times. However, in reality, the loop detector data collected from the field show serious violation of this constraint. The N-D model was illustrated using data for a short period of time (4 hours), and hence used adjustment

factors for correcting this discrepancy. In this dissertation, a more systematic method based on a nonlinear optimization by GRG method is used for correcting this discrepancy.

2. The relation for travel time estimation during normal-flow conditions is modified such that the travel time will be estimated based on all the vehicles entering in that time period, instead of considering only those vehicles which enter and exit in the same period, as in the N-D model.
3. The N-D model calculated density from the cumulative flow values. This was found to be a good method to calculate density, only if the quality of the flow data is assured. In cases where the ILD data have errors, the calculation of density from occupancy is a better choice and hence this method is adopted in this dissertation.
4. The use of an extrapolation method is suggested for very low traffic flow conditions so that accuracy can be maintained under varying traffic flow conditions.

5.4 RESULTS AND DISCUSSION

The results are illustrated using the data collected from link 1 and link 2 of the I-35 test bed shown in Figures 3.10. The ILD data from all the 5 days from February 10 to February 14, 2003 are used. The effects of each of the suggested modifications will be illustrated first using the ILD data. Next, the validation of the modified model will be carried out using AVI data collected from the same location as the ILD data. Validation will also be carried out using simulated data generated using CORSIM. Finally, results obtained from a comparative study of the performance of the proposed model with the extrapolation method using both field data and simulated data are shown.

5.4.1 Influence of the Modifications on Travel Time Estimation

Figures 5.5 and 5.6 show sample plots of travel time estimated for link 1 and 2 by N-D model using unmodified ILD data before the optimization is carried out.

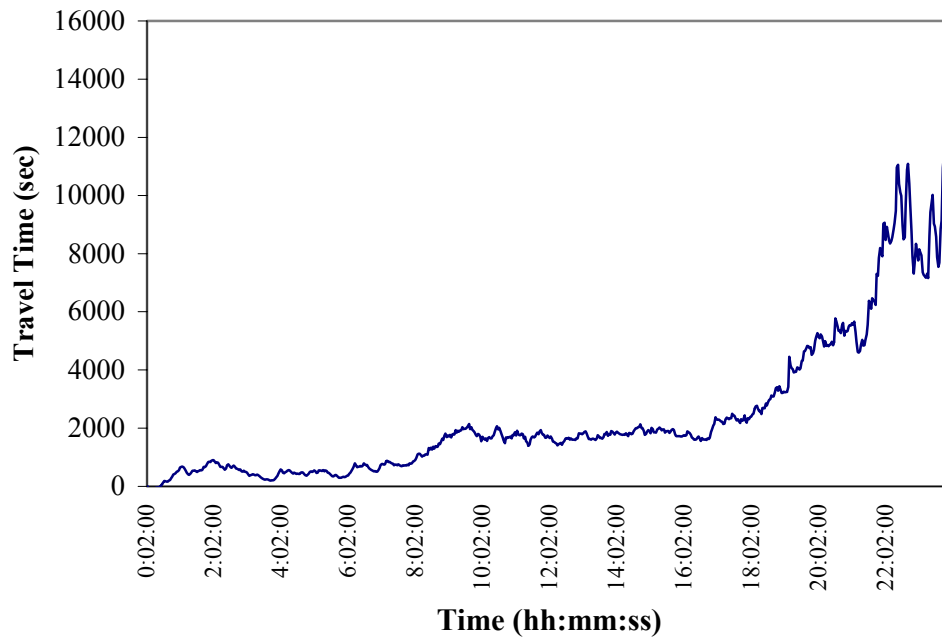


Fig. 5.5 Travel time estimated by the N-D model using actual data for link 1 on February 11, 2003, for 24 hours

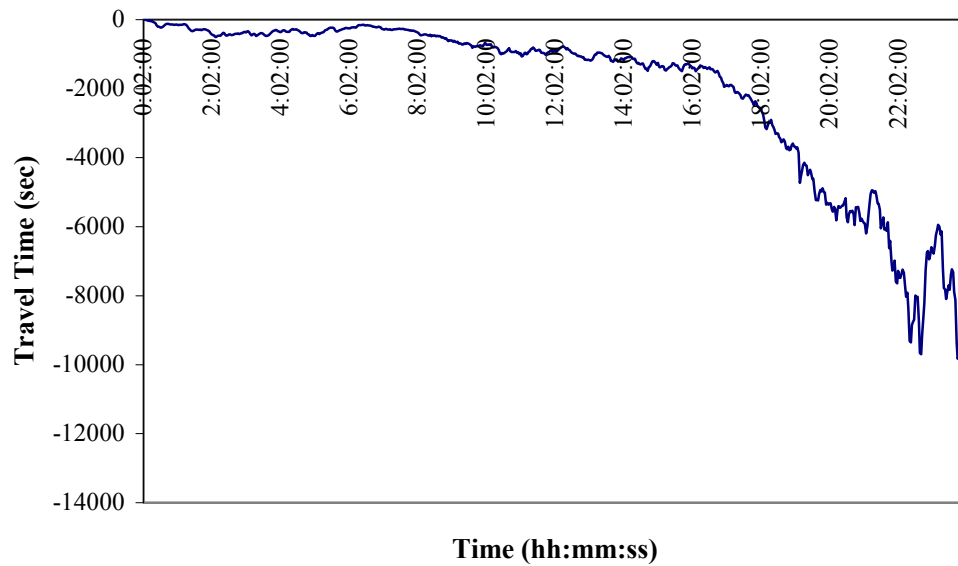


Fig. 5.6 Travel time estimated by the N-D model using actual data for link 2 on February 11, 2003, for 24 hours

It can be seen that the estimated travel time for link 1 varies from 0 to 15,000 seconds and that the travel time for link 2 came out to be negative for the whole 24 hours.

Figures 5.7 and 5.8 depict the same estimated travel time calculated using the data after the optimization using the GRG technique described in Chapter IV. It can be seen that the range of travel time has improved, even though the values are still unreasonably high. In the field data, the speed variation was from 5 mph to 80 mph, and the corresponding travel time can only vary from 22.5 seconds to 360 seconds for a 0.5-mile section. In Figures 5.7 and 5.8, it can be seen that the travel time estimated varies from 0 to 600 seconds, showing the need for further improvement.

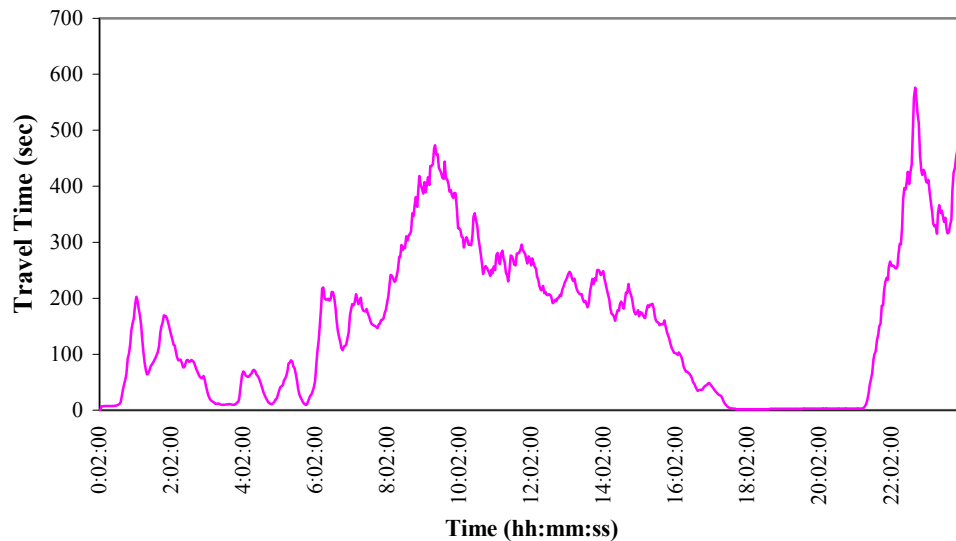


Fig. 5.7 Travel time estimated by the N-D model using optimized data for link 1 on February 11, 2003, for 24 hours

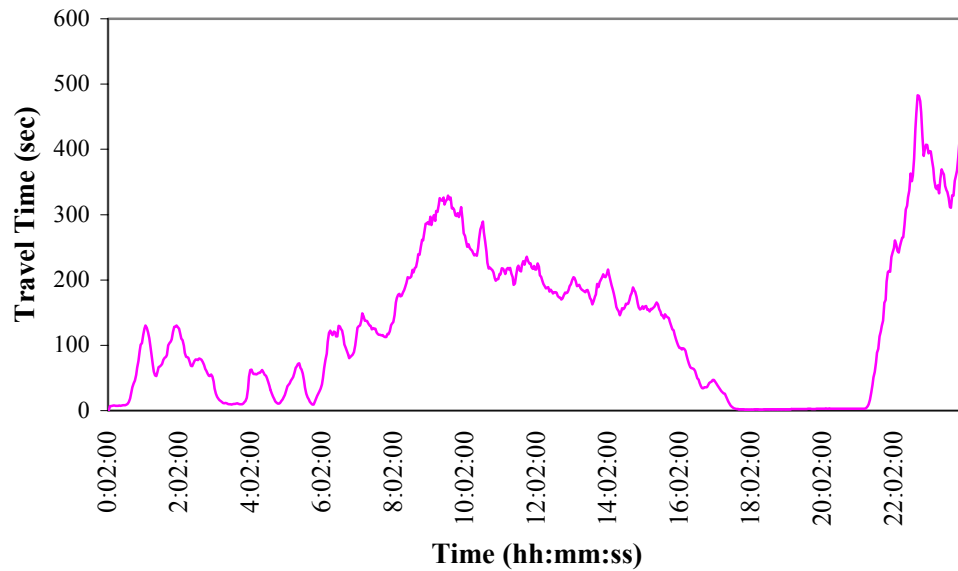


Fig. 5.8 Travel time estimated by the N-D model using optimized data for link 2 on February 11, 2003, for 24 hours

Modifications II and III were carried out next by replacing the normal-flow model of Nam and Drew by Equation 5.21 and by calculating the density from occupancy. The resulting graphs are shown in Figures 5.9 and 5.10. It can be seen that the results have improved and the estimated travel times are within reasonable limits of 22.5 and 360 seconds calculated earlier. However, it can be seen that there is a large fluctuation in the estimated travel time under very low flow conditions. The corresponding flow values from locations 1, 2 and 3 are shown in Figures 5.11, 5.12, and 5.13. It can be seen that the fluctuations in travel time happen when the volume is less than 50 vehicles per 2-minute interval.

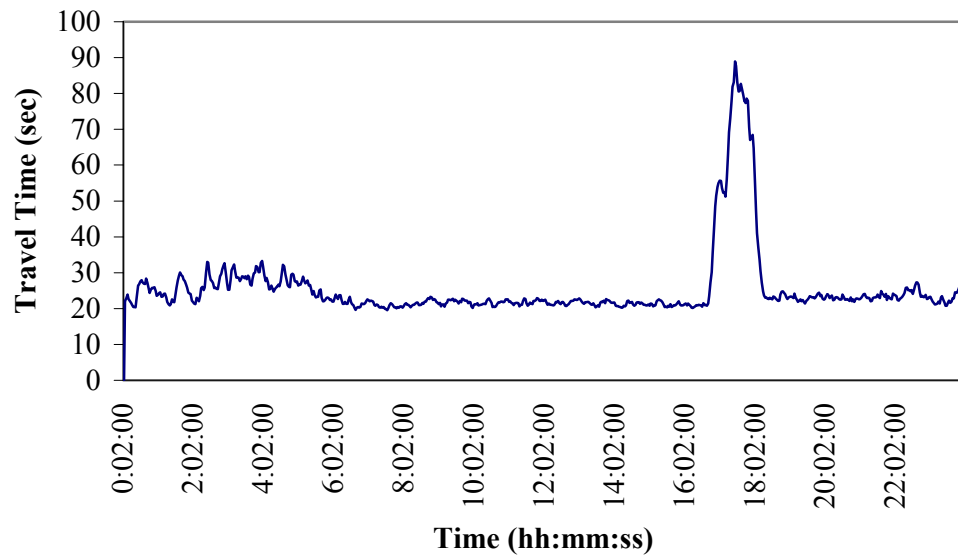


Fig. 5.9 Estimated travel time on link 1 with density calculated from occupancy values on February 11, 2003, for 24 hours

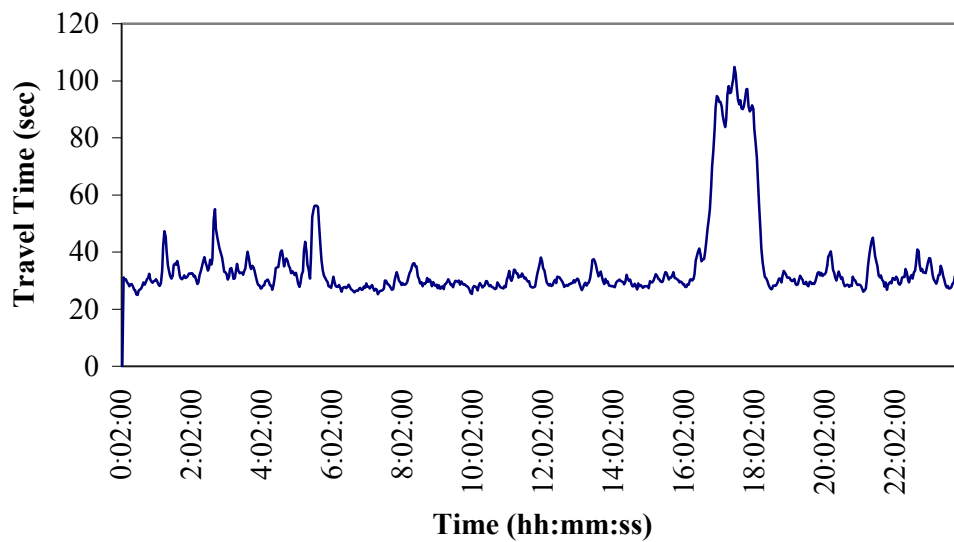


Fig. 5.10 Estimated travel time on link 2 with density calculated from occupancy values on February 11, 2003, for 24 hours

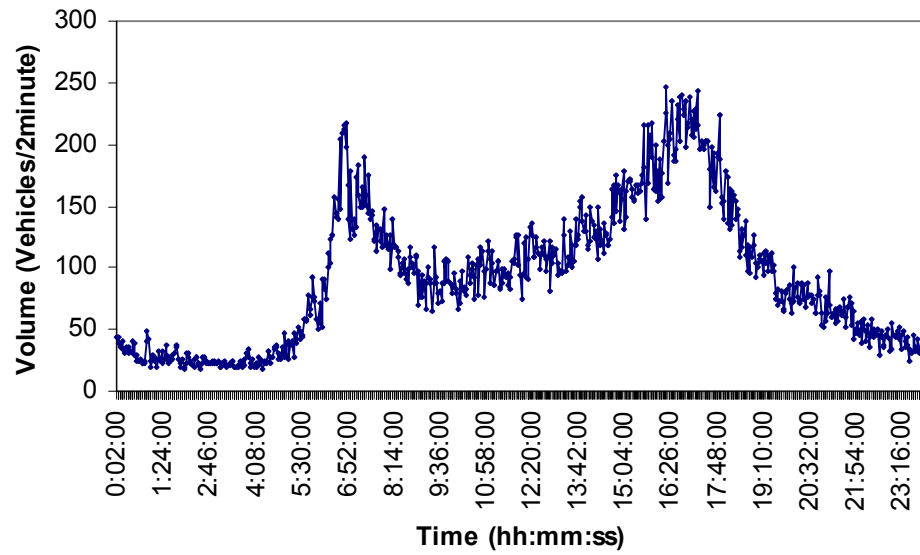


Fig. 5.11 Volume distribution on February 11, 2003, for 24 hours at location 1

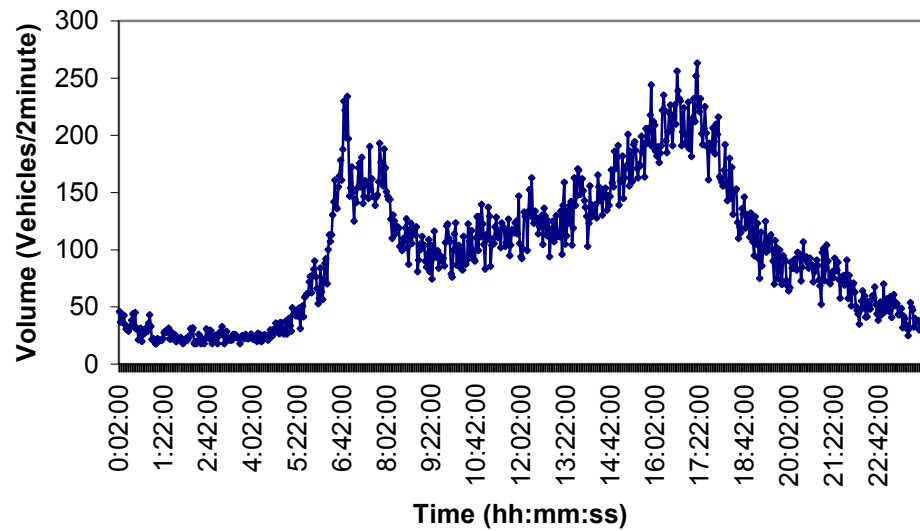


Fig. 5.12 Volume distribution on February 11, 2003, for 24 hours at location 2

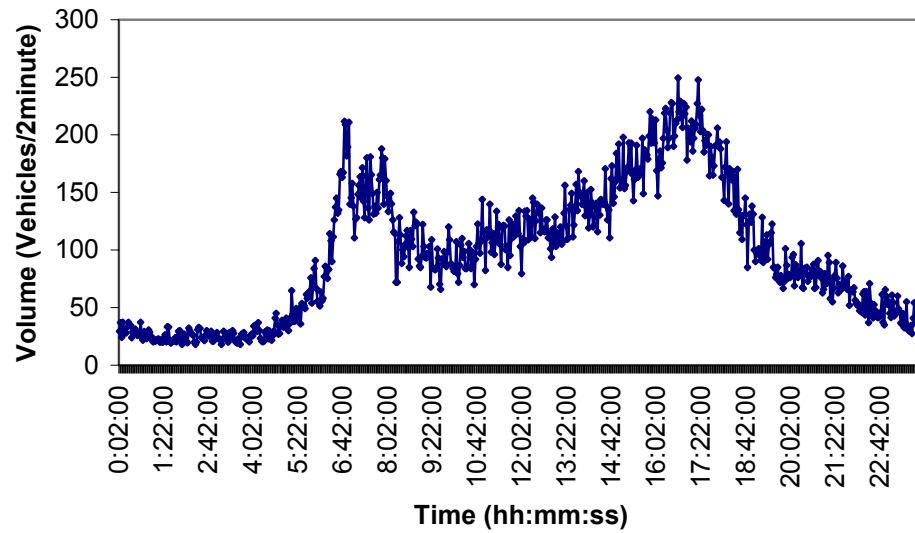


Fig. 5.13 Volume distribution on February 11, 2003, for 24 hours at location 3

To take into account this fluctuation in the estimated travel time, the modification of combining the extrapolation method at low traffic flow conditions (modification IV) was carried out. The resulting travel time values are shown in Figures 5.14 and 5.15.

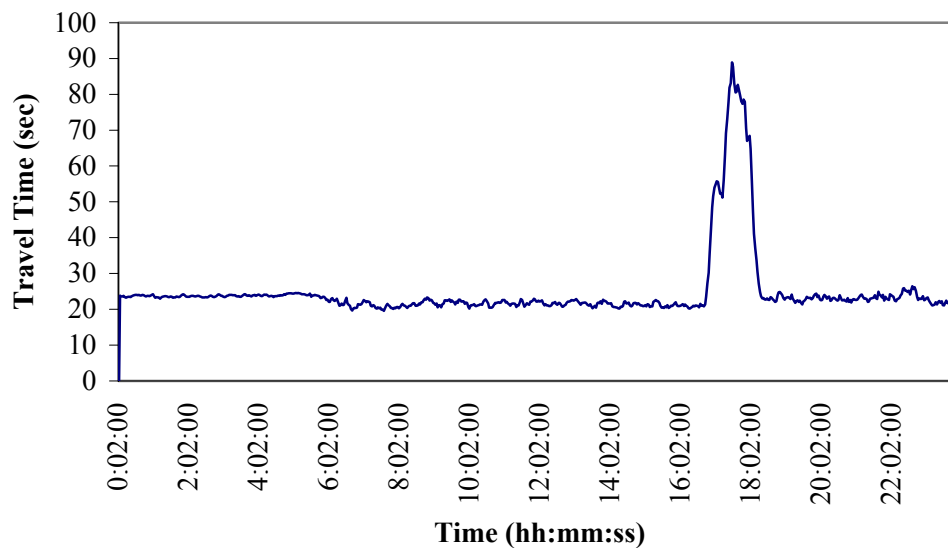


Fig. 5.14 Effect of combining extrapolation method on estimated travel time for low flow conditions on link 1 on February 11, 2003

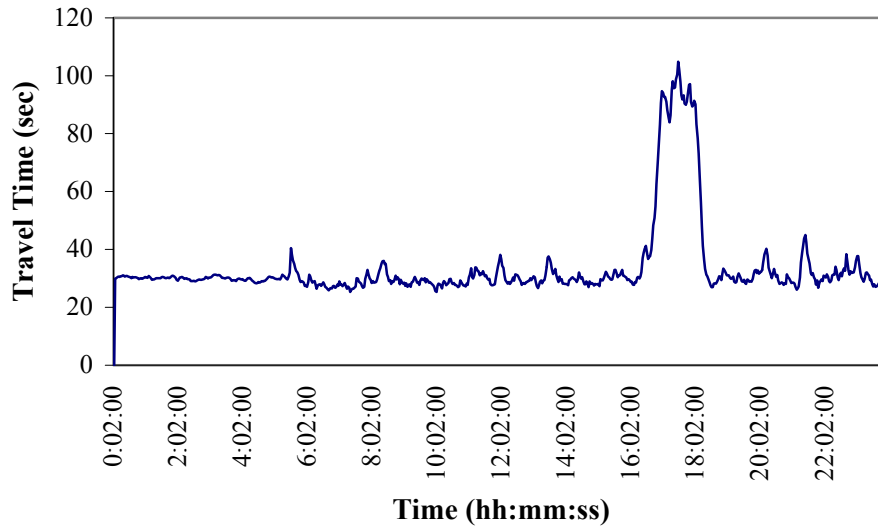


Fig. 5.15 Effect of combining extrapolation method on estimated travel time for low flow conditions on the link 2 on February 11, 2003

An overall comparison of the performance of the N-D model and the proposed model using field data after optimization for a 24-hour period is shown in Figure 5.16 for February 11, 2003. It is clear from the graph that the performance has improved by adopting the suggested modifications. Corresponding AVI data are also plotted in Figure 5.16 to illustrate the improvement in the quality of the estimated travel time. A comparison with the corresponding AVI data shows that the travel time estimated by the proposed model captures similar trends during the whole 24-hour period. MAPE was calculated for the estimated travel time using the N-D model and the proposed model with respect to AVI data and it was found that the error reduced from 98.82% to 3.91%.

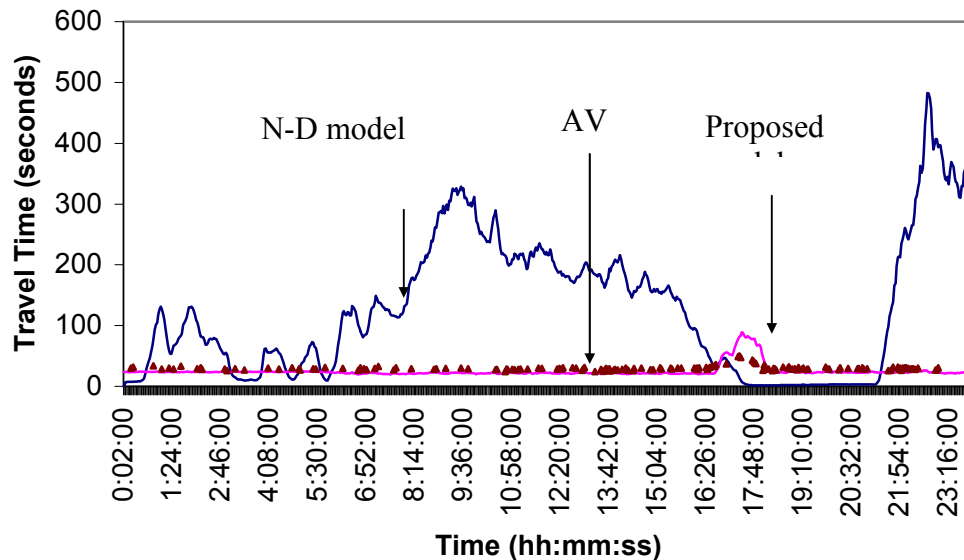


Fig. 5.16 Comparison of N-D model and proposed model using optimized field data for February 11, 2003

A comparison of results was carried out using simulated data also. Some of the specific results related to the effect of the selected modifications on the travel time estimation will be shown in the following section using simulated data. As explained in modification II, the N-D model ignores a portion of the vehicles from the estimation of travel time at the transition period from off-peak to peak. This leads to more error in the estimated travel time at the transition period. $m(t_n)$, which is defined as the number of vehicles which enter and exit the link under consideration in the same time period (Equation 5.13), is the measure used by Nam and Drew to classify normal and congested-flow. Thus, as the transition period start, the value of $m(t_n)$ should start decreasing. Once the value of $m(t_n)$ is less than zero, the flow is considered as congested-flow. The transition period, hence, is considered as normal-flow in the N-D model, and the travel time is calculated based on those vehicles that were able to enter and exit in the same time period. Thus, the portion of vehicles that were not able to exit in the same period gets ignored, leading to more error in the estimated travel time. This was taken into account in the proposed model by using the modified Equation 5.21. The variation in the value of $m(t_n)$ and the corresponding error in the estimated travel time are plotted in Figure 5.17 for both the N-D

model and the developed model using simulated data from CORSIM. The error in estimated travel time is calculated as the absolute difference between the estimated travel time and the travel time calculated directly from simulation. The travel time was estimated using the N-D model and the developed model, and the errors were calculated. It can be seen that the error values increase with decrease in $m(t_n)$ in the case of N-D model, whereas the error of the proposed model remains approximately constant over time.

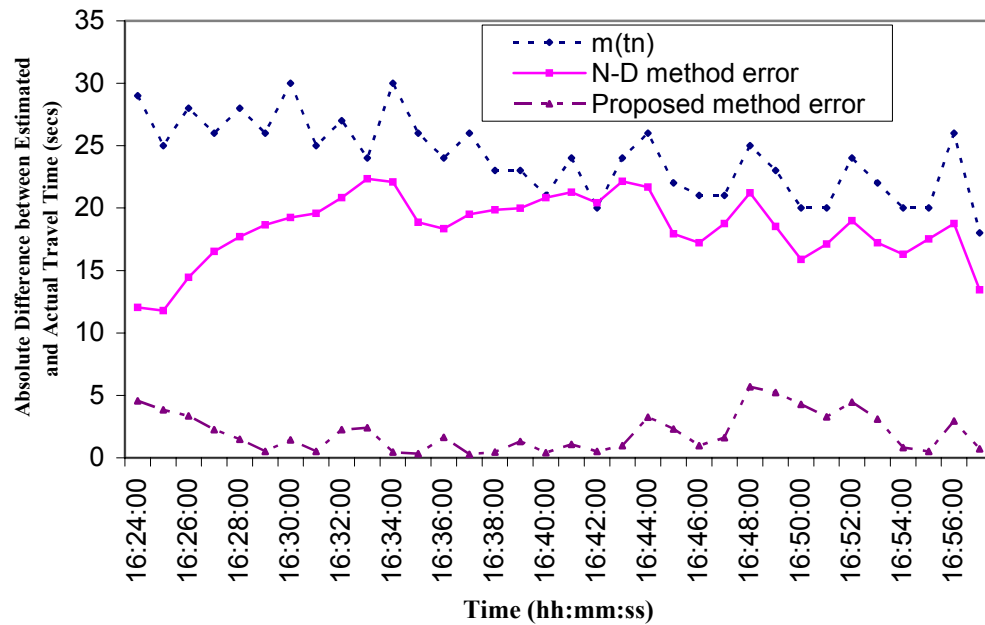


Fig. 5.17 Variation in the performance of the N-D model and the developed model with varying values of $m(t_n)$ during transition from off-peak to peak condition

Similarly, the effect of modification I is tested using simulated data, and the results are shown in Figure 5.18. This figure illustrates the effect of optimization on the accuracy of the estimated travel time using simulated data. The travel time calculated before and after the optimization along with the actual travel time obtained from simulation is shown. This illustration is for an introduced error of 10% in the flow values. The optimization was carried out as detailed in Chapter IV on the data with introduced error. The improvement in the performance and the

increase in the accuracy of the estimated travel time can be observed in the diagram. The MAPE value was calculated and was found to be decreasing from 15.94% to 2.91% with the use of optimized data.

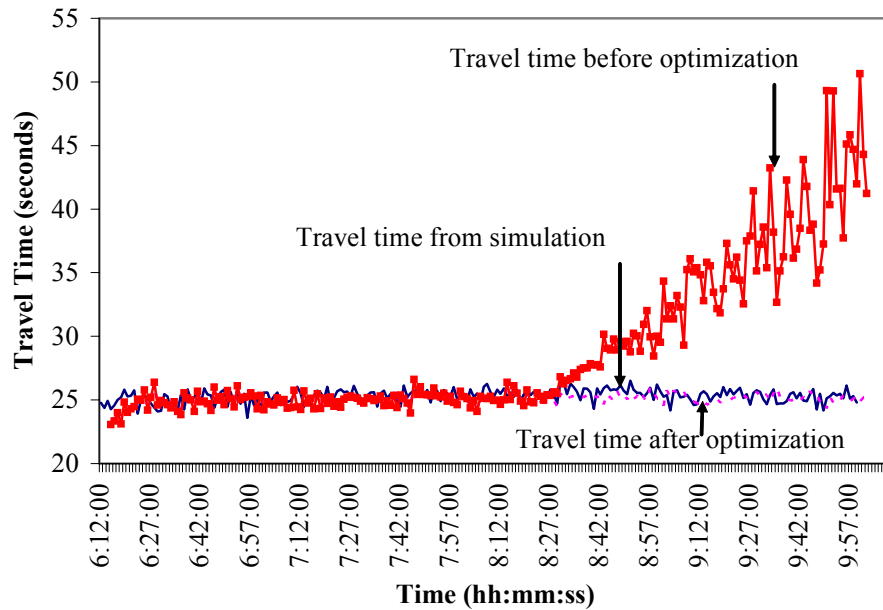


Fig. 5.18 Effect of optimization on the estimated travel time using simulated data

An overall comparison of the performance of the N-D model and the proposed model using simulated data is shown in Figure 5.19 and it illustrates the performance of the proposed model under a transition period using simulated data. The analysis was carried out for a 2-hour period, and the flow values were generated based on field values. The true travel time from the simulation is plotted along with the values estimated by the models. Estimation was carried out using the N-D model and the proposed model. It can be seen that the travel time estimated by the proposed model is in close agreement with the simulation travel time, with an MAPE of 6.58%. In the case of the N-D model, the MAPE was considerably higher at 48.97%.

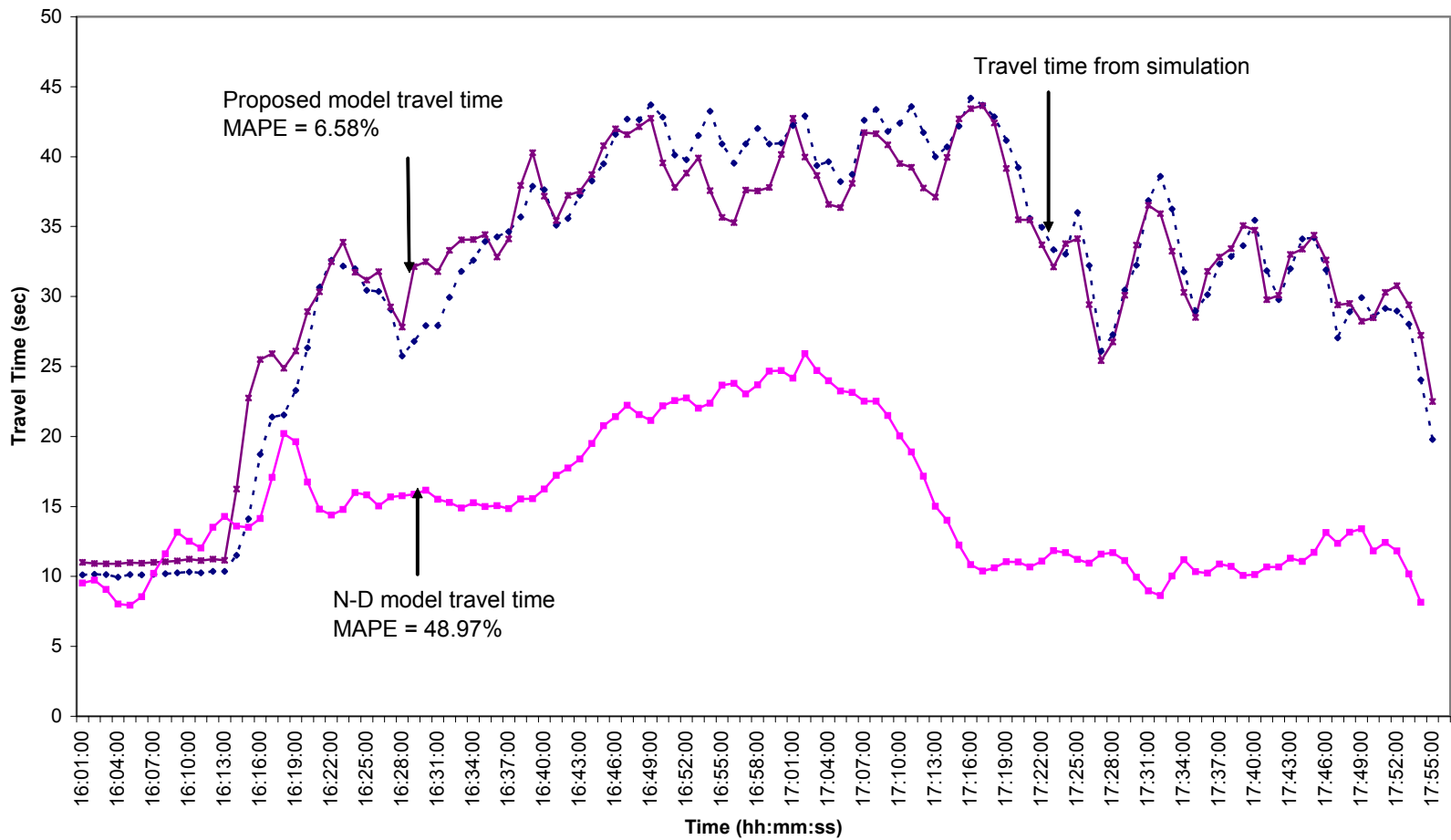


Fig. 5.19 Overall comparison of the proposed model with N-D model using simulated data

5.4.2 Validation of the Developed Model Using Field Data

The model results were validated using field data by comparing them with the corresponding direct travel time obtained from AVI. The results obtained for selected dates are shown in Figures 5.20 to 5.22. It may be seen that the travel time obtained from AVI and that calculated using the developed model are in good agreement for all days. The MAPE between the estimated travel time and AVI travel time were 1.54, 2.53, and 2.38 % for February 10th, 13th and 14th respectively, which are shown in Figures 5.20 to 5.22.

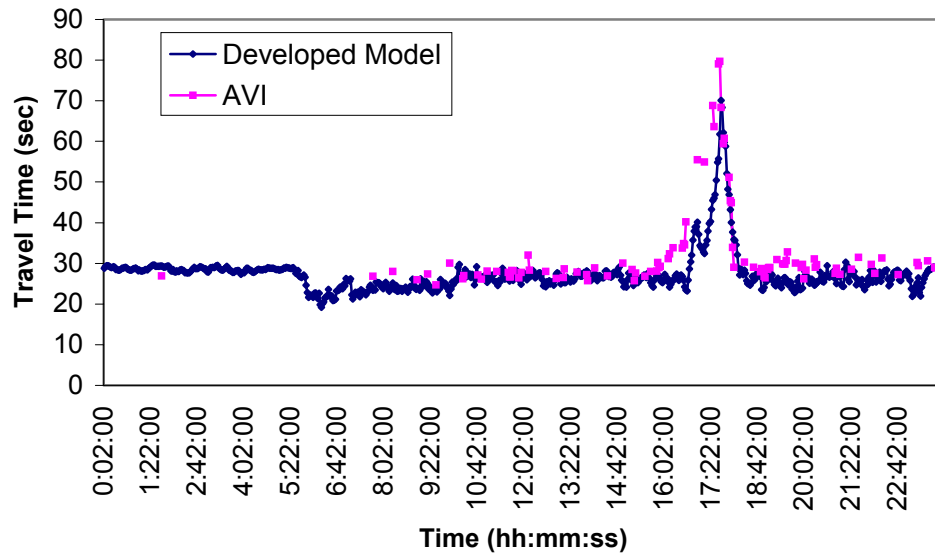


Fig. 5.20 Estimated travel time with AVI for 24 hours on February 10, 2003 in link 1

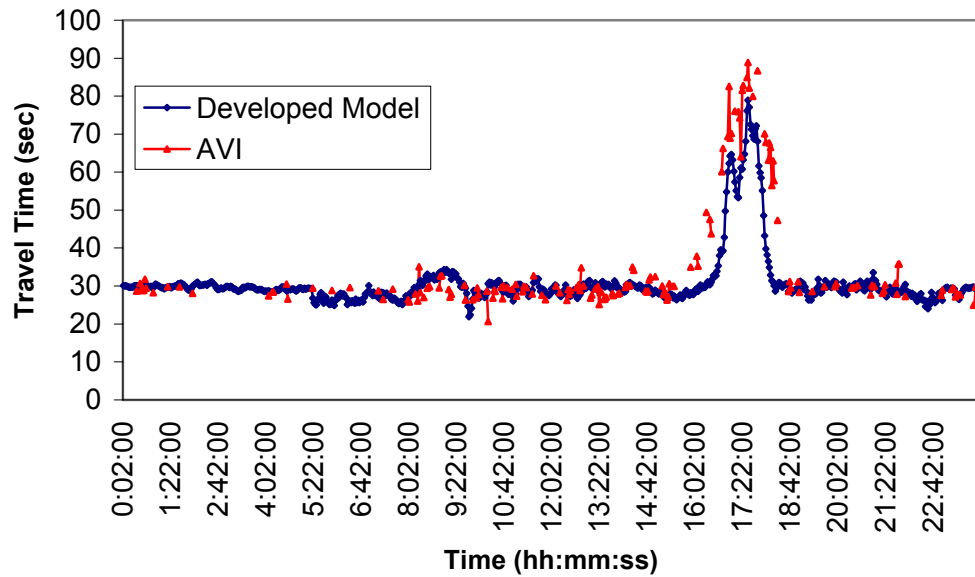


Fig. 5.21 Estimated travel time with AVI for 24 hours on February 13, 2003 in link 1

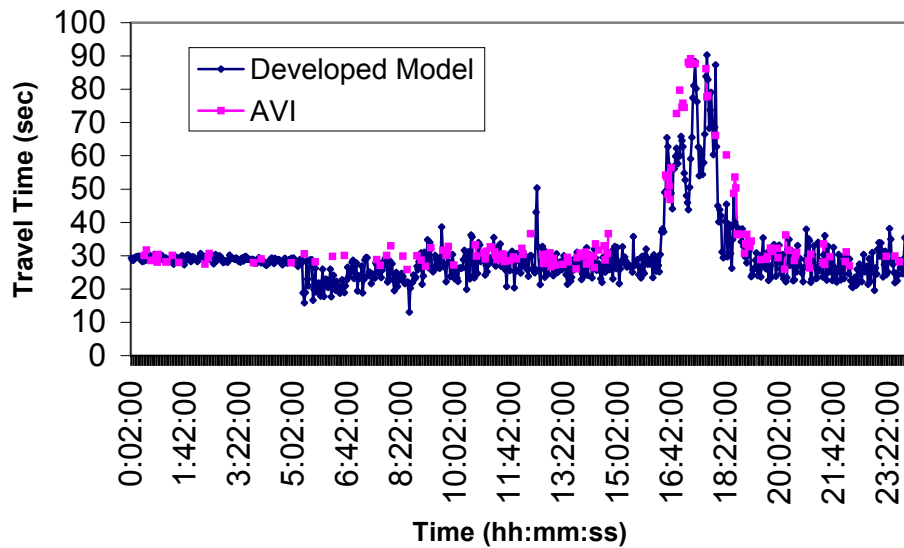


Fig. 5.22 Estimated travel time with AVI for 24 hours on February 14, 2003 in link 1

The performance of the model during the peak and transition periods is enlarged and shown in Figure 5.23 for February 10, 2003. The MAPE was calculated and was found to be 3.87%. It can be seen that the performance of the model is consistently satisfactory under peak and transition periods.

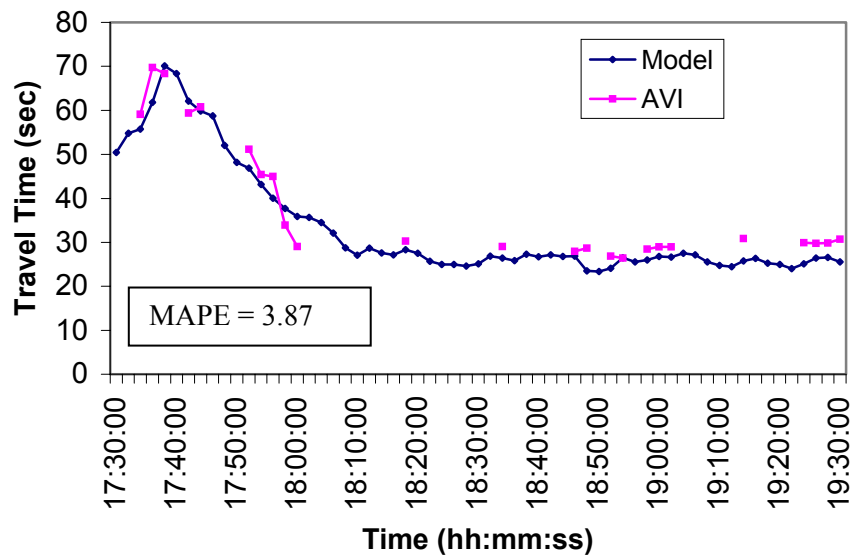


Fig. 5.23 Estimated travel time with AVI for peak and transition periods (February 10, 2003) in link 1

The performance of the model during normal-flow condition is shown in Figure 5.24. The calculated MAPE was 0.75%, showing a good performance of the model under off-peak period.

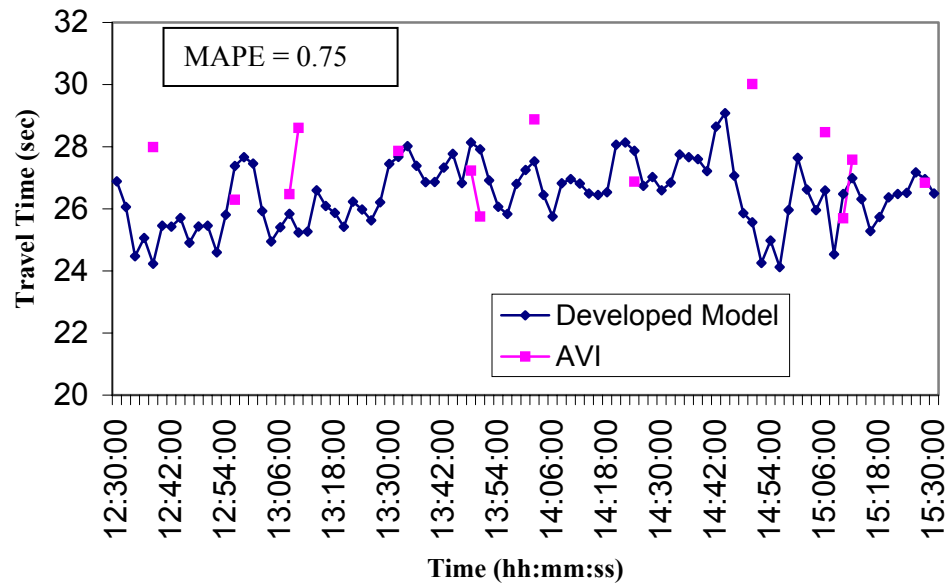


Fig. 5.24 Estimated travel time with AVI for an off-peak period (February 10, 2003) in link 1

However, it should be noted that some assumptions are needed to compare AVI travel time with the travel time estimated from the loop detector data. First of all, the AVI data samples a percentage of the vehicle population and gives the travel time of these selected vehicles. In the case of loop detectors, the data are collected from all the vehicles that cross it, and an average travel time for the interval under consideration is calculated. For example, in the present study for the analyzed 5 days data, the ILD data available at 2-minute interval is 720 observations per day and the corresponding AVI data available varied from 100 to 200.

Also, the time interval of the reported loop data and the time of the AVI data may not match exactly. For example, in the February 10, 2003 data, the loop data are collected from midnight 12:00:00 at 2-minute intervals. The first AVI data reported on that day entered the link at 01:38:22 and exited at 01:42:52. The corresponding loop data available are from 01:38:00 to 01:42:00. Also, the detector location and the AVI location may not match exactly. For example, the starting milepost of the ILD in the present study was 159.500, and the nearest AVI station

was at 158.989. Thus, the data need to be extrapolated to match with each other spatially and temporally.

5.4.3 Validation of the Model Using Simulated Data

Due to the above-mentioned reasons, validation of the models was carried out using simulated data also. A traffic network similar to the field test bed was created in CORSIM and “ILDs” were placed every 0.5 miles, to be comparable to field condition. The vehicles were also generated based on the field values to mimic the field scenario. Traffic volumes from the field were given as input to CORSIM at every 30-minute interval. Detectors were placed in each link to collect the flow, speed, and occupancy rate. Data were generated for 2 hours, which included both peak and off-peak flows. These data were used for checking the validity of the proposed model. The detector output was reported in the OUT file of CORSIM and was used to get the flow, occupancy, and speed values. Travel time was estimated based on these flow, occupancy and speed values and was compared to the travel time given by the simulation. The binary .TSD file from CORSIM, which contains the snap shot data at every time step, was used to calculate the real travel time of individual vehicles from simulation as detailed in Chapter III.

Figure 5.25 illustrate the performance of the developed model during the off-peak period using simulated data. The data were simulated for 4 hours during evening off-peak flow. It can be seen that the travel time estimated by the proposed model follows the travel time calculated directly from CORSIM. The MAPE was found to be 1.8% in this case.

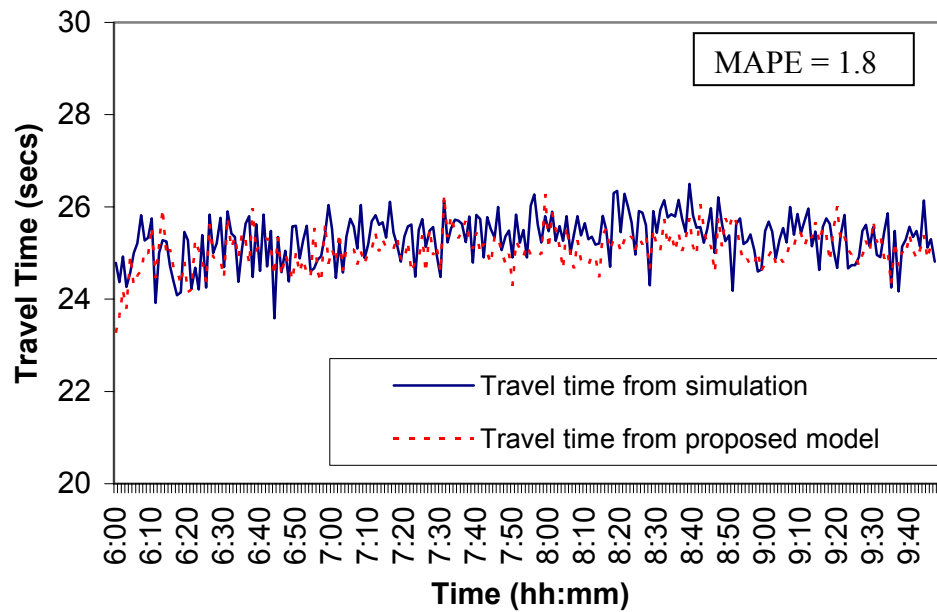


Fig. 5.25 Validation of the Travel time estimation model using simulation data for the off-peak condition

Figure 5.26 shows a similar comparison where the data were simulated for 2 hours. The true travel time from the simulation is plotted along with the estimated values. Again, the estimated travel time by the developed model follows the trends in the actual data. The MAPE came to be 6.58 % in this case.

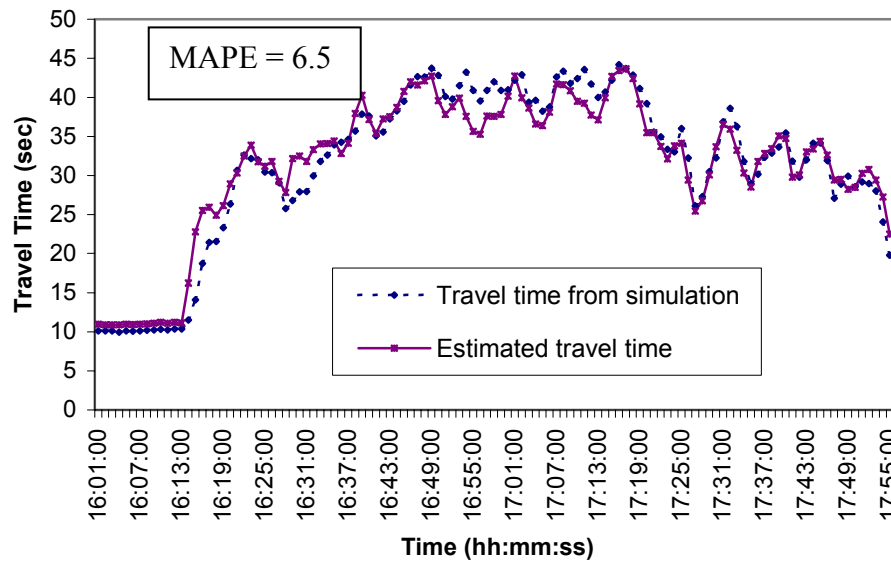


Fig. 5.26 Validation of the travel time estimation model using simulation data for peak condition

5.4.4 Comparison with Extrapolation Results

Though the extrapolation methods, which were discussed in Chapter II, have many drawbacks, they are the most popular methods adopted in the field, and hence a comparison was carried out with the results obtained from the extrapolation as well. The travel time estimated using the proposed model is compared with the results from extrapolation methods. The three different extrapolation methods as discussed in Chapter II (Equation 2.1, 2.2, or 2.3) were analyzed, and the most suitable one was chosen for further comparison. The travel time estimated by the three extrapolation methods is shown in Figures 5.27 and 5.28 for links 1 and 2. Method 1 (Equation 2.1) assumes the effect of speed from each detector for half the distance, method 2 (Equation 2.2) considers the average speed, and method 3 (Equation 2.3) take the minimum speed out of the two detectors, as explained in Chapter II.

From the results obtained at different sites it was found that method 3 tended to overestimate the travel time compared to methods 1 and 2. However, there was no significant difference between the performance of method 1 and 2, and any one of these can be used for further comparison (Eisele 2001). In this dissertation, method 2, which considers the average speed of the two detectors, is used.

A comparison of the travel time estimated by the proposed method and the extrapolation method with the AVI travel time is shown in Figure 5.29 for February 13, 2003. As discussed previously, it may be difficult to reach any solid conclusions by comparing the AVI travel time and the travel time calculated from loop data. However, it can be used for checking whether the estimated data follow the trend in the actual data. It can be seen that the travel time estimated by the developed model is able to capture the variations in the travel time more efficiently than the extrapolation methods. Also, it can be seen that at peak flow conditions, the extrapolation method overestimated the travel time due to the failure to capture the change in speed within the section.

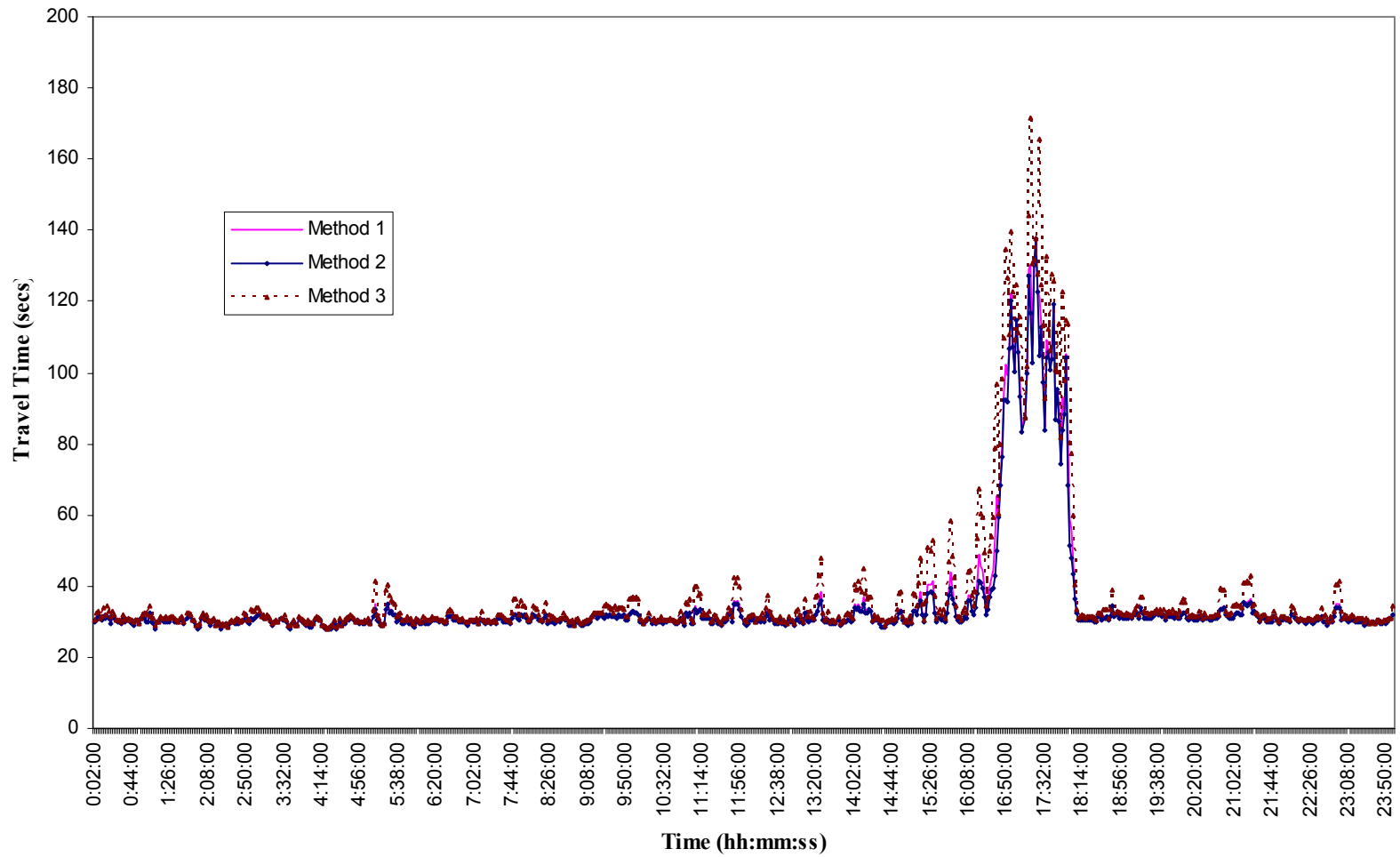


Fig. 5.27 Travel time estimated by different extrapolation methods for link 1 on February 11, 2003

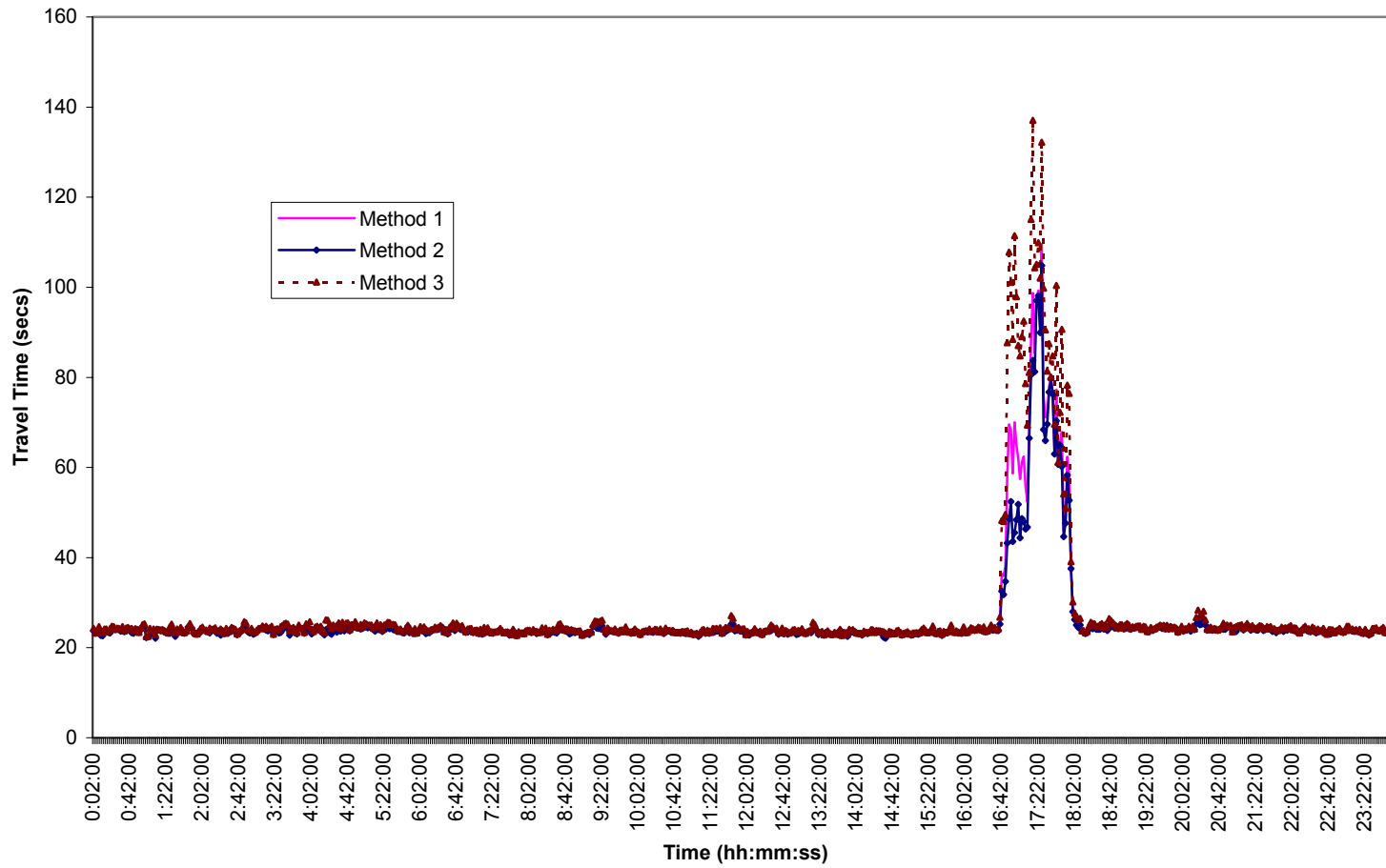


Fig. 5.28 Travel time estimated by different extrapolation methods for link 2 on February 11, 2003

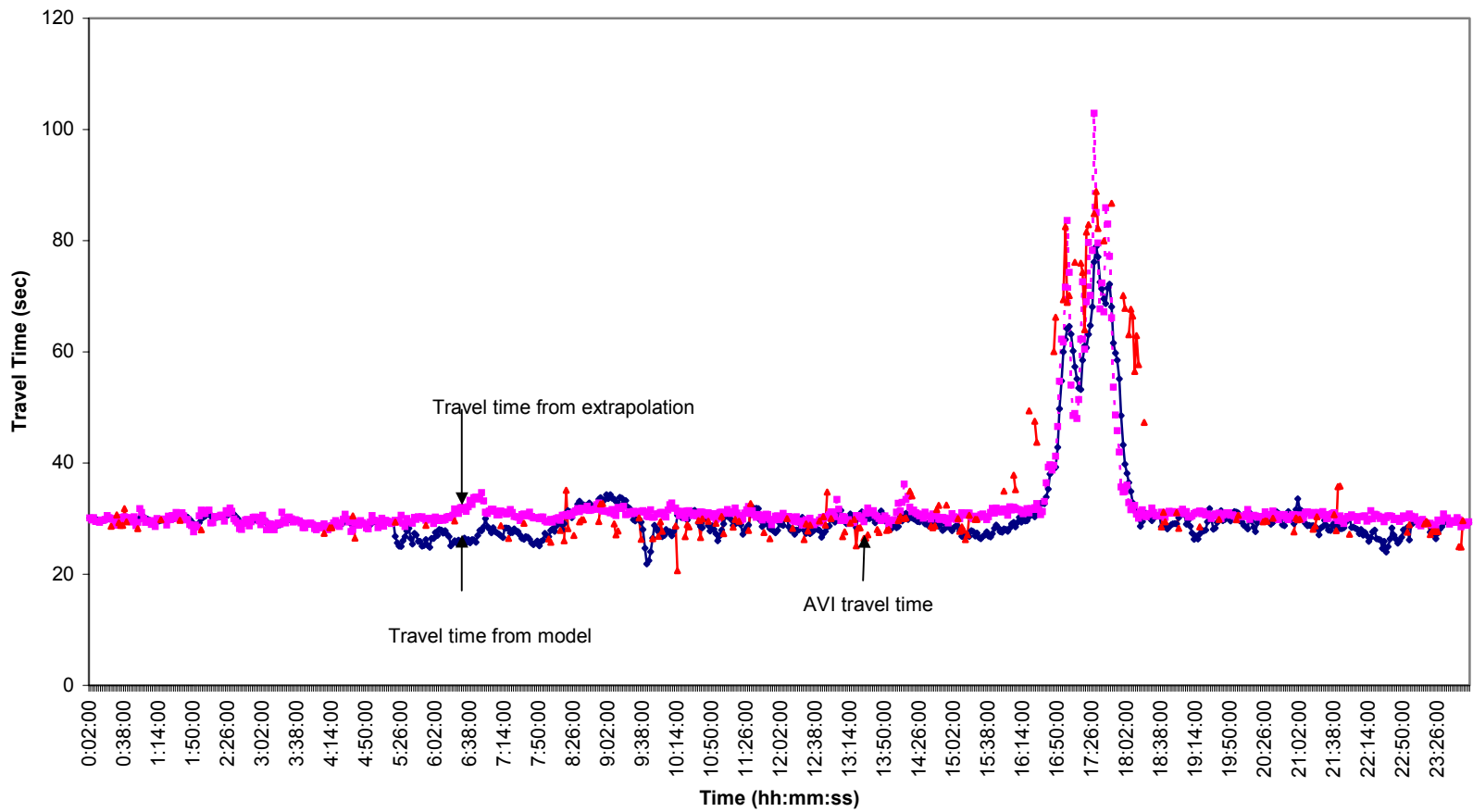


Fig. 5.29 Comparison of estimated travel time from extrapolation method, developed method, and AVI using field data on February 13, 2003

Figures 5.30 to 5.33 show graphs comparing the estimated travel time by extrapolation method and the developed method separately for the off-peak, peak, and transition periods on February 11, 2003 on link 2. The plots are continuous from 15:00:00 to 19:00:00, during which time the flow varied from off-peak to peak and then to off-peak values. From the plots it can be seen that the two values match under off-peak conditions. The mean absolute difference (MAD) between the travel time estimated by the extrapolation method and the proposed model is calculated using Equation 5.27.

$$MAD = \sum \frac{|extrapolated - model|}{N}. \quad (5.27)$$

The MAD was found to be 2.71 from 14:00:00 to 15:00:00 as shown in Figure 5.30. However, during the transition period and peak flow conditions, the values differ reasonably, with the MAD going up to 14.29. This agrees with the findings from previous studies that the extrapolation method fails to capture the changes in flow during congested conditions. The availability of AVI data for the corresponding hours was scarce, and so these data were not included in the plots.

Figure 5.30 displays the travel time estimated by the selected extrapolation method and the travel time estimated by the developed method in the afternoon off-peak hours from 15:00:00 to 16:00:00 on February 11, 2003. It can be seen that both the travel times are close to each other, with an absolute difference of 2.71 between the values.

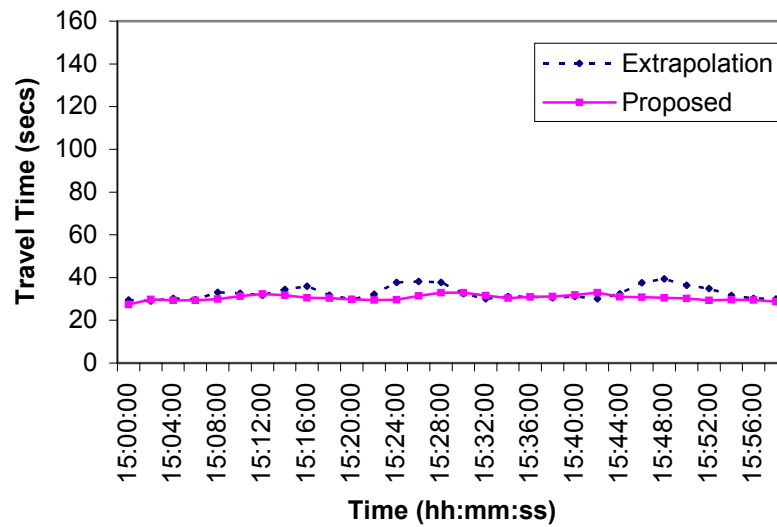


Fig. 5.30 Comparison of extrapolation and developed model results during afternoon off-peak hours

Figure 5.31 displays the travel time estimated by the selected extrapolation method and the travel time estimated by the developed method in the afternoon transition period from off-peak to peak on February 11, 2003. The MAD value was 7.01 and it can be seen that both the travel times are close to each other until the flow increases.

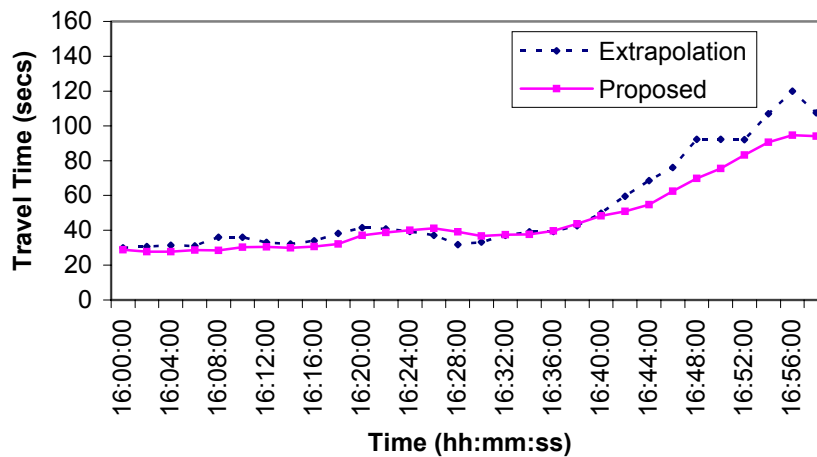


Fig. 5.31 Comparison of extrapolation and developed model results during the start of evening peak hours

Figure 5.32 displays the travel time estimated by the selected extrapolation method and the travel time estimated by the developed method in the afternoon peak period from 16:00:00 to 17:00:00 on February 11, 2003. The MAD came to be 14.29, showing that both the travel times differ considerably from each other.

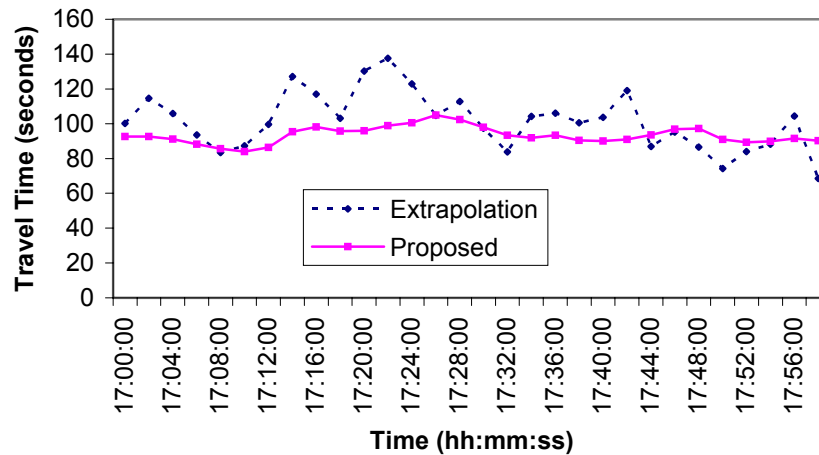


Fig. 5.32 Comparison of extrapolation and developed model results during evening peak hours

Figure 5.33 shows the travel time estimated by the extrapolation method and the travel time estimated by the developed method in the transition period from peak to off-peak on February 11, 2003. The MAD in this case was 7.23, and it can be seen that both the travel times agree with each other after the peak flow is over.

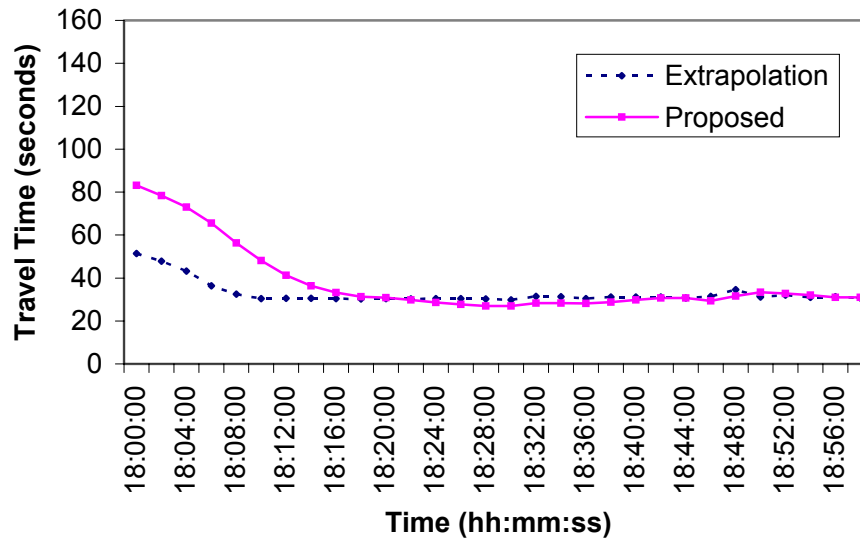


Fig. 5.33 Comparison of extrapolation and developed model results during transition to evening off peak hours

The data pertaining to February 10, 2003 were analyzed in a similar manner and the AVI data were also included. Because the number of AVI data is less compared to ILD data, the analysis was carried out for a longer duration and the results are shown in Figures 5.34 and 5.35. The results again confirm that the performance of extrapolation is reducing at peak flow conditions, whereas the developed model is able to perform uniformly during varying traffic flow conditions. This can be seen from the calculated MAPE of 1.21 and 1.84 for the developed model and extrapolation method respectively under off-peak condition and the corresponding MAPE under congested-flow condition being 4.39 and 6.35.

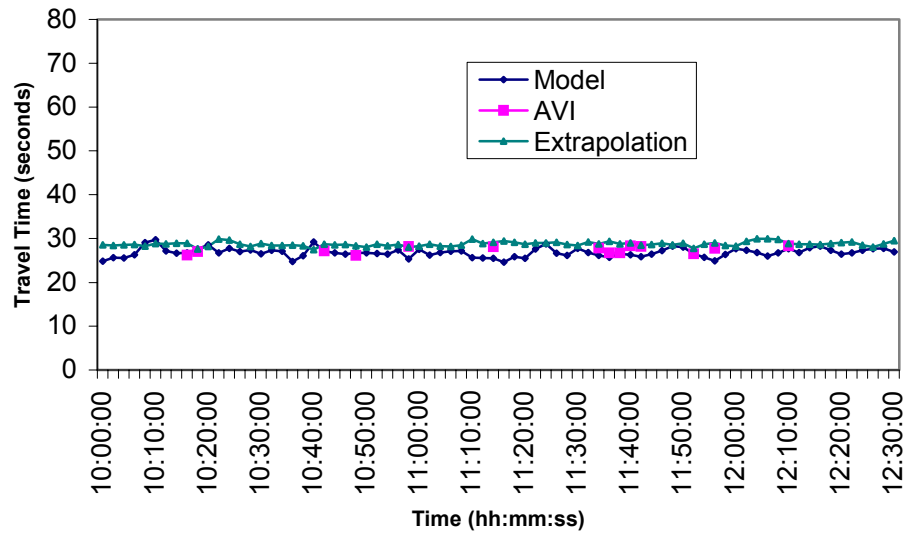


Fig. 5.34 Comparison of extrapolation and developed model results with AVI values during off peak hours on February 10, 2003

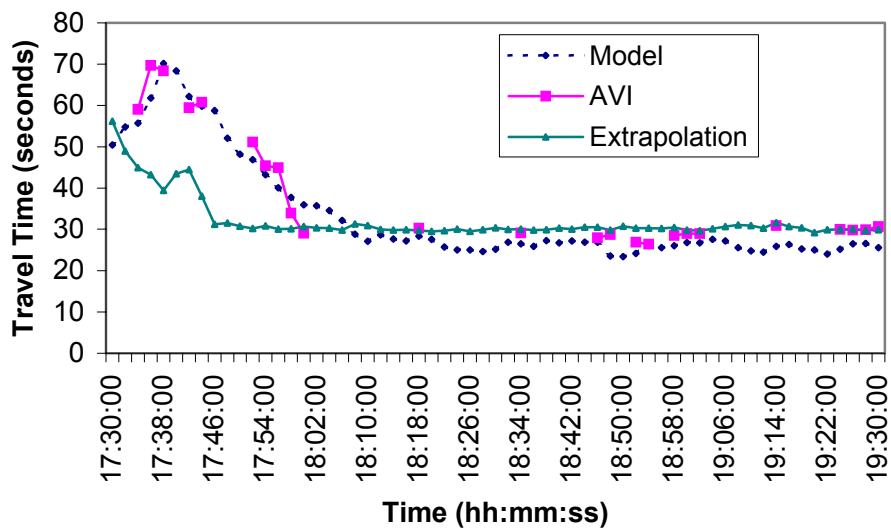


Fig. 5.35 Comparison of extrapolation and developed model results with AVI values during peak and transition periods on February 10, 2003

Similar comparisons were also carried out using simulated data. Figure 5.36 shows a comparison of the extrapolation method and the developed method for the simulated data by CORSIM. The MAPE values in this case were 6.5 and 48.97% respectively, for the proposed method and extrapolation method. It can be seen that, as expected, the performance of the extrapolation reduces as the flow value increases.

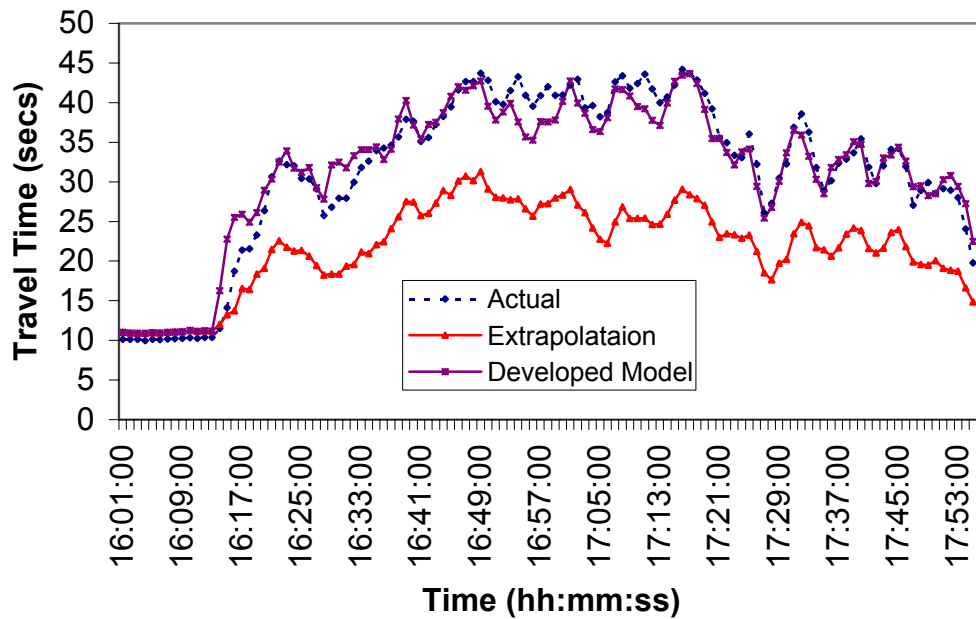


Fig. 5.36 Comparison of the extrapolation method with the developed method using simulated data

Finally, a comparison of the estimated travel time with the variables obtained from field was carried out to check the trends in the values. A plot of the estimated travel time using the developed model is made along with the corresponding occupancy and speed values obtained from the field and is shown in Figure 5.37. It can be seen that the developed model was able to estimate the travel time under varying traffic flow conditions.

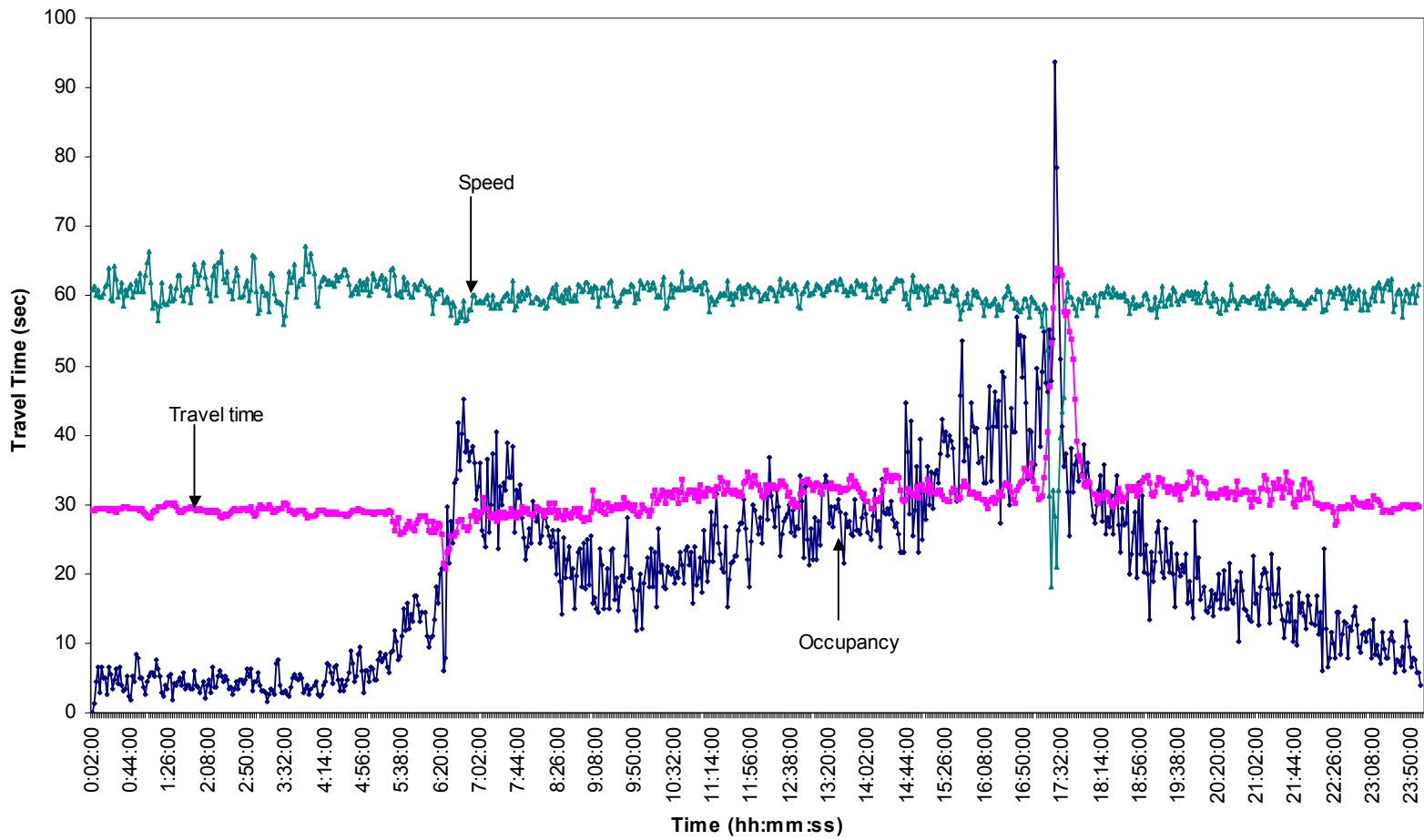


Fig. 5.37 Relation between speed, occupancy, and travel time from February 10, 2003

5.5 CONCLUDING REMARKS

Travel time estimation from loop detector data has achieved increasing interest with the development of ITS applications such as in-vehicle route guidance systems and advanced traveler information systems. Accurate and timely information has to be obtained in a quick fashion to meet the demands of these real-time applications. At present, travel time estimation is carried out in the field based on extrapolation methods, assuming a constant speed for the distance between the detector stations. Studies have shown that the accuracy of the extrapolation method reduces as the flow increases. This is due to the inability of these methods to capture the dynamics of traffic in congested conditions. Thus, there is a need for models that can take into account varying traffic flow conditions.

This dissertation presented several modifications to an existing theoretical model for travel time estimation on freeways, such that the model can estimate travel time for varying traffic flow conditions directly from the loop detector data. The approach was designed for analyzing ILD data for longer intervals of time and was robust enough to suspect or missing data. The system is based on detector data obtained from the field and the travel time estimation is based on the traffic flow theory. Simulated data using CORSIM simulation software was used for validating the results. After the validation, the model was used to estimate travel time from field data. The travel time estimated is compared with the AVI data collected from the field. The travel time estimated was also compared to the results obtained from different available methods such as the extrapolation method. The results indicate the developed model as a promising method to estimate travel time from loop detector data under varying traffic flow conditions.

CHAPTER VI

SHORT-TERM TRAVEL TIME PREDICTION

6.1 INTRODUCTION

After the estimation of travel time from loop detector data was carried out, as explained in Chapter V, the next and final stage in this dissertation was the prediction of travel time. Travel time prediction refers to predicting the travel time before a vehicle traverses the link or route of interest. The ability to predict travel time based on real-time data and historic data, collected by various systems in transportation networks, is vital to many Intelligent Transportation Systems (ITS) applications, such as Route Guidance Systems (RGS), Advanced Traveler Information Systems (ATIS), and Advanced Traffic Management Systems (ATMS).

One of the applications of the above ITS applications is to provide real-time traffic information to traffic management centers, using which traffic information can be provided back to the travelers in real-time. The accuracy of this information is important since travelers make appropriate decisions to bypass congested segments of the network, to change departure times or destinations etc., based on the information. The travel time information provided to travelers through ATIS can be classified into three distinct groups: historic, real-time, and predictive. Historic, as its name implies, is based on archived data, while real-time is based on the current values obtained from the system. Predictive is the predicted future values calculated using the real-time or historic information. For pretrip planning and en-route decisions, it is argued that predicted information would be more useful than real-time or historic information. If the current or historic traffic values are used, the performance of a given application will be constrained because by the time the user makes the trip, the situation would have changed. The travel time prediction becomes very important under such situations where traffic conditions are changing, such as during transition periods. Then the travel time will be a function of 1) when the driver arrives at the link in question and 2) how fast travel times are changing. Thus, the methodology should anticipate the values in the next few minutes under dynamic traffic conditions and inform travelers accordingly.

Previous traffic prediction efforts have used historic and real-time algorithms, time-series and Kalman filtering models, and Artificial Neural Network (ANN) models. More details of these methods and the literature related to the application of these methods on travel time prediction is detailed in Chapter II. However, there is no consensus on the “best” method for travel time forecasting because all the above methods have both advantages and disadvantages. Also, most of the results reported are data specific and cannot be used for choosing one single method that can be applied in all situations. Thus, based on the data characteristics and the specific application requirements, different methods are adopted in different studies.

The objective of the study in this chapter is to investigate the potential of a recently developed pattern classification and regression technique called Support Vector Machines (SVM) for the short-term prediction of travel time. A multilayer perceptron ANN model as well as historic and real-time methods are also developed for comparison purposes. The analysis considered forecasts ranging from a few minutes ahead up to an hour into the future. Up to 4 day’s data were used for training the networks and 1 day’s data were left for cross validation to evaluate the prediction errors. The data used were the estimated travel time obtained from the models described in the previous chapter.

In the following sections, a brief discussion of the historic method, real-time method, ANN, and SVM methods will be given followed by the implementation details for the application of travel time prediction.

6.2 MODELS FOR TRAFFIC PREDICTION

6.2.1 Historic and Real-time Methods

The historic approach is based on the assumption that the historic profile can represent the traffic characteristics for a given time of the day. Thus, a historical average value will be used for predicting future values. This method can be valuable in the development of prediction models since they explain a substantial amount of the variation in traffic over many days. However, for the same reason, the reliability of the prediction is limited because of its implicit assumption that the projection ratio remains constant (Hoffman and Janko 1990). Commuters, in general, have an

idea about the average traffic conditions and will be more interested in abnormal conditions. That is, they are most interested in conditions when average values are not representative of the current or future traffic conditions.

In the real-time approach, it is assumed that the travel time from the data available at the instant when prediction is performed represents the future condition. This method can perform reasonably well for the prediction into the immediate future under traffic flow conditions without much variation (Thakuria et al. 1992). More details of the historic and real-time methods and the literature related to the application of these two methods on travel time prediction are detailed in Chapter II.

6.2.2 ANN

ANN in the most general sense is an information processing structure whose design is motivated by the design and functioning of human brains and components thereof. Thus, ANNs are computing techniques, which can be trained to learn a complex relationship in a data set. Basically it is a parallel computing system composed of interconnected simple processing nodes, which are non-algorithmic, non-parametric, and intensely parallel (Kecman 2001; Haykin 1999).

Over the past several years, both in research and in practical applications, neural networks have proven to be a very powerful method of mathematical modeling. In particular, neural networks are well suited for pattern recognition and classification and to model nonlinear relationships effectively. The use of neural networks has been proven successful in a number of applications where the input-output mapping is highly non-linear and where the functional form of the underlying distributions of the data is difficult to reach. ANNs are applied typically in areas such as sensor processing, pattern recognition, and data analysis and control, which may require information processing structures for which the algorithms or rules are not known.

One major application area of ANNs is forecasting. Several features of ANNs make them valuable and attractive for a forecasting task. First, as opposed to the traditional model-based methods, ANNs are data-driven, self-adaptive methods where very few *a priori* assumptions

about the models are needed. Also, because they “learn” from example data, they can capture subtle functional relationships among the data even if the underlying relationship is unknown or hard to describe. Thus ANNs are suited for problems whose solutions require knowledge that is difficult to specify, but for which there are enough data available. This modeling approach with the ability to learn from experience is very useful for many practical problems because it is often easier to obtain data than to have a good theoretical understanding about the underlying laws governing the system from which data are generated. These abilities of ANNs make them a good tool for forecasting.

Neural networks have been widely used in transportation studies, and a review of these applications can be found in Dougherty (1995), Faghri and Hua (1992), and Nakatsuji and Shibuya (1998). The ANN model, with its learning capabilities, is suitable for solving complex problems like prediction of traffic parameters. ANN models were chosen for traffic prediction mainly because of their ability to take into account spatial and temporal information simultaneously (Park and Rilett 1999). Some of the applications of ANN in the prediction of speed, flow and occupancy in traffic forecasting can be found in Dougherty and Cobbett (1997), Smith and Demetsky (1994), Park et al. (1998), Yun et al. (1998), Dia (2001), Mahalel and Hakkert (1985), Mc Fadden et al. (2001), Nair et al. (2001), Xiao et al. (2003), Ishak et al. (2003), Huang and Ran (2003), and Lee et al. (1998). The literature related to the use of ANN for travel time prediction is reviewed in detail in Chapter II. A brief description of the ANN technique and how it works is detailed below.

ANNs are the primary information processing structures of a technological discipline called neuro-computing (Simon 1993). Neuro-computing is concerned with parallel, distributed, adaptive information processing systems. The difference between neuro-computing and other branches of computing is that in neuro-computing the algorithms are “data driven.” Rather than the computer working through lists of instructions written by a programmer, it learns the strengths of different relationships by being exposed to a set of examples of the behavior concerned. By absorbing the pattern in the data, the network “learns” to generalize (Dougherty et al. 1994).

There are two main groups of ANNs, namely continuous and discrete. As their names imply, the former can take continuous valued input and output, whereas the latter's input and output space are discrete in nature. Different types of discrete/binary neural nets include hopfield net, hamming net, carpenter/grossberg classifier, etc. The networks that can take continuous input include perceptrons, multi-layer perceptrons, Kohonen self-organizing maps etc. (Lippman 1987). The details of most of these networks can be found in any of the standard textbooks on ANN (Haykin 1994; Wasserman 1989; Dayhoff 1990; Beale and Jackson 1990).

Perceptrons are one of the most widely used ANNs, and since it is used in this dissertation also, it is briefly discussed here. A simple perceptron consist of an input layer and an output layer. Each neuron in the input layer will be connected to each neuron in the output layer, and these connections between the input and output layers are adjusted as the network is trained. The multi-layer perceptron (MLP) is based on the original simple perceptron model but with additional hidden layers of neurons between the input and output layers (Lippman 1987). Figure 6.1 shows a schematic diagram of a single perceptron, and Figure 6.2 show a multi-layer perceptron.

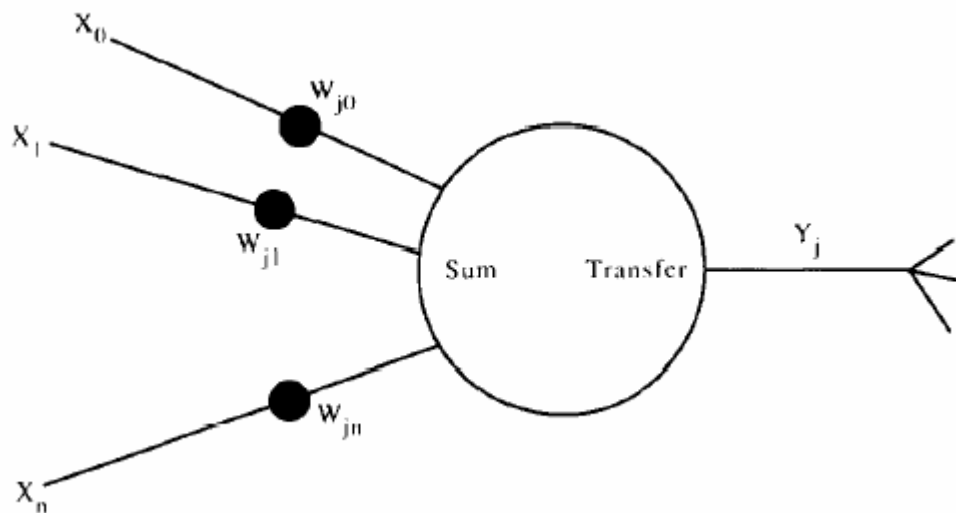


Fig. 6. 1. Schematic diagram of a perceptron
(Source: Dougherty 1995)

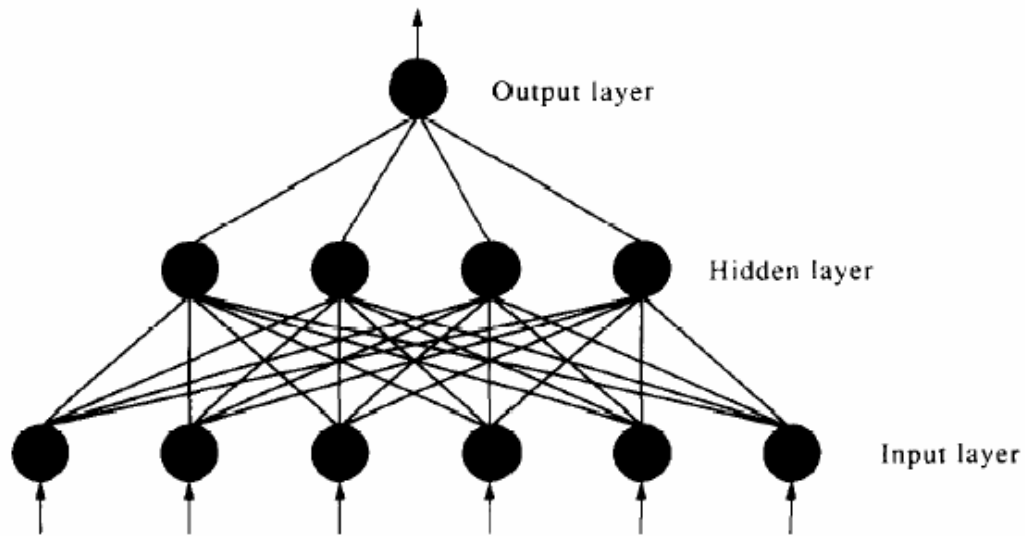


Fig. 6. 2. Multi-layer perceptron
(Source: Dougherty 1995)

A neural network consists of the following elements (Dougherty et al., 1994):

Nodes: The basic building block of ANNs is the neuron, also known as a node or processing element. A node takes in a set of inputs and computes an output according to a transfer function. This is carried out by multiplying each input by a corresponding weight and then summing up all these weighted inputs to determine the activation level of the neuron.

Connection weights: A neural network is composed of many nodes joined together by connections, making the outputs of some nodes as the inputs to others. These connections are of varying strength, and each connection has a weight associated with it.

Bias: The bias is a shifting function that is much like a weight, except that it has a constant input of 1. The bias has the effect of lowering or increasing the net input of the activation function, depending on whether it is negative or positive, respectively.

Transfer function/Activation function: Typically the output state of a single neuron can be characterized as either “on” or “off.” A change from one state to the other is triggered when the sum of the inputs (weighted by the strength of their respective connections) exceeds some threshold. This threshold is usually represented by transfer functions such as sigmoid, logistic, hyperbolic, linear, etc.

Layers: In theory, any topological arrangement of nodes and connections will be sufficient. However, to make the visualization easier, it is usual to arrange the neurons in layers, with all nodes in adjacent layers connected to each other. A neural network thus has an input layer, an output layer, and one or more hidden layers.

Figure 6.3 below shows the model of a neuron with all the above elements. In the case of perceptrons, an input vector p is transformed to an intermediate vector of hidden variables n using an activation function f .

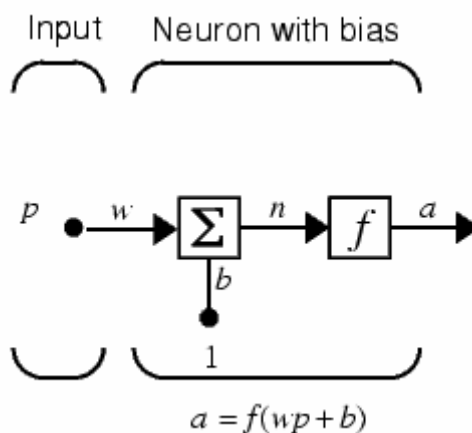


Fig. 6. 3. Model of a neuron
(Source: MathWorks, Inc. 2003)

The output of the j^{th} node in a hidden layer can be mathematically represented as:

$$n_j = f_1 \left(\sum_{i=1}^N w_{i,j}^1 p_i + b_j^1 \right), \quad (6.1)$$

where,

b_j^1 = bias of the j^{th} node in the hidden layer, and

$w_{i,j}^1$ = weight of the connection between the j^{th} node in the hidden layer and the i^{th} input node.

The superscript 1 denotes that the connections are between the input layer and the hidden layer. The output vector \mathbf{a} of the network is obtained from the vector of intermediate variables through a similar transformation using the activation function as:

$$a_k = f_2 \left(\sum_{l=1}^M w_{l,k}^2 n_l + b_k^2 \right), \quad (6.2)$$

where, the superscript 2 denotes that the connections are between the hidden layer and the output layer. The training of an MLP network involves finding values of the connection weights that minimize the error function between the actual network output and the corresponding target values in the training set.

Thus, the performance of ANNs mainly depends on the training rules used. There are different training rules available to train neural networks. These training rules specify an initial set of weights (usually random in the range of [-0.5, 0.5]) and indicate how the weights should be adapted during the training to improve the performance. In other words, the purpose of the learning algorithm is to adjust the network so that the network produces the “correct” outputs for the given set of examples. The learning methods are mainly categorized into supervised and unsupervised. Supervised learning consists of training a network with a set of examples for which the desired outputs are known. In each step, the calculated output is compared with the desired output and a global error function is computed. The weights are then adjusted to reduce the error, and this process occurs over and over as the weights are continually tweaked. The set of data, which enables the training, is called the “training set.” During the training of a network the same set of data is processed many times as the connection weights are refined. In

unsupervised learning, the training of the network is entirely input data-driven and no target results for the input data vectors are provided.

The learning algorithm used can be non-constructive or constructive in nature. Non-constructive means the algorithms for which the topology of the network has to be fixed a priori, while in constructive ones the algorithm itself automatically determines the topology of the network. Most of the learning algorithms used in ANN are non-constructive and supervised. Some of the more popular non-constructive supervised learning algorithms are the perceptron learning algorithm (Rosenblatt 1962) and the back propagation algorithm (Rumelhart et al. 1986)

Back propagation is one of the earliest, most widely used, and the most successful learning algorithms. The present study also uses this algorithm, and so it will be described in more detail. Back propagation is a supervised learning algorithm that provides a method to adjust the weights in a multilayer network of connected processing units. The back propagation algorithm is an extension of the least mean square (LMS) algorithm, which will minimize the errors between the actual and the desired output.

A gradient based approach is used to minimize the error at the outputs in the back propagation method. This is done by calculating the error function for each input pattern and then back propagating the error from one layer to the previous one. The weights of a node are adjusted in direct proportion to the error in the units to which it is connected. Any of the measures of error such as the sum of the mean square error can be used for this purpose.

The steps involved in training a back propagation network are as follows:

1. Initialize weights;
2. Present input and desired output pair to the network;
3. Compute an output which emerges from the output layer (forward pass) using the starting connection weights;
4. Compare this output with the value of output that was expected for this example by computing an error function;
5. Update the connection weights by a small amount to displace the output towards the desired output. This updating starts from the output layer and works backwards to adapt

weights. This is achieved by “back propagating” the global error function (backward pass). The weights are updated as given in Equation 6.3:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x'_j, \quad (6.3)$$

where,

$w_{ij}(t)$ = weight of the connection between the node i and node j at time t ,

x'_j = either output from node j or the input to the network,

η = gain term, and

δ = error term for node j .

If node j is an output node, the error term is calculated as:

$$\delta_j = y_j(1 - y_j)(d_j - y_j), \quad (6.4)$$

where,

d_j = desired output of node j , and

y_j = actual output of node j .

If node j is an internal hidden node, then the error term is:

$$\delta_j = x'_j(1 - x'_j) \sum_k \delta_k w_{jk}, \quad (6.5)$$

where,

k = all nodes in the layers above node j .

6. Present the next input pattern;
7. Calculate total error by calculating the outputs for all training patterns; and
8. Adapt weights starting from the output layer.

If the training is successful, the squared difference reduces over time, as the algorithm continuously iterates through the example data. The convergence can be checked by checking the Root Mean Square (RMS) error values. The rate of convergence varies greatly, and there are various methods to increase it, such as the use of variable momentum term and learning rate. The

variable momentum helps to update the weights during iteration as a function of the previous weight. The learning rate is used to identify the step size to be used for updating the weights. Hence, the selection of these two should be carried out judiciously. A large momentum and large learning rates may lead to local minima, rather than the global minimum.

If a momentum term is added, Equation 6.3 becomes:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x'_j + \alpha (w_{ij}(t) - w_{ij}(t-1)) , \quad (6.6)$$

where,

α = the momentum term.

The main disadvantage of the back propagation algorithm is the step size problem to find the global minimum in the overall error function. If the step size is too small, a local minimum will be reached. If the step size is too large, the network may oscillate around the global minimum without reaching it. Also this algorithm assumes that the changes in one weight have no effect on the error gradient of other weights, which may not be true.

Several variations of the back propagation algorithm were developed to take into account the above-mentioned drawbacks. Some of the examples include the quickprop algorithm, bold driver method, Levenberg-Marquardt (LM) algorithm, etc. The LM algorithm appears to be the fastest method for training moderate-sized feed forward neural networks (MathWorks, Inc. 2003). It also has a very efficient MATLAB implementation because the solution of the matrix equation is a built-in function, and hence its positive attributes become even more pronounced in a MATLAB programming environment.

A major perceived disadvantage of ANN models is that, unlike other statistical models, they provide no information about the relative importance of the various parameters (Dougherty et al. 1994). In ANNs, as the knowledge acquired during training is stored in an implicit manner, it is very difficult to come up with a reasonable interpretation of the overall structure of the network. This has led to the term “black box,” which many researchers use while referring to ANNs behavior (Speed and Spiegelman 1998; Kecman 2001).

6.2.3 SVM

At present ANN is one of the most popular methods in use for the prediction of traffic parameters. However, there are numerous practical shortcomings associated with conventional ANNs including the difficulty in selecting the optimum number of hidden layers and hidden neurons. Another common concern about ANN is the difficulty in providing a reasonable interpretation of the overall design of the ANN network, as discussed previously. In response, a number of modifications have been proposed to alleviate these shortcomings and some of them were applied for the problem of travel time prediction by Park and Rilett (1998), Park et al. (1999), Rilett and Park (2001), Kisgyorgy and Rilett (2002).

These shortcomings also led to explore alternative techniques for the prediction of traffic parameters. In this dissertation one such alternative technique, namely SVM, is explored for the prediction of travel time. The performance of SVM for the prediction of traffic speed is also explored in this dissertation to check whether the results are data specific. Several studies compared the performance of ANN and SVM in other applications. Gunn (2003) reported that the traditional neural network approaches have limitations on generalization, giving rise to models that may over-fit the training data. This deficiency is due to the optimization algorithm used in ANN for the selection of parameters and the statistical measure used for selecting the model (Gunn 2003). Valyon and Horvath (2002) also discussed the issue of poor generalization and over fitting of ANN when presented with noisy training data. Samanta (2004) and Jack and Nandi (2002) compared the performance of SVM and ANN for the application of gear fault detection. Samanta (2004) reported almost equal performance from both the methods, with slightly better performance from SVM. However, Jack and Nandi (2002) reported that the generalization from ANN was better than SVM.

The main difference between SVM and ANN is in the principle of risk minimization (RM). In the case of SVM, the structural risk minimization (SRM) principle is used, which minimizes an upper bound on the expected risk, whereas in ANN, traditional empirical risk minimization (ERM) is used which minimizes the error in the training data. Training in SVM involves the optimization of a convex cost function without any local minima to complicate the learning

process (Campbell 2002). The comparison between ANN and SVM was addressed by Kecman (2001) as: “NNs had a more heuristic origin. This does not mean that NNs are of lesser value for not being developed from clear theoretical considerations. It just happens that their progress followed an experimental path, with a theory being evolved in the course of time. SVMs had a reverse development: from theory to implementation and experiments. It is interesting to note that the very strong theoretical underpinnings of SVMs did not make them widely appreciated at first.”

SVM has been successfully applied to a number of applications ranging from particle identification to database marketing (Campbell 2002). The approach is systematic and is motivated by statistical learning theory. Support vector machines are constructed from a unique learning algorithm that extracts training vectors that lie closest to the class boundary, and makes use of them to construct a decision boundary that optimally separates the different classes of data. These sets of training patterns which carry all relevant information about the classification problem are called support vector (Hearst 1998). Thus, the model constructed has an explicit dependence on a subset of the data points (the support vectors). SVMs represent novel learning techniques that have been introduced in the framework of structural risk minimization (SRM).

Support vector algorithms can be used in the case of problems which are complex, yet the method is simple enough to be analyzed mathematically, because it can be shown to correspond to a linear method in a high dimensional feature space non-linearly related to input space. But, it does not involve any computations in the high dimensional space. By the use of kernels, all the necessary computations are performed directly in the input space.

In the case of a binary classification problem, SVM attempts to place a linear boundary between two different classes, and orient it in such a way that the margin is maximized. In essence, the learning problem is cast as a constrained nonlinear optimization problem. In the case of classification of linearly separable data, the approach is to find among the hyperplanes the ones that minimize the training error as shown in Figure 6.4. The SVM tries to orient the boundary such that the distance between the boundary and the nearest data point in each class is maximal as shown in Figure 6.5. The boundary is then placed in the middle of this margin between the two points. The maximal margin is used for better classification of new data (generalize). The

nearest data points are used to define the margins and are known as support vectors. Once the support vectors are selected, the rest of the data can be discarded (Samanta 2004). Thus, SVM uses the strategy of keeping the error fixed and minimizing the confidence interval.

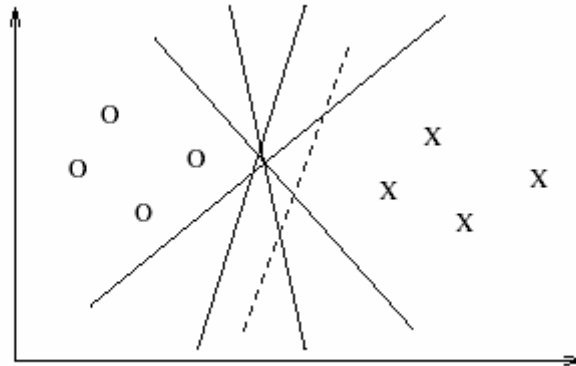


Fig. 6. 4. Separating hyperplanes

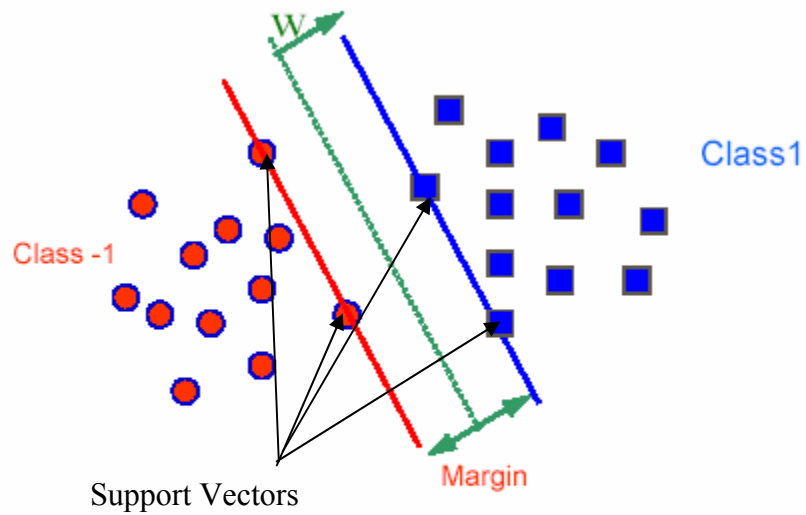


Fig. 6. 5. Support vectors with maximum margin boundary

In the following section, a simple model of SVM for a classification problem of two separate classes is illustrated. This model problem gives an overview of how SVM works. For more detailed explanations any of the tutorials or standard textbooks can be referred (Vapnik 1998; Burges 1998; Smola and Scholkopf 1998; Cristianini and Shawe-Taylor 2000; Kecman 2001).

Let the binary classification data points be

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\}, \quad \mathbf{x} \in \mathfrak{R}^n, \mathbf{y} \in \{-1, 1\} \quad (6.7)$$

where,

y = a binary value representing the two classes, and

\mathbf{x} = the input vector.

As explained previously, there are a number of hyperplanes that can separate these two sets of data and the problem is to find out the one with the largest margin. The SV classifiers are based on the class of hyperplanes called boundary lines,

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0, \quad \mathbf{w} \in \mathfrak{R}^n, b \in \mathfrak{R}, \quad (6.8)$$

where,

\mathbf{w} = the boundary,

\mathbf{x} = the input vector, and

b = the scalar threshold.

To remove redundancy, the hyperplane is considered in canonical form defined by a unique pair of values (\mathbf{w}, b) at the margins satisfying the condition:

$$(\mathbf{w} \cdot \mathbf{x}) + b = 1, \quad (6.9)$$

$$(\mathbf{w} \cdot \mathbf{x}) + b = -1. \quad (6.10)$$

The quantities \mathbf{w} and b will be scaled for this to be true, and therefore the support vectors correspond to the extremities of the data. Thus, the decision function that can be used to classify the data is:

$$\mathbf{y} = \text{sign}((\mathbf{w} \cdot \mathbf{x}) + b). \quad (6.11)$$

Thus, a separating hyper plane in canonical form must satisfy the following constraints:

$$y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1, \quad i = 1, \dots, l \quad (6.12)$$

where,

l = the number of training sets.

There can be many possible hyperplanes that can separate the training data into the two classes. However, the optimal separating hyperplane is the unique one that not only separates the data without error but also maximizes the margin. This means that it should maximize the distance between the closest vectors in both classes to the hyperplane. This margin, ρ is the sum of the absolute distance between the hyperplane and the closest training data points in each class.

This distance $d(w, b; x)$ of a point x from the hyperplane (w, b) is:

$$d(w, b; x) = \frac{|(\mathbf{w} \cdot \mathbf{x}_i) + b|}{\|\mathbf{w}\|}. \quad (6.13)$$

Thus, the sum of the absolute distance between the hyperplane and the closest training data points in each class i and j , ρ is calculated as given in Equation 6.14.

$$\rho = \min \frac{|(\mathbf{w} \cdot \mathbf{x}_i) + b|}{\|\mathbf{w}\|} + \min \frac{|(\mathbf{w} \cdot \mathbf{x}_j) + b|}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \quad (6.14)$$

The optimal canonical hyperplane is the one that maximizes the above margin. Thus, the optimal hyperplane, with the maximal margin of separation between the two classes can be uniquely

constructed by solving a constrained quadratic optimization whose solution is in terms of a subset of training patterns that lie on the margin. These training patterns, called support vectors, carry all relevant information about the classification problem.

In cases where the given classes cannot be linearly separated in the original input space, the SVM first non-linearly transforms the original input space into a higher dimensional feature space as shown in Figure 6.6. This transformation is carried out by using various non-linear mappings: polynomial, sigmoidal, radial basis, etc. After the non-linear transformation step, SVM finds a linear optimal separating hyperplane in this feature space (Kecman 2001; Campbell 2002). Thus, a non-linear function is learned by a linear learning machine in a kernel induced feature space.

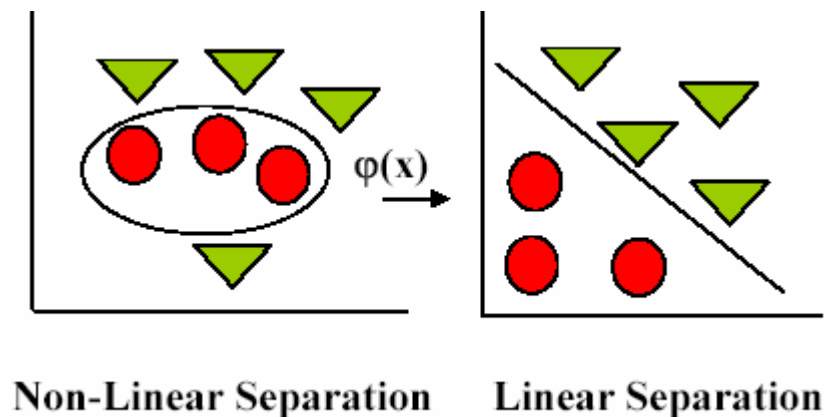


Fig. 6. 6. The kernel method for classification

In Support Vector regression (SVR) the basic idea is to map the data into a high-dimensional feature space F via a non-linear mapping ϕ and to do linear regression in this space.

$$f(x) = (w \cdot \phi(x)) + b, \quad (6.15)$$

$$\text{with } \phi: R^n \rightarrow F, \quad w \in F, \quad (6.16)$$

where,

b = the threshold.

Thus, linear regression in a high dimensional (feature) space corresponds to non-linear regression in the low dimensional input space R^n (Kecman 2001).

Overall, the construction of an SVM incorporates the idea of structural risk minimization. According to this principle, the generalization error rate is upper bounded by a formula. By minimizing this formula, an SVM can assure a known upper limit of the generalization error. The primary advantage of the SVM method is that it automatically calculates the optimal (with respect to generalization error) network structure for a given problem. In practice it means that a lot of questions that had to be answered during the design of a traditional NN (e.g., the number of neurons, the length and structure of the learning cycle, etc.) are eliminated. However, some other questions arise, namely the proper selection of some other parameters that are used in SVM. Such parameters are the loss function (ϵ), which determines the cost of deviation from the training sample, the width of the Gaussian radial bases, σ and C , which is a trade off between the minimization of the training error and the number of training points falling outside the error boundary.

The drawbacks of the SVM method are addressed in some of the previous literature (Talukder and Casasent 2001), and they are discussed briefly here. SVMs for classification involve designing classifiers based on only a few so-called support vectors that lie close to the decision boundary between the two classes. Linear SVM calculates a linear basis function that maximizes the minimum distance between the classes. It is a linear combination of the training vectors. Thus, when the training data set is large (>5000), the problem cannot be solved on a PC or equivalent computer without data and problem decomposition. Another reported drawback of SVM is that when the data classes overlap, a user-defined cost parameter to measure the amount of misclassification is needed.

SVMs have been successfully applied to a number of applications ranging from face identification to time series prediction. Some of the recent applications for the pattern recognition case are: handwritten digit recognition (Cortes and Vapnik 1995; Scholkopf et al. 1995, 1996; Burges and Scholkopf 1997), object recognition (Blanz et al. 1996), speaker identification (Schmidt 1996), face detection (Osuna et al. 1997a) and text recognition (Joachims

1997). For the regression estimation case, SVMs have been compared with time series prediction sets (Muller et al. 1999; Mukherjee et al. 1997; Osuna et al. 1997b).

Reported applications of SVM in the field of transportation engineering are very few and are discussed below. Yuan and Cheu (2003) used SVM for incident detection in an arterial network (simulated) and a freeway network (actual). Two different non-linear kernels were trained and tested. The method was compared to a multi-layer feed forward (MLF) ANN and a probabilistic neural network. Based on their results they reported that SVM had a lower misclassification rate, higher correct detection rate and slightly faster detection time than the multi-layer feed forward neural network and probabilistic neural network models while using simulated data. While using real data from the field, the detection performance was reported as equal to that of the MLF network. Ding et al. (2002) proposed a traffic time series prediction based on the SVM theory. Another reported application of SVM in the traffic engineering area is for vehicle detection (Sun et al. 2002a, 2002b). Vanajakshi and Rilett (2004a) studied the application of SVR in traffic speed prediction and compared the results with the performance of a multi-layer feed forward neural network, and real-time and historic methods.

6.3 MODEL PARAMETERS

6.3.1 ANN

In this dissertation a multi-layer perceptron network with back propagation algorithm is used because of its excellent predictive capacities as reported in previous studies for similar applications (Smith and Demetsky 1994, 1997; Lee et al. 1998). In particular, multi-layer feed forward neural networks that utilize a back propagation algorithm have been applied successfully for forecasting traffic parameters (Mc Fadden et al. 2001; Huang and Ran 2003; Park and Rilett 1999).

In this dissertation, programs were developed in MATLAB for the neural network application. For an application using ANN, first the network needs to be trained, where the weights and node biases are calculated. For this the available data set is divided into a training set and testing set. A training set is used to estimate the arc weights and node biases, and the testing data are used

for measuring the generalization ability of the network. The parameter selection was carried out carefully to get the best results, the details of which are given below.

6.3.1.1 Number of Hidden Layers and Nodes

Because most theoretical works show that a single hidden layer is sufficient for ANNs to approximate any complex non-linear function with any desired accuracy (Cybenko 1989; Hornik et al. 1989) most of the forecasting applications use only one hidden layer. In this dissertation also, a single hidden layer was selected. The issue of determining the optimal number of hidden nodes was a more complicated one. Networks with fewer hidden nodes are preferable as they usually have better generalization ability and less overfitting problems. But networks with too few hidden nodes may not have enough power to model and learn the data. The most common way of determining the optimal number of hidden nodes is by a sensitivity analysis. In this dissertation, 10 neurons in the hidden layer were found to be the optimum.

6.3.1.2 Number of Input Nodes

The number of input nodes corresponds to the number of variables in the input vector used for forecasting the future values. In the case of travel time prediction from loop detector data, either the travel time can be estimated from the detector data variables like speed flow or occupancy and then can be predicted to future time steps, or the detector data can be first predicted to future time steps and then the corresponding travel time can be calculated. The first method was adopted in this dissertation, since it gave better results in previous studies compared to the indirect method of predicting speed, flow or occupancy and then calculating the corresponding travel time (Kisgyorgy and Rilett 2002). A fixed number of lagged observations of the travel times from the same link were selected as the input variables as in any time series forecasting problems. Travel time information from the previous five time periods was selected as input, based on previous studies (Park and Rilett 1998, 1999). Data normalization was performed to standardize the data and to avoid computational problems. If the data are not normalized, inputs with higher values will drive the training process, masking the contribution of lower valued inputs (Desa 2001).

6.3.1.3 The Number of Output Nodes

The number of output nodes is relatively easy to specify as it is directly related to the problem under study. For a time series forecasting problem, the number of output nodes often corresponds to the forecasting horizon. The forecasting can be one-step-ahead or multi-step-ahead prediction. In this dissertation multi-step-ahead forecasting was adopted and prediction up to 30 time steps ahead were attempted to see how many time steps ahead the prediction performance is better than the historic method. There are two ways of performing multi-step-ahead forecasting. The first method is called the iterative forecasting method where the forecast values are used as input for the next forecast. In this case, only one output node is necessary. The second method, namely the direct method, is to let the neural network have several output nodes to directly forecast each step into the future. Zhang et al. (1998) reported that the direct method performed better than the iterative method whereas Weigend et al. (1992) reported that the direct method performed worse than the iterative method. An advantage of using the direct method is that the neural network can be built directly to forecast multi-step-ahead values. In the case of iterative method, only a single function is used to predict one point each time and then iterates this function on its own outputs to predict points in the future. As the forecast moves forward, past observations are dropped. Instead, forecasts are used to forecast further future points. Hence, it is typical that the longer the forecasting horizon, the less accurate the iterative method is (Zhang et al. 1998). In this dissertation the direct method was chosen. The normalized output values obtained from the ANN were transferred back to the actual values.

6.3.1.4 Interconnection of the Nodes

The network architecture is also characterized by the interconnection of the nodes in different layers. For most forecasting applications the networks are fully connected to all the nodes in the next higher layer and this approach was adopted in this dissertation.

6.3.1.5 Activation Function

Different activation functions such as sigmoid, logistic, hyperbolic, linear etc. have been used in previous studies. In this dissertation, a logistic sigmoid activation function, which makes the input and output spaces continuous, was used. Figure 6.7 shows the sigmoid activation function

with its mathematical form in Equation 6.17. This transfer function takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1.

$$f(n) = a = \frac{1}{1 + e^{-n}} \quad (6.17)$$

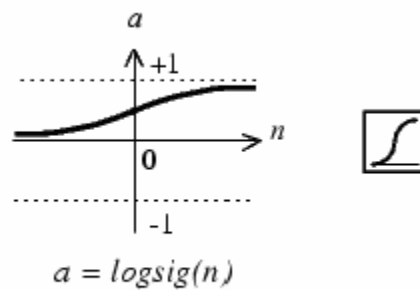


Fig. 6. 7. Log-sigmoid transfer function

6.3.1.6 Training Algorithm

As discussed previously, the ANN training is an unconstrained non-linear minimization problem where the weights are iteratively modified to minimize the overall error between desired output and actual output. The most popular algorithm for this is the back propagation algorithm, which requires the selection of a step size (learning rate). Small rates lead to a slow learning process whereas large rates will lead to oscillations around a global minimum. To improve this, a momentum parameter can be used, which selects the next weight change in more or less the same direction as the previous one and hence reduces the oscillation effect of larger learning rates. Standard back propagation with momentum is selected in most studies. The momentum parameter and learning rate are usually selected through trial and error. However, there is no consistent conclusion with regard to the best learning parameter combination.

Hence, more high performance algorithms that can converge from 10 to 100 times faster than the conventional back propagation methods were developed (MathWorks, Inc. 2003). These faster

algorithms fall into two main categories. The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm. One heuristic modification is the momentum technique. The second category of fast algorithms uses standard numerical optimization techniques. Conjugate gradient, quasi-Newton, and Levenberg-Marquardt (LM) are some of the examples that fall in this category. Their faster convergence, robustness and ability to find good local minima make them attractive in ANN training. In this dissertation the LM method was adopted. However, its use is restricted to small networks (less than a few hundred weights) with a single output layer (Statsoft Pacific Pty Ltd. 2004).

6.3.1.7 Training and Testing Data

The training sample is used to train the data and testing data to evaluate the forecasting ability of the model. The main point here is to have both the training and testing data representative of the population data. Most researchers select them based on the rule of 90% vs. 10%, 80% vs. 20%, 70% vs. 30% etc. In this dissertation, 80% vs. 20% was used for training and testing.

6.3.1.8 Performance Measures

Commonly adopted measures for checking the accuracy of the predicted data are the mean absolute error, sum of squared error, root mean squared error, mean absolute percentage error (MAPE) etc. In this dissertation MAPE was used as given in Equation 3.7.

6.3.2 SVM

The SVM toolbox for Matlab developed by Steve Gunn (2003) was used in the present study. The parameters to be chosen are the loss function (ϵ), cost function C and kernel function (Gunn 2003).

6.3.2.1 Loss Function, ϵ

The choice of loss function determines the approximation error achieved, the training time, and the complexity of the solutions; the last two depend directly on the number of support vectors. A given training example becomes a support vector only if the approximation error on that example

is larger than ϵ . Therefore, the number of support vectors is a decreasing function of ϵ . In practice one should ensure that the value of ϵ is sufficiently small so that the theoretical risk it defines constitutes a reasonable measure of the approximation error. A robust compromise suggested is the percentage of support vectors is equal to 50% (Mattera and Haykin 1999). A larger value of ϵ can be utilized to reduce the training time and the network complexity. Thus, the loss function determines the measure of accuracy of the result in the regression. Each choice of loss function will result in a different overall strategy for performing regression. A loss function that ignores errors that are within a certain distance of the true value is referred to as the ϵ -insensitive loss function (Cristianini and Shawe-Taylor 2000). In this dissertation the ϵ -insensitive loss function with an ϵ of 0.05 was selected.

6.3.2.2 Cost Function C

The choice of cost function involves a tradeoff between the minimization of the training error and the number of training points falling outside the error boundary. C defines the range of the values assumed by linear coefficients, and its choice affects the range of the possible output. If the range of output is $[0, B]$ and if C is very small compared to B , it would be impossible to obtain a good approximation. A value of C that is very large compared to B will lead to numerical instability. Therefore, a value of C that is approximately equal to B is suggested as a robust choice (Mattera and Haykin 1999). Cost function C signifies the tolerance to misclassification errors. If the value of C is high, the tolerance will be less (Talukder and Casasent 2001). In this dissertation a C of 100 was selected by trial and error.

6.3.2.3 Kernel Function

The kernel function implicitly maps the input vector into the feature space and calculates their inner product in the feature space. Any symmetric function such as linear spline, B-spline, sigmoidal, polynomial, radial basis function etc., can be used as a kernel function. In the present study, the SVR model used a radial basis kernel function. The parameter σ determines the width of the Gaussian radial bases. A σ value of 15 was selected based on a preliminary analysis.

6.4 RESULTS

The results are illustrated using the data collected from the I-35 test bed shown in Figures 3.10a and 3.10b. The ILD data from all 5 days from February 10 to February 14, 2003, are used. Travel time was predicted into future time steps using the historic method, real-time method, ANN method, and SVM method, and the results are compared. The analysis considered prediction times ranging from 2 minute ahead up to 1 hour ahead. Up to 4 day's data was used for training, and 1 day's data was left for cross validation and to evaluate the prediction errors. First, the 2-minute aggregated data were normalized based on the range of the travel time values. The input and output data were selected as the travel time for the five previous time step values and the travel time for the next time step value, respectively. Thus, for a 3-day data for training will have a training matrix of size 2155×5 and a testing matrix of size 2155×1 . Because the data was grouped in 2-minute intervals, five time steps correspond to a 10-minute interval. Thus, the prediction was based on the previous 10-minute travel time values. The model then predicts the next 2-minute travel time as shown in the following equation:

$$T(k+\Delta t) = f(T(k-4 \Delta t), T(k-3 \Delta t), T(k-2 \Delta t), T(k-\Delta t), T(k)) , \quad (6.18)$$

where,

Δt = time interval,

T = travel time, and

k = current time interval.

The prediction was subsequently carried out to 4 minutes, 6 minutes, etc., up to 1 hour ahead. The training data were varied from 1 day's data to four day's data, and testing was done for a separate day.

6.4.1 All Day Data from Link 1

The travel time from link 1 on all five days was analyzed first. The training was carried out based on data from February 10 to 13, 2003, (Monday to Thursday). The data from Friday, February 14 was kept for validation. Figure 6.8 shows the travel time distribution on all 5 days on link 1. It can be seen that the Tuesday, February 11, 2003, data is showing less magnitude throughout the day compared to all the other days. Also, it can be seen that in February 12, 2003 data the peak in the travel time is small compared to other days. All the other days shows similar travel time values. The MAD as given in Equation 5.27 was calculated between each day's data with Friday data. The MAD came to be 3.85, 7.85, 4.87, and 3.99 for Monday, Tuesday, Wednesday, and Thursday data, respectively.

First the historic method, which assumes that the historic average represents the future travel time, was used. The results obtained using a single day's data (Monday data alone) for prediction is shown in Figure 6.9. In this case, since only the Monday data were used, the historic value equals the Monday travel time. It can be seen that the patterns of travel time from these two days are very similar, with an MAD of 3.85, except for the magnitude at the peak period. The MAPE between the predicted travel time and the actual travel time was calculated for the 24-hour period and was 9.36%.

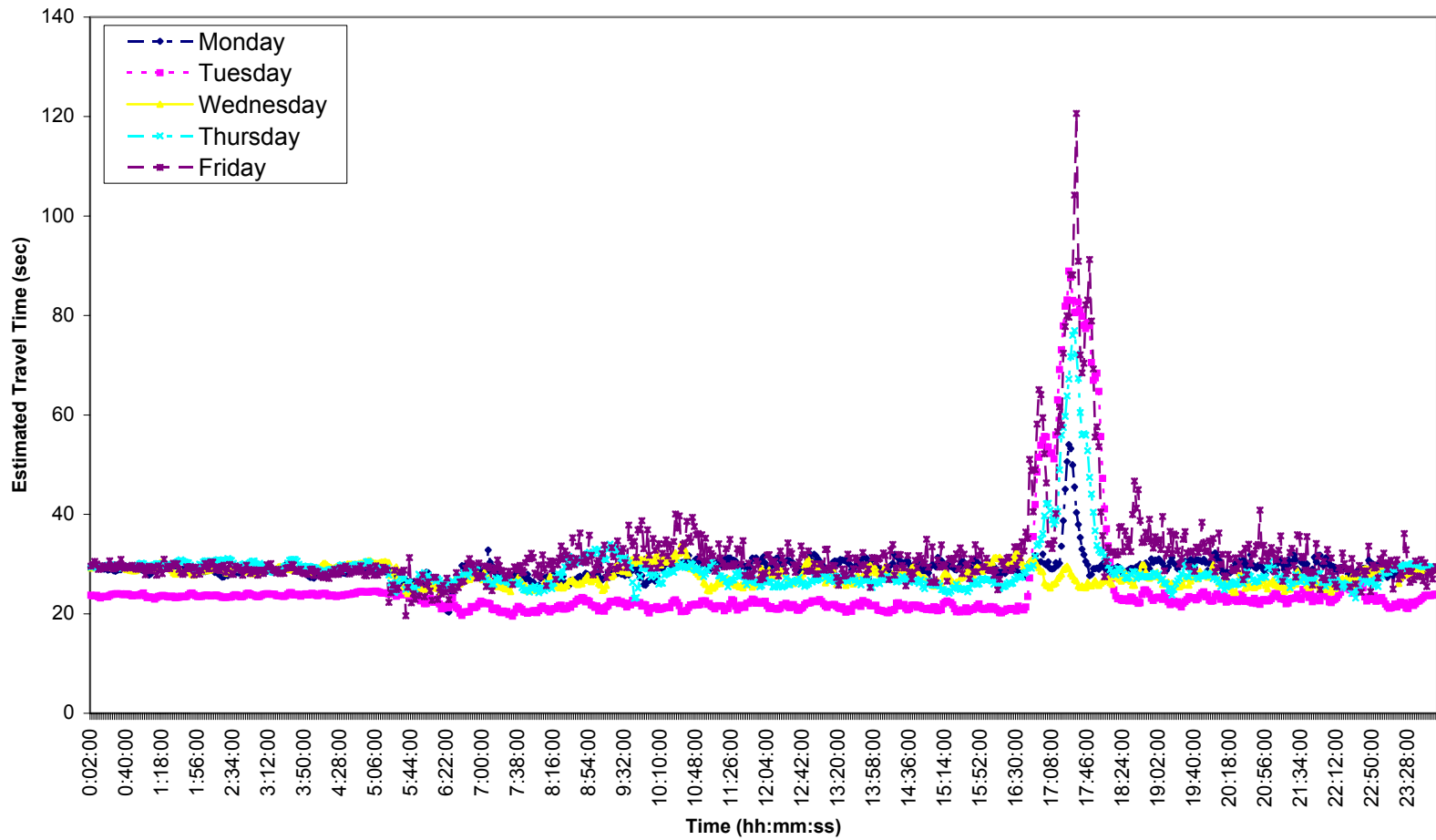


Fig. 6. 8. Travel time distribution on link 1 on all 5 days

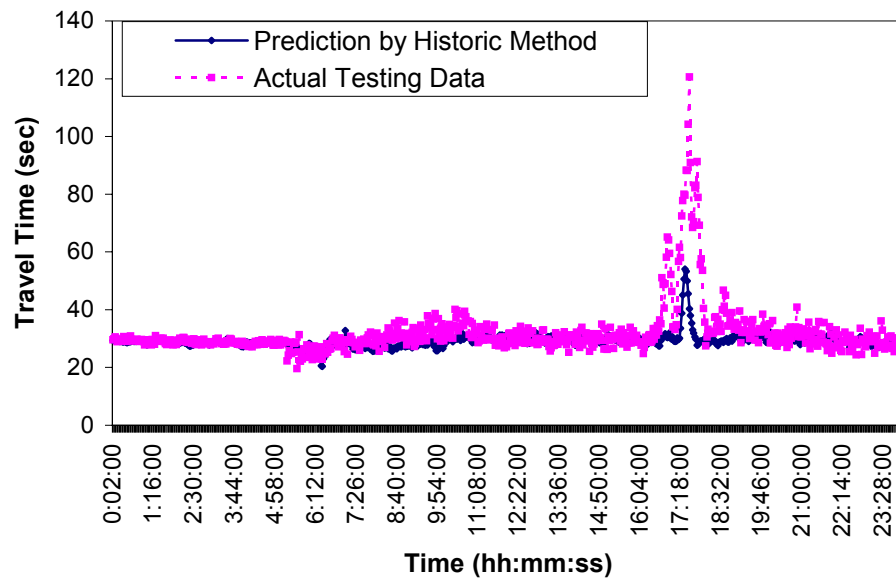


Fig. 6. 9. Travel time predicted by historic method for link 1 on February 10, 2003

Figure 6.10 shows the predicted travel time using the real-time method, which assumes that the current travel time is going to continue to the future time step using a single day's data for prediction as detailed in 2.4.1. As expected, the predicted travel time leads the actual travel time by the 2-minute prediction interval. The corresponding MAPE for the whole 24-hour period came to be 9.66 %.

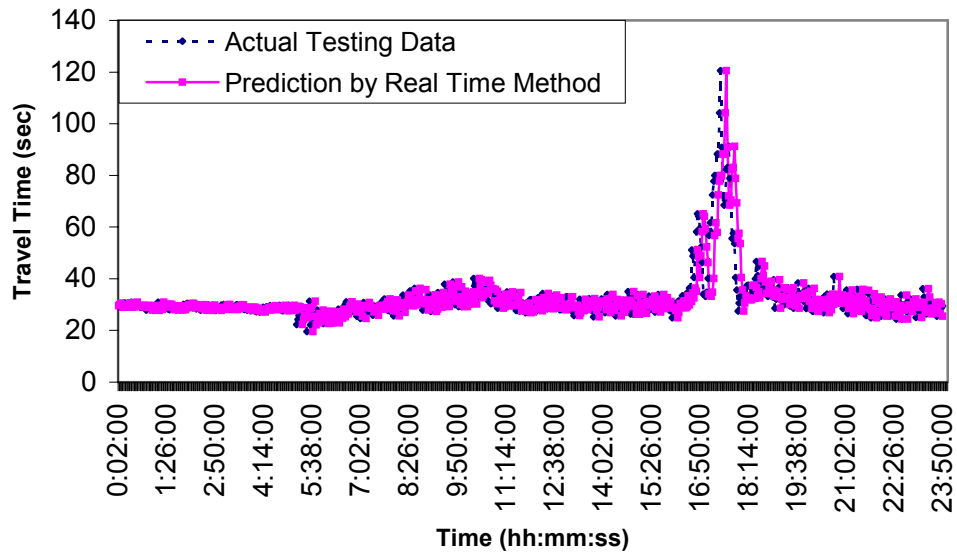


Fig. 6. 10. Travel time predicted by real-time method for link 1 on February 10, 2003

Figures 6.11 and 6.12 show the predicted travel time using ANN method and SVM method using a single day's data for training. It can be seen that the travel time predicted by both SVM and ANN were able to follow the trends in the actual data, with MAPE values of 8.64% and 7.38% respectively.

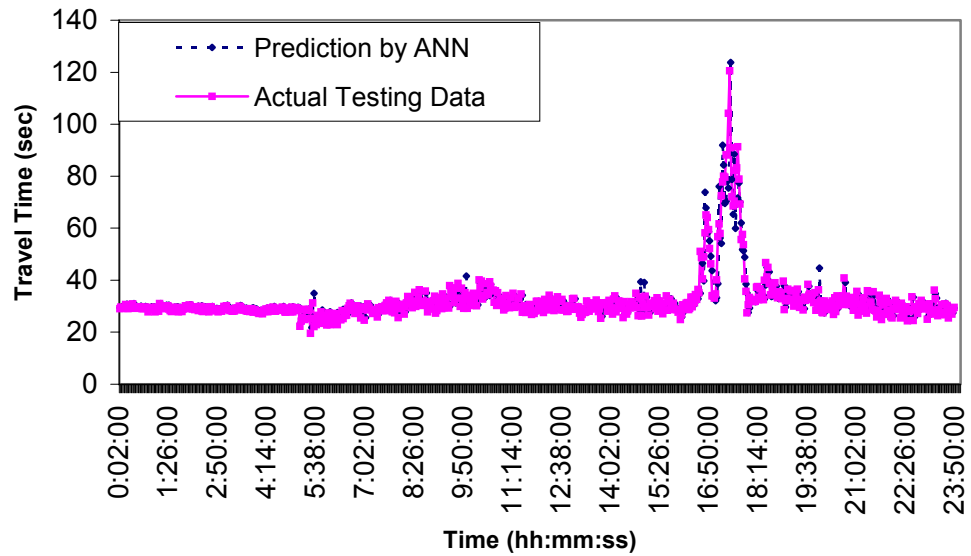


Fig. 6. 11. Travel time predicted by ANN method for link 1 on February 10, 2003

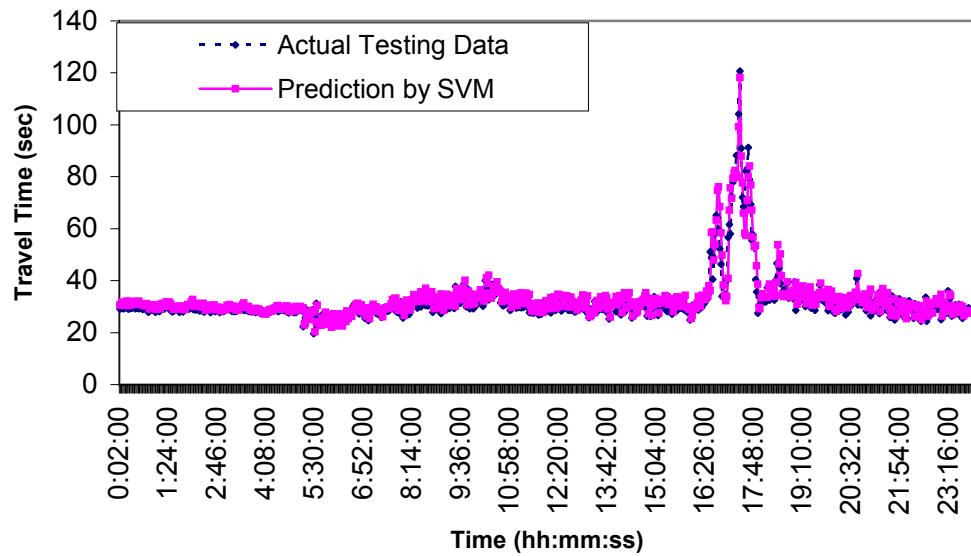


Fig. 6. 12. Travel time predicted by SVM method for link 1 on February 10, 2003

An enlarged view of the actual travel time values and the corresponding predicted values for a 2-hour evening peak and off-peak for a 2-minute ahead prediction using all the four methods is shown in Figure 6.13. The training data used in this particular example is from Monday data alone. This figure clearly illustrates the historic method with Monday data alone for training performing very poorly for the prediction of peak period. Also it can be seen that in the case of the real-time method the predicted travel time leads the actual travel time by the 2-minutes prediction interval. The SVM and ANN followed the trends in the actual travel time. The MAPE for this 2 hour prediction was calculated separately and were 26.74%, 15.90%, 12.18%, and 11.35% for the historic method, real-time method, ANN, and SVM respectively.

To illustrate the technique further, the travel time prediction was extended to 4-minutes ahead, 6-minutes ahead etc., up to 1 hour into the future for the Friday data. The prediction was carried out for the full 24-hour data. The performance measure used was Mean Absolute Percentage Error (MAPE). This was calculated based on the difference between the predicted travel time by each of the methods and the actual travel time of Friday for the 24-hour period.

Figure 6.14 shows the error in prediction when 1 day's data (Monday) was used for training the network and Friday travel time was predicted. MAPE values are shown from 2-minutes ahead up to 1 hour ahead prediction. The MAPE for the historic method, the real-time method, the ANN, and SVM methods are shown in this figure. It can be seen that the historic method outperformed the real-time method throughout the prediction. SVM performed better than historic only up to 6 minutes of prediction and ANN performed better up to 10 minutes of prediction ahead. Thus, it can be seen that historic method outperformed the other methods in this case after 10 minutes of prediction time ahead, which can be explained based on Figure 6.8. As it can be seen in Figure 6.8, both the training data (Monday) and testing data (Friday) had similar pattern with an MAD of 3.85. It can also be observed that ANN performed better than the SVM in this case.

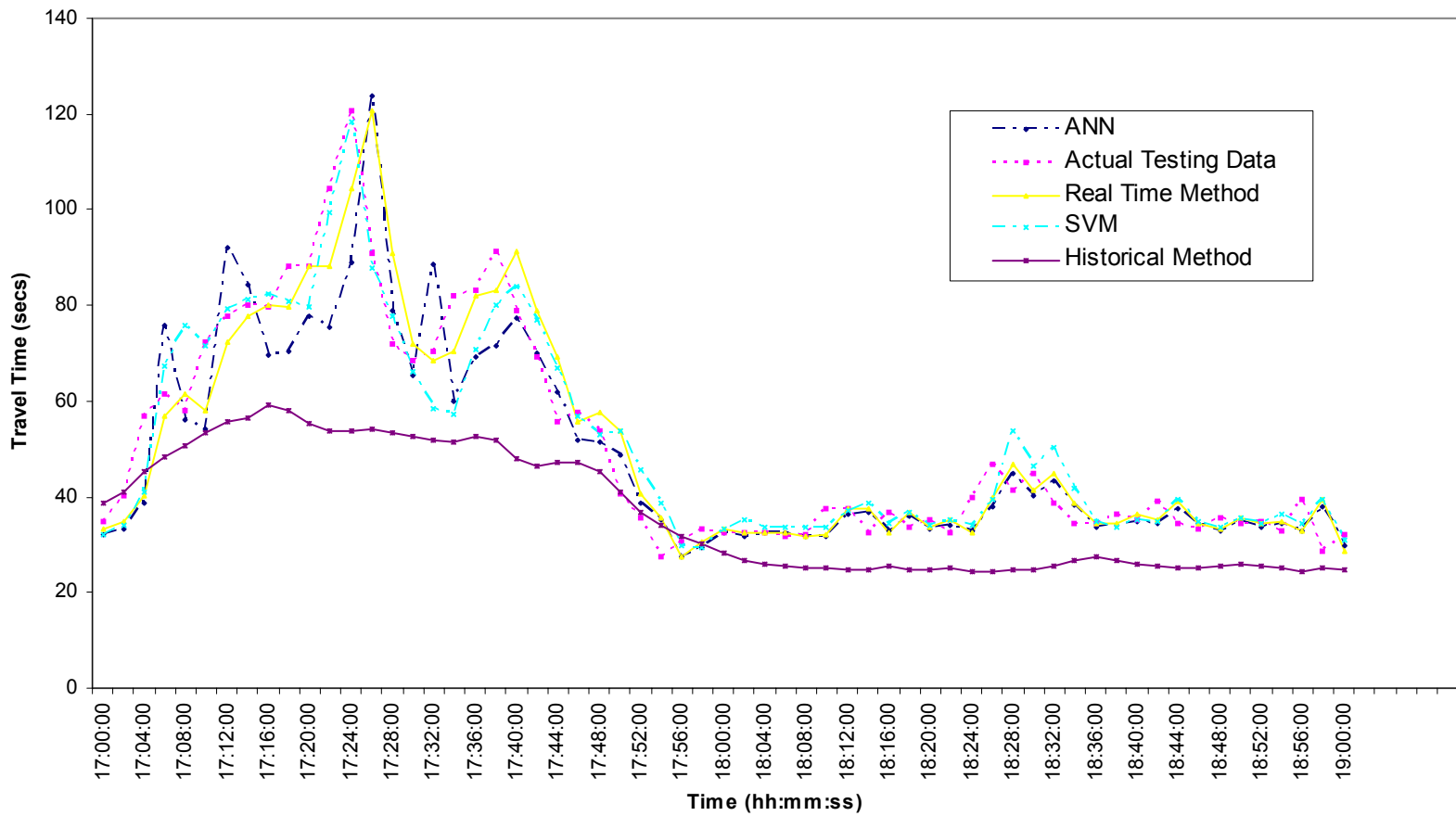


Fig. 6. 13. Comparison of the predicted values with 1 day training data for link 1 on February 10, 2003

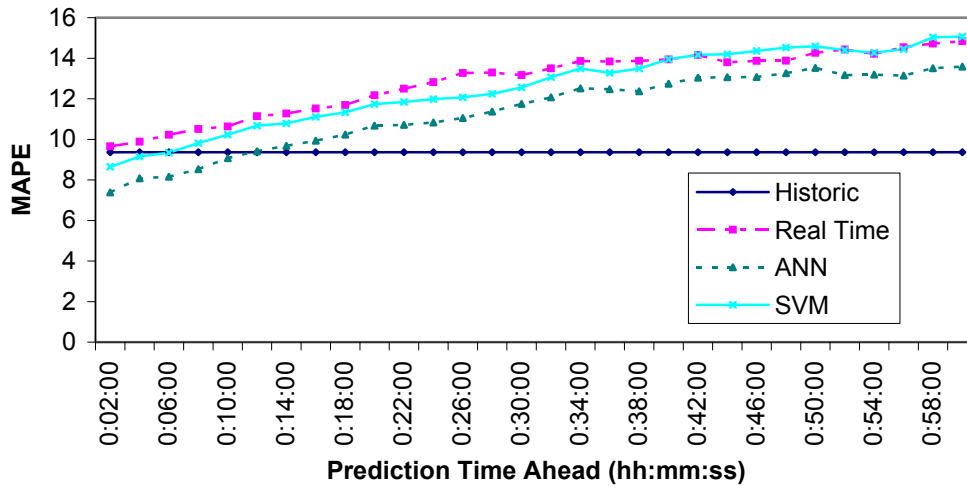


Fig. 6. 14. MAPE for prediction using 1 day's data for training

Figure 6.15 shows the MAPE values when 2 day's data were used for training (Monday and Tuesday) and when the 24-hour Friday data was predicted.

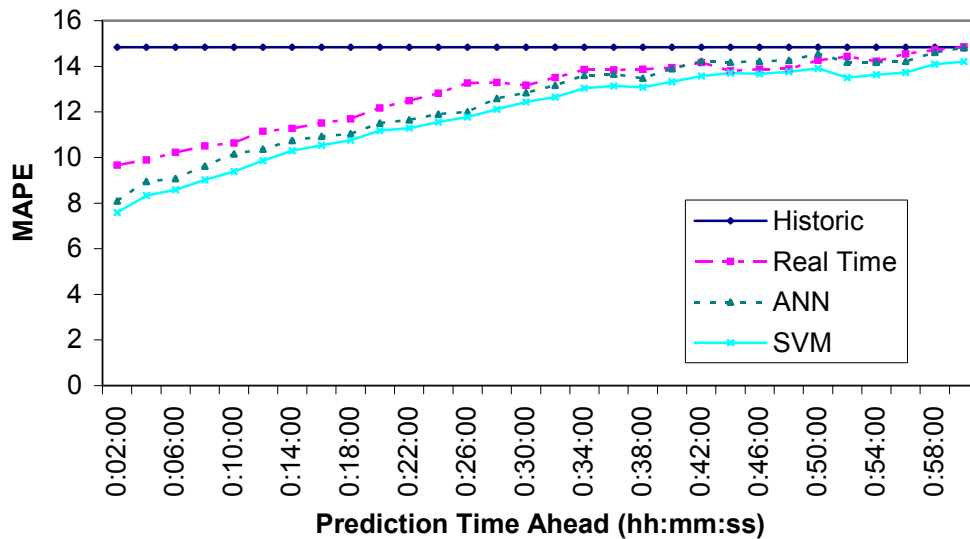


Fig. 6. 15. MAPE for prediction using 2 day's data for training

Comparing Figure 6.14 with Figure 6.15, it is seen that there is an increase in the prediction error using the historic method from 9.3% to 14.8% when the training data were changed from Monday data alone to Monday and Tuesday data together. This is due to the fact that the Tuesday travel time data differed in magnitude when compared to Monday and Friday data and this is illustrated in Figure 6.16. Figure 6.16 shows the travel time values on Monday, Tuesday and Friday for a 5-hour period from 11:00:00 to 16:00:00. It can be seen that the Monday and Friday data have very similar trends throughout. The MAD of 3.85 between Monday and Friday data opposed to the MAD of 7.84 between Tuesday and Friday data for the 24-hour period also illustrate this fact.

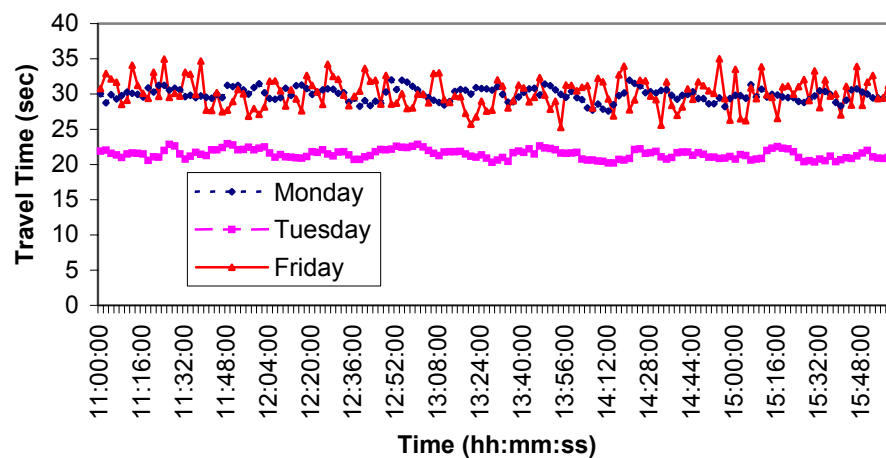


Fig. 6.16. Travel time pattern of Monday, Tuesday, and Friday

This difference of Tuesday data makes the training data different from testing data, reducing the performance of the historic method. It can be seen that this reduced the performance of ANN also. The SVM method out-performed all other methods in this case.

Figure 6.17 shows similar results when 3 day's data were used for training (Monday, Tuesday, and Wednesday) and the Friday data was predicted. It can be seen that with more data being

added to the training set, the effect of Tuesday data is declining. As in the previous case, here also the SVM performed better than all the other methods. Up to 30 minutes of prediction ahead, the other methods performed better than the historic method.

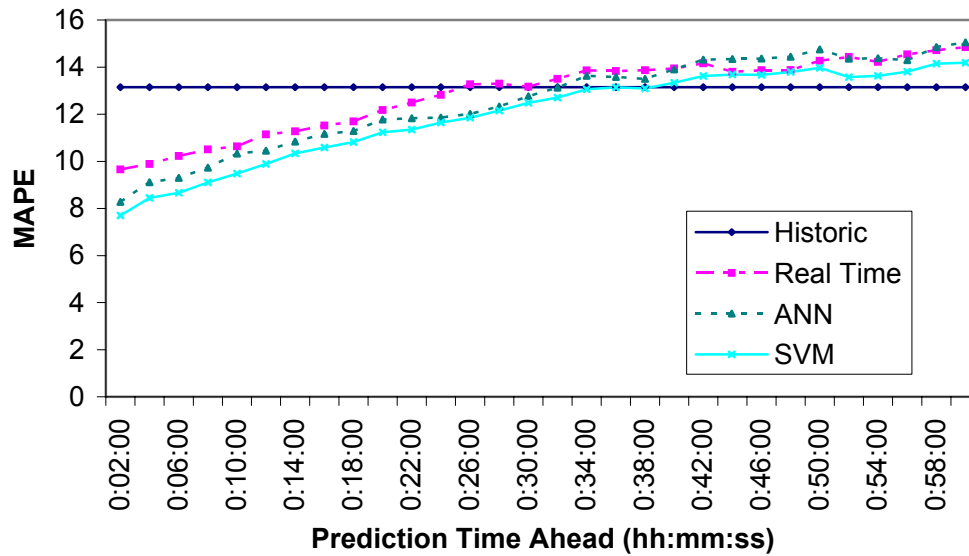


Fig. 6.17. MAPE for prediction using 3 day's data for training

Figure 6.18 shows similar result when 4 day's data were used for training (Monday, Tuesday, Wednesday, and Thursday) and the Friday data was predicted. Here also for approximately up to 30 minutes of prediction time, the other methods performed better than the historic method.

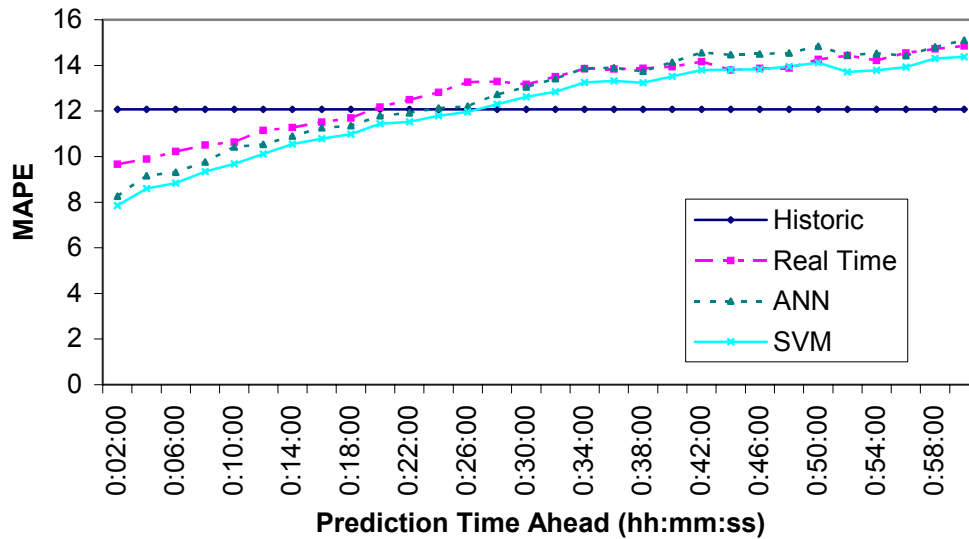


Fig. 6. 18. MAPE for prediction using 4 day's data for training

As more and more data are added to the training set, the influence of the Tuesday data declines and this is reflected in the reduction in error for the historic and ANN methods. Comparison of the performance of ANN with SVM for prediction using 4 day's data for training shows a slight advantage to SVM. This performance of SVM can be explained based on the inherent nature of the SVM training process. Once SVM chooses the data points, which can represent the input data (support vectors), its performance is more or less independent of the amount of training

data. Hence, if the support vectors selected from the training data are not affected, SVM's performance may not get affected by the amount of training data. However, in the case of ANN, the network can learn more about the data as the amount of training data increases, and this changes the results for the better up to an extent.

Overall, the performance of both SVM and ANN were comparable to each other. The historic method is a better choice when the training data and the testing data have the same magnitude as well as the same pattern. SVM becomes a better choice for the short-term prediction of travel time if the training data have more variations compared to the testing data. Also, it was found that the influence of the amount of training data used is greater on the ANN method than on the SVM method. To check the validity of these conclusions, travel time prediction was carried out for link 2 where all days travel time had similar trends.

6.4.2 All Day Data from Link 2

Data from February 10 to February 14, 2003 were used for link 2 also. The travel time from all five days for link 2 is shown in Figure 6.19, and it is seen that all days have similar trends in the data except Wednesday, where the peak was relatively low. The MAD between Friday data and Monday, Tuesday, Wednesday and Thursday data were 4.2, 4.2, 5.7, and 4.2, respectively.

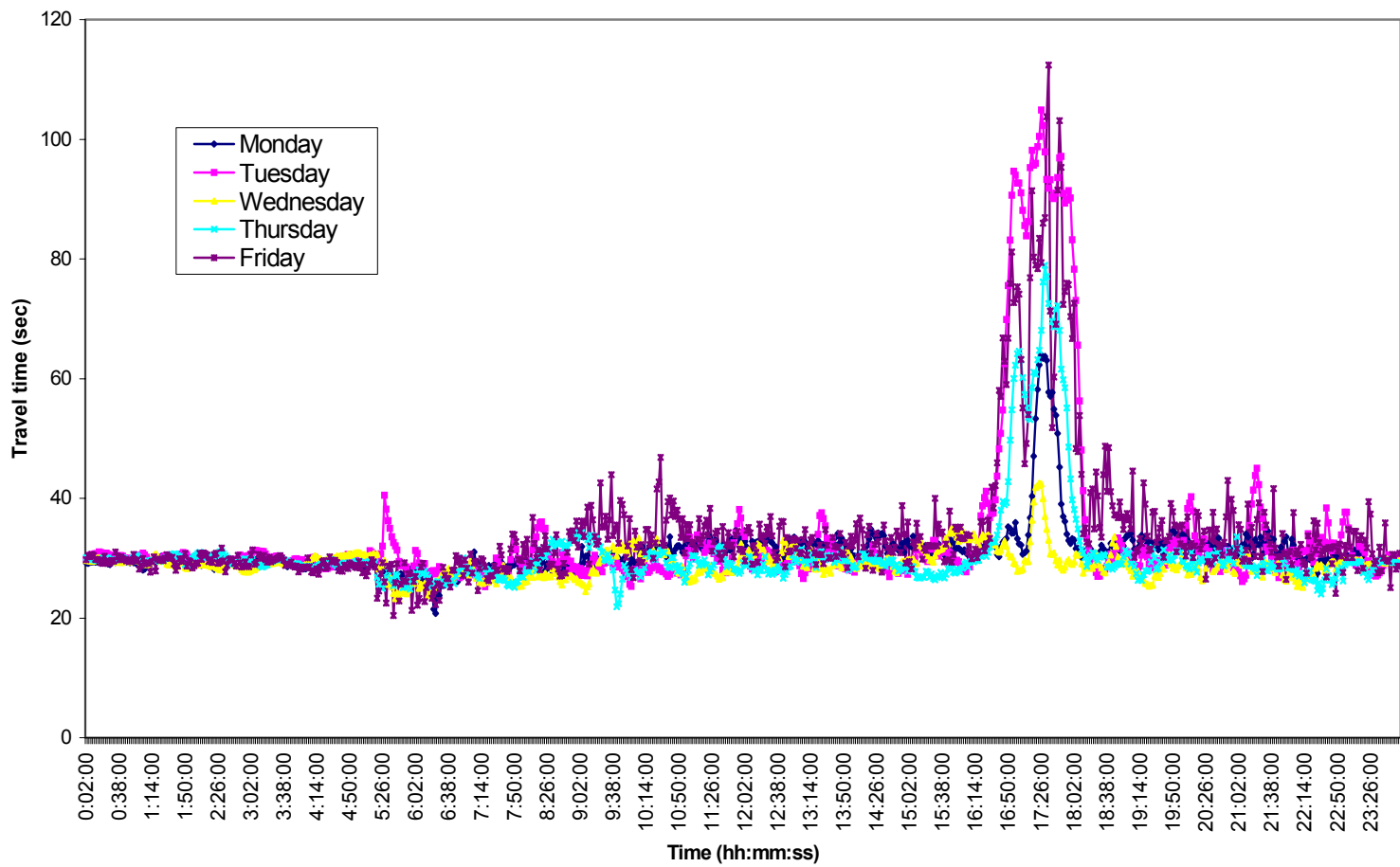


Fig. 6. 19. Travel time distribution for link 2 from February 10 to February 14, 2003

The prediction interval was varied in this case also from 2 minutes ahead up to 1 hour ahead, and the MAPE at each time step was calculated as in the case of link 1. Figure 6.20 shows the MAPE values for the prediction of 24-hour Friday data when only 1 day's data were used for training. As expected, the error from the historic method is very small, and the performance of ANN and SVM are similar, with a slight advantage to ANN.

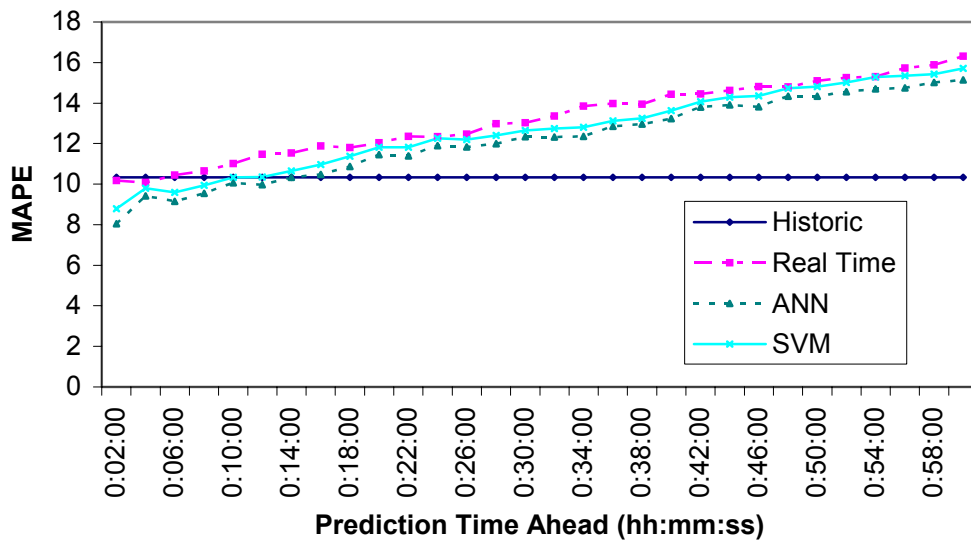


Fig. 6.20. MAPE for prediction using 1 day's data for training

Figure 6.21 shows the MAPE values when 2 day's data were used for training. As can be seen from Figure 6.19, Tuesday data also represent the Friday data (testing data) very well with MAD of 4.2. Hence the prediction result remains same as in Figure 6.20, with historic method performing better than the other methods after 10 minutes of prediction ahead. Also ANN outperforms SVM in this case also.

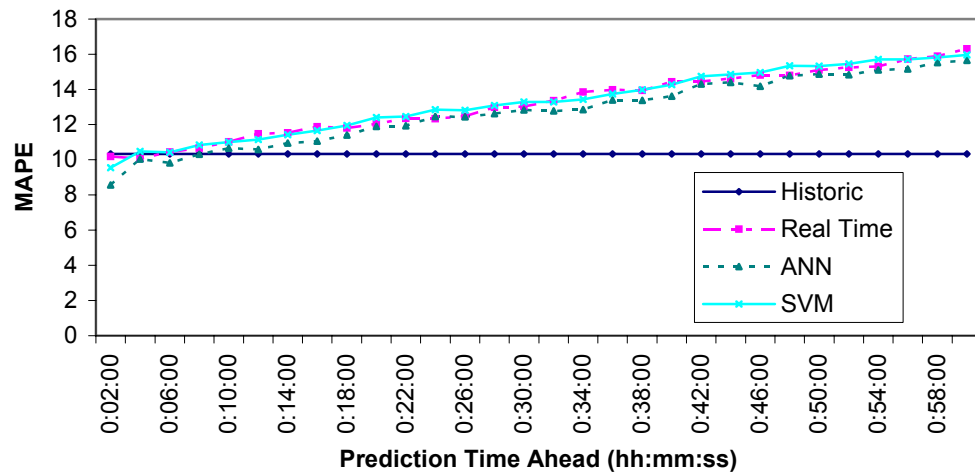


Fig. 6. 21. MAPE for prediction using 2 day's data for training

Figure 6.22 shows the MAPE values when 3 day's data were used for training. Here it can be seen that SVM had a slight advantage over ANN with the MAPE values being smaller than that of ANN. This may be due to the small variation in the Wednesday data from other days.

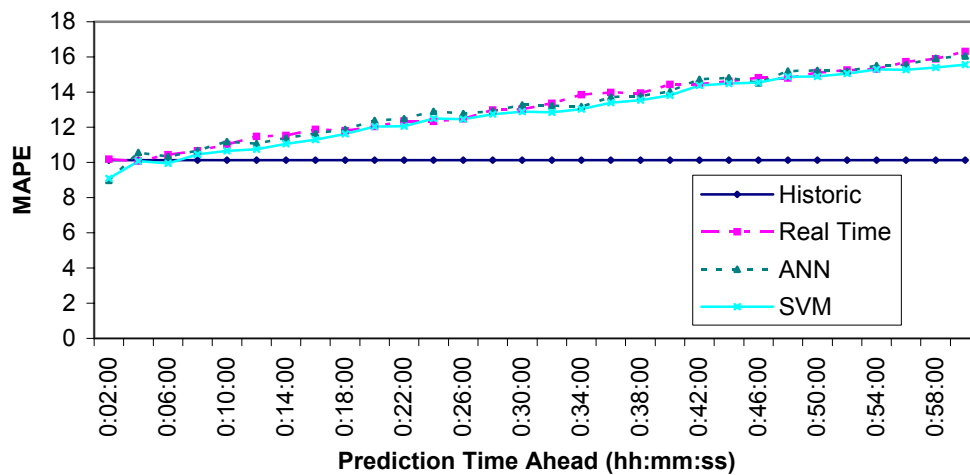


Fig. 6. 22. MAPE for prediction using 3 day's data for training

Figure 6.23 shows the MAPE values when 4 day's data were used for training. As expected, when the travel time data do not have much variation, historic and real-time method are able to predict the future conditions well. The performance of ANN and SVM are comparable, with ANN being slightly better in the case where the data did not have much variation.

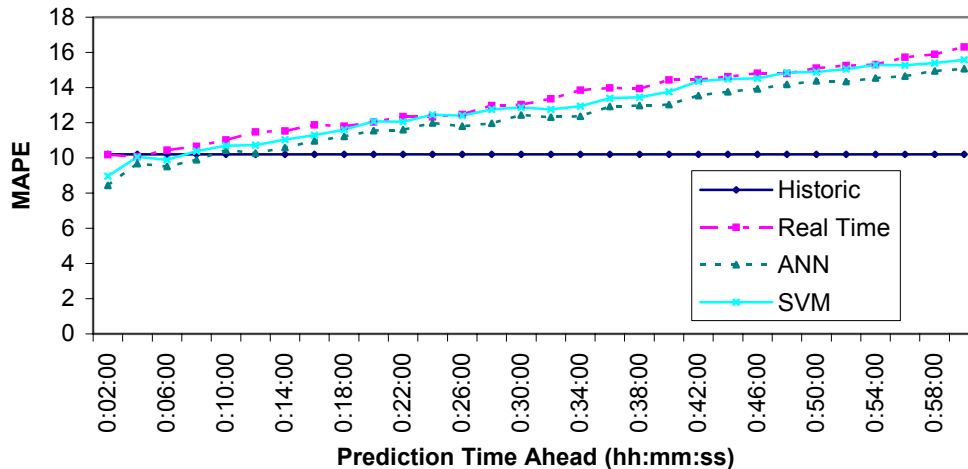


Fig. 6. 23. MAPE for prediction using 4 day's data for training

It should also be noted that both ANN and SVM performed better than the real-time and historic methods when the testing data had variations from the training data as indicated by a bigger magnitude of MAD between the training and testing data. From the results obtained for the two links described above, one can see that SVM has a better predictive capability when the training data has lot of variations. As discussed earlier, the accuracy of the SVM prediction does not depend on the amount of data used once the support vectors are selected. Hence, in scenarios where the training data has variations (as in the example of link 1) and the availability of data is limited, SVM will be a better choice than ANN. On the other hand, in cases where large amounts of data are available and the training and testing data have similar trends, ANN is a better predictive algorithm. Links 3 and 4 had very similar trends in the travel time values for all 5 days, similar to link 2, and hence the results obtained are not repeated here.

6.4.3 Speed Prediction

The conclusions drawn from the results of travel time prediction need to be checked to find out whether they are data specific. To ensure that the above conclusions relate as well to other traffic parameters, investigation related to the prediction of speed was also carried out. Field data from detector number 159.998, as shown in Figure 3.10, are analyzed from August 4 to 8, 2003. The speed distribution for the 5 days is shown in Figure 6.24.

From Figure 6.24 it can be seen that the speed data pertaining to all days except Monday have similar trends. Monday data do not show an evening peak but show a morning peak. The MAD was calculated for each of the days with the Friday data and was 3.5, 2.9, 3.3, and 2.4, respectively for the 24-hour period. This one week data was used for predicting speed on Friday. The MAPE obtained is plotted in Figure 6.25 when the Monday data alone are used for training the network.

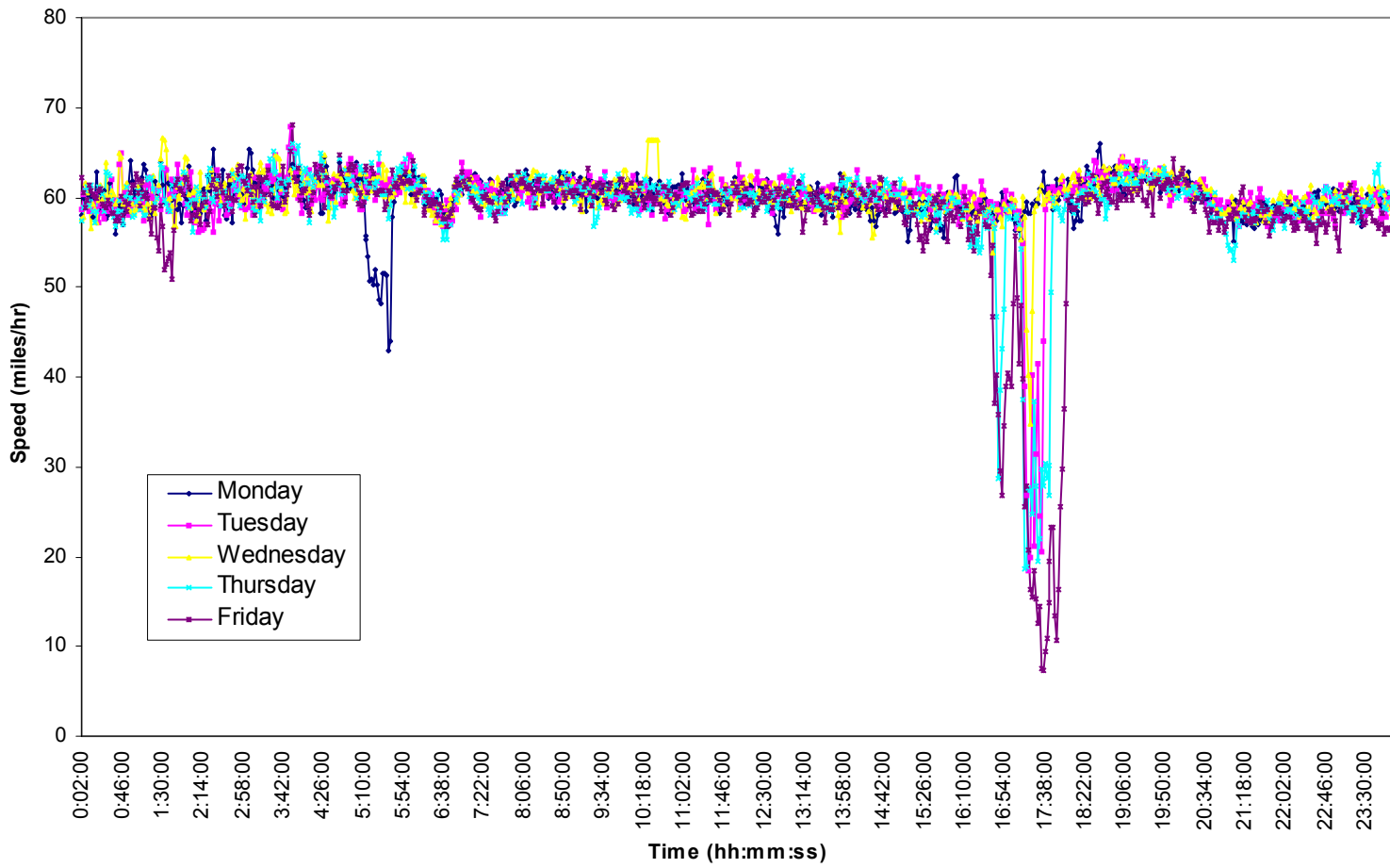


Fig. 6. 24. Speed distribution at 159.998 for 1 week from August 4 to 8, 2003

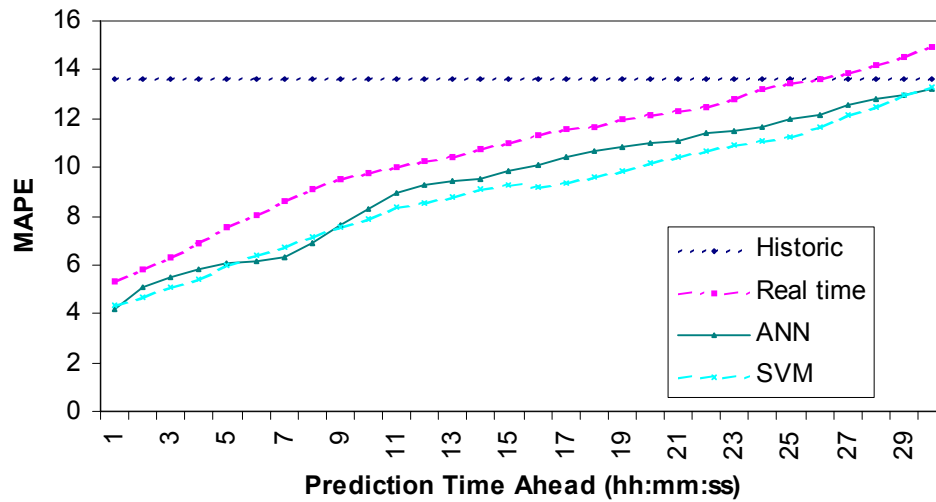


Fig. 6. 25. Performance comparison using 1 day's data for training

It can be seen that SVM performs better than all the other methods in this case. As discussed previously, the training data used was the one that was having the maximum difference from the testing data. Figure 6.26 show the MAPE when 2 day's data were used for training.

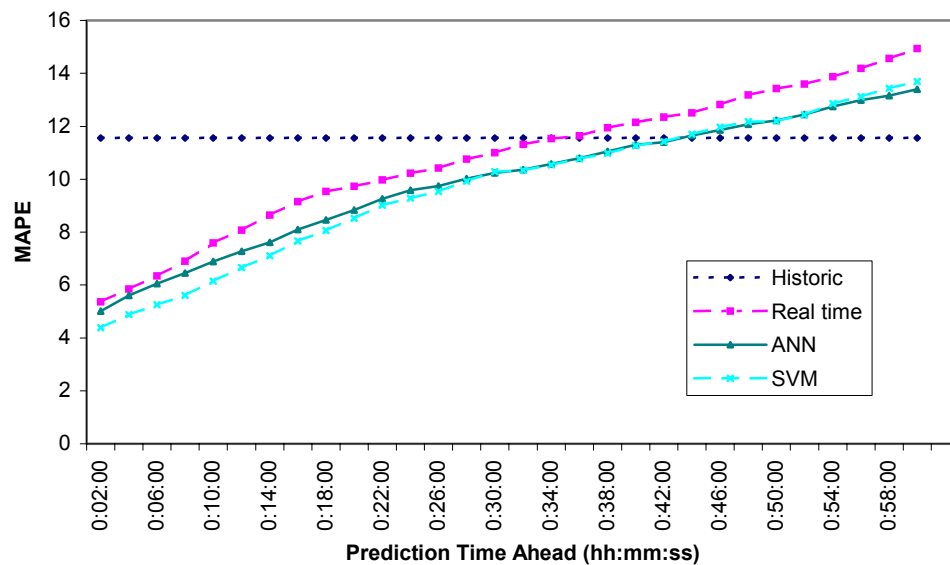


Fig. 6. 26. Performance comparison with 2 day's training data

It can be noted that Tuesday data were in agreement with the Friday data, and hence the effect of variation declines. The results obtained with 3 day's data for prediction are shown in Figure 6.27.

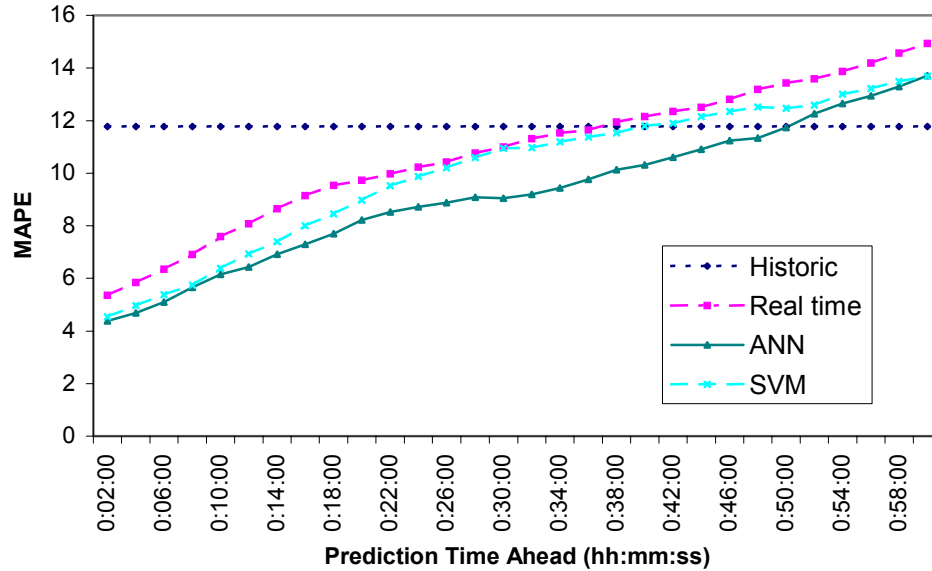


Fig. 6. 27. Performance comparison with 3 day's data for training

Figure 6.27 shows that ANN started performing better than all the other methods as the quality and quantity of the training data are increased. The errors of historic method as well as the ANN method declines as more data are included in the training set, which reduces the variation from the testing data. Results obtained from 4 day's of data for training are shown in Figure 6.28.

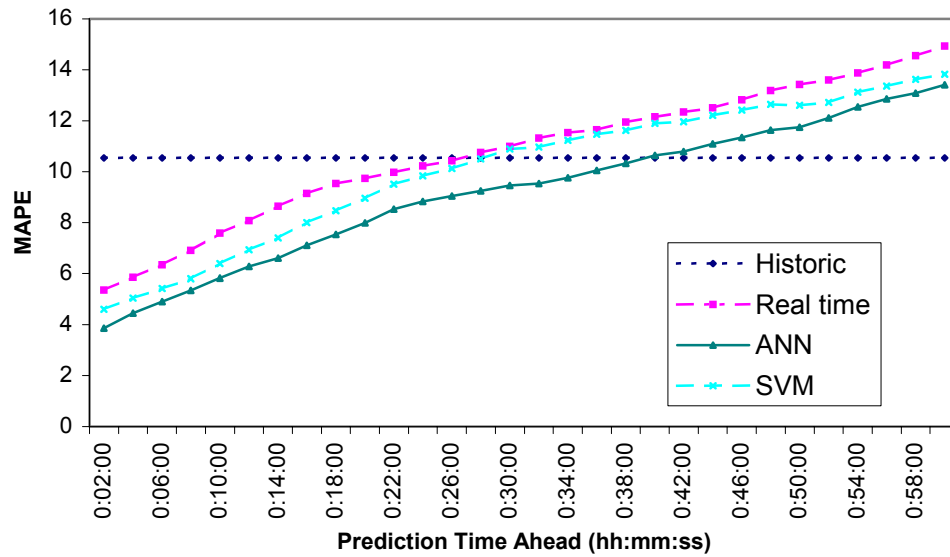


Fig. 6. 28. Performance comparison with 4 day's data for training

From the above figures (Figures 6.25 to 6.28), the conclusions drawn from the travel time estimation have been confirmed. The results obtained confirmed that ANN and SVM are powerful tools with performance better than the real-time or historic methods under varying traffic conditions. Also, it was found that SVM is a powerful tool for the prediction of traffic parameters with the performance comparable to ANN under most of the situations. When the training data were non-representative of test data, SVM outperformed ANN, showing that it can be considered as a viable alternative to ANN under situations with a lesser quantity of quality training data.

6.5 CONCLUDING REMARKS

This chapter presented a comparison of the performance of two machine-learning techniques, namely, ANN and SVM for the short-term prediction of travel time. The ANN model used is a multi-layer feed forward neural network and the SVM model used was a support vector regression with a radial basis kernel function. The analysis considered forecasts ranging from 2 minutes ahead up to 1 hour into the future. One day's data was left for crossvalidation to

evaluate the prediction errors. The training data were varied from 1 day's data to 4 day's data. The results were compared with historic and real-time approach results.

Results of this comparison indicate that the explanatory power of SVR is comparable to ANN. Also, SVR performed better than ANN when the training data had more variations. To check whether the results were data specific, speed predictions were also carried out using the field data. Based on the investigation conducted in this dissertation, it was found that SVR is a viable alternative to ANN for short-term prediction, especially when the training data have variations and the amount of training data is less. The performance of ANN depends largely on the amount of data available for training the network. Thus, in situations where there is less available data, there is a need for an alternative method for prediction. Due to the characteristic nature of the SVM method, the performance of SVM is almost independent of the number of data available, once the network chooses the support vectors. Hence, SVM can be used for the prediction of traffic parameters when the amount of data available for training is less. In cases where the training data is more, the performance of SVM is comparable to ANN, thus making it as an alternative option for prediction problems.

Comparison of the overall performance showed ANN and SVM outperforming the traditional methods, namely real-time and historic methods, especially under varying traffic flow conditions. However, for long-range predictions, the use of historic data proved to be more useful. The study also showed that current traffic conditions are good predictors while the traffic conditions are not having variations. The ANN and SVM methods performed well for some range into future. Also, both these methods have good dynamic response and show better performance compared to the traditional models. The training of both SVM and ANN may not make them attractive for online applications. However, both of them can be trained offline, and then used for on-line prediction. Once the networks are trained and the network parameters are stored off-line, the system can be used for online-applications, where the travel time corresponding to the incoming data needs to be predicted quickly.

As discussed earlier in this chapter, to the knowledge of this author there have been very few studies that explored the use of SVM in transportation applications and there have been none that used SVM for the prediction of traffic variables. A lot more work is needed to exploit the

explanatory power of this powerful tool to the fullest. Also, more work is needed to explore the effect of each of the different parameters of SVM, such as kernel function and cost function on the prediction performance.

As explained already, the present study used the SVM toolbox developed by Steve Gunn (2003) for MATLAB. The running time required by this toolbox was relatively high, taking two hours for training 1 day's data and going up to 4 day's for training a 4-day data set. The running time of the corresponding ANN model was in the order of 5 to 10 minutes. This may be due to the fact that the toolbox for SVM may not be using the best optimization technique. The performance of this toolbox is yet to be tested by MATLAB, and clearly some optimization techniques similar to that used by ANN are needed to increase the computational efficiency in the MATLAB environment. On the other hand, the ANN toolbox used in this dissertation is developed and distributed as part of the MATLAB package, which is standardized and optimized for fast and optimum performance. Being a new technique, SVM is yet to be explored fully to get the best performance in terms of training time. Since the aim of this work was to investigate the potential of SVM for the prediction of travel time, these issues are clearly out of scope of this dissertation and hence not considered.

CHAPTER VII

SUMMARY AND CONCLUSIONS

7.1 SUMMARY

The problem statement of this dissertation identified three main needs: 1) the need to perform data quality control of loop detector data at system level by analyzing the detectors as a series; 2) the need to estimate travel time from loop detector data under varying traffic flow conditions; and 3) the need to predict travel time to future time steps in an accurate way. A summary of how each of these problems is addressed in this dissertation and the conclusions reached with recommendations for further researches are provided in the following subsections.

Overall, this dissertation developed a comprehensive automated technique that is comprised of different techniques at each individual stage, to predict travel time from the ILD data collected from the field. The first step in this multi-step analysis was to carry out quality control of the ILD data. Since in this dissertation the detectors were analyzed as a series, in addition to the usual tests for checking the data discrepancies, quality control tests using the constraints based on conservation of vehicles was also carried out. A non-linear constrained optimization technique was adopted for correcting the discrepancy when there was a violation of the conservation of vehicles. After correcting the discrepancies, the data were used for the estimation of travel time. A methodology based on traffic flow theory was developed for the estimation of travel time from ILD data. Finally the travel time was predicted to the future time steps using the techniques, support vector machines, and artificial neural networks. Each of these steps is briefly detailed below.

7.1.1 Data Reduction and Quality Control

Traditionally gross errors in loop detector data are identified using threshold checking on the speed, volume, or occupancy observations, either individually or in combination. All of these tests analyze and correct data at individual locations and therefore cannot account for systematic problems over a series of detectors. While substantial failures in loop detector data are easily identified using these existing methodologies; some other failures such as biases in volume

counts may go unnoticed, which can be identified if the detectors are analyzed as a series. Also, for an application like estimation of travel time in a link, as in this present study, data from consecutive ILDs need to be considered. In such cases when the detectors are analyzed as a series, it is necessary to check the accuracy of the data based on the conservation of vehicles, in addition to the individual location checks, since it is a basic condition that the data as a series must follow.

Even though the violation of conservation of vehicles principle is a common problem with detector data, this requirement has received little attention. Common applications of ILD data such as incident detection may not get affected by this type of error and that may be a reason for ignoring these errors in the earlier studies. However, if the loop detector data are to be successfully used for applications such as O-D estimation or travel time estimation, these issues of system data quality need to be addressed. As discussed in the literature review in Chapter II, very few studies have been reported which systematically analyzed a series of detector locations for a long interval of time to check whether the collected data follow the conservation of vehicles. Most of those studies, when faced with a violation of conservation of vehicles, suggested applying adjustment factors to rectify it, rather than applying any systematic methodology.

In this dissertation the conservation of vehicles is checked, by comparing the cumulative flow curves from consecutive detector stations. One week's loop detector data from February 10 to 14, 2003, from the I-35 freeways of San Antonio was used. Systematic examination of the data revealed that the conservation of vehicles principle was violated on many days. This may be due to systematic errors such as some detectors under- or over-counting the vehicles.

This dissertation used a constrained non-linear optimization approach for systematically identifying and correcting loop detector data obtained from the field, in situations where the data violated the conservation of vehicles principle. The generalized reduced gradient method is adopted with the objective function and constraints selected in such a way that the result will follow the conservation of vehicles principle with least change of the original data. The objective function was chosen to minimize the error from violation of conservation of vehicles principle and the constraints were selected to keep the difference between the entry-exit observations

within the allowable maximum. Simulated data using CORSIM simulation software were used for validating the methodology. This method of correcting the loop detector data is more useful and convenient than the application of volume adjustment factors, when dealing with large amount of data for a longer duration and having large discrepancies. Also, the optimization technique proved to be very useful for imputing missing data as well as to prioritize the detector stations for maintenance as illustrated in Chapter IV. This dissertation represents the first application of this kind of an optimization technique for quality control of the freeway ILD data.

7.1.2 Estimation of Travel Time

The ILD data corrected using the optimization procedure can be used as input for the next stage, which is the estimation of travel time. There are different methods available for the estimation of travel time from loop detector data, the most popular among them being the extrapolation of the point speed values. However, the accuracy of the speed-based methods declines as the flow becomes larger. Other methods available are statistical and traffic flow theory based models, the majority of which are developed for either the free-flow condition or the congested-flow condition.

This dissertation presented several modifications to an existing traffic flow theory based model for travel time estimation on freeways, such that the model can estimate travel time for varying traffic flow conditions directly from the loop detector data. The approach was designed for analyzing ILD data for longer intervals of time under varying traffic flow conditions. The input used includes speed, flow, and occupancy obtained from field and the travel time estimation is based on the area between the cumulative flow curves at entry and exit. Simulated data using CORSIM simulation software were used for validating the results. After the validation, the model was used for estimating travel time from field data. The travel time estimated was compared with the AVI data collected from the field. The model result was also compared to the results obtained from different available methods such as extrapolation method. The results showed the developed model performing better under varying traffic flow conditions.

7.1.3 Prediction of Travel Time

Real-time information on current travel time can be useful to drivers in making their route decisions if the traffic conditions are stable without much fluctuation. However, there are fluctuation in traffic resulting in a substantial difference between the current link travel time and the travel time on the link when traversed after a short time. Hence, accurate predictions are more beneficial than current travel time information.

The present work introduced the application of a recently developed pattern classification and regression technique called support vector machines (SVM) for travel time prediction. An Artificial Neural Network (ANN) method was also developed in this dissertation for comparison. It is also aimed at comparing and contrasting the performance of SVM, ANN, historic, and real-time methods. Up to 4 days data were used for training the networks and 1 day's data was left for crossvalidation. The data used were the estimated travel time obtained from the model described in the previous section.

The ANN model used was a multi-layer feed forward neural network and the SVM model used was a support vector regression with radial basis kernel function. The analysis considered forecasts ranging from 2 minutes ahead up to 1 hour into the future. The training data were varied from 1 day's data to 4 day's data. The results were compared with historic and real-time approaches.

Results of this comparison indicated that the explanatory power of SVR is comparable to ANN. Also, SVR performed better than ANN when the training data is having more variations. To check whether the results are data specific, speed predictions were also carried out using the field data. Based on the investigation conducted in this dissertation, it was found that SVR is a viable alternative to ANN for short-term prediction, especially when the training data are not a good representative sample and when the amount of training data is less. In cases where enough training data were available, the performance of SVM was comparable to ANN. Overall, it was found that SVR is a good alternative option for prediction of traffic variables such as travel time.

The study also showed that current traffic conditions are good predictors, while long-range predictions need the use of historical data. The ANN and SVM methods performed well for some ranges into the future. Also, both these methods have good dynamic response and show better performance compared to the traditional models. The training of both SVM and ANN may not make them attractive for on-line applications. However, both of them can be trained off-line, and then used for on-line prediction. Once the networks are trained and the network parameters are stored off-line, the system can be used for online applications, where the travel time corresponding to the incoming data needs to be predicted quickly.

As discussed in Chapter VI, to the knowledge of this author there have been very few studies that explored the use of SVM in transportation applications and there have been none which used SVM for the prediction of traffic variables. Thus, this dissertation is the first attempt to use SVM technique for the prediction of vehicle travel time.

7.2 CONCLUSIONS

This dissertation resulted in a number of conclusions and they are listed as follows:

- There are unidentified discrepancies in the ILD data even after the usual error checking algorithms are applied. The data quality control can depend on the particular application for which the data are used and for an application such as travel time estimation, more data quality control is required than the usual error checking methods.
- Majority of the ILD data collected from field violate the conservation of vehicles principle when analyzed as a series for a long time. Thus, if the ILD data are used at a system level, where the data from one detector are compared with that of its neighboring detectors, checks should be conducted for conservation of vehicles.
- The non-linear optimization technique adapted, namely, generalized reduced gradient method, is found to be a suitable technique for removing the discrepancies in the ILD data when the conservation of vehicles principle is violated.
- Systematic correction of ILD data, such as using the optimization method is more useful and convenient than the application of volume adjustment factors when dealing with large amount of data for a longer duration and having large discrepancies.

- The proposed generalized reduced gradient method also proved to be very useful for imputing missing data as well as to prioritize the detector stations for maintenance.
- It was found that the travel time estimation model proposed in this dissertation estimated the travel time with considerable accuracy under varying traffic flow conditions. The model was first validated using simulated data from CORSIM and it was found that the estimated travel time is in good agreement with the actual travel time from simulation, under congested and un-congested-flow conditions. The estimated travel time from ILD data was compared with AVI data and the performance was found to be very satisfactory. Thus, the developed theoretical model is found to be a promising method to estimate travel time from loop detector data under varying traffic flow conditions.
- A comparison of the developed model for the estimation of travel time with the extrapolation method, which is the current field method, showed that the accuracy of the performance of the developed model results increased with increasing flow values. It was also found that the biggest differences in performance were observed during transition and congested conditions. This is not unexpected because these conditions are more difficult to model. In contrast it was found that both methods gave similar results for un-congested conditions.
- Support Vector Regression is a promising tool for the short-term prediction of travel time with performance comparable to that of ANN, when the traffic condition is stable. SVR performed better than ANN when the training data had more variations and the amount of training data is relatively less.
- Both ANN and SVM methods have good dynamic response and showed better performance compared to the traditional models.

7.3 FUTURE RESEARCH

- The optimization method for the data quality control used in this dissertation analyzed up to five detectors in series. Future studies can check the performance with longer sections having more detectors in series. Also, the optimization was carried out based on objective function and constraints that will make sure that conservation of vehicles principle is not violated. It is hypothesized that more rigorous objective function incorporating more constraints may give better results and reduce more discrepancies in the data. Also, in this dissertation the optimization was validated using simulated data. Future work can be

performed along a section where ground truth flow data can be collected for the same locations as the detectors. This will provide the added benefit of a direct comparison of the performance of the optimization method using field data.

- Travel time estimation from loop detector data is an important component for the successful use of ATIS as discussed in this dissertation. A model based on traffic flow theory was used to obtain the travel time from field data corrected in the first step of the research. The validation of the model was checked mainly using simulated data from CORSIM. Validation using field data made use of AVI data. However, the sample size of AVI data was very less. Future similar work should be performed along a section from where more ground truth travel time data can be collected for the same location as the detector points. This would provide a direct comparison of the performance of the theoretical model used in this dissertation. Also, the data used in the present study added up data from all the lanes of the road at a detector location and assumed it as a single lane. Future work is needed where the analysis of the data is carried out at lane-by-lane level, and this needs the development of a model that can also take into account the lane changing characteristics.
- More research concerning the use of SVM for travel time prediction is necessary, especially when one is interested in getting the best performance in terms of training time. For example, exploring the effect of each of the different parameters of SVM such as kernel function and cost function on the prediction performance is needed. Also, a more computationally efficient and standardized toolbox is needed for fast and optimum performance.

REFERENCES

- Abadie, J. (1970), "Application of the GRG algorithm to optimal control problems." *Integer and nonlinear programming*, J. Abadie, ed., Norton-Holland Publishing Company, Amsterdam, The Netherlands, 191-211.
- Abadie, J. (1978), "The GRG method for nonlinear programming." in *Design and implementation of optimization software*, H. J. Greenberg, ed., Sijthoff and Noordhoff, Leyden, The Netherlands, 335-362.
- Abadie, J., and Carpenter, J. (1969), "Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints." in *Optimization-symposium of the institute of mathematics and its applications*, R. Fletcher, ed., University of Keele, Academic Press, U. K., 37-47.
- Abdulhai, B., and Tabib, S. M. (2003), "Spatio-temporal inductance pattern recognition for vehicle re-identification." *Transp. Res. C*, 11, 223-239.
- Al-Deek, H. M. (1998), "Travel time prediction with non-linear time series." *Proc. of the 5th Int. Conf. on Applications of Advanced Technologies in Transportation Engineering (AATT-5)*: ASCE, Newport Beach, California, 317-324.
- Ametha, J. (2001), "Development and implementation of algorithms used in ground and internet traffic monitoring." Master's thesis, Department of Mechanical Engineering, Texas A&M University, College Station, Texas.
- Anderson, J. M., Bell, M. G. H., Sayers, T. M., Busch, F. M., and Heymann, G. (1994), "The short term prediction of link travel times in signal controlled road networks." *Transp. systems: Theory and Application of Advanced Technology, IFAC symposium*, Tianjin, PRC, 621-626.
- Beale, R., and Jackson, T. (1990), *Neural computing: An introduction*. IOP Publishing Ltd., Bristol, U.K.
- Bellamy, P. H. (1979), "Undercounting of vehicle with single-loop-detector systems." *TRRL Supplementary Rep. 473*, Transport and Road Research Laboratory, Berkshire, U. K.
- Bellemans, T., Schutter, B. D., and Moor, B. D. (2000), "On data acquisition, modeling, and simulation of highway traffic." *Proc. of the 9th IFAC Control in Transp. Systems-Vol 1*, E. Schnieder and U. Becker, eds., Braunschweig, Germany, 22-27.

- Bender, J., and Nihan, L. (1988), "Inductive loop detector failure identification: A state of the art review." *Final Interim Rep., Research Project GC8286*, Task 24, Item no. 204 C, Washington State Transp. Center, Washington.
- Berka, S., and Lall, K. B. (1998), "New perspectives for ATMS: advanced technologies in traffic detection." *J. of Transp. Engineering, ASCE*, 1, 9-15.
- Bikowitz, E. W., and Ross, S. P. (1985), "Evaluation and improvement of inductive loop traffic detectors." *Transp. Res. Rec. 1010*, Transportation Research Board, Washington, D. C., 76-80.
- Blanz, V., Scholkopf, B., Bulthoff, H., Burges, C., Vapnik, V., and Vetter, T. (1996), *Comparison of view-based object recognition algorithms using realistic 3D models*. In Artificial Neural Networks - ICANN'96, Berlin, Springer Lecture Notes in Computer Science, 1112, 251-256.
- Blue, V., List, G. F., and Embrechts, M. J. (1994), "Neural net freeway travel time estimation." in *Intelligent Engineering Systems through Artificial Neural Networks*, C.H. Dagli, B.R. Frenandez, J. Ghosh, and R.T.S. Kumara, eds., Vol. 4, ASME Press, New York, 1135-1140.
- Bovy, P. H. L., and Thijs, R. (2000), *Estimators of travel time for road networks: New developments, evaluation results, and applications*. Delft University Press, The Netherlands.
- Boyce, D., Roupail, N., and Kirson, A. (1993), "Estimation and measurement of link travel times in the ADVANCE project." in *Proc. of the Vehicle Navigation and Information Systems Conf., IEEE*, New York, 62-66.
- Brydia, R. E., Turner, S. M., Eisele, W. L., and Liu, J. C. (1998), "Development of intelligent transportation system data management." *Transp. Res. Rec. 1625*, Transportation Research Board, Washington, D.C., 124-130.
- Burges, C. J. C. (1998), *A tutorial on support vector machines for pattern recognition*. Kluwer Academic Publishers, Boston.
- Burges, C. J. C., and Scholkopf, B. (1997), "Improving the accuracy and speed of support vector learning machines." in *Advances in Neural Information Processing Systems*, Vol. 9, MIT Press, Cambridge, Massachusetts, 375-381.
- Campbell, C. (2002), "Kernel methods: A survey of current techniques." *Neuro Computing*, 48, 63-84.

- Cassidy, M. J. (1998), "Bivariate relations in nearly stationary highway traffic." *Transp. Research B*, 32, 49-59.
- Chen, L., and May, A. D. (1987), "Traffic detector errors and diagnostics." *Transp. Res. Rec. 1132*, Transportation Research Board, Washington, D.C., 82-93.
- Chen, L., Kwon, J., Rice, J., Skabardonis, A., and Varaiya, P. (2003), "Detecting errors and imputing missing data for single-loop surveillance systems." *Presented at the TRB 82nd Annual Meeting* (CDROM), Transportation Research Board, Washington D.C.
- Chen, M. and Chien, S. I. J. (2001), "Dynamic freeway travel time prediction using probe vehicle data: Link based vs. path based." *Presented at the 80th Annual Meeting* (CD-ROM), Transportation Research Board, Washington D.C.
- Cherrett, T. J., Bell, H. A., and Mc Donald, M. (1996), "The use of SCOOT type single-loop detectors to measure speed, journey time and queue status on non- SCOOT controlled links." *8th Int. Conf. on Road Traffic Monitoring and Control*, 23-25.
- Chien, S., Liu, X., and Ozbay, K. (2003), "Predicting travel times for the South Jersey real-time motorist information system." *Presented at the 82nd Annual Meeting* (CD-ROM), Transportation Research Board, Washington D.C.
- Chien, S. I. J., and Kuchipudi, C. M. (2002), "Dynamic travel time prediction with real-time and historical data", *Presented at the 81st TRB Annual Meeting* (CD-ROM), Transportation Research Board Washington D. C.
- Cleghorn, D., Hall, F. L., and Garbuio, D. (1991), "Improved data screening techniques for freeway traffic management systems." *Transp. Res. Rec. 1320*, Transportation Research Board, Washington, D.C., 17-23.
- Coifman, B. (1998), "Vehicle re-identification and travel time measurement in real-time on freeways using existing loop detector infrastructure." *Transp. Res. Rec. 1643*, Transportation Research Board, Washington, D.C., 181-191.
- Coifman, B. (1999), "Using dual-loop speed traps to identify detector errors." *Transp. Res. Rec. 1683*, Transportation Research Board, Washington, D.C., 47-58.
- Coifman, B. (2001), "Improved Velocity Estimation Using Single-loop Detectors." *Transp. Research A*, 35(10), 863-880.
- Coifman, B. (2002), "Estimating travel times and vehicle trajectories on freeways using dual-loop detectors." *Transp. Research A*, 36, 351-364.

- Coifman, B., and Cassidy, M. (2002), "Vehicle re-identification and travel time measurement on congested freeways." *Transp. Research A*, 36, 899-917.
- Coifman, B., and Dhoorjaty, S. (2002), "Event data based traffic detector validation tests." *Presented at the TRB 81st Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- CORSIM User's Guide* (Software Help Menu) (2001), FHWA, U.S. Department of Transportation, Washington, D.C.
- Cortes, C., and Vapnik, V. (1995), "Support vector networks." *Machine Learning*, 20, 273-297.
- Cortes, C. E., Lavanya, R., Oh, J. S., and Jayakrishnan, R. (2002), "A general purpose methodology for link travel time estimation using multiple point detection of traffic." *Presented at the 81st Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Courage, K. G., Bauer, C. S., and Ross, D. W. (1976), "Operating parameters for main line sensors in freeway surveillance systems." *Transp. Res. Rec. 601*, Transportation Research Board, Washington, D.C., 19-26.
- Cristianini, N. and Shawe-Taylor, J. (2000), *An introduction to support vector machines and other kernel based learning methods*, Cambridge University Press, Cambridge, New York.
- Cybenko, G. (1989), "Approximation by superimposition of a sigmoidal function." *Mathematical Control Signals Systems*, 2, 303-314
- D'Angelo, M. P., Al-Deek, H. M., and Wang, M. C. (1998), "Travel time prediction for freeway corridors." *Transp. Res. Rec. 1676*, Transportation Research Board, Washington, D.C., 184-191.
- Daganzo, C. (1997), *Fundamentals of transportation and traffic operations*. Pergamon-Elsevier, Oxford, U.K.
- Dia, H. (2001), "An object oriented neural network approach to short term traffic forecasting." *European J. of Operational Res.*, 131, 253-261.
- Dailey, D. J. (1993), "Travel time estimation using cross-correlation techniques." *Transp. Res. B*, 27(2), 97-107.
- Dailey, D. J. (1997), "Travel time estimates using a series of single-loop volume and occupancy measurements." *Presented at the 76th Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.

- Dayhoff, J. E. (1990), *Neural network architectures: An introduction*. Van Nostrand Reinhold, New York.
- Desa, J. P. M. (2001), *Pattern recognition, concepts, methods and applications*. Springer, New York.
- Dharia, A., and Adeli, H. (2003), "Neural network model for rapid forecasting of freeway link travel time." *Engineering. Applications of Artificial Intelligence*, 16(7-8), 617-613.
- Dhulipala, S. (2002), "A system for travel time estimation on urban freeways." Master's thesis, Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
- Ding, A., Zhao, X., and Jiao, L. (2002), "Traffic flow time series prediction based on statistics learning theory." *IEEE 5th Int. Conf. on Intelligent Transp. Systems*, Singapore, 727-730.
- Dougherty, M. (1995), "A review of neural networks applied to transport." *Transp. Res. C*, 3(4), 247-260.
- Dougherty, M. S., and Cobbett, M. R. (1997), "Short term inter urban traffic forecasts using neural networks." *Int. J. of Forecasting*, 13, 21-31.
- Dougherty, M. S., Kirby, H. R., and Boyle, R. D. (1994), "Using neural networks to recognize, predict and model traffic." in *Artificial intelligence applications to traffic engineering*, M. Bielli, G. Ambrosino and M. Boero, eds., Utrecht, The Netherlands, 233-250.
- Drew, D. R. (1968), *Traffic flow theory and control*. McGraw Hill series in Transp., McGraw Hill Book Company, New York.
- Dudek, C. L., Messer, C. J., and Dutt, A. K. (1974), "Study of detector reliability for a motorist information system on the gulf freeway." *Transp. Res. Rec. 495*, Transportation Research Board, Washington, D.C., 35-43.
- Eisele, W. L. (2001), "Estimating travel time mean and variance using intelligent transportation systems data for real-time and off-line transportation applications." Doctoral dissertation, Department of Civil Engineering, Texas A&M University, College Station, Texas.
- Eiselt, H. A., Pederzoli, G., and Sandblom, C. L. (1987), *Continuous optimization models*. Walter De Gruyter Inc., Berlin, Germany.
- Ezforecaster. (2003). < <http://www.ezforecaster.com/bestfit.htm> > (Dec. 19, 2003).
- Faghri, A., and Hua, J. (1992), "Evaluation of artificial neural network applications in transportation engineering" *Transp. Res. Rec. 1358*, Transportation Research Board, Washington, D.C., 71-80.

- Faouzi, N. E., and Lesort, J. B. (1995), "Travel time estimation on urban networks from traffic data and on-board trip characteristics." *Proc. of the 2nd World Congress on Intelligent Transp. Systems*, Yokohama, Japan.
- Fenton, R. E. (1980), "On future traffic control: advanced systems hardware." *IEEE Transactions on Vehicular Technology*, VT-29, 200-207.
- Ferrier, P. J. (1999), "Comparison of vehicle travel times and measurement techniques along the I-35 corridor in San Antonio, Texas." Master's thesis, Department of Civil Engineering, Texas A&M University, College Station, Texas.
- Gabriele, G. A., and Beltracchi, T. J. (1987), "Resolving degeneracy in the generalized reduced gradient method." *J. of Mechanics, Transmissions and Automation in Design*, 109(2), 263-267.
- Gabriele, G. A., and Ragsdell, K. M. (1977), "The generalized reduced gradient method: A reliable tool for optimal design." *J. of Engineering for Industry: Transactions of the ASME*, 99, 394-400.
- Gabriele, G. A., and Ragsdell, K. M. (1980), "Large-scale non-linear programming using the generalized reduced gradient method." *Transactions of the ASME*, 102(3), 566-573.
- Gibson, D., Mills, M. K., and Rekenhaller, D. (1998), "Staying in the loop: The search for improved reliability of traffic sensing systems through smart test instruments." *Public Roads*, 62(2), <<http://www.tfhrc.gov/pubrds/septoct98/loop.htm>> (Feb. 12, 2004).
- Gold, D. L., Turner, S. M., Gajewski, B. J., and Spiegelman, C. (2001), "Imputing missing values in its data archives for intervals under 5 minutes." *Presented at the 80th Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Gunn, S. R. (2003), "Support vector machines for classification and regression." <<http://www.ecs.soton.ac.uk/~srg/publications/pdf/SVM.pdf>> (Nov. 23, 2003).
- Gupta, S. (1999), "A new algorithm for detecting erroneous loop detector data." Master's thesis, Department of Mechanical Engineering, Texas A& M University, College Station, Texas.
- Hauslen, R. A. (1977), "The promise of automatic vehicle identification." *IEEE Transactions on Vehicular Technology*, VT-26, 30-38.
- Haykin, S. S. (1994), *Neural networks: A comprehensive foundation*. Prentice Hall, N.J., 1999.
- Hearst, M. A. (1998), "Trends and controversies: Support vector machines", *IEEE Intelligent Systems*, 13(4), 18-28.
- Highway Capacity Manual*, (2000), Transportation Research Board, Washington, D.C.

- Himmelblau, D. M. (1972), *Applied Nonlinear Programming*. McGraw-Hill, New York.
- Hoffman, C., and Janko, J. (1990), "Travel time as a basis of the LISB guidance strategy", in *Proc. of IEEE Road Traffic Control Conf., IEEE*, New York, 6-10.
- Hoogendoorn, S. P. (2000), "Model-based multiclass travel time estimation." *Presented at the 79th Annual Meeting* (CD-ROM), Transportation Research Board, Washington D.C.
- Hornik, K., Stinchcombe, M., and White, H. (1989), "Multilayer feed forward networks are universal approximators." *Neural Networks*, 2, 359-366
- Huang, S. H., and Ran, B. (2003), "An application of neural network on traffic speed prediction under adverse weather condition." *Presented at the 82nd TRB Annual Meeting* (CD-ROM), Transportation Research Board, Washington D.C.
- Huisken, G., and van Berkum, E. (2002), "Short-term travel time prediction using data from induction loops." *9th World Congress on Intelligent Transport Systems* (CD-ROM), Chicago, Illinois.
- Innama, S. (2001), "Short term prediction of highway travel time using MLP neural networks." *8th World Congress on Intelligent Transp. systems*, Sydney, Australia, 1-12.
- Ishak, S., and Al-Deek, H. (2002), "Performance evaluation of short term time series traffic prediction model." *J. of Transp. Engineering, ASCE*, 128(6), 490-498.
- Ishak, S., Kotha, P., and Alecsandru, C. (2003), "Optimization of dynamic neural networks performance for short term traffic prediction." *Presented at the 82nd TRB Annual Meeting* (CD-ROM), Transportation Research Board, Washington D.C.
- Iwasaki, M., and Shirao, K. (1996), "A short term prediction of traffic fluctuations using pseudo traffic patterns." *3rd World Congress on Intelligent Transport Systems Conference* (CD-ROM), Orlando, Florida.
- Jack, L. B., and Nandi, A. K. (2002), "Fault detection using support vector machines and artificial neural networks augmented by genetic algorithms." *Mechanical Systems and Signal Processing*, 16(2-3), 373-390.
- Jacobson, L. N., Nihan, N. L., and Bender, J. D. (1990), "Detecting erroneous loop detector data in a freeway traffic management system." *Transp. Res. Rec. 1287*, Transportation Research Board, Washington, D.C., 151-166.
- Jeffrey, D. J., Russam, K., and Robertson, D. I. (1987), "Electronic route guidance by AUTOGUIDE: The research background." *Traffic Engineering and Control*, 28(10), 525-529.

- Joachims, T. (1997), "Text categorization with support vector machines." *Technical Rep., LS VIII no. 23*, University of Dortmund, Dortmund, Germany.
- Kaysi, I., Ben-Akiva, M., and Koutsopoulos, H. (1993) "An integrated approach to vehicle routing and congestion prediction for real-time driver guidance." *Transp. Res. Rec. 1408*, Transportation Research Board, Washington, D.C., 66-74.
- Kecman, V. (2001), *Learning and soft computing: Support vector machines, neural networks, and fuzzy logic models*, The MIT Press, Cambridge, Massachusetts.
- Kikuchi, S. (2000), "A method to defuzzify the fuzzy numbers: Transportation problem application." *Fuzzy Sets and Systems*, 116, 3-9.
- Kikuchi, S., and Miljkovic, D. (1999), "A method to pre-process traffic data: Application of fuzzy optimization concept." *Presented at the 78th Annual Meeting of the Transportation Research Board (CD-ROM)*, Transportation Research Board, Washington, D.C.
- Kikuchi, S., Miljkovic, D., and Van Zuylen, H. J. (2000), "Examination of methods that adjust observed traffic volumes on a network." *Transp. Res. Rec. 1717*, Transportation Research Board, Washington, D. C., 109-119.
- Kisgyorgy, L., and Rilett, L. R. (2002), "Travel time prediction by advanced neural network." *Periodic Polytechnica Series in Civil Engineering*, 46 (1), 15-32.
- Klein, L. A. (2001), *Sensor technologies and data requirements for ITS*. Artech House, Boston, London.
- Kreer, J. B. (1975), "A comparison of predictor algorithms for computerized control." *Traffic Engineering*, 45 (4), 51-56.
- Kuchipudi, C. M., and Chien, S. I. J. (2003), "Development of a hybrid model for dynamic travel time prediction." *Presented at the 82nd Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Kuhne, R., and Michalopoulos, P. (2003), "Continuum flow models, Traffic flow theory," in *The Revised Monograph on Traffic Flow Theory*, N. Gartner, C. J. Messer, and A. K. Rathi, eds. <<http://www.tfrc.gov/its/tft/tft.htm>> (May 21, 2003).
- Kuhne, R. D., Palen, J., Gardner, C., and Ritchie, S. (1997), "Loop based travel time measurement: Fast incident detection using traditional loops." *Traffic Technology Int. Annual*, 157-161.

- Kwon, J., Coifman, B., and Bickel, P. (2000), "Day-to-day travel time trends and travel time prediction from loop detector data." *Transp. Res. Rec. 1717*, Transportation Research Board, Washington, D.C., 120-129.
- Labell, L. N., Spencer, M., Skabardonis, A., and May, A. D. (1989), "Detectors for freeway surveillance and control." *Working Paper UCB-ITS-WP-89-1*, University of California, Berkeley.
- Lasdon, L. S., and Waren, A. D. (1978), "Generalized reduced gradient software for linearly and non linearly constrained problems." *Design and implementation of optimization software*, H. J. Greenberg, ed., Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 363-396.
- Lasdon, L. S., Waren, A. D., Jain, A., and Ratner, M. (1978), "Design and testing of a generalized reduced gradient code for nonlinear programming." *ACM Transactions on Mathematical Software*, 4, 34-50.
- Lee, S., Kim, D., Kim, J., and Cho, B. (1998), "Comparison of models for predicting short-term travel speeds." *5th World Congress on Intelligent Transp. Systems(CD-ROM)*, Seoul, Korea.
- Lighthill, M. J., and Whitham, G. B. (1955), "On kinematic waves: II. A theory of traffic flow on long crowded roads." *Proc., Royal Society, A 229*, 1178, 317-345.
- Lindveld, C. D. R., Thijis, R., Bovy, P. H., and Van der Zijpp, N. J. (2000), "Evaluation of online travel time estimators and predictors." *Transp. Res. Rec. 1719*, Transportation Research Board, Washington, D.C., 45-53.
- Lindveld, C. D. R., and Thijs, R. (1999), "On-line travel time estimation using inductive loop data: The effect of instrumentation peculiarities." *6th Annual World Conf. on Intelligent Transp. Systems (CD-ROM)*, Toronto.
- Lippman, R. P. (1987), "An introduction to computing with neural nets." *IEEE ASSP Magazine*, 4-22.
- Liu, T. K. (2000), *Travel time data needs, applications, and data collection*. <http://www.Nmsu.edu/Research/traffic/public_html/NATDAC96/authors/liu.htm> (Aug. 25, 2000).
- Mahalel, D., and Hakkert, A. S. (1995), "Time series for vehicle speeds." *Transp. Research B*, 19(3), 217-225.

- Manfredi, S., Salem H. H., and Grol, H. J. M. (1998), "Development and application of coordinated control of corridors."
ftp://ftp.cordis.lu/pub/telematics/docs/taptransport/daccord_d9.1.pdf (Nov. 8, 2003)
- The MathWorks, Inc. (2003), *MATLAB documentation, Version 6.5.0.180913a, Release 13*, Natick, Massachusetts.
- Matsui, H., and Fujita, M. (1998), "Travel time prediction for freeway traffic information by neural network driven fuzzy reasoning." in *Neural networks in transp. applications*, V. Himanen, P. Nijkamp, A. Reggiani, and J. Raitio, eds., Ashgate Publishers, Burlington, Vermont, 355-364.
- Mattera, D., and Haykin, S. (1999), "Support vector machines for dynamic reconstruction of a chaotic system." *Advances in kernel methods: Support vector learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, eds., MIT Press, Cambridge, Massachusetts.
- May, A. D. (1990), *Traffic flow fundamentals*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- May, A. D., Cayford, R., Coifman, B., and Merritt, G. (2003), "Loop detector data collection and travel time measurement in the Berkeley highway laboratory." *California PATH Research Rep. UCB-ITS-PRR-2003-17*, Institute of Transp. Studies, Berkeley, California.
- Mc Fadden, J., Yang, W. T., and Durrans, S. R. (2001), "Application of artificial neural networks to predict speeds on two-lane rural highways." *Presented at the 80th TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington, D.C.
- Middleton, D., Jasek, D., and Parker, R. (1999), "Evaluation of some existing technologies for vehicle detection." *Rep. No. FHWA/TX-00/1715-S*, Texas Transportation Institute, College Station, Texas.
- Miller, J. C., and Miller, J. N. (1993), *Statistics for analytical chemistry*, Ellis Horwood PTR Prentice Hall, Englewood Cliffs, New Jersey.
- Mukherjee, S., Osuna, E., and Girosi, F. (1997), "Nonlinear prediction of chaotic time series using a support vector machine," *Proc. of the IEEE Workshop on Neural Networks and for Signal Processing*, Amelia Island, Florida, 511-519.
- Muller, K. R., Smola, A., Ratsch, G., Scholkopf, B., Kohlmorgen, J., and Vapnik, V. (1999), "Using support vector machines for time series prediction." in *Advances in kernel methods: Support vector learning*, B. Scholkopf, C. J. C. Burges, A. Smola, eds., MIT Press, Cambridge, Massachusetts.

- Nair, A. S., Liu, J. C., Rilett, L. R., and Gupta, S. (2001), "Non linear analysis of traffic flow." *4th Int. IEEE Conf. on Intelligent Transp. Systems*, Oakland, California, 681-685.
- Nakatsuji, T., and Shibuya, S. (1998), "Neural network models applied to traffic flow problems." in *Neural networks in transport applications*, V. Himanen, P. Nijkamp, A. Reggiani, and J. Raitio, eds., Ashgate Publishers, Burlington, Vermont, 249-262.
- Nam, D. H. (1995), "Methodologies for integrating traffic flow theory, ITS and evolving surveillance technologies." Doctoral dissertation, Department of Civil Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
- Nam, D. H., and Drew, D. R. (1996), "Traffic dynamics: Methods for estimating freeway travel times in real-time from flow measurements." *J. of Transp. Engineering, ASCE*, 122(3), 185-191.
- Nam, D. H., and Drew, D. R. (1998), "Analyzing freeway traffic under congestion: Traffic dynamics approach." *J. of Transp. Engineering, ASCE*, 124(3), 208-212.
- Nam, D. H., and Drew, D. R. (1999), "Automatic measurement of traffic variables for intelligent transportation systems applications." *Transp. Research B*, 33, 437-457.
- Nanthawichit, C., Nakatsuji, T., and Suzuki, H. (2003), "Application of probe vehicle data for real-time traffic state estimation and short term travel time prediction on a freeway." *Presented at the 82nd Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- NEMA, Traffic control systems (1983)*, Standard Publication Number TS-1, National Electrical Manufacturers Association, Washington, D.C.
- Nihan, L., Jacobson, L. N., and Bender, J. D. (1990), "Detector data validity.", *Final Rep., WARD 208.1*, Washington State Transp. Center, Washington.
- Nihan, N., and Wong, M. (1995), "Improved error detection using prediction techniques and video imaging", *Final Technical Rep., Research Project T9233*, Washington State Transp. Center, Washington.
- Oda, T. (1990), "An algorithm for prediction of travel time using vehicle sensor data." *IEEE 3rd Int. Conf. on Road Traffic Control*, 40-44,
<<http://ieeexplore.ieee.org/servlet/opac?punumber=1222>>.
- Oh, J., Jayakrishnan, R., and Recker, W. (2003), "Section travel time estimation from point detection data." Center for Traffic Simulation Studies, *Paper VCI-ITS-TS-WP-02-15*, <<http://repositories.cdlib.org/itsirvine/ctss/UCI-ITS-TS-WP-02-15>>, (July 4, 2003).

- Ohba, Y., Koyama, T., and Shimada, S. (1997), "Online learning type of traveling time prediction model in expressway." *IEEE Conf. on Intelligent Transp. Systems*, Boston, Massachusetts, 350-355.
- Osuna, E., Freund, R., and Girosi, F. (1997a), "Training support vector machines: An application to face detection." *IEEE Conf. on Computer Vision and Pattern Recognition*, Juan, Puerto Rico, 130-136.
- Osuna, E., Freund, R., and Girosi, F. (1997b), "Nonlinear prediction of chaotic time series using support vector machines." *Proc. of the IEEE Workshop on Neural Networks for Signal Processing*, Amelia Island, Florida, 276-285.
- Palacharla, P. V., and Nelson, P. C. (1999), "Application of fuzzy logic and neural networks for dynamic travel time estimation." *Int. Transactions in Operational Research*, 6, 145-160.
- Park, B., Messer, C. J., and Urbanik, T. II. (1998), "Short term traffic volume forecasting using radial basis function neural network." *Transp. Res. Rec. 1651*, Transportation Research Board, Washington, D.C., 39-47.
- Park, D., and Rilett, L. R. (1998), "Forecasting multiple period freeway link travel times using modular neural networks." *Transp. Res. Rec. 1617*, Transportation Research Board, Washington, D.C., 163-170.
- Park, D., and Rilett, L. R. (1999), "Forecasting freeway link travel times with a multi-layer feed forward neural network." *Computer Aided Civil and Infrastructure Engineering*, 14, 357-367.
- Park, D., Rilett, L. R., and Han, G. (1999), "Spectral basis neural networks for real-time link travel times forecasting." *J. of Transp. Engineering, ASCE*, 125(6), 515-523.
- Park, E. S., Turner, S., and Spiegelman, C. H. (2003), "Empirical approaches to outlier detection in ITS data." *Presented at the TRB 82nd Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Payne, H. J., Helfenbein, E. D., and Knobel, H. C. (1976), "Development and testing of incident detection algorithms", 2, Research methodology and detailed results, *Rep. No. FHWA-RD-76-20*, McLean, Virginia.
- Payne, H. J., and Thompson, S. (1997), "Malfunction detection and data repair for induction loop sensors using I-880 database." *Transp. Res. Rec. 1570*, Transportation Research Board, Washington, D. C., 191-201.

- Peeta, S., and Anastassopoulos, I. (2002), "Automatic real-time detection and correction of erroneous detector data using Fourier transforms for on-line traffic control architectures." *Presented at the TRB 81st Annual Meeting (CD-ROM)*, Transportation Research Board Washington D.C.
- Persaud, B. N., and Hurdle, V. F. (1988), "Some new data that challenge some old ideas about speed flow relationships." *Transp. Res. Rec. 1194*, Transportation Research Board, Washington, D.C., 191-198.
- Petty, K. (1995), "Freeway service petrol 1.1 the analysis software for the FSP project." *California PATH Research Rep., UCB-ITS-PRR-95-20*, Berkeley, California.
- Petty, K. F., Bickel, P., Ostland, M., Rice, J., Schoenberg, F., Jiang, J., and Ritov, Y. (1998), "Accurate estimation of travel times from single-loop detectors." *Transp. Res. - A*, 32(1), 1-17.
- Pfannerstill, E. (1989), "Automatic monitoring of traffic conditions by re-identification of vehicles." *Institution of Electrical Engineers 2nd Int. Conf. on Road Traffic Monitoring*, Publication # 299, London. U. K.
- Pinnell-Anderson-Wilshire and Associates, Inc. (1976), *Inductive loop detectors: Theory and practice*, U.S. Department of Transportation, Federal Highway Administration, Springfield, Washington, D.C.
- Quiroga, C. (2000), "Assessment of dynamic message travel time information accuracy." *Proc. of the North American Travel Monitoring Conf. and Exposition*, Middleton, Wisconsin, 1-13.
- Raj, J., and Rathi, A. (1994), "Inductive loop tester – ILT II." *Summary Report*, FHWA-SA-94-077, Washington D. C.
- Rice, J., and van Zwet, E. (2002), "A simple and effective method for predicting travel times on freeways." *IEEE Intelligent Transp. Systems Conf. Proc.*, Piscataway, New Jersey, 227-232.
- Richards, P. I. (1956), "Shock waves on the highway." *Operations Research*, 4(1), 42-51.
- Rilett, L. R., and Park, D. (2001), "Direct forecasting of freeway corridor travel times using spectral basis neural networks." *Transp. Res. Rec. 1752*, Transportation Research Board, Washington, D.C., 140-147.

- Rilett, L. R., Kim, K., and Raney, B. (2000), "Comparison of low-fidelity TRANSIMS and high-fidelity CORSIM highway simulation models with intelligent transportation system data." *Transp. Res. Rec. 1739*, Transportation Research Board, Washington, D.C., 1-8.
- Rosenblatt, F. (1962), *Principles of neuro dynamics*. Spartan Books, New York.
- Roth, S. H. (1977), "History of automatic vehicle monitoring (AVM)." *IEEE Transactions on Vehicular Technology* VT-26, 2-6.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986), "Learning representations by back propagating errors." *Nature*, 323(9), 533-536.
- Saito, M., and Watanabe, T. (1995), "Prediction and Dissemination system for travel time utilizing vehicle detectors." *Proc. of the 2nd World Congress on Intelligent Transp. Systems conference*, Yokohama, Japan.
- Samanta, B. (2004), "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms." *Mechanical Systems and Signal Processing*, 18, 625-644.
- Schmidt, M. (1996), "Identifying speaker with support vector networks." *Interface 96 Proc.*, Sydney, Australia.
- Scholkopf, B., Burges, C., and Vapnik, V. (1995), "Extracting support data for a given task." *Proc. of first Int. Conf. on Knowledge Discovery and Data Mining*, U. M. Fayyad and Uthurusamy, R., eds., AAAI Press, CA.
- Scholkopf, B., Burges, C., and Vapnik, V. (1996), "Incorporating invariances in support vector learning machines." *Artificial neural networks-ICANN 1996*. 47-52.
- Scott, B. M. (1992), "Automatic vehicle identification: A test of theories of technology." *University of Wollongong*, Published in *Science, Technology, & Human Values*, 17 (4), 485-505, < <http://www.uow.edu.au/arts/sts/bmartin/pubs /92sthv.html>> (Nov. 23, 2003).
- Seki, S. (1995), "Travel time measurement and provision system using AVI units." *Proc. of the 2nd World Congress on Intelligent Transp. Systems*, Yokohama, Japan.
- Sen, A., Liu, N., Thakuriah, P., and Li, J. (1991), "Short-term forecasting of link travel times: A preliminary proposal." *ADVANCE Working Paper Series*, Number 7, Illinois, Chicago.
- Sharma, S., Lingras, P., and Zhong, M. (2003), "Effect of missing value imputations on traffic parameters estimations from permanent traffic counts." *Presented at the TRB 82nd Annual Meeting (CD-ROM)*, Transportation Research Board Washington D.C.

- Shbaklo, S., Bhat, C., Koppelman, F., Li, J., Thakuriah, P., Sen, A., and Roupail, N. (1992), "Short-term travel time prediction." *ADVANCE Project Rep., TRF-TT-01*, University Transp. Research Consortium, Illinois, Chicago.
- Simon, N. (1993), "Constructive supervised learning algorithms for artificial neural networks." Master's Thesis, Delft University, The Netherlands.
- Singleton, M., and Ward, J. E. (1977), "A comparative study of various types of vehicle detectors." *Rep. No. DOT-TSC-OST-77-9*, Springfield, Virginia.
- Sisiopiku, V. P., Roupail, N. M., and Santiago, A. (1994a), "Analysis of correlation between arterial travel time and detector data from simulation and field studies." *Transp. Res. Rec. 1457*, Transportation Research Board, Washington, D.C., 166-173.
- Sisiopiku, V. P., Roupail, N. M., and Tarko, A. (1994), "Estimating travel times on freeway segments." *ADVANCE Working Paper Series Number 32*, Urban Transp. Center, University of Illinois at Chicago.
- Smith, B. L., and Conklin, J. H. (2002), "Use of local lane distribution patterns to estimate missing data values from traffic monitoring systems." *Transp. Res. Rec. 1811*, Transportation Research Board, Washington, D. C., 50 - 56.
- Smith, B. L., and Demetsky, M. J. (1994), "Short term traffic flow prediction: Neural network approach." *Transp. Res. Rec. 1453*, Transportation Research Board, Washington, D. C. 98-104.
- Smith, B. L., and Demetsky, M. J. (1997), "Traffic flow forecasting: Comparison of modeling approaches." *J. of Transp. Engineering, ASCE*, 123 (4), 261-266.
- Smith, B. L., Scherer, W. T., and Conklin, J. H. (2003), "Exploring imputation techniques for missing data in transportation management systems." *Presented at the 82nd TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D. C.
- Smola, A. J., and Scholkopf, B. (1998), "A tutorial on support vector regression." *NeuroCOLT2 Technical Rep. Series, NC2-TR-1998-030*, < <http://www.kernelmachines.org/tutorial.html>> (Feb. 12, 2004)
- Son, B. (1996), "Discussion on 'Traffic dynamics: Methods for estimating freeway travel times in real-time from flow measurements, by Nam and Drew.'" *J. of Transp. Engineering, ASCE*, 122(3), 519-520.

- Speed, M., and Spiegelman, C. (1998), "Evaluating black boxes: An ad-hoc method for assessing nonparametric and non-linear curve fitting estimators." *Communications in Statistics—Simulation and Computation* 27(3), 699-710.
- Sreedevi, I., and Black, J. (2001), <http://www.path.berkeley.edu/~leap/TTM/Incident_Manage/Detection/loopdet.html#Houston> (May 15, 2003).
- Statsoft Pacific Pty Ltd. (2004). <<http://www.statsoftinc.com/textbook/glos1.html>> (Jan. 31, 2004).
- Stephanedes, Y. J., Michalopoulos, P. G., and Plum, R. A. (1981) "Improved estimation of traffic flow for real-time control." *Transp. Res. Rec. 795*, Transportation Research Board, Washington, D.C., 28-39.
- Sun, C., Arr, G., and Ramachandran, R. P. (2003), "Vehicle re-identification as method for deriving travel time and travel time distributions." *Transp. Res. Rec. 1826*, Transportation Research Board, Washington, D. C., 25-31.
- Sun, Z., Bebis, G., and Miller, R. (2002a), "Quantized wavelet features and support vector machines for on-road vehicle detection." *IEEE Int. Conf. on Control, Automation, Robotics and Vision*, Singapore, <<http://www.cs.unr.edu/~bebis/vehicleICARCV02.pdf>> (Feb. 13, 2004).
- Sun, Z., Bebis, G., and Miller, R. (2002b), "On-road vehicle detection using gabor filters and support vector machines." *IEEE Int. Conf. on Digital Signal Processing*, Santorini, Greece, <<http://www.cs.unr.edu/~bebis/vehicleDSP02.pdf>> (Feb. 13, 2004).
- Sun, C., Ritchie, S. G., and Tsai, K. (1998), "Algorithm development for derivation of section related measures of traffic system performance using inductive loop detectors." *Transp. Res. Rec. 1643*, Transportation Research Board, Washington, D.C., 171-180.
- Sun, C., Ritchie, S. G., Tsai, K., and Jayakrishnan, R. (1999), "Use of vehicle signature analysis and lexicographic optimization for vehicle re-identification on freeways." *Transp. Research C*, 7, 167-185.
- Takahashi, K., Inoue, T., Yokota, T., Kobayashi, Y., and Yamane, K. (1995), "Measuring travel time using pattern matching technique." *Proc. of the 2nd World Congress on Intelligent Transp. Systems* (CD-ROM), Yokohama, Japan.
- Talukder, A., and Casasent, D. (2001), "A closed form neural network for discriminatory feature extraction from high dimensional data." *Neural Network*, 14, 1201-1218.

- Tarko, A., and Roupail, N. M. (1993), "Travel time data fusion in ADVANCE." *In Proc. of the 3rd Int. Conf. on Applications of Advanced Technologies in Transp. Engineering*, ASCE, New York, 36-42.
- Taylor, B. N., Parker, W. H., and Langenberg, D. N. (1969), "The fundamental constants and quantum electrodynamics." in *Reviews of modern physics monograph*, Academic Press, New York.
- Texas Department of Transportation. (TxDOT). (2000).
<<http://www.tongji.edu.cn/~yangdy/books/TransGuide.PDF>> (Nov. 23, 2003).
- Texas Department of Transportation. (TxDOT). (2003).
<http://www.transguide.dot.state.tx.us/docs/atms_info.html> (Nov. 23, 2003).
- TexHwyMan. (2003). <<http://home.att.net/~texhwyman/transgd.htm>> (Nov. 23, 2003).
- Thakuriah, P., Sen, A., Li, J., Liu, N., Koppelman, F. S., and Bhat, C. (1992), "Data needs for short term link travel time prediction." *Advance Working Paper Series Number 19*, Urban Transp. Center, University of Illinois, Chicago.
- Traffic Detector Handbook (1991)*, Second edition, Institute of Transportation Engineers, McLean, Virginia.
- Transguide Model Deployment Initiative Design Rep. and Transguide Technical Paper (2002)*, Texas Department of Transportation,
<<http://www.transguide.dot.state.tx.us/PublicInfo/papers.php>>, (Nov. 1, 2003).
- Travel Time Data Collection Handbook (1998)*, Rep. no. FHWA-PL-98-035, Texas Transportation Institute, College Station, Texas.
- Turner, S. M. (1996), "Advanced techniques for travel time data collection." *Transp. Res. Rec. 1551*, Transportation Research Board, Washington, D.C., 51-58.
- Turner, S. M., Albert, L., Gajewski, B., and Eisele, W. (2000), "Archived ITS data quality: Preliminary analysis of San Antonio Transguide data." *Transp. Res. Rec. 1719*, Transportation Research Board, Washington, D. C., 77-84.
- Turner, S. M., Eisele, W. L., Gajewski, B. J., Albert, L. P., and Benz, R. J. (1999), "ITS data archiving: Case study analyses of San Antonio TransGuide data." *Rep. No. FHWA A-PL-99-024*, Federal Highway Administration, Texas Transportation Institute, College Station, Texas.

- Turochy, R. E., and Smith, B. L. (2000), "New procedure for data screening in traffic management systems." *Transp. Res. Rec. 1727*, Transportation Research Board, Washington, D. C., 127-131.
- Valyon, J., and Horvath, G., A. (2002), "Comparison of the SVM and LS-SVM regression, from the viewpoint of parameter selection." *IEEE Hungary Section Proc. Mini-Symposium*, <<http://www.mit.bme.hu/events/minisy2002/ValyonJozsef.pdf>> (Feb. 2, 2004).
- Van Aerde, M., and Yagar, S. (1983), "Volume effects on speeds of 2-lane highways in Ontario." *Transp. Res. A*, 17, 301-313.
- Van Arem, B., Van der vlist, M. J. M., Muste, M. R., and Smulders, S. A. (1997), "Travel time estimation in the GERDIEN Project", *Int. J. of Forecasting*, 13, 73 - 85.
- Van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2000), "Robust and adaptive travel time prediction with neural networks." TRAIL Research School, Delft, <vk042.citg.tudelft.nl/.../staff/lint/papers/Robust%20and%20adaptive%20Travel%20Time%20prediction.pdf> (Dec. 3, 2003).
- Van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2002), "Freeway travel time prediction with state space neural networks." *Presented at the 81st TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D. C.
- Van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2003), "Towards a robust framework for freeway travel time prediction: Experiments with simple imputation and state space neural networks." *Presented at the TRB 82nd Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D. C.
- Van Lint, J. W. C., and van der Zijpp, N. J. (2003), "An improved travel time estimation algorithm using dual-loop detectors." *Presented at the 82nd TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D. C.
- Vanajakshi, L., and Rilett, L. R. (2004a), "A comparison of the performance of artificial neural networks and support vector machines for the prediction of vehicle speed," *Accepted for IEEE Intelligent Vehicles Symposium*, Parma, Italy.
- Vanajakshi, L., and Rilett, L. R. (2004b), "Loop detector data diagnostics based on conservation of vehicle principle." Accepted for publication in *Transp. Res. Rec.*, Transportation Research Board, Washington, D.C.
- Vanderplaats, G. N. (1984), *Numerical optimization techniques for engineering design*. McGraw-Hill, Inc., New York.

- Vapnik, V. N. (1998), *Statistical learning theory*. John Wiley and Sons, Inc., New York.
- Venkataraman, P. (2001), *Applied optimization with Matlab programming*. John Wiley and Sons, Inc., New York.
- Wall, Z., and Dailey, D. J. (2003), "An algorithm for the detection and correction of errors in archived traffic data." *Presented at the 82nd TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Wasserman, P. D. (1989), *Neural computing: Theory and practice*. Van Nostrand Reinhold, New York.
- Weigend, A. S., Huberman, B. A., and Rumelhart, D. E. (1992), "Predicting sun pots and exchange rates with connectionist networks." in *Non Linear Modeling and Forecasting*, M. Casdagli and S. Eubank, eds. Addison Wesley, Menlo Park, California, 395-432.
- Windover, J. R., and Cassidy, M. J. (2001), "Some observed details of freeway traffic evolution." *Transp. Research A*, 35, 881-894.
- Wolfe, P. (1963), "Methods of nonlinear programming." in *Recent Advances in Mathematical Programming*, R. L. Graves and P. Wolfe, eds. Mc-Graw Hill, New York, 67-86.
- Wolfe, P. (1967), "Methods for linear constraints." in *Nonlinear Programming*, J. Abadie, ed., John Wiley & Sons, New York, 121-125.
- Xiao, H., Sun, H., and Ran, B. (2003), "The fuzzy-neural network traffic prediction framework with wavelet decomposition." *Presented at the 82nd TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Yasui, K., Ikenoue, K., and Takeuchi, H. (1995), "Use of AVI information linked up with detector output in travel time prediction and O-D flow estimation." *Proc. of the 2nd World Congress on Intelligent Transp. Systems (CD-ROM)*, Yokohama, Japan.
- You, J., and Kim, T. J. (2000), "Development of hybrid travel time forecasting model." *Transp. Research C*, 8, 231-256.
- Yuan, F., and Cheu, L. (2003), "Incident detection using support vector machines." *Transp. Research C*, 11, 309-328.
- Yun, S. Y., Namkoong, S., Rho, J. H., Shin, S. W., and Choi, J. U. (1998), "A performance evaluation of neural network models in traffic volume forecasting." *Mathematical and Computer Modeling*, 27(9-11), 293-310.
- Zhang, G., Patuwo, E., and Hu, M. Y. (1998), "Forecasting with artificial neural networks: The state of the art." *Int. J. of Forecasting*, 14, 35-62.

- Zhang, X., and Rice, J. (2003), "Short term travel time prediction." *Transp. Research C*, 11, 187-210.
- Zhang, X., Wang, Y., Nihan, N. L., and Hallenbeck, M. E. (2003), "Development of a system to collect loop detector event (individual vehicle) data." *Proc. 80th TRB Annual Meeting (CD-ROM)*, Transportation Research Board, Washington D.C.
- Zhao, M., Garrick, N. W., and Achenie E. K. (1998), "Data reconciliation – based traffic count analysis system." *Transp. Res. Rec. 1625*, Transportation Research Board, Washington, D. C., 12-17.
- Zhu, F. (2000), "Locations of AVI system and travel time forecasting." Master's Thesis, Department of Civil Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
- Zuylen, H. J., and Brantson, D. M. (1982), "Consistent link flow estimation from counts." *Transp. Research B*, 16, 473-476.

APPENDIX A**NOTATIONS**

q	- Flow in vehicles per unit time
D	- Distance
T	- Travel time
t_{occ}	- Occupancy time of detectors
t_{on}	- Instant of time the detector detects a vehicle
t_{off}	- Instant of time the vehicle exits the detector
v	- Vehicle speed
L_n	- Vehicle length
L_d	- Detection zone length
O	- Percent occupancy time
Δt	- Time period
k	- Density in vehicles per unit distance
\overline{L}_v	- Average vehicle length
Δ	- Step size
Q	- Cumulative flow in vehicles
β	- Bias
w	- Weight
v_f	- Free-flow speed
S^2	- Variance

APPENDIX B

GLOSSARY OF FREQUENTLY USED TERMS AND ACRONYMS

B.1 FREQUENTLY USED TERMS

Advanced Traffic Management System (ATMS): The location, usually centralized, where intelligent transportation systems data are collected and the transportation system is monitored.

Advanced Traveler Information System (ATIS): The use of intelligent transportation systems technologies and communication methods for providing information to motorists.

Artificial Neural Network (ANN): An information-processing structure whose design is motivated by the design and functioning of human brains and components thereof.

Automatic Vehicle Identification (AVI): A system where probe vehicles equipped with electronic toll tags communicate with roadside antennas to identify unique vehicles and collect travel time data between the antenna locations.

Automatic Vehicle Location (AVL): An automatic vehicle location enables to remotely track the location of a vehicle with the use of mobile radio receiver, GPS receiver, GPS modem, GPS antenna etc.

Conservation of Vehicles Principle: The concept of conservation of vehicles states that the difference between the number of vehicles entering and leaving a link during a specific time interval corresponds to the change in the number of vehicles traveling on the link.

CORSIM: CORridor SIMulation software package developed by the Federal Highway Administration (FHWA).

Density: A measure of the concentration of vehicles, stated as the number per unit distance per lane.

Detector Failures: The occurrence of detector malfunctions including nonoperation, chattering, or other intermittently erroneous detections.

Detectors: A system for indicating the presence or passage of vehicles.

Deterministic Model: A mathematical model that enables one to compute precisely what will happen to one variable if a specified value is chosen for another variable. This model has no random variables, and all entity interactions are defined by exact relationships (mathematical, statistical, or logical).

Distance Measuring Instrument (DMI): An electronic device connected to the transmission of a vehicle that can be used to determine travel time along a corridor based on the speed and distance information.

Estimation: Calculation of traffic state variables, for the most recent period for which measurements are available.

Extrapolation Method: Method to calculate travel time from detector data by dividing the distance between the detectors by the speed obtained from the detectors.

Freeway Surveillance: Process or method of monitoring freeway traffic performance and control system operation.

Generalized Reduced Gradient (GRG): A non-linear optimization technique, which can take non-linear objective function and non-linear constraints into account.

Imputation: The process of calculating the missing detector data using techniques such as interpolation.

Inductance: The property of an electric circuit whereby an electromotive force is generated by a change of current.

Inductance Loop Detectors (ILD): Traffic monitoring technique, where wire loops buried below road surface, detect vehicles as they cross the loop, due to change in inductance.

Intelligent Transportation Systems (ITS): Application of advanced technologies and communication methods to the transportation sector to improve the efficiency or safety of a surface transportation system.

Loop Detector Unit: An electronic device which is capable of energizing the sensor loops, of monitoring the sensor loops inductance, and of responding to a pre-determined decrease in inductance with an output which indicates the passage or presence of vehicles zone of detection.

Machine Learning: Machine learning involves adaptive mechanisms that enable computers to learn from experience, learn by example and learn by analogy.

Macroscopic Model: Macroscopic models describe the behavior of the average vehicle driver units in the traffic stream, based on the aggregate behavior of drivers.

Mean Absolute Difference (MAD): A statistical measure used to determine the difference between two sets of data.

Mean Absolute Percentage Error (MAPE): A statistical measure used to determine the error in a set of data in comparison with a correct set of data.

Microscopic Model: Microscopic flow models aim to describe the behavior of individual vehicle driver units with respect to other vehicles in the traffic stream.

Occupancy: The proportion of time period a detector is occupied by vehicles (vehicles are above the detectors).

Prediction/Forecasting: Calculation of future traffic state variables.

Probe Vehicles: Vehicles used for travel time data collection techniques in which the vehicles travels along the corridor for the exclusive purpose of data collection and records travel time data between points of interest.

Route Guidance System (RGS): The use of intelligent transportation systems technologies and communication methods for guiding the vehicles to select the optimum route.

Stochastic Model: A model that uses a random process subjected to probability to formulate the system.

Support Vector Machine (SVM): A recently developed pattern classification and regression technique based on statistical learning theory.

Travel Time: Time to traverse a route between any two points of interest.

Validation: The process to determine whether a model provides an accurate representation of the real-world system under study. It involves comparing the model output to generated analytical solutions or to collected field data.

B.2 ACRONYMS

TABLE B.1. List of Frequently Used Acronyms

Acronym	Title
ANN	Artificial Neural Network
ATIS	Advanced Traveler Information System
ATMS	Advanced Traffic Management System
AVI	Automatic Vehicle Identification
AVL	Automatic Vehicle Location
CORSIM	CORridor SIMulation
DMI	Distance Measuring Instrument
FHWA	Federal HighWays Administration
FRESIM	FRieway SIMulation
GRG	Generalized Reduced Gradient
HCM	Highway Capacity Manual
ILD	Inductance Loop Detector
ITS	Intelligent Transportation System
MAD	Mean Absolute Difference
MAPE	Mean Absolute Percentage Error
NEMA	National Electrical Manufacturers Association
NETSIM	NETwork SIMulation
RGS	Route Guidance System
SVM	Support Vector Machine
SVR	Support Vector Regression
TRB	Transportation Research Board
TSIS	Traffic Software Integration System
TMC	Traffic Management Center
TransGuide	Transportation Guidance System
WIM	Weigh In Motion

APPENDIX C
MICROSCOPIC TRAFFIC SIMULATION

CORSIM INPUT FILES

.TRF File

```

ITRAF 2.0
1 1 5 7981 21 80600 7781 7581
72007200
60
8001 1 2 0 1 1 19
1 2 3 25000 1 1 19
2 3 4 25000 1 1 19
3 48002 25000 1 1 19
8001 1 11070 20
1 2 11070 20
2 3 11070 20
3 4 11070 20
8001 1 2 100 25
1 2 3 100 25
2 3 4 100 25
3 48002 100 25
2 3 1 10 6 15 2 1 28
3 4 1 10 6 15 2 2 28
3 4 12490 6 15 2 3 28
8001 11088 100 50
8001 1 0 001358 101430 301650 601760 90 1 53
10 0 60 20 1 0 64
1 2 3 67
0 170
1 0 8000 195
2 2500 8000 195
3 5000 8000 195
4 7500 8000 195
0 8 210
8001 11848 1201918 1501637 1801390 210 1 53
0 170
1 210

```

```

ITRAF 2.0
1 1 10 7981 21 80600 7781 7581 00
1800180018001800 2
60 3
4
5
8001 1 2 0 1 1 19
1 2 3 25000 1 19
2 3 4 25000 1 19
8003 5 4 1 1 19
5 4 6 5091 1 19
3 4 6 10300 1 19
4 68002 25000 1 91 100 19
8001 1 11070 20
1 2 11070 20
2 3 11070 20
8003 5 11070 20
5 4 11070 20
3 4 11070 20
4 6 11070 20
8001 1 2 100 25
1 2 3 100 25
2 3 4 100 25
8003 5 4 100 25
5 4 6 100 25
3 4 1 10 6 15 2 1 28
3 4 11020 6 15 2 2 28
4 6 11000 6 15 2 3 28
4 6 12490 6 15 2 4 28
5 4 1 400 6 15 2 5 28
8001 11783 100 50
8003 5 50 100 50
8001 1 0 001783 041953 102198 161976 222085 28 1 53
8003 5 0 00 130 04 150 10 170 16 190 22 210 28 1 53
10 0 60 20 1 0 64
1 2 3 4 5 67
0 70
1 0 8000 95
2 2500 8000 95
3 5000 8000 95
4 6030 8000 95
5 5930 7500 95
6 8530 8000 95
0 8 10
8001 11909 342113 402199 461980 521811 58 1 53
8003 5 250 34 270 40 300 46 310 52 300 58 1 53
0 70
0 8 10
8001 11877 642165 701868 761932 821770 88 1 53
8003 5 380 64 300 70 290 76 285 82 220 88 1 53
0 70
0 8 10
8001 11876 961836 1021770 1081767 1141836 120 1 53
8003 5 200 96 180 102 160 108 150 114 140 120 1 53
0 70
1 10

```


APPENDIX D

PROGRAMS DEVELOPED

MATLAB FILES

Data Retrieval

% This program read the data file for the selected day and make a new file with the details of the %interested detectors alone. The input is the data file with the field data and the output will be the details %of the selected detectors specified in the 'deal' command.

```
clear;
```

```
[date, time, det, speed, vol, occ] = textread('PollServerFLaneData1_jan2002.txt', '%s %s %s %s %s %s');
```

```
str1 = strrep(speed, 'Speed=', ''); %remove the letters
```

```
str2 = strrep(vol, 'Vol=', '');
```

```
str3 = strrep(occ, 'Occ=', '');
```

```
[str{1:10}] = deal('L1-0035S-163.421', 'L2-0035S-163.421', 'L3-0035S-163.421', 'EX1-0035S-163.328', 'L1-0035S-162.899', 'L2-0035S-162.899', 'L3-0035S-162.899', 'L1-0035S-162.482', 'L2-0035S-162.482', 'L3-0035S-162.482');
```

```
for m=(1:10)
    itn(m,1) = 0;
end
```

```
for m = 1:10
    [str4] = ('jan2002');
    tempstr1 = char(str(m));
    tempstr2 = char(str4);
    str5 = [tempstr1(end-2:end), '_', tempstr2, '_', tempstr1(1:2), '.txt'];
```

```
fid(m) = fopen(char(str5), 'w');
```

```
    for n=1:length(date)
        if (strmatch(str(m), det(n), 'exact') ) %checking for match
            itn(m,1) = itn(m,1)+1
            str4 = hour(time(n))*3600+minute(time(n))*60+second(time(n));
            fprintf(fid(m), '%s\t %f\t %s\t %s\t %s\t %s\n', date{ n}, str4, det{ n}, str1 { n}, str2 { n}, str3 { n});
        end
    end
    fclose(fid(m));
end
```

Data Averaging

% averages 20 sec actual volume for 2 minute intervals.

clear;

```
[str{1:15}] = deal('500_feb11N_L1.txt','500_feb11N_L2.txt','500_feb11N_L3.txt','998_feb11N_L1.txt',...
'998_feb11N_L2.txt','998_feb11N_L3.txt','504_feb11N_L1.txt','504_feb11N_L2.txt','504_feb11N_L3.txt',
..
'892_feb11N_L1.txt','892_feb11N_L2.txt','892_feb11N_L3.txt','405_feb11N_L1.txt','405_feb11N_L2.txt',
405_feb11N_L3.txt');
```

for zz=1:15

```
    [date, time, detector, speed1, vol1, occ1] = textread(char(str(zz)), '%s %s %s %s %s %s');
```

```
    t = str2num(char(time));
```

```
    speed = str2num(char(speed1));
```

```
    length(speed)
```

```
    vol = str2num(char(vol1));
```

```
    occ = str2num(char(occ1));
```

```
    if (vol(1) == 0)
```

```
        vol(1) = 1;
```

```
    end
```

```
    if (occ(1) == 0)
```

```
        occ(1) = 1;
```

```
    end
```

% check for unreasonable combinations and threshold values

```
for i=1:11
```

```
    itn(i) = 0;
```

```
end
```

```
for n = 2:length(t)
```

```
    if(speed(n) > 0 && speed(n) < 100 && vol(n) > 0 && vol(n) <= 17 && occ(n) > 0 && occ(n) < 90)
```

```
        itn(4) = itn(4) + 1;
```

```
    end
```

```
    if(speed(n) == 0 && vol(n) == 0 && occ(n) == 0)
```

```
        itn(5) = itn(5) + 1;
```

```
    end
```

```
    if(vol(n) > 17)
```

```
        itn(1) = itn(1) + 1;
```

```
        vol(n) = vol(n-1);
```

```
    end
```

```
    if(speed(n) > 100)
```

```
        itn(2) = itn(2) + 1;
```

```
        speed(n) = (speed(n-1));
```

```
    end
```

```

if(occ(n)>90)
    itn(3) = itn(3) +1;
    occ(n) = occ(n-1);
end

if(speed(n) == 0 && vol(n) ~= 0 && occ(n) ~= 0)
    itn(6) = itn(6) +1;
    speed(n) = speed(n-1);
end

if(speed(n) ~= 0&& vol(n) ==0 && occ(n) ~= 0)
    itn(7) = itn(7) +1;
    vol(n) = vol(n-1);
end
if(speed(n) ~= 0 && vol(n) ~= 0 && occ(n) ==0)
    itn(8) = itn(8) +1;
    occ(n) = occ(n-1);
end

if(speed(n) == 0 && vol(n) == 0 && occ(n) ~= 0)
    itn(9) = itn(9) +1;
    speed(n) = speed(n-1);
    vol(n) = vol(n-1);
end
if(speed(n)~= -1)
if(speed(n) ~= 0 && vol(n) == 0 && occ(n) == 0)
    itn(10) = itn(10) +1;
    vol(n) = vol(n-1);
    occ(n) = occ(n-1);
end
end
if(speed(n) == 0 && vol(n) ~= 0 && occ(n) ==0)
    itn(11) = itn(11) +1;
    occ(n) = occ(n-1);
    speed(n) = speed(n-1);
end

check(n,:) = itn;
if(n>2)
    if(check(n,:) == check(n-1,:))
        fprintf('none of the above at %d\n',n);
    end
end

end
itn

%cumulate to 2 mts

start_t(1) = 0; %data collection started at time 0
i=1;
j=1;

```

```

for n = 1:length(t)
    end_t(i) = start_t(i) + 120;
    if t(n) >= end_t(i) % 1st data after 15 mt interval
        if (t(n)-end_t(i) < 20) %within +30
            time_2mt(i,1) = t(n);
            vol_2mt(i,1) = mean(vol(j:n))*6;%vol is the sum for all 6 - 20 sec intervals in the 2mt interval
            if(nnz(speed(j:n)) == 0)
                speed_2mt(i,1) = 0;
            else
                speed_2mt(i,1) = sum(speed(j:n))/nnz(speed(j:n));%average of all non zero speeds
            end
            occ_2mt(i,1) = sum(occ(j:n))/6;%occupancy calculated for the 2 mt from the 20 sec
            %occ is percentage value and hence each number to be
            %multiplied by 20 and divide by 100 to get the actual time
            %occupied. then sum it up and divide by 120 and make it
            %percent. the whole calculation comes out as divide by 6.
            start_t(i+1) = end_t(i);
            i = i+1;
            j = n;
            continue
        end
        if(n~=1)
            if (abs(end_t(i)-t(n-1))<20) % within one time step
                time_2mt(i,1) = t(n-1);
                vol_2mt(i,1) = mean(vol(j:n))*6;%vol is the sum for all 6 - 20 sec intervals in the 2mt interval
                if(nnz(speed(j:n)) == 0)
                    speed_2mt(i,1) = 0;
                else
                    speed_2mt(i,1) = sum(speed(j:n))/nnz(speed(j:n));%average of all non zero speeds
                end
                occ_2mt(i,1) = sum(occ(j:n))/6;
                start_t(i+1) = end_t(i);
                n=n-1;
                i = i+1;
                j = n;
                continue
            end
        end
        if t(n) >= end_t(i) + 120 %if the time is more than 4 mt interval
            x = (t(n)-end_t(i))/120;
            y = round(x);
            if(vol(n-1,1)>0 & vol(n,1) > 2*vol(n-1,1))
                for (z=i:i+y)
                    time_2mt(z,1) = end_t(z);
                    vol_2mt(z,1) = vol(n,1)/y;
                    speed_2mt(z,1) = (speed(n,1)+speed(n-1,1))/2;
                    occ_2mt(z,1) = (occ(n,1)+occ(n-1,1))/2;
                    start_t(z+1) = end_t(z);
                    end_t(z+1) = start_t(z+1) + 120;
                end
            end
        end
    end
end

```

```

for (z=i:(i+y))
    time_2mt(z,1) = end_t(z);
    vol_2mt(z,1) = ((vol(n-1,1)+vol(n,1))/2)*6;
    speed_2mt(z,1) = (speed(n-1,1)+speed(n,1))/2;
    occ_2mt(z,1) = (occ(n-1,1)+occ(n,1))/2;
    start_t(z+1) = end_t(z);
    end_t(z+1) = start_t(z+1) + 120;
end

end_t(z+1) = 0;
i=z+1;
j=n;
continue
end

time_2mt(i,1) = t(n); % otherwise
vol_2mt(i,1) = mean(vol(j:n))*6; %vol is the sum for all 6 - 20 sec intervals in the 2mt interval
if(nnz(speed(j:n)) == 0)
    speed_2mt(i,1) = 0;
else
    speed_2mt(i,1) = sum(speed(j:n))/nnz(speed(j:n)); %average of all non zero speeds
end
occ_2mt(i,1) = sum(occ(j:n))/6;
start_t(i+1) = end_t(i);
i = i+1;
j = n;
end
end

tempstr = char(str{zz});
str1 = [tempstr(1:end-4),'_2mt.txt'];

fid = fopen(char(str1),'w');
for k=1:length(vol_2mt)
    fprintf(fid,'%f\t%f\t%f\t%f\t%f\t%f\n',start_t(k), end_t(k), time_2mt(k), speed_2mt(k),
vol_2mt(k), occ_2mt(k));
end
fclose(fid);
end %for zz loop

```

AVI Data

% File to get travel time from AVI data file between two selected points. Input is the tag data of only AVI % stations . In this example AVI number 142 and 144 sorted based on vehicle ID and then on avi station % number is given as input and the travel time of vehicles is obtained as output.

```
clear;
```

```
[AVInum, vehid, time1, date] = textread(char('feb11_avi.txt'),'%s %s %s %s ', 'whitespace','t');
```

```
AVI1 = (char(AVInum));
```

```
AVI = str2num(AVI1);
```

```
t = strrep(time1, '&', ''); %remove the $ from time
```

```
for n = 1:length(date)
```

```
    n
```

```
    time(n) = hour(t(n))*3600+minute(t(n))*60+second(t(n));
```

```
end
```

```
fid = fopen('avi_tt.txt','w');
```

```
for n = 2:length(date)
```

```
    if ((AVI(n) == 144) && (AVI(n-1) == 142))% for loops 159-164
```

```
        b(n)=1;
```

```
        if (strmatch(vehid(n), vehid(n-1),'exact'))
```

```
            a(n)=1;
```

```
            if (a(n)==1 && b(n) == 1)
```

```
                tt(n) = time(n)-time(n-1);
```

```
                if (tt(n) > 0 && tt(n) < 1800) %assuming a 10mph min speed
```

```
                    n
```

```
                    fprintf(fid,'%s\t %s\t %f\t %s\t %f\t %f\n', vehid{n}, t{n-1}, time(n-1), t{n}, time(n), tt(n)
```

```
);
```

```
                end
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
fclose(fid);
```

Optimization

%The program for optimizing three detectors data. Input is the cumulative flow at three consecutive detectors and the out is the corresponding optimized values.

```

clear
format compact
format short e

%*****
%* define analytical functions
%* remember to use vectors for g and h if more than one of them
%* and modify code
%*****
syms f g1 g2 g3 g4 g5 g6 g7 c1i c2i c3i c1j c2j c3j x1 x2 x3 x4 x5 x6 x7
syms gradc1i gradc2i gradc3i gradc1j gradc2j gradc3j
syms gradx1 gradx2 gradx3 gradx4 gradx5 gradx6 gradx7
syms h1 h1c1i h1c2i h1c3i h1c1j h1c2j h1c3j h1x1 h1x2 h1x3 h1x4 h1x5 h1x6 h1x7
syms h2 h2c1i h2c2i h2c3i h2c1j h2c2j h2c3j h2x1 h2x2 h2x3 h2x4 h2x5 h2x6 h2x7
syms h3 h3c1i h3c2i h3c3i h3c1j h3c2j h3c3j h3x1 h3x2 h3x3 h3x4 h3x5 h3x6 h3x7
syms h4 h4c1i h4c2i h4c3i h4c1j h4c2j h4c3j h4x1 h4x2 h4x3 h4x4 h4x5 h4x6 h4x7
syms h5 h5c1i h5c2i h5c3i h5c1j h5c2j h5c3j h5x1 h5x2 h5x3 h5x4 h5x5 h5x6 h5x7
syms h6 h6c1i h6c2i h6c3i h6c1j h6c2j h6c3j h6x1 h6x2 h6x3 h6x4 h6x5 h6x6 h6x7
syms h7 h7c1i h7c2i h7c3i h7c1j h7c2j h7c3j h7x1 h7x2 h7x3 h7x4 h7x5 h7x6 h7x7

% the functions
f = ((c1j-c2j)^2 + (c2j-c3j)^2);
g1 = c1j-c2j;
h1 = g1-x1;
g2 = c1j-c2j-500;
h2 = g2 + x2;
g3 = c2j-c3j;
h3 = g3-x3;
g4 = c2j-c3j-500;
h4 = g4 + x4;
g5 = c1j - c1i;
h5 = g5 - x5;

g6 = c2j - c2i;
h6 = g6 - x6;

g7 = c3j - c3i;
h7 = g7 - x7;

%*****
% input the design vector

load 'data.txt'
data1 = data(:,1); %4th column is the L1 cum volume
data2 = data(:,2); % L2 cumu vol
data3 = data(:,3); %L3 cumu vol
count = 0;

```

```

count1 = 0;

for n = 1:length(data)
    if ( ((data1(n,1)-data2(n,1))>0) && ((data1(n,1)-data2(n,1))<500) && ((data2(n,1)-data3(n,1))>0) &&
        ((data2(n,1)-data3(n,1))<500))
        status(n) = 1;
    else
        status(n) = 0;
    end
end
check = min(status);
for n = 1:length(data)
    if(check == 1)
        fprintf('no need for optimization, the data is good:-)n');
        break
    end

    for(i=1:13)
        xn(1,i) = -1;
    end

    fn(1) = 1;
    flag = 1;
    itn = 1;
    diffF(1) = 1;
    threshold = -1e-4;
    threshold1 = 1e-4;

    while (itn <40 & fn(itn) > threshold1 & min(xn(itn,:)) <threshold)
        if (flag == 1 & n==1)
            xs = [data1(n) data2(n) data3(n) 0 0 0];
        elseif (flag == 1 & n>1)
            xs = [data1(n) data2(n) data3(n) optmddata(n-1,1) optmddata(n-1,2) optmddata(n-1,3)];
        elseif (flag~=1 & n>1)
            xs = [xn(itn,1) xn(itn,2) xn(itn,3) optmddata(n-1,1) optmddata(n-1,2) optmddata(n-1,3)];
        else
            xs = [xn(itn,1) xn(itn,2) xn(itn,3) 0 0 0];
        end

        n
        itn

        if(itn > 1)
            if (diffF(itn-1) < threshold1 & diffF(itn) < threshold1)
                break
            end
        end

        xs(7) = subs(g1, {c11j,c12j}, {xs(1),xs(2)});
        xs(8) = -subs(g2, {c11j,c12j}, {xs(1),xs(2)});

```



```

xs(9) = subs(g3, {cl2j, cl3j}, {xs(2), xs(3)});
xs(10) = -subs(g4, {cl2j, cl3j}, {xs(2), xs(3)});
xs(11) = subs(g5, {cl1j, cl1i}, {xs(1), xs(4)});
xs(12) = subs(g6, {cl2j, cl2i}, {xs(2), xs(5)});
xs(13) = subs(g7, {cl3j, cl3i}, {xs(3), xs(6)});

%fprintf('\n\nThe start design vector [%10.4f %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f %10.4f
%10.4f %10.4f %10.4f %10.4f %10.4f ]\n', xs);

% the gradients
gradcl1j = diff(f, cl1j);
gradcl2j = diff(f, cl2j);
gradcl3j = diff(f, cl3j);

h1cl1j = diff(h1, cl1j);
h1cl2j = diff(h1, cl2j);
h1x1 = diff(h1, x1);

h2cl1j = diff(h2, cl1j);
h2cl2j = diff(h2, cl2j);
h2x2 = diff(h2, x2);

h3cl2j = diff(h3, cl2j);
h3cl3j = diff(h3, cl3j);
h3x3 = diff(h3, x3);

h4cl2j = diff(h4, cl2j);
h4cl3j = diff(h4, cl3j);
h4x4 = diff(h4, x4);

h5cl1j = diff(h5, cl1j);
h5cl1i = diff(h5, cl1i);
h5x5 = diff(h5, x5);

h6cl2j = diff(h6, cl2j);
h6cl2i = diff(h6, cl2i);
h6x6 = diff(h6, x6);

h7cl3j = diff(h7, cl3j);
h7cl3i = diff(h7, cl3i);
h7x7 = diff(h7, x7);

% evaluate the function, gradients, and hessian at the current design
fn(1) = double(subs(f, {cl1j, cl2j, cl3j}, {xs(1), xs(2), xs(3)}));
g1v = double(subs(g1, {cl1j, cl2j}, {xs(1), xs(2)}));
h1v = double(subs(h1, {cl1j, cl2j, x1}, {xs(1), xs(2), xs(7)}));
g2v = double(subs(g2, {cl1j, cl2j}, {xs(1), xs(2)}));
h2v = double(subs(h2, {cl1j, cl2j, x2}, {xs(1), xs(2), xs(8)}));
g3v = double(subs(g3, {cl2j, cl3j}, {xs(2), xs(3)}));
h3v = double(subs(h3, {cl2j, cl3j, x3}, {xs(2), xs(3), xs(9)}));
g4v = double(subs(g4, {cl2j, cl3j}, {xs(2), xs(3)}));
h4v = double(subs(h4, {cl2j, cl3j, x4}, {xs(2), xs(3), xs(10)}));

```

```

g5v = double(subs(g5, {cl1j, cl1i}, {xs(1), xs(4)}));
h5v = double(subs(h5, {cl1j, cl1i, x5}, {xs(1), xs(4), xs(11)}));
g6v = double(subs(g6, {cl2j, cl2i}, {xs(2), xs(5)}));
h6v = double(subs(h6, {cl2j, cl2i, x6}, {xs(2), xs(5), xs(12)}));
g7v = double(subs(g7, {cl3j, cl3i}, {xs(3), xs(6)}));
h7v = double(subs(h7, {cl3j, cl3i, x7}, {xs(3), xs(6), xs(13)}));

%fprintf('\n start function and constraints(f h1 h2 h3 h4 h5 h6 h7):\n '), disp([fn(1) h1v h2v h3v h4v
h5v h6v h7v])

dfcl1j = double(subs(gradcl1j, {cl1j, cl2j, cl3j}, {xs(1), xs(2), xs(3)}));
dfcl2j = double(subs(gradcl2j, {cl1j, cl2j, cl3j}, {xs(1), xs(2), xs(3)}));
dfcl3j = double(subs(gradcl3j, {cl1j, cl2j, cl3j}, {xs(1), xs(2), xs(3)}));
dfcl1i = 0;
dfcl2i = 0;
dfcl3i = 0;
dfx1 = 0;
dfx2 = 0;
dfx3 = 0;
dfx4 = 0;
dfx5 = 0;
dfx6 = 0;
dfx7 = 0;

dh1cl1j = double(subs(h1cl1j, {cl1j, cl2j, cl3j, x1}, {xs(1), xs(2), xs(3), xs(7)}));
dh1cl2j = double(subs(h1cl2j, {cl1j, cl2j, cl3j, x1}, {xs(1), xs(2), xs(3), xs(7)}));
dh1x1 = double(subs(h1x1, {cl1j, cl2j, cl3j, x1}, {xs(1), xs(2), xs(3), xs(7)}));
dh1cl3j = 0;
dh1cl1i = 0;
dh1cl2i = 0;
dh1cl3i = 0;
dh1x2 = 0;
dh1x3 = 0;
dh1x4 = 0;
dh1x5 = 0;
dh1x6 = 0;
dh1x7 = 0;

dh2cl1j = double(subs(h2cl1j, {cl1j, cl2j, cl3j, x2}, {xs(1), xs(2), xs(3), xs(8)}));
dh2cl2j = double(subs(h2cl2j, {cl1j, cl2j, cl3j, x2}, {xs(1), xs(2), xs(3), xs(8)}));
dh2x2 = double(subs(h2x2, {cl1j, cl2j, cl3j, x2}, {xs(1), xs(2), xs(3), xs(8)}));
dh2cl3j = 0;
dh2cl1i = 0;
dh2cl2i = 0;
dh2cl3i = 0;
dh2x1 = 0;
dh2x3 = 0;
dh2x4 = 0;
dh2x5 = 0;
dh2x6 = 0;
dh2x7 = 0;

dh3cl2j = double(subs(h3cl2j, {cl1j, cl2j, cl3j, x3}, {xs(1), xs(2), xs(3), xs(9)}));

```

```

dh3cl3j = double(subs(h3cl3j, {cl1j,cl2j,cl3j,x3}, {xs(1),xs(2),xs(3),xs(9)}));
dh3x3 = double(subs(h3x3, {cl1j,cl2j,cl3j,x3}, {xs(1),xs(2),xs(3),xs(9)}));
dh3cl1j =0;
dh3cl1i =0;
dh3cl2i =0;
dh3cl3i =0;
dh3x2 =0;
dh3x1 =0;
dh3x4 =0;
dh3x5 =0;
dh3x6 =0;
dh3x7 =0;

dh4cl2j = double(subs(h4cl2j, {cl1j,cl2j,cl3j,x4}, {xs(1),xs(2),xs(3),xs(10)}));
dh4cl3j = double(subs(h4cl3j, {cl1j,cl2j,cl3j,x4}, {xs(1),xs(2),xs(3),xs(10)}));
dh4x4 = double(subs(h4x4, {cl1j,cl2j,cl3j,x4}, {xs(1),xs(2),xs(3),xs(10)}));
dh4cl1j =0;
dh4cl1i =0;
dh4cl2i =0;
dh4cl3i =0;
dh4x2 =0;
dh4x3 =0;
dh4x1 =0;
dh4x5 =0;
dh4x6 =0;
dh4x7 =0;

dh5cl1j = double(subs(h5cl1j, {cl1j,cl1i,x5}, {xs(1),xs(4),xs(11)}));
dh5cl1i = double(subs(h5cl1i, {cl1j,cl1i,x5}, {xs(1),xs(4),xs(11)}));
dh5x5 = double(subs(h5x5, {cl1j,cl1i,x5}, {xs(1),xs(4),xs(11)}));
dh5cl2j =0;
dh5cl3j =0;
dh5cl2i =0;
dh5cl3i =0;
dh5x2 =0;
dh5x3 =0;
dh5x4 =0;
dh5x1 =0;
dh5x6 =0;
dh5x7 =0;

dh6cl2j = double(subs(h6cl2j, {cl2j,cl2i,x6}, {xs(2),xs(5),xs(12)}));
dh6cl2i = double(subs(h6cl2i, {cl2j,cl2i,x6}, {xs(2),xs(5),xs(12)}));
dh6x6 = double(subs(h6x6, {cl2j,cl2i,x6}, {xs(2),xs(5),xs(12)}));
dh6cl1j =0;
dh6cl3j =0;
dh6cl1i =0;
dh6cl3i =0;
dh6x2 =0;
dh6x3 =0;
dh6x4 =0;
dh6x5 =0;
dh6x1 =0;

```

```

dh6x7 =0;

dh7cl3j = double(subs(h7cl3j, {cl3j,cl3i,x7}, {xs(3),xs(6),xs(13)}));
dh7cl3i = double(subs(h7cl3i, {cl3j,cl3i,x7}, {xs(3),xs(6),xs(13)}));
dh7x7 = double(subs(h7x7, {cl3j,cl3i,x7}, {xs(3),xs(6),xs(13)}));
dh7cl1j =0;
dh7cl2j =0;
dh7cl1i =0;
dh7cl2i =0;
dh7x2 =0;
dh7x3 =0;
dh7x4 =0;
dh7x5 =0;
dh7x6 =0;
dh7x1 =0;

%matrix A and B
A = [dh1cl1j dh1cl2j dh1cl3j dh1cl1i dh1cl2i dh1cl3i; dh2cl1j dh2cl2j dh2cl3j dh2cl1i dh2cl2i
dh2cl3i;
dh3cl1j dh3cl2j dh3cl3j dh3cl1i dh3cl2i dh3cl3i; dh4cl1j dh4cl2j dh4cl3j dh4cl1i dh4cl2i dh4cl3i;
dh5cl1j dh5cl2j dh5cl3j dh5cl1i dh5cl2i dh5cl3i; dh6cl1j dh6cl2j dh6cl3j dh6cl1i dh6cl2i dh6cl3i;
dh7cl1j dh7cl2j dh7cl3j dh7cl1i dh7cl2i dh7cl3i];
B = [dh1x1 dh1x2 dh1x3 dh1x4 dh1x5 dh1x6 dh1x7; dh2x1 dh2x2 dh2x3 dh2x4 dh2x5 dh2x6
dh2x7;
dh3x1 dh3x2 dh3x3 dh3x4 dh3x5 dh3x6 dh3x7; dh4x1 dh4x2 dh4x3 dh4x4 dh4x5 dh4x6 dh4x7;
dh5x1 dh5x2 dh5x3 dh5x4 dh5x5 dh5x6 dh5x7; dh6x1 dh6x2 dh6x3 dh6x4 dh6x5 dh6x6 dh6x7;
dh7x1 dh7x2 dh7x3 dh7x4 dh7x5 dh7x6 dh7x7];
C = inv(B)*A;
Gr1 = ([dfcl1j;dfcl2j;dfcl3j;dfcl1i; dfcl2i;dfcl3i] - C*[dfx1; dfx2; dfx3; dfx4; dfx5; dfx6; dfx7]);
S1 = -Gr1;

alpha = 0;

for jj = 1:3

    if (jj < 3)
        %string1 = ['\nInput the stepsize for evaluation.\n'];
        %alpha = input(string1)
        alpha = alpha + 0.05;
    aa(jj+1) = alpha;
    end

%*****
% for a given stepsize - Y calculation
%*****

dz1 = S1*alpha;

xn1 = xs(1) + dz1(1);
xn2 = xs(2) + dz1(2);
xn3 = xs(3) + dz1(3);
xn4 = xs(4);%+ dz1(4);
xn5 = xs(5);%+ dz1(5);

```

```

xn6 = xs(6);%+ dz1(6);

dy1 = -C*dz1;

    yn1 = xs(7);
    yn2 = xs(8);
    yn3 = xs(9);
    yn4 = xs(10);
    yn5 = xs(11);
    yn6 = xs(12);
    yn7 = xs(13);

        for i = 1: 40
            yn1 = yn1 + dy1(1);
            yn2 = yn2 + dy1(2);
            yn3 = yn3 + dy1(3);
            yn4 = yn4 + dy1(4);
            yn5 = yn5 + dy1(5);
            yn6 = yn6 + dy1(6);
            yn7 = yn7 + dy1(7);

            xxn=[xn1 xn2 xn3 xn4 xn5 xn6 yn1 yn2 yn3 yn4 yn5 yn6 yn7];

            h1n = double(subs(h1, {cl1j,cl2j,x1}, {xxn(1),xxn(2),xxn(7)}));
            h2n = double(subs(h2, {cl1j,cl2j,x2}, {xxn(1),xxn(2),xxn(8)}));
            h3n = double(subs(h3, {cl2j,cl3j,x3}, {xxn(2),xxn(3),xxn(9)}));
            h4n = double(subs(h4, {cl2j,cl3j,x4}, {xxn(2),xxn(3),xxn(10)}));
            h5n = double(subs(h5, {cl1j,cl1i,x5}, {xxn(1),xxn(4),xxn(11)}));
            h6n = double(subs(h6, {cl2j,cl2i,x6}, {xxn(2),xxn(5),xxn(12)}));
            h7n = double(subs(h7, {cl3j,cl3i,x7}, {xxn(3),xxn(6),xxn(13)}));

            hsq = h1n*h1n + h2n*h2n + h3n*h3n + h4n*h4n + h5n*h5n + h6n*h6n + h7n*h7n;
                if hsq <= 1.0e-08
                    break
                else
                    dy1 = inv(B)*[-h1n -h2n -h3n -h4n -h5n -h6n -h7n]';
                end
            end
            %fprintf("\nNo. of iterations of dy for same dz, alpha and constraint error: '),disp(i),disp([alpha
hsq]);
            %fprintf("\n improved design vector: '),disp(xxn)
            fn(itn+1) = double(subs(f, {cl1j,cl2j,cl3j}, {xxn(1),xxn(2),xxn(3)}));
            %fprintf("\n improved function and constraints (f h1 h2 h3 h4 h5 h6 h7)n '),disp([fn(itn+1) h1n
h2n h3n h4n h5n h6n h7n])

            flag = 2;

                ff(jj+1)=fn(itn+1);

                    if (jj == 2)
                        aa(1) = 0; rhs(1) = fn(1);
                        amat = [1 0 0; 1 aa(2) aa(2)^2; 1 aa(3) aa(3)^2];

```

```

    rhs=[fn(1) ff(2) ff(3)];
    xval = inv(amat)*rhs;
    alpha = -xval(2)/(2*xval(3));
    %alpha = .05; %0.25
end

end % jj loop
%fprintf('\n improved design vector: '),disp(xxn)
%xxn(4) = xs(4);
%xxn(5) = xs(5);
%xxn(6) = xs(6);

xn(itn+1,:) = xxn;

F(1) = fn(1);
F(itn+1) = fn(itn+1);
diffF(itn+1) = abs(F(itn+1) - F(itn));

itn = itn + 1;

end % while loop

if (min(xn(itn,:)) < 0)
    count1 = count1 + 1;
    fid1 = fopen('infeasible1.txt','a');
    fprintf(fid1,'%d %d\n', count1,n);
    fclose(fid1);
end

%fprintf('\n final design vector: '),disp(xn(itn,:))
%fprintf('\n final function and constraints (f h1 h2 h3 h4 h5 h6)\n '),disp([fn(itn) h1n h2n h3n h4n h5n
h6n])
newx(n,:) = xn(itn,:);
for k = 1:3
    optmddata(n,k) = newx(n,k);
end

save feb10th_345with50_optmd.txt optmddata -ascii

end % for loop for number of data

%optmddata
%plot(optmddata)

```

N-D Model

% Program to calculate travel time based on Nam and Drew model. The input is the flow values at % consecutive points and the output will be the travel time.

```

clear;
format compact
format short e

load 'voldataformodel.txt' % read data

vol1 = voldataformodel(:,1); % speed
vol2 = voldataformodel(:,2);
vol3 = voldataformodel(:,3);

delta_x1 = .522; % given that section is ~.5 miles
delta_x2 = .417;
delta_t = 2; % choose based on delta_x/free_flow_speed relation
%cumu = 4; % 30 sec data to be cumulated to 2 mt data. hence 4 set has to be added
no_in_start1 = -2; % not known. trial and error and choose the best number
no_in_start2 = -4;

for i=1:length(voldataformodel)
    %act_vol(i,:)=temp((i-1)*2+1:i*2)'; %given data
    act_vol(i,1) = vol1(i);
    act_vol(i,2) = vol2(i);
    act_vol(i,3) = vol3(i);
end

figure
plot(act_vol) % to check any unreasonable data
title('actual volume')

for i=1:length(voldataformodel)
    vol(i,:)= act_vol(i,:);
end
figure
plot(vol)
title('aggregated actual volume')

for i = 1:length(voldataformodel)
    q(i,:) = vol(i,:)/delta_t;%number per unit time
end
figure
plot(q)
title('q')

cum_vol(1,1) = q(1,1);
cum_vol(1,2)=(no_in_start1/2) + q(1,2);%cumulated q
cum_vol(1,3) = (no_in_start2/2) + q(1,3);

```

```

for i=2:length(voldataformodel)
    cum_vol(i,:)=q(i,:)+cum_vol(i-1,:);
end
figure
plot(cum_vol)
title('cumulated q')

for i=1:length(voldataformodel)
    Q(i,:)=cum_vol(i,:)*delta_t;%cumulated volume(not per unit time)
end
figure
plot(Q)
title('Q')

for i= 1:length(voldataformodel)
    no_in_link1(i,1)=Q(i,1)-Q(i,2);
    no_in_link2(i,1)=Q(i,2)-Q(i,3);
    density(i,1)=no_in_link1(i,1)/delta_x1;
    density(i,2)=no_in_link2(i,1)/delta_x2;
end

figure
plot(density)
title('density')

m(1,1)=Q(1,2);
m(1,2)=Q(1,3);
m_percent(1,1)=m(1,1)/vol(1,1);
m_percent(1,2)=m(1,2)/vol(1,2);

for i=2:length(voldataformodel)
    m(i,1)=Q(i,2)-Q(i-1,1);
    m(i,2)=Q(i,3)-Q(i-1,2);
    m_percent(i,1)=m(i,1)/vol(i,1);
    m_percent(i,2)=m(i,2)/vol(i,2);
end
figure
plot(m(:,1))
hold on
plot(m(:,2),'-r')
title('m')

for i =2:length(voldataformodel)
    if (m(i,1)>0)
        %tt_mts(i,1) = (delta_x1/2)*((q(i,1)*density(i-1,1))+q(i,2)*density(i,1))/(q(i,1)*q(i,2)); %drew's
original
        tt_mts(i,1) = m_percent(i,1)*((delta_x1/2)*((q(i,1)*density(i-
1,1))+q(i,2)*density(i,1)))/(q(i,1)*q(i,2)) + (1-m_percent(i,1))*((delta_x1/2)*((density(i-
1,1)+density(i,1))/q(i,2)));
    else
        tt_mts(i,1) = ((delta_x1/2)*((density(i-1,1)+density(i,1))/q(i,2))); % same eqn can be written as the
next line
    end
end

```



```

    %tt_mts(i,1) = ((2*(density(i-1,1)*delta_x))+((q(i,1)-q(i,2))*delta_t))/(2*q(i,2));
end
if(m(i,2) > 0)
    %tt_mts(i,2) = (delta_x/2)*((q(i,2)*density(i-1,2))+q(i,3)*density(i,2))/(q(i,3)*q(i,2));%drew's
original
    tt_mts(i,2) = m_percent(i,2)*((delta_x/2)*((q(i,2)*density(i-
1,2))+q(i,3)*density(i,2)))/(q(i,2)*q(i,3)) + (1-m_percent(i,2))*((delta_x/2)*((density(i-
1,2)+density(i,2))/q(i,3)));
    else
        tt_mts(i,2) = ((delta_x/2)*((density(i-1,2)+density(i,2))/q(i,3))); % same eqn can be written as the
next line
        %tt_mts(i,2) = ((2*(density(i-1,2)*delta_x))+((q(i,2)-q(i,3))*delta_t))/(2*q(i,3));
    end
end

% smoothing of the data
alpha1 = 0.3; % can be varied, do trial and error
alpha2=0.3;
tt_smoothed(2,1) = tt_mts(2,1);
tt_smoothed(2,2) = tt_mts(2,2);

for n=3:length(voldataformodel)
    tt_smoothed(n,1) = alpha1 *tt_mts(n,1)+(1-alpha1)*tt_smoothed(n-1,1);
    tt_smoothed(n,2) = alpha2*tt_mts(n,2)+(1-alpha2)*tt_smoothed(n-1,2);
end

%for i =2:length(dataformodel)/cumu,
% tt_t(i,1) = ((delta_x/2)*((q(i,1)*density(i-1,1))+q(i,2)*density(i,1)))/(q(i,1)*q(i,2));
%end

figure
plot(tt_mts(:,1)*60,'-g')
hold on
plot( tt_mts(:,2)*60,'-b')

title('travel time in seconds')

figure
plot(tt_smoothed(:,1)*60,'-r')
hold on
plot(tt_smoothed(:,2)*60, '-y')

title('smoothed travel time')

save tt_model.txt tt_mts -ascii
save tt_model_smoothed.txt tt_smoothed -ascii

```

Travel Time Estimation

% Program to calculate travel time based on the model proposed in this dissertation. Input includes the %flow, speed and density at consecutive points and the travel time will be the output.

```

clear;
format compact
format short e

load 'voldataformodel.txt' % read data
load 'densitydataformodel.txt'
load 'speeddataformodel.txt' % read data

speed1 = speeddataformodel(:,1); % speed
speed2 = speeddataformodel(:,2);
speed3 = speeddataformodel(:,3);
speed4 = speeddataformodel(:,4);
speed5 = speeddataformodel(:,5);

vol1 = voldataformodel(:,1); % speed
vol2 = voldataformodel(:,2);
vol3 = voldataformodel(:,3);
vol4 = voldataformodel(:,4);
vol5 = voldataformodel(:,5);

density1 = densitydataformodel(:,1);%no_in_link1(i,1)/delta_x;
density2 = densitydataformodel(:,2);%no_in_link2(i,1)/delta_x;
density3 = densitydataformodel(:,3);%no_in_link1(i,1)/delta_x;
density4 = densitydataformodel(:,4);%no_in_link2(i,1)/delta_x;

delta_x1 = .498; % given that section is ~ .5 miles
delta_x2 = .506; % .47;
delta_x3 = .388;
delta_x4 = .513;

delta_t = 2; % choose based on delta_x/free_flow_speed relation
%cumu = 4; % 30 sec data to be cumulated to 2 mt data. hence 4 set has to be added
no_in_start1 = 0; % not known. trial and error and choose the best number
no_in_start2 = 0;
no_in_start3 = 0; % not known. trial and error and choose the best number
no_in_start4 = 0;

for i=1:length(voldataformodel)
    %act_vol(i,:)=temp((i-1)*2+1:i*2)'; %given data
    act_vol(i,1) = vol1(i);
    act_vol(i,2) = vol2(i);
    act_vol(i,3) = vol3(i);
    act_vol(i,4) = vol4(i);
    act_vol(i,5) = vol5(i);
end

```

```

figure
plot(act_vol) % to check any unreasonable data
title('actual volume')

for i=1:length(voldataformodel)
    vol(i,:)= act_vol(i,:);
end
figure
plot(vol)
title('aggregated actual volume')

for i = 1:length(voldataformodel)
    q(i,:) = vol(i,:)/delta_t;%number per unit time
end
figure
plot(q)
title('q')

cum_vol(1,1) = q(1,1);
cum_vol(1,2)=(no_in_start1/2) + q(1,2);%cumulated q
cum_vol(1,3) = (no_in_start2/2) + q(1,3);
cum_vol(1,4)=(no_in_start3/2) + q(1,4);%cumulated q
cum_vol(1,5) = (no_in_start4/2) + q(1,5);

for i=2:length(voldataformodel)
    cum_vol(i,:)=q(i,:) + cum_vol(i-1,:);
end
figure
plot(cum_vol)
title('cumulated q')

for i=1:length(voldataformodel)
    Q(i,:) = cum_vol(i,:)*delta_t;%cumulated volume(not per unit time)
end
figure
plot(Q)
title('Q')

for i= 1:length(voldataformodel)
    no_in_link1(i,1) = Q(i,1)-Q(i,2);
    no_in_link2(i,1) = Q(i,2)-Q(i,3);
    no_in_link3(i,1) = Q(i,3)-Q(i,4);
    no_in_link4(i,1) = Q(i,4)-Q(i,5);
end

figure
plot(density1, '-b')
hold on
plot(density2, '-r')
hold on
plot(density3, '-g')

```

```

hold on
plot(density4, '-y')
title('density')

m(1,1) = Q(1,2);
m(1,2) = Q(1,3);
m(1,3) = Q(1,4);
m(1,4) = Q(1,5);

m_percent(1,1) = m(1,1)/vol(1,1);
m_percent(1,2) = m(1,2)/vol(1,2);
m_percent(1,3) = m(1,3)/vol(1,3);
m_percent(1,4) = m(1,4)/vol(1,4);

for i=2:length(voldataformodel)
    m(i,1)=Q(i,2)-Q(i-1,1);
    m(i,2) = Q(i,3) - Q(i-1,2);
    m(i,3)=Q(i,4)-Q(i-1,3);
    m(i,4) = Q(i,5) - Q(i-1,4);

    m_percent(i,1)= m(i,1)/vol(i,1);
    m_percent(i,2)= m(i,2)/vol(i,2);
    m_percent(i,3)= m(i,3)/vol(i,3);
    m_percent(i,4)= m(i,4)/vol(i,4);
end
figure
plot(m(:,1))
hold on
plot(m(:,2),'-r')
hold on
plot(m(:,3),'-g')
hold on
plot(m(:,4),'-y')
title('m')

for i = 1:length(speeddataformodel)
    if (act_vol(i,:) < 50)
        %tt1_method1(i) = ((delta_x/(2*speed1(i)))+(delta_x/(2*speed2(i))))*3600;
        tt_mts(i,1) = delta_x/((speed1(i)+speed2(i))/2)*60;
        %tt1_method3(i) = delta_x/(min(speed1(i),speed2(i)))*3600;
        %tt2_method1(i) = ((delta_x/(2*speed2(i)))+(delta_x/(2*speed3(i))))*3600;
        tt_mts(i,2) = delta_x/((speed2(i)+speed3(i))/2)*60;
        %tt2_method3(i) = delta_x/(min(speed2(i),speed3(i)))*3600;
        tt_mts(i,3) = delta_x/((speed3(i)+speed4(i))/2)*60;
        tt_mts(i,4) = delta_x/((speed4(i)+speed5(i))/2)*60;
    else
        if (m(i,1) > 0)

            tt_mts(i,1) = m_percent(i,1)*((delta_x/2)*((q(i,1)*density1(i-1,1))+(q(i,2)*density1(i,1))))/(q(i,1)*q(i,2)) + (1-m_percent(i,1))*((delta_x/2)*((density1(i-1,1)+density1(i,1))/q(i,2)));

```

```

else
    tt_mts(i,1) = ((delta_x1/2)*((density1(i-1,1)+density1(i,1))/q(i,2))); % same eqn can be written
    as the next line
    %tt_mts(i,1) = ((2*(density(i-1,1)*delta_x))+((q(i,1)-q(i,2))*delta_t))/(2*q(i,2));
end
if(m(i,2)>0)

    tt_mts(i,2) = m_percent(i,2)*((delta_x2/2)*((q(i,2)*density2(i-
    1,1))+q(i,3)*density2(i,1)))/(q(i,2)*q(i,3)) + (1-m_percent(i,2))*((delta_x2/2)*((density2(i-
    1,1)+density2(i,1))/q(i,3)));
    else
    tt_mts(i,2) = ((delta_x2/2)*((density2(i-1,1)+density2(i,1))/q(i,3))); % same eqn can be written as
    the next line
    %tt_mts(i,2) = ((2*(density(i-1,2)*delta_x))+((q(i,2)-q(i,3))*delta_t))/(2*q(i,3));
end

if(m(i,3)>0)

    tt_mts(i,3) = m_percent(i,3)*((delta_x3/2)*((q(i,3)*density3(i-
    1,1))+q(i,4)*density3(i,1)))/(q(i,3)*q(i,4)) + (1-m_percent(i,3))*((delta_x3/2)*((density3(i-
    1,1)+density3(i,1))/q(i,4)));
    else
    tt_mts(i,3) = ((delta_x3/2)*((density3(i-1,1)+density3(i,1))/q(i,4))); % same eqn can be written as
the next line
end

if(m(i,4)>0)

    tt_mts(i,4) = m_percent(i,4)*((delta_x4/2)*((q(i,4)*density4(i-
    1,1))+q(i,5)*density4(i,1)))/(q(i,4)*q(i,5)) + ...
    (1-m_percent(i,4))*((delta_x4/2)*((density4(i-1,1)+density4(i,1))/q(i,5)));
    else
    tt_mts(i,4) = ((delta_x4/2)*((density4(i-1,1)+density4(i,1))/q(i,5))); % same eqn can be written as
the next line
end
end
end

% smoothing of the data
alpha1 = 0.3; % can be varied, do trial and error
alpha2 = 0.3;
alpha3 = 0.3; % can be varied, do trial and error
alpha4 = 0.3;
tt_smoothed(1,1) = tt_mts(1,1);
tt_smoothed(1,2) = tt_mts(1,2);
tt_smoothed(1,3) = tt_mts(1,3);
tt_smoothed(1,4) = tt_mts(1,4);

for n=2:length(tt_mts)
    if(act_vol(i,:)<50)
        tt_smoothed(n,1) = alpha1*tt_mts(n,1)+(1-alpha1)*tt_smoothed(n-1,1);
        tt_smoothed(n,2) = alpha2*tt_mts(n,2)+(1-alpha2)*tt_smoothed(n-1,2);
        tt_smoothed(n,3) = alpha3*tt_mts(n,3)+(1-alpha3)*tt_smoothed(n-1,3);
    end
end

```

```

    tt_smoothed(n,4) = alpha4*tt_mts(n,4)+(1-alpha4)*tt_smoothed(n-1,4);
else
    tt_smoothed(n,1) = tt_mts(n,1);
    tt_smoothed(n,2) = tt_mts(n,2);
    tt_smoothed(n,3) = tt_mts(n,3);
    tt_smoothed(n,4) = tt_mts(n,4);
end
end

figure
plot(tt_mts(:,1)*60,'-g')
hold on
plot(tt_mts(:,2)*60,'-b')
hold on
plot(tt_mts(:,3)*60,'-r')
hold on
plot(tt_mts(:,4)*60,'-y')
title('travel time in seconds')

figure
plot(tt_smoothed(:,1)*60,'-r')
hold on
plot(tt_smoothed(:,2)*60,'-y')
hold on
plot(tt_smoothed(:,3)*60,'-b')
hold on
plot(tt_smoothed(:,4)*60,'-g')
title('smoothed travel time')

%save tt_model.txt tt_mts -ascii
%save tt_model_smoothed.txt tt_smoothed -ascii

fid = fopen('tt_frommodel.txt','w');

    for n=1:length(tt_mts)

        fprintf(fid,'%f\t%f\t%f\t%f\n', tt_smoothed(n,1)*60, tt_smoothed(n,2)*60, tt_smoothed(n,3)*60,
tt_smoothed(n,4)*60 );
    end

fclose(fid);

```

Extrapolation Method

%different extrapolation methods to calculate travel time. Input the speed values and travel time will be %calculated.

```
clear;
format compact
format short e

load 'speeddataformodel.txt' % read data

speed1 = speeddataformodel(:,1); % speed
speed2 = speeddataformodel(:,2);
speed3 = speeddataformodel(:,3);
speed4 = speeddataformodel(:,4);
speed5 = speeddataformodel(:,5);

delta_x1 = .498; % given that section is ~ .5 miles
delta_x2 = .506;
delta_x3 = .388; % given that section is ~ .5 miles
delta_x4 = .513;
delta_t = 2; % choose based on delta_x/free_flow_speed relation
%cumu = 4; % 30 sec data to be cumulated to 2 mt data. hence 4 set has to be added
no_in_start = 0; % not known. trial and error and choose the best number

% following are the two methods applied in the field

for(n=1:length(speed1))
    tt1_method1(n) = ((delta_x1/(2*speed1(n)))+(delta_x1/(2*speed2(n))))*3600;
    tt1_method2(n) = delta_x1/((speed1(n)+speed2(n))/2)*3600;
    tt1_method3(n) = delta_x1/(min(speed1(n),speed2(n)))*3600;
    tt2_method1(n) = ((delta_x2/(2*speed2(n)))+(delta_x2/(2*speed3(n))))*3600;
    tt2_method2(n) = delta_x2/((speed2(n)+speed3(n))/2)*3600;
    tt2_method3(n) = delta_x2/(min(speed2(n),speed3(n)))*3600;
    tt3_method2(n) = delta_x3/((speed3(n)+speed4(n))/2)*3600;
    tt4_method2(n) = delta_x4/((speed4(n)+speed5(n))/2)*3600;

end

figure
plot(tt1_method2, '-g')
title('travel time1 from method2 in seconds')
figure
plot(tt2_method2, '-r')
title('travel time2 from method2 in seconds')
figure
plot(tt3_method2, '-r')
title('travel time3 from method2 in seconds')

figure
plot(tt4_method2, '-b')
title('travel time4 from method2 in seconds')
%figure
```

```
%plot(tt2_method2, '-y')
%title('travel time2 from method2 in seconds')
%figure
%plot(tt2_method3, '-y')
%title('travel time2 from method3 in seconds')

fid = fopen('tt_fromspeed.txt','w');

    for n=1:length(speed1)
        fprintf(fid,'%f\t%f\t%f\t%f\n',tt1_method2(n), tt2_method2(n), tt3_method2(n), tt4_method2(n));
    end

fclose(fid);
```


Travel Time Prediction

Real-time method

% To predict the travel time using real-time method. Input is the previous 5 time steps travel time and the %travel time up to the next 30 time steps will be calculated.

```
clear;
real_
load tst.mat
save tst1.mat x y mx mn
load real_res.mat
errl(1)=ers;
errl1(1) = ers1;

for i=2:30,
    N=length(y);
    y=y(2:N);
    x=x(1:N-1,:);
    save tst.mat x y mx mn
    real_
    load real_res.mat
    errl(i)=ers;
    errl1(i) = ers1;
end

clear x,y;
load tst1.mat
save tst.mat x y mx mn

save realres.mat errl errl1
plot(errl)
figure
plot(errl1)

*****

function real_

load tst.mat
load norm.mat

N=length(y);
ye=x(:,5);

ers=sum(abs(ye-y)./y)*100/(N-1);

actual_ye=((ye-mm)*nx1)+ nx2;
actual_y = ((y-mm)*nx1)+ nx2;
ers1=sum(abs(actual_y-actual_ye)./actual_y)*100/N;

save real_res.mat actual_y actual_ye ers ers1
```

ANN

% Program to predict travel time using ANN method. Input the previous 5 time steps travel time values
%and get the travel time up to 30 time steps ahead

```

clear;
ntr_
nntst_

load tst.mat
save tst1.mat x y mx mn

load nnres.mat
ernn(1)=ers;
ernn1(1) = ers1;
for i=2:30,
    N=length(y)
    x=x(1:N-1,:);
    %x(2:end,4)=ye(1:end-1)';
    y=y(2:N);
    %x=x(2:end,:);
    save tst.mat x y mx mn
    nntst_
    load nnres.mat
    ernn(i)=ers;
    ernn1(i) = ers1;
end

clear x,y;
load tst1.mat
save tst.mat x y mx mn
% cd ..

save nnres.mat ernn ernn1

figure
plot(ernn)
figure
plot(ernn1)

*****
function ntr_
load tr.mat

size(x);
fcn_init='rands';
mi=round(min(x)*10)/10;
ma=round(max(x)*10)/10;

net = newff([mi ma],[10 1],{'logsig' 'purelin'});

```

```

net.initFcn='initlay';
net.layers{1}.initFcn='initwb';
net.layers{2}.initFcn='initwb';

for i=1:2,
    net.inputWeights{i}.initFcn=fcn_init;
end
for i=1:2,
    net.layerWeights{1,i}.initFcn=fcn_init;
end
net.layerWeights{2,1}.initFcn=fcn_init;
net.biases{1}.initFcn=fcn_init;
net.biases{2}.initFcn=fcn_init;
net=init(net);
net.trainFcn='trainlm';
net.trainParam.epochs = 1000;
net.trainParam.mu=1;

%disp(net.trainParam)
%pause
%net.lw{2,1}

net = train(net,x',y');

save nnwt.mat net

*****
function nntst_
load tst.mat
load norm.mat %data coming from SVMdata
load nnwt.mat

ye = sim(net,x');
aa=ye;
ye=aa';
N=length(y);
ers=sum(abs(y-ye)./y)*100/N

actual_y=((y-mm)*nx1)+ nx2; %mean(testx1)
actual_je=((ye-mm)*nx1)+ nx2; %mean(testx1)
ers1=sum(abs(actual_y-actual_je)./actual_y)*100/N
save nnres.mat actual_y actual_je ers ers1

```

SVM

% Program to predict travel time using SVM method. Input the previous time steps values and get the %future travel time.

```

clear;
svmdata
trainsvm
testsvm

load tst.mat
save tst1.mat x y mn mx

load svmres.mat
ersvm(1)=ers;
ersvm1(1)=ers1;
for i=2:30,
    i
    N=length(y);
    x=x(1:N-1,:);
    %x(2:end,4)=ye(1:end-1)';
    y=y(2:N);
    %x=x(2:end,:);
    %size(y)
    save tst.mat x y mn mx
    testsvm
    load svmres.mat
    ersvm1(i)=ers1;
    ersvm(i)=ers;
end

clear x,y;
load tst1.mat x y mn mx
%load tst1.mat
save tst.mat x y mn mx
%cd ..

save svmres.mat ersvm ersvm1

figure
plot(ersvm)
figure
plot(ersvm1)

*****

function svmdata

load 'train_original.txt'
trainx = cat(1, train_original(:,1));%, train_original(:,2));%, train_original(:,3));
load 'test_original.txt'
testx1 = test_original;
```

```

St = 1;
en = length(trainx);
St1 = 1%500;
en1 = length(testx1); %600

nx=max(trainx(St:en))-min(trainx(St:en));
nx1=max(testx1(St1:en1))-min(testx1(St1:en1));
nx2 = mean(testx1(St1:en1));
mm=max(mean(trainx(St:en)),mean(testx1(St1:en1)));
x_norm=((trainx-mean(trainx))/nx)+mm;
x1_norm=((testx1-mean(testx1))/nx1)+mm;
x_final=x_norm(St:en); %nomalised input data for training
x1_final=x1_norm(St1:en1); %nomalised input data for testing
P=5; % take five numbers as input and the 6th number as the output
Ntr=719;%;%1438 %2157;
%x_final=svdatanorm(trainx,'rbf');
%x1_final=svdatanorm(testx1,'rbf');
%Ntst=length(x)-Ntr-P-1;
count=1;
for i=P+1:Ntr,
    for j=1:P
        X(count,j)=x_final(i-j);
    end
    Y(count,1)=x_final(i);
    count=count+1;
end
x=X;
y=Y;
mn=min(x);
mx=max(x);
save tr.mat x y mn mx
save norm.mat mm nx1 nx2

% training data
count=1;
Ntst = 719;
for i=P+1:Ntst,
    for j=1:P
        Xtst(count,j)=x1_final(i-j);
    end
    Ytst(count,1)=x1_final(i);
    count=count+1;
end
x=Xtst;
y=Ytst;
mn=min(x);
mx=max(x);
save tst.mat x y mn mx

```

```
*****
```

```

function testsvm
global C P p1 p2 sep beta nsv bias;

load tr.mat %variable name is x and y
X=x;
Y=y;
load tst.mat %variable name is x and y

load norm.mat %for denormalising the data
load svmresult.mat

C=Inf;%C=500;%
P=1;
e=0.05;%e=0.1;%
ker='erbf';%'rbf';
p1=15;
p2=0;
sep=1;

%save tsdata_svm.mat X Y C P e ker p1 p2 sep
%size(beta)
err=svrerror(X,x,y,ker,beta,bias,'eInsensitive',e)
out=svroutput(X,x,ker,beta,bias);

N=length(y);
ers=sum(abs(y-out)./y)*100/N

actual_out=((out-mm)*nx1)+ nx2;
actual_y = ((y-mm)*nx1)+ nx2;
ers1=sum(abs(actual_y-actual_out)./actual_y)*100/N
save svmres.mat actual_y actual_out ers ers1

*****

function trainsvm
global C P p1 p2 sep nsv beta bias
load tr.mat

C=Inf;%C=500;%
P=1;
e=0.05;%e=0.1;%
ker='erbf';
p1=15;
p2=0;
sep=1;
%save tsdata_svm.mat X Y C P e ker p1 p2 sep

[nsv beta bias] = svr(x,y,ker,C,'eInsensitive',e);

save svmresult.mat nsv beta bias

```

C Programs for Extracting Simulation Data

```

/*-----program to get the entry exit details from tsd_text file----*/

#include <stdio.h>
#include <stdlib.h>
#define size 121856
/*.....swapping function starts-----*/

void swap(int *x, int *y) /*function for swapping*/
{
    int temp;
    temp = *x;
    *x=*y;
    *y=temp;
}

/*-----main program starts-----*/

int main()
{
    FILE *inf = NULL;
    FILE *outf = NULL;
    FILE *inf1 = NULL;
    int time[size],id[size],tempid[size]; /* reading data as 2 one dim arrays*/
    int speed[size],tempspeed[size];
    int tempcount[size],count[size];
    int i,j,k,l,m,counter=0,cum_count=-1;

    inf=fopen("datafile","r"); /* datafile name is "datafile" */
    outf=fopen("outfile1","w"); /* want the output in the file "outfile"*/

    /*fscanf(inf,"%d",&size);*/ /*specify the size of file*/
    for(i=0;i<size;i++)
    {
        fscanf(inf,"%5d %d %d",&time[i], &id[i], &speed[i]); /*read data*/
    }

    /*-----swapping and sorting-----*/

    for(i=0;i<size;i++)
    {
        if (time[i] == time[i+1])
        {
            tempid[i] = id[i]; /* if time is same save id */
            counter ++;
            cum_count++;
            tempspeed[i] = speed[i];
        }
        else
        {

```

```

tempid[i] = id[i];
tempspeed[i]=speed[i];
cum_count++;
for (k=i-counter;k<=i-1;k++)
{
    for(l=k+1;l<i+1;l++)
    {
        if(tempid[k] > tempid[l])
        {
            swap(&tempid[k],&tempid[l]); /*ascendingly order saved id's*/
            swap(&tempspeed[k],&tempspeed[l]);
        }
    }
    tempcount[i]=cum_count;
} counter = 0;
}
}

/*-----getting points where the time changes.....*/

count[0]=0; /*count array start at zero*/
m=1;
for(i=0;i<size;i++)
{
    if(tempcount[i]!=0) /*points where time changes*/
    {
        count[m]=tempcount[i]; /*time counts where time changes*/
        m++;
    }
}

/*--getting exit details ---if a vehicle id is missing in the second time group it exited in the previous time
step-----*/

i = 0; /*first time group checked separately*/
for(j=count[i];j<=count[i+1];j++)
{
    for(k=count[i+1]+1; k<=count[i+2]; k++)
    {
        if(tempid[j] > tempid[k]) continue; /*check the next id in the 2nd group*/
        else
        {
            if(tempid[j] == tempid[k]) break; /*vehicle continue in next time*/
            else
            {
                fprintf(outf,"%d\t %d\t %d\t %d\n", time[j],tempid[j],tempspeed[j],0); /*exit is 0*/
                break;
            }
        }
    }
}
}

for(i=1;i<m-2;i++) /*from time 1 to last but one*/
{

```



```

for(j=count[i]+1;j<=count[i+1];j++)
{
for(k=count[i+1]+1;k<=count[i+2];k++)
{
if(tempid[j]>tempid[k]) continue;
else
{
if(tempid[j] == tempid[k]) break;
else
{
fprintf(outf,"%d\t %d\t %d\t %d\n", time[j], tempid[j],tempid[k],0);
break;
}
}
}
}
}
}
}

```

/*--finding out entries-check with the previous time group and if a new number is there it is an entry...represented as 1..*/

```

i=1; /*from group 1, group 1 done separately*/
for(j=count[i]+1; j<=count[i+1]; j++)
{
for(k=count[i-1]; k<count[i]; k++)
{
if(tempid[j] == tempid[k]) break;
else if(tempid[j] > tempid[k]) continue;
}
if(tempid[j]!=tempid[k])
fprintf(outf,"%d\t %d\t %d\t %d\n", time[j],tempid[j],tempid[k],1);
}

for(i=2;i<m-1;i++) /*from group 2 to last*/
{
for(j=count[i]+1; j<=count[i+1]; j++)
{
for(k=count[i-1]+1; k<count[i]; k++)
{
if(tempid[j] == tempid[k]) break;
else if (tempid[j] > tempid[k]) continue;
}
}
if(tempid[j]!=tempid[k])
fprintf(outf,"%d\t %d\t %d\t %d\n", time[j],tempid[j],tempid[k],1);
}
}

/* for(j=0;j<m;j++)
fprintf(outf,"count[%d]=%d\n",j,count[j]);

for(i=0;i<size;i++)

```

```

    fprintf(outf,"%d\t %d\t %d\n",time[i],id[i],tempid[i]); */

fclose(infl);
fclose(inf);
fclose(outf);
return 0;
}

*****

/*-----program to calculate entry and exit volume and average entry and exit speeds in every one minute
interval obtained from the entryexit.c program-----*/

#include <stdio.h>
#include <stdlib.h>
#define size 8421
int main()
{

FILE *inf = NULL;
FILE *outf = NULL;
int time[size],id[size], speed[size], traveltime[size],status[size];
int entry[size],exit[size];
float entrspeed[size],exitspeed[size];
int i,j,k,l,m;

inf=fopen("outfile1","r"); /* datafile name is "datafile" */
outf=fopen("outfile3","w"); /* want the output in the file "outfile"*/

for(i=0;i<size;i++)
    fscanf(inf,"%5d %d %d %d",&time[i], &id[i], &speed[i], &status[i]); /*read data*/

for (k =0 ; k <size ; k++)
{
for(j=0;j<size;j++)
{
if(time[j]>k*60 && time[j]<=(k+1)*60)
{
if(status[j] == 0) { exit[k]++; exitspeed[k]+=speed[j]; }
else {entry[k]++; entrspeed[k]+=speed[j];}
}
}
}
fprintf(outf,"entryvol\t exitvol\t avgentrspeed\t avgexitspeed\n");

for(k=0;k<size;k++)
{
if (exit[k]!=0 || entry[k]!=0)
fprintf(outf,"%d\t\t %d\t\t %f\t %f\n",entry[k]*60,exit[k]*60, (entrspeed[k]/entry[k])* .682,
(exitspeed[k]/exit[k])* .682);
}
}

```

```

fclose(inf);
fclose(outf);
return 0;
}

*****

/*---program to get exit time and travel time for each vehicle obtained from the entry exit.c program-----*/

#include <stdio.h>
#include <stdlib.h>
#define size 7493

void swap(int *x, int *y) /*function for swapping*/
{
    int temp;
    temp = *x;
    *x=*y;
    *y=temp;
}

int main()
{
    FILE *inf = NULL;
    FILE *outf = NULL;
    int time[size],id[size], speed[size],traveltime[size],status[size];
    int i,j,k,l,m;

    inf=fopen("outfile1","r"); /* datafile name is "datafile" */
    outf=fopen("outfile2","w"); /* want the output in the file "outfile"*/

    for(i=0;i<size;i++)
        fscanf(inf,"%5d %d %d %d",&time[i], &id[i], &speed[i], &status[i]); /*read data*/

    for (k =0 ; k <size-2 ; k++)
    {
        for (l = k + 1; l < size; l++)
        {
            if(id[k] > id[l])
            {
                swap(&id[k],&id[l]); /*sorting ascending order*/
                swap(&time[k],&time[l]);
                swap(&speed[k],&speed[l]);
                swap(&status[k],&status[l]);
            }
        }
    }

    /*for(i=0;i<size;i++)
    fprintf(outf,"%d\t %d\t %d\t %d\n",id[i],time[i],speed[i],status[i]);
    */
    for(i=0;i<size;i++)

```

```

    {
    if (id[i] == id[i+1])
    {
    travelttime[i] = time[i]-time[i+1]; /* if id is same find travelttime */
    fprintf(outf,"%d\t %d\t\t %d\n",id[i], time[i],travelttime[i]);
    i++;
    }
    }

fclose(inf);
fclose(outf);
return 0;
}

*****

/*-----program to get the density values from tsd_text file----*/

#include <stdio.h>
#include <stdlib.h>
#define size 146614

/*-----main program starts-----*/

int main()
{
FILE *inf = NULL;
FILE *outf = NULL;
int time[size],id[size],tempid[size]; /* reading data as 2 one dim arrays*/
int speed[size],temptime[size];
int density[size],d,den1, den2;
int i,j,k,l,m,counter=0,cum_count=-1;

inf=fopen("datafile","r"); /* datafile name is "datafile" */
outf=fopen("outfile","w"); /* want the output in the file "outfile"*/

for(i=0;i<size;i++)
{
fscanf(inf,"%5d %d %d",&time[i], &id[i], &speed[i]); /*read data*/
}

d=0;
den1 = 1;
den2 =1;
for(i=0;i<size;i++)
{
if (time[i] == time[i+1])
{
den1++;
}
else
{

```

```

        density[d]=den1;
        temptime[d] = time[i];
        d ++;
        den1=1;
        den2++;
    }
}
for (i=0;i<(den2-1);i++)
{
    fprintf(outf,"%d\t %d\n", temptime[i],density[i]);
}
fclose(inf);
fclose(outf);
return 0;
}

*****

/*-----program to calculate average density in every one minute interval  obtained from the outfile-----
*/

#include <stdio.h>
#include <stdlib.h>
#define size 7033

int main()
{
    FILE *inf = NULL;
    FILE *outf = NULL;
    int time[size], density[size];
    float average[size];
    int j,k,counter[size],id[size];

    inf=fopen("outfile","r"); /* datafile name is "datafile" */
    outf=fopen("outfile5","w"); /* want the output in the file "outfile"*/

    for(k=0;k<size;k++)
        fscanf(inf,"%d %d", &time[k], &density[k] ); /*read data*/
    for(k=0;k<size;k++)
    {
        counter[k]=0;
        average[k]=0;
    }

    for (k =0 ; k <size ; k++)
    {
        for(j=0;j<size;j++)
        {
            if(time[j]>k*60 && time[j]<=(k+1)*60)
            {
                counter[k]++;
            }
        }
    }
}

```

```

        average[k] += density[j];
    }
}
}
fprintf(outf,"totalden\t count\t averageden\n");

for(k=0;k<size;k++)
{
    if(counter[k] !=0)
        fprintf(outf,"%f\t\t %d\t %f\n",average[k],counter[k],average[k]/counter[k]);
}

fclose(inf);
fclose(outf);
return 0;
}

*****

/*-----program to calculate average travel time of the vehicles in every one minute interval obtained
from the traveltime.c program-----*/

#include <stdio.h>
#include <stdlib.h>
#define size 3742

int main()
{
    FILE *inf = NULL;
    FILE *outf = NULL;
    int exittime[size], traveltime[size];
    float average[size];
    int j,k,counter[size],id[size];

    inf=fopen("outfile2","r"); /* datafile name is "datafile" */
    outf=fopen("outfile4","w"); /* want the output in the file "outfile"*/

    for(k=0;k<size;k++)
        fscanf(inf,"%d %d %d", &id,&exittime[k], &traveltime[k] ); /*read data*/
    for(k=0;k<size;k++)
    {
        counter[k]=0;
        average[k]=0;
    }

    for (k =0 ; k <size ; k++)
    {
        for(j=0;j<size;j++)
        {
            if(exittime[j]>k*60 && exittime[j]<=(k+1)*60)

```

```
        {
            counter[k]++;
            average[k] += traveltime[j];
        }
    }
}
fprintf(outf, "total\t\t count\t averaget\n");

for(k=0;k<size;k++)
{
    if(counter[k] !=0)
        fprintf(outf, "%f\t\t %d\t %f\n", average[k], counter[k], average[k]/counter[k]);
}

fclose(inf);
fclose(outf);
return 0;
}
```

VITA

Lelitha Devi Vanajakshi

Permanent Address

Prabhanilayam, Neerkunnam, Alleppey, Kerala, India 688 005, e-mail: lelitha@yahoo.com

Education

Ph.D., Civil Engineering, Texas A&M University, August 2004

M.Tech., Civil Engineering, Government College of Engg., Trivandrum, India, September 1995

B.Tech., Civil Engineering, Government College of Engg., Trivandrum, India, September 1993

Publications and Presentations

1. Vanajakshi, L. D., and Rilett, L. R. (2004), "Loop detector data diagnostics based on vehicle conservation principle." Accepted for publication in *Transportation Research Record*, Transportation Research Board, Washington, D.C.
2. Vanajakshi, L. D. (2003), "Loop detector data screening and diagnostics based on conservation of vehicles." *Proceedings of the IGERT Student Research Conference* (CD-ROM), Institute of Transportation Studies, University of California, Davis.
3. Vanajakshi, L. D., and Rilett, L. R. (2004), "Some issues in using loop detector data for ATIS applications." *ITS Safety and Security Conference* (CD-ROM), Miami, Florida.
4. Vanajakshi, L. D., and Rilett, L. R. (2004), "Travel time estimation from loop detector data." *ITS Safety and Security Conference* (CD-ROM), Miami, Florida.
5. Vanajakshi, L. D., and Rilett, L. R. (2003), "Estimation and prediction of travel time from loop detector data for intelligent transportation systems applications." Presented at the *TAMUS Pathways Students Research Symposium*, Galveston, Texas.
6. Vanajakshi, L. D. (2004), "Estimation and prediction of travel time from loop detector data for intelligent transportation systems applications." Presented at the Ph.D. Dissertation Seminar of the 83rd TRB Annual Meeting, Washington, D.C.
7. Vanajakshi, L. D., and Rilett, L. R. (2004), "A comparison of the performance of artificial neural networks and support vector machines for the prediction of vehicle speed." Accepted for *IEEE Intelligent Vehicles Symposium*, Parma, Italy.