2003

Exploring geometric, kinematic and behavioral scalability of microscopic traffic simulation systems

Srikanth Chakravarthy
Louisiana State University and Agricultural and Mechanical College, schakr2@lsu.edu

EXPLORING GEOMETRIC, KINEMATIC AND BEHAVIORAL SCALABILITY OF MICROSCOPIC TRAFFIC SIMULATION SYSTEMS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
In partial fulfillment of the
Requirements for the degree of
Master of Science in
Civil Engineering

in

The Department of
Civil and Environmental Engineering

by
Srikanth Chakravarthy
B.Tech, Kakatiya University, 2002
December 2003

ACKNOWLEDGEMENTS

I wish to deeply acknowledge my advisor, Dr. Sherif Ishak, for supporting me morally and financially throughout my master's degree. He will stand out most notably in my mind for his motivation and contribution to the contents of this study. Special thanks to Dr. Chester Wilmot and Dr. Brian Wolshon for their suggestions and advice. Deep appreciation goes to my colleague, Ciprian Alecsandru, for helping me by developing the program code much needed for this study.

I am especially grateful to my family members for their untiring support and belief in me and encouraging my work. Deep appreciation goes to my brother, Srinivas, and my uncle, whom I respect for their insight and guidance. A big thank you to my friends (high school, college and friends here) for supporting me at all times. I wish to thank all of them for their role in my life. Thanks are due to all those people who, directly or indirectly, contributed to my work.

I wish to dedicate this work to my grandparents who always wanted me to pursue higher studies and excel in all my endeavors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Even with today's remarkable advancement in computing power, microscopic simulation modeling remains a computationally intensive process that imposes limitations on its potential use for modeling large-scale transportation networks. Research and practice have repeatedly demonstrated that microscopic simulation runs can be excessively time-consuming, depending on the network size, the number of simulated entities (vehicles), and the computational resources available. While microscopic features of a simulated system collectively define the overall system characteristics, it is argued that the microscopic simulation process itself is not necessarily free of redundancy, which if reduced, could substantially improve the computational efficiency of simulation systems without compromising the overall integrity of the simulation process. This research study explores the concept of scalability for microscopic traffic simulation systems in order to improve their computational efficiency and cost-effectiveness. More specifically, we present an optimized downsampling procedure for transforming the full-scale simulation system (*prototype*) into a geometrically, kinematically, and behaviorally equivalent reduced-scale system (*microcosm*). The ultimate goal is to execute the microscopic simulation process in the *microcosm* environment, observe all necessary macroscopic characteristics and performance measures, and upsample the results back to the *prototype* environment. Experimental analysis was conducted on a homogeneous freeway corridor to examine the effect of different operating conditions on the optimal solutions for the downsampling procedure. The study also investigates the tradeoff between performance and scalability of microscopic simulation systems.

1. INTRODUCTION AND PROBLEM STATEMENT

1.1.INTRODUCTION

The past two decades have witnessed remarkable advancement in the development and application of microscopic simulation models in the field of transportation engineering. This is partly attributed to the significant improvements in microscopic traffic flow models and driver behavioral models, coupled with the remarkable advancement in computational resources. Lately, interest in microscopic simulation modeling resurfaced as a support system for real-time traffic control and management functions, and other Intelligent Transportation Systems (ITS) applications, in addition to its traditional role in off-line design and operational analysis. Nevertheless, microscopic simulation models are intrinsically computationally intensive and even with today's computing powers our abilities to deploy such models in large scale transportation system simulation have been extremely limited. This exploratory research proposes a mathematical approach to improve the computational efficiency of existing traffic simulation systems.

1.2.PROBLEM STATEMENT

Research and practice have repeatedly demonstrated that microscopic simulation of large transportation networks is a very time consuming process. While such computational burden is tolerable in offline analysis, it poses a major obstacle to applications where microscopic simulation has to be carried out in real time environments repeatedly to test various alternate what-if scenarios.

Parallel computing of segmented networks and simple distributed computing architectures are being suggested as possible alternatives to meet this heavy computational demand (Grama, et.al, 1994). These approaches, however, might incur heavy infrastructural costs. Discouraged by the computational requirements and associated costs, modelers often resort to macroscopic simulation models that require far less computational resources. While macroscopic models can provide quick solutions to many practical and localized problems, they reportedly fail to account for the stochastic variation in transportation system operation and are therefore not suitable for modeling large-scale transportation networks. On the contrary, microscopic simulation modeling has the ability to reveal intricate details of the simulated system and replicate real-world well through tracking of individual vehicles over time and space. The entire process is, however, not necessarily free of redundancy, which contributes to the overall system complexity and requires an intensive amount of computational resources. Redundancy in the context of this study refers to a proportion of the simulated system entities, which when removed in a systematic manner, will not result in a substantial loss of critical information and will thus retain most of the system operating characteristics.

This research addresses the complexity and limitations of current microscopic simulation modeling procedures by downsampling (downscaling) the simulation process itself and thus reducing the amount of redundancy in the simulated system while preserving most of the system operating characteristics. The developed approach has the potential to allow modelers to simulate large-scale transportation systems with the least effort possible and using reasonable amount of computational resources without compromising the simulation system performance. In simple terms, this approach transforms the original full-scale system (referred to as “prototype” henceforth) into a reduced scale system (henceforth referred to as “microcosm”). Microscopic simulation modeling is then carried out for the microcosm environment to collect all necessary properties and characteristics, which in turn are transformed back to the prototype. The developed approach has the potential to impact the next-generation traffic management systems by overcoming the current limitations to the online use of microscopic simulation in decision support systems.

1.3.OBJECTIVES

The main goal of this research is to improve the efficiency and cost-effectiveness of microscopic simulation modeling of large-scale transportation networks. This goal can be achieved through the following objectives:

1) Investigate the role and limitations of existing microscopic simulation models in terms of applications and computational requirements.
2) Develop an effective mathematical approach for downsampling microscopic simulation processes.
3) Evaluate the performance of the approach under different scenarios and downsampling ratios.

This research, if successful, can impact the capabilities of the next generation traffic simulation models.

2. LITERATURE REVIEW

This chapter presents an overview of traffic microscopic simulation modeling, their classification and a short description of some of the widely used microscopic simulation models. A section describing the car-following model used in this study and its historical perspective is also presented.

2.1.OVERVIEW OF MICROSCOPIC SIMULATION MODELS

Traffic simulation models are logical and mathematical representations of real-world systems that are designed into software and can be run on a digital computer in an experimental fashion. These simulation models are characterized by complex inter-component interaction within the system, which cannot be adequately described by existing analytical forms owing to their complexity. The underlying theories in microscopic traffic simulation such as car-following, lane-changing and gap acceptance models that govern the behavior of vehicle entities in the system, address the complexity of this interaction very well. Thus microscopic traffic simulation tools, developed in the last one to two decades, can assist in testing, verification and improvement of different traffic management strategies and what-if scenarios prior to implementation, and then allow us to select an optimal strategy based on a set of Measures of Effectiveness (MOEs).

2.1.1. ROLE OF MICROSCOPIC SIMULATION MODELS

Analytical modeling of the random processes associated with traffic and driver behavior is complicated. For this reason, microscopic traffic simulation modeling has been put to extensive use, for the past few years, in modeling transportation networks and evaluating various traffic management systems prior to implementation. Simulation modeling also has attracted a fair amount of research with efforts put into bridging the gap between observed (realistic) data and data obtained from simulation and hence to closely represent actual traffic behavior. They are suitable for a great diversity of applications such as evaluation of ramp metering (Chu, et.al 2002, Hourdakis, et.al, 1999), signal control strategies and are generally capable of replicating the real world given that they are well calibrated (Revised monograph, 2000).

With the advent of Intelligent Transportation Systems (ITS), new and innovative functionalities such as Advanced Travel Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS), have been introduced to improve the efficiency and productivity of existing transportation systems. ATIS and ATMS include real-time route guidance, mainline traffic control and incident management. Microscopic simulation models are considered as viable tools to reproduce the behavior of such complex systems; they can address the inherent randomness in individual behavior and variability in traffic stream characteristics. Nevertheless, the advantage of microscopic simulation models is discounted by the burden of computation power that appears to increase exponentially with the size of the network.

2.1.2. LIMITATIONS OF MICROSCOPIC SIMULATION MODELS

Several microscopic simulation models have been proposed for offline design and real-time analysis of complex traffic systems such as Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS). To investigate these demand management strategies, simulation runs need to be carried out simultaneously in real time so as to assess present flow conditions and evaluate strategies for predicted future flow patterns over short-term horizons. This will enable practitioners to adopt a suitable strategy to effectively manage traffic congestion. However, the open literature suggests that while microscopic simulation models provide a sufficiently realistic approach for analysis of complex systems, an efficient and practical computational environment for online assessment of various alternatives is still lacking (Kwon, 1997). This may partly be attributed to the vast amount of computational resources that the microscopic simulation models demand.

Microscopic simulation models compute the characteristics such as the speed, position, and acceleration of individual components in the network on a second by second basis. These computations appear to increase the required computational resources exponentially with more number of vehicles in the system (Leonard II, 2000). With limited computational resources available, modeling a large transportation network becomes an extremely time consuming process. This problem becomes more critical when it comes to online implementation of ITS technologies, which requires continuous assessment of traffic conditions for effective system management.

2.2.CLASSIFICATION OF SIMULATION MODELS

Traffic simulation models are classified based on the level of detail, degree of randomness, how the system is updated, flow input, objective of the analysis and mode of analysis (HCM 2000).

2.2.1. CLASSIFICATION BY LEVEL OF DETAIL

Microscopic (high-fidelity) models describe both the system entities and their interaction at a high level of detail. Such interaction is often represented by car-following models that predict the response of the following vehicle to the stimulus caused by the lead vehicle. The response is generally expressed as a function of the type of stimulus and driver's sensitivity. The response of the following vehicle is expressed as acceleration or deceleration that is required to maintain a safe distance to avoid collision. Other models that are instrumental to microscopic simulation modeling are derived from the behavior associated with lane-changing, passing and turning maneuvers, and gap acceptance (Velan and Van Aerde, 1996, Nakayama et.al 1999 and Levison et.al 1998). Hence, it is evident that microscopic simulation models incorporate random parameters derived from assumed or observed probability distribution functions. These models are costly to develop, execute and maintain due to complexity of logic and larger number of parameters that need to be calibrated. Examples of such models are CARSIM, CORSIM, AIMSUN2, PARAMICS, MITSIM, ROADSIM and VISSIM.

Macroscopic (low-fidelity) models employ the fundamental traffic flow relationships between flow, density, and speed. The interactions of entities and their activities are described at a low level of detail. They rely on the assumption that traffic flow can be modeled as one-dimensional continuous fluid flow with more emphasis on aggregated characteristics of traffic flow. Macroscopic models are easier and less costly to develop, execute and maintain. These models are best suited when the availability of model development time and resources are limited. However, their representation of the real world is less accurate. Models that fall in this category are CORFLO, KWAVES, KRONOS, PASSER IV, AUTOS, TEXAS and FREQ.

Mesoscopic (mixed-fidelity) models, which encompass the features of both microscopic and macroscopic models, incorporate the movement of platoons of vehicles and employ equations that describe the interaction among these clusters. Most entities are represented in detail, but their interaction is predominantly macroscopic. For example, individual vehicles at a high level of detail can represent car-following behavior, but the safe spacing between vehicles is determined by the total flow rate rather than individual vehicle interactions. Examples of mesoscopic models are CORFLO/NETFLO, DYNASMART, DYNEMO, and INTEGRATION.

2.2.2. CLASSIFICATION BY LEVEL OF STOCHASTICITY

Simulation models can also be classified based on the characteristics of the components that make up a transportation system. In a *deterministic* environment all the parameters used in the simulation model are preset to a fixed value and models have no random variables. Similar simulation configurations essentially lead to the same output. All interactions among system entities are described by exact mathematical, statistical, or logical relationships. Examples of such models are TRANSYT, KWAVES and FREQ.

In a *stochastic* environment, the model parameters are treated as random variables derived from probability distribution functions. With such random parameters, the simulation system itself becomes stochastic and simulation runs with different random number seeds will yield different results. Individual simulation runs are treated as random experiments and hence simulation runs are repeated as many times as required to obtain reasonable statistical confidence bounds (e.g. 95%). Example models that fall in this category are PARAMICS, AIMSUN2, CORSIM, INTEGRATION and TRANSIMS.

2.2.3. CLASSIFICATION BY SYSTEM UPDATING METHOD

Traffic simulation models can be further classified based on the method of updating the system. Continuous models update the system continuously over time in response to continuous stimuli and, therefore, change in state is gradual rather than abrupt. Discrete models, on the other hand, update the system at specified time intervals (Discrete-time model) or when certain events are triggered (Discrete-event model) such as change in traffic signal indication. Discrete models are close to real world systems that are characterized by abrupt changes in their states.

2.2.4. CLASSIFICATION BY FLOW INPUT

In static-flow models, traffic flows remain constant for the entire simulation period. In other words, the O-D matrix is fixed and is insensitive to the time of simulation. Time-varying simulation models, on the other hand, have varying flow rates during the simulation period. This implies that the O-D matrix varies over time. However, this classification only means the variation of flow by input and does not consider the time-varying flows that are caused internally due to accidents and other simulated events.

2.2.5. CLASSIFICATION BY OBJECTIVE OF THE ANALYSIS

Descriptive models describe the response of the traffic in a given situation but do not optimize the parameters to obtain the best network performance. For example, a simulation model describing the response of the drivers under different traffic flow conditions. Normative models identify a set of parameters that facilitates best system performance, signal optimization, for instance.

2.2.6. CLASSIFICATION BY MODE OF ANALYSIS

Real-time models simulate traffic conditions in real-time. One second in real-time is equal to one second of simulation time. These models require more computational requirements during heavy traffic conditions. The input to these models is provided by sensors and other equipment that operate in real-time. Off-line modeling finds its use in evaluating different what-if scenarios by specifying different traffic inputs. These models are often faster than real-time simulation models.

2.3.STATE-OF-THE-ART MICROSCOPIC SIMULATION MODELS

PARAMICS (PARAllel MICroscopic Simulator) comprises of high performance software tools that are capable of modeling the movement of individual vehicles on urban and highway road networks (PARAMICS V 3.0 User Guide, 2000). The PARAMICS

Suite consists of Modeller, Processor, and Analyser. The car following and lane changing models are highly sophisticated and can manage up to 32 lanes in width. It has an open architecture, and parameters that affect driver behavior are accessible to the modeler. Modelers can override and extend many features of the underlying car following and lane changing models using Application Programming Interfaces (APIs) (Chu et.al. 2002). Visualization is possible via a Graphical User Interface (GUI). The model is also capable of interfacing drivers and Intelligent Transportation Systems (ITS) and hence is rendered extremely suitable for planning and policy evaluation. PARAMICS uses the discrete-time based simulation approach with a default time step (system updating interval) of 0.5 seconds that can be altered by the user. It does not have any limitation on the network size but is constrained by the memory and processor constraints of the machine.

AIMSUN2 (Advanced Interactive Microscopic Simulation for Urban and Non-urban networks) simulates individual vehicles and their interactions in fine detail using several driver behavior models such as car following, lane changing, and gap acceptance, etc. This feature allows AIMSUN2 to simulate any measurable traffic detector like counts, occupancy, and speed (Adams et. al, 2000). Unlike PARAMICS, AIMSUN2 is a discrete-continuous modeler where some elements of the system are updated discretely at specific time intervals while the state of some elements changes continuously over time (Hourdakis et.al, 1999). Because of its open architecture, all models like traffic control, vehicle behavior, route calculation and route choice models are independent of the simulation logic and hence can be replaced by alternative ones. It also has the ability to distinguish different types of vehicles and drivers. AIMSUN2 has no limitations on network size but the speed of execution depends on the available computer memory (RAM).

CORSIM (CORridor microscopic SIMulation) is a microscopic, discrete time, stochastic simulation model that is used to model traffic operations strategies (ITT 1998). CORSIM is based on the car following theory that relates the vehicle operating behavior to the characteristics of vehicles ahead of and beside them. Unlike PARAMICS and AIMSUN2, CORSIM has a fixed updating time step embedded in the model. CORSIM's capability can handle a network up to 500 nodes, 1000 links and a maximum of 20,000 vehicles. CORSIM is a combination of two simulation models: arterial network microscopic model NETSIM and freeway microscopic model FRESIM. While NETSIM is designed to simulate interrupted flow conditions, FRESIM is preferred in simulating uninterrupted flow conditions. A more comprehensive description of traffic simulation models can be found in Adams et al. (2002) and Bernauer et al. (1997). Table 2-1 summarizes the classification of some of the simulation models in use.

2.4.CAR FOLLOWING MODELS

Car following models describe the process of one vehicle following another vehicle in the same lane. These models explain the interaction between adjacent vehicles in the same lane and control their motion in a traffic network. In the 1950s and 1960s, Reuschel and Pipes were pioneers in the development of car-following models, along with parallel efforts by Kometani and Sasaki in Japan, Forbes at Michigan State University, and General Motors (GM) Researchers (May, A.D, 1990). A review of the research development in car-following models can be found in the open literature (e.g. Hatipkarasulu et.al 2000, Chundury and Wolshon, 2000, Brackstone and McDonald, 1999).

Table 2-1: Traffic Simulation Models (Revised Monograph, 2000)

Name	Discrete Time	Discrete Event	Micro	Mesoscopic	Macro	Deterministic	Stochastic
NETSIM	X	X	X				X
NETFLO1		X		X			X
NETFLO2	X				X	X	
FRESIM	X		X				X
DYNASMART	X			X		X	
TRANSIMS	X			X			X
PARAMICS	X		X				X
AIMSUN2	Continuous	Continuous	X				X
ROADSIM	X		X				X

As car following models have the ability to describe inter-component interaction precisely, they have become an integral process in microscopic simulation modeling. For this exploratory type of research, the GM car-following models have been chosen arbitrarily and any other car-following model can be similarly applied. A general expression of a car-following model is given by:

Response = Function (Sensitivity, Stimulus)

Here, response is the acceleration/deceleration of the following vehicle due to the stimulus caused by a difference in speeds and/or spacing between the two vehicles. Sensitivity is a factor that depends on speed difference, distance headway and the characteristics of the driver. Chandler et.al (1958) proposed the first linear car following model expressed mathematically as:

$$a_{n+1}(t+\Delta t) = \alpha(\dot{x}_n(t) - \dot{x}_{n+1}(t)) \tag{1}$$

Where,

$a_{n+1}(t+\Delta t)$ = Acceleration/ deceleration of the following vehicle

α = Sensitivity parameter

Δt = Reaction time of the following vehicle driver

$\dot{x}_n(t)$ = Velocity of lead vehicle

$\dot{x}_{n+1}(t)$ = Velocity of following vehicle

$\dot{x}_n(t) - \dot{x}_{n+1}(t)$ = Stimulus expressed by the relative velocity between lead vehicle and following vehicle

A simple graphical illustration of the first GM-I car-following model is shown in Figure 2-1.

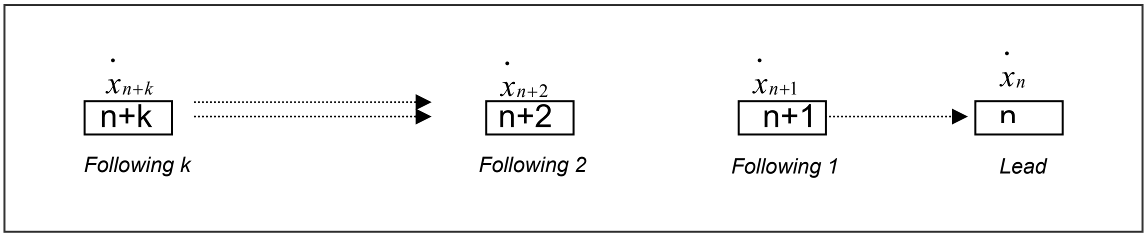


Figure 2-1: Illustration of GM-I car-following model

From their observations, they concluded that average values of reaction time and sensitivity were 1.5 seconds and 0.37 sec^{-1}, respectively. This was the first General Motors model that was launched in an effort to describe car-following theory.

Subsequent research efforts by the General Motors research group (Gazis et al. 1959, 1961 and Herman et al. 1959) paved the way to a generalized model with an improved sensitivity factor that depends on spacing and speed of the following vehicle. Mathematical expression is given by:

$$a_{n+1}(t+\Delta t) = \frac{\alpha_{l,m}\left[\dot{x}_{n+1}(t+\Delta t)\right]^{m}}{\left[x_n(t) - x_{n+1}(t)\right]^{l}}\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right] \qquad (2)$$

Where,

$\left[x_n(t) - x_{n+1}(t)\right]$ is the spacing between the lead vehicle and the following vehicle, respectively, and *l* and *m* are distance headway and speed exponents.

Models developed by Chandler et al. (1958) and Gazis et al. (1959) can be seen as special cases of the generalized model {m=0, l=0}. Also, by setting *m*=0 and *l*=1, we obtain the third GM model (Equation(3)).

$$a_{n+1}(t+\Delta t) = \frac{\alpha_{\circ}}{x_n(t) - x_{n+1}(t)}\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right] \qquad (3)$$

Where,

$\alpha_{\circ}$= Sensitivity parameter in ft/sec

For the following 15 years several attempts were made to determine exact values of *l* and *m*. Treiterer and Myers (1974) proposed values of *l* = 1.6, *m* = 0.2 and *l* = 2.5, *m* = 0.7 for acceleration and deceleration respectively by using a microscopic approach. Hoefs (1972) also employed a microscopic approach to arrive at a different set of *l* and *m* values for acceleration, deceleration with braking and deceleration without breaking, respectively. More recently, Ozaki (1993) attempted to estimate α , *l* and *m* but his values were contradictory to those observed by earlier researchers. He, however, observed that space headway and acceleration of the lead vehicle have an effect on reaction time.

2.5.SUMMARY

Microscopic simulation modeling is being extensively used to come up with innovative solutions for transportation related problems. The availability of a wide range of simulation models allows the modeler to choose a simulation model that answers the specific problem at hand. However, the lack of sufficient computational resources severely restricts the modeler either to simulate a smaller network or reduce the desired level of accuracy. Even though some microscopic simulation models (AIMSUN2 and PARAMICS) neither have limitations on network size or level of accuracy, they depend on the performance capabilities of the machine on which they are run. No previous studies seeking a solution to improve the computational efficiency of microscopic simulation systems have been found in the literature. Therefore, an exploratory research is done to develop an approach to improve the performance of traffic simulation systems.

3. METHODOLOGY

3.1.INTRODUCTION

In order to reduce the computational resources associated with microscopically simulating a large transportation network, a methodology is developed that simulates a reduced network (referred to as microcosm in the context of this study) obtained by systematic elimination of vehicles (referred to as downsampling) in the original network (referred to as prototype). The results of the simulation are then transformed back to the prototype environment. The entire process consists of downsampling the prototype, simulating the downsized network and upsampling the results to the prototype environment. The downsampling and upsampling process is shown in Figure 3-1.

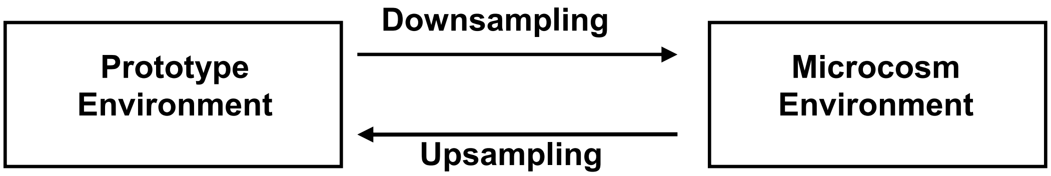


Figure 3-1: Downsampling - Upsampling Process

Though the vehicles are eliminated in a systematic way, the process still remains entirely random since the distribution of vehicle and driver characteristics in the traffic stream is essentially random. For example, if we generate a stream of random numbers, then one can easily prove that the stream of every other number in that stream is random as well, and follows the same probability distribution function. This observation is very critical to ensure that in the process of network resizing, the distribution of driver and vehicle characteristics remains essentially the same in prototype and microcosm environments (e.g. percentages of turning movements at intersections and percentage of trips by destination). The following sections contain more detailed explanation of the approach.

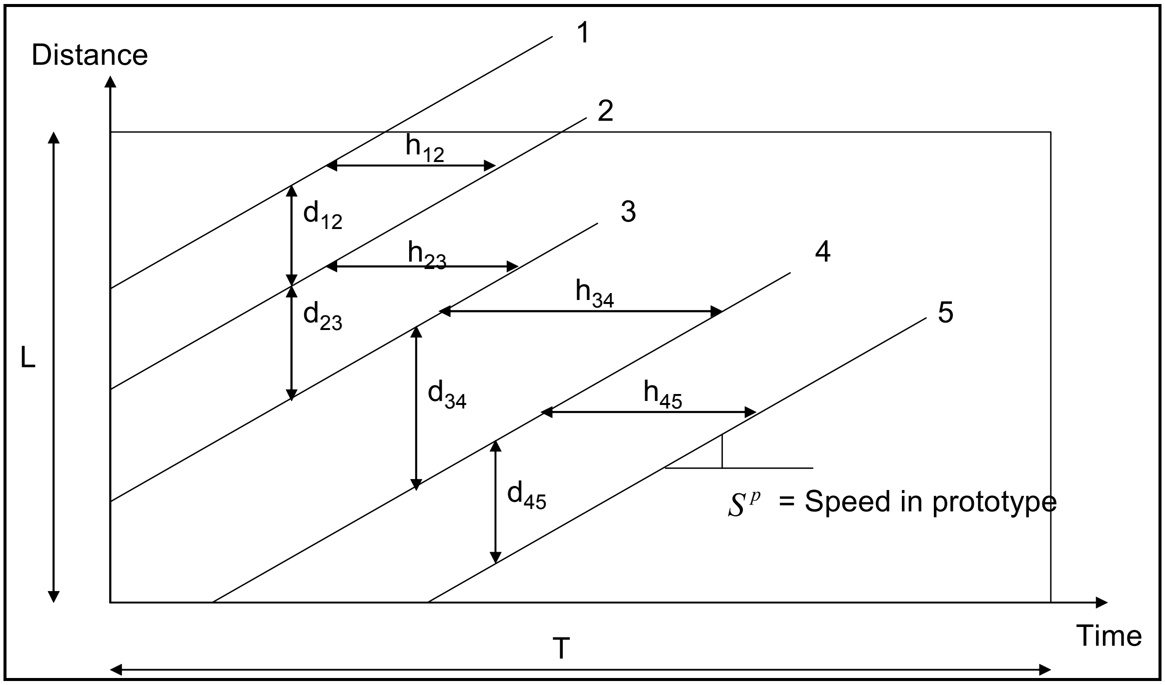
3.2.ILLUSTRATIVE EXAMPLE

The methodology developed for this study is a four step process and can be best described by an illustrative example. Let us consider the case of a one-lane freeway section of length L, as shown in Figure 3-2(a). The figure shows the trajectories of five hypothetical vehicles in a traffic stream with spacing d_{ij} and headway h_{ij} between two consecutive vehicles i and j. This is the prototype network and all the vehicles are in car following mode. For illustrative purposes only, let us assume that all vehicles are traveling at the same speed S_p. However, this assumption is not realistic and is relaxed in the mathematical formulation of the methodology as described later on.

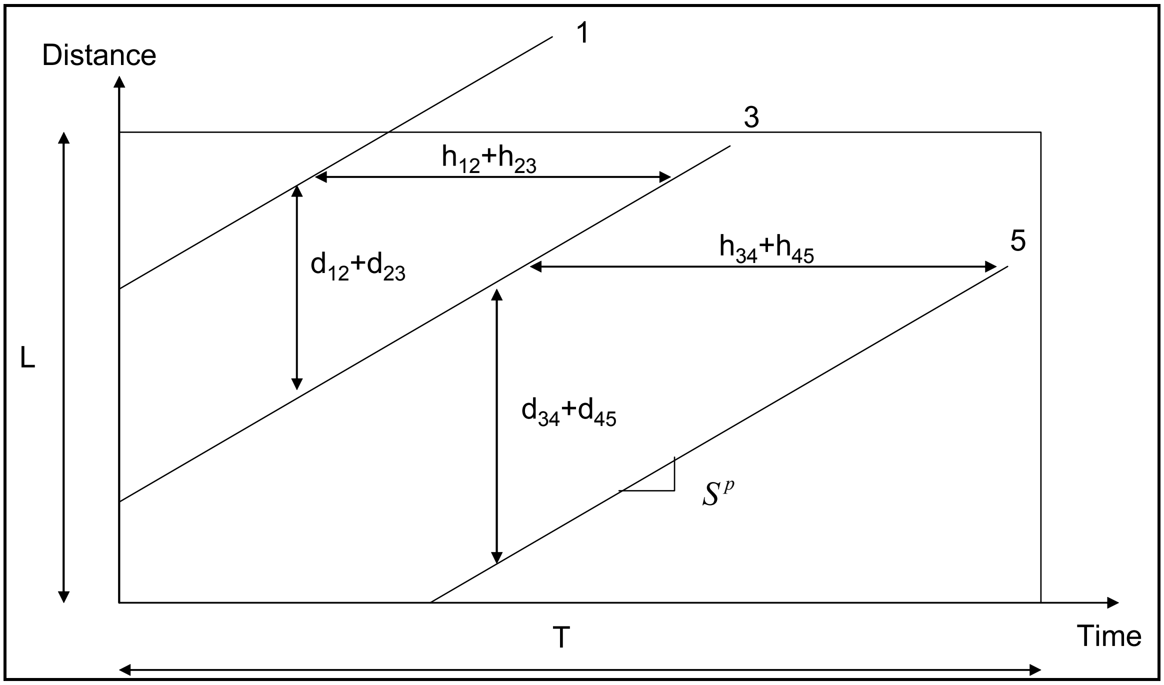
Let us apply a 50% downsampling ratio to the network in Figure 3-2 (a), where a downsampling ratio is defined as the ratio of retained vehicles in the microcosm environment to the total number of vehicles in the prototype. Here, a 50% downsampling ratio implies systematic elimination of every other vehicle in the network (or in this case, the even-numbered vehicles 2, 4, etc.). Spacing between successive vehicles is now d_{12}+ d_{23}, d_{34}+ d_{45}, and so on as shown in Figure 3-2 (b). The elimination process of every other vehicle causes nearly a 50% reduction in the density, as a result of the increased spacing between vehicles. To restore the density in the traffic stream and to preserve geometric similarity between the prototype and the microcosm environments, we now apply 50% geometric downscaling of the section vehicular trajectories. This causes the segment length and the relative positions of vehicles with respect to the downstream end of the link to be reduced by 50%, as shown in Figure 3-2 (c). This shift in positions, without altering vehicular speeds, will reduce both spacing and headway between vehicles by nearly 50% of their values in Figure 3-2 (b), restoring the density between vehicles to the prototype values.

However, geometric downscaling also restores the prototype headways and the flow rate as well. This violates the initial 50% downsampling ratio of flow rate, and therefore, necessitates simultaneous kinematic downscaling of trajectories. This can be achieved by downsampling the kinematic characteristics by 50% (speed and acceleration, but only speed in this simplistic example). Downscaling the vehicular speeds by 50% (i.e. $S^m = \frac{1}{2} S^p$, where S^m and S^p are the speeds in microcosm and prototype environments, respectively) will increase the headways by 100%, and consequently, result in reducing the flow rate by 50%, as shown in Figure 3-2 (d). One can observe in this figure that the downsampling process has the tendency to preserve the link travel time in the microcosm environment (note that no temporal scaling is applied). This is an important characteristic that has been preserved to ensure that route choice models that are based on minimum travel times are still applicable in the microcosm environment.

However, retaining only the geometric and kinematic similitude in both the prototype and the microcosm environments will not preserve all system characteristics due to other stochastic behavioral implications that result from car-following and lane-changing behavior. This makes the complete downsampling process far more complicated than it appeared in the illustration shown. Moreover, in microscopic simulation, most of the system characteristics are essentially random variables that are derived from some probability distribution functions. For instance, in stochastic

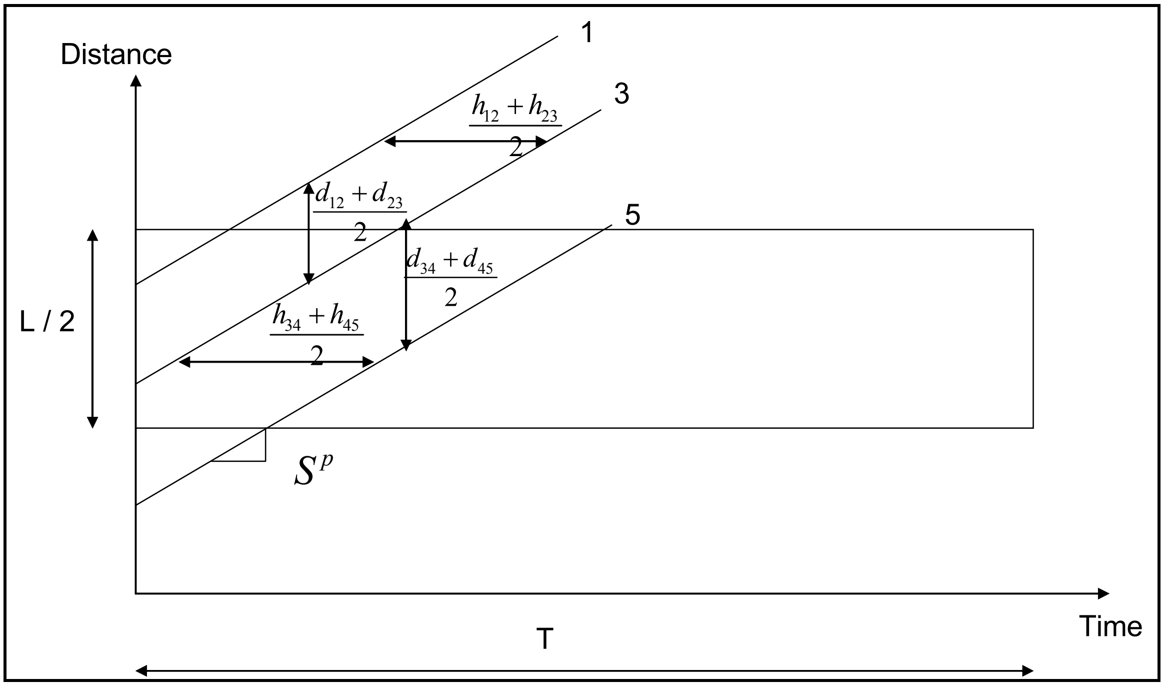


(a) Step 1: Vehicle Trajectories in Prototype Environment

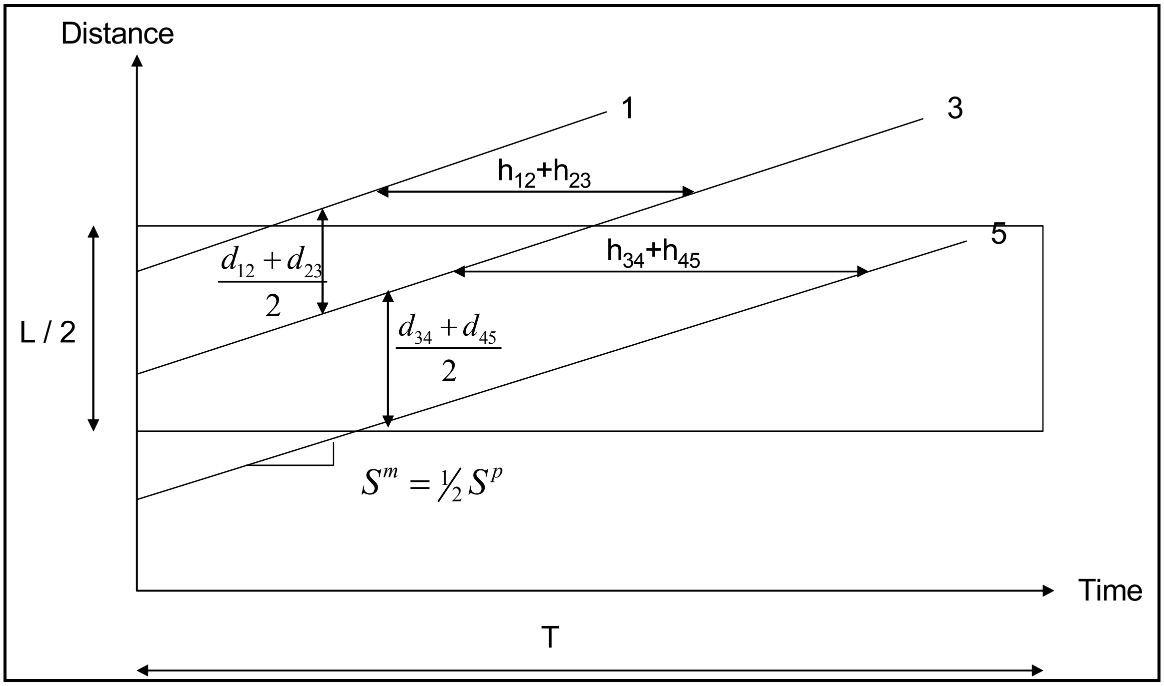


(b) Step 2: Elimination of Even-Numbered Vehicles

Figure 3-2: Steps Illustrating Downsampling Procedure


Distance
1
3
5
$\frac{h_{12}+h_{23}}{2}$
$\frac{d_{12}+d_{23}}{2}$
$\frac{d_{34}+d_{45}}{2}$
$\frac{h_{34}+h_{45}}{2}$
L / 2
S^p
Time
T

(c) Step 3: Geometric Downscaling of *Prototype* (*r* = 0.5)



(d) Step 4: Kinematic Downscaling of *Prototype* (*r* = 0.5)

Figure 3-2 (Continued)

conditions, where spacing is considered a random variable with some probability distribution function, defined by ($Mean = E[d^p], Variance = V[d^p]$) in the prototype environment, one can easily prove statistically that after vehicle elimination and geometric downscaling, the spacing probability distribution parameters in the microcosm environment, ($E[d^m], V[d^m]$), can be defined as $E[d^m] = E[d^p]$ and $V[d^m] = V[d^p]$. Statistically, this procedure applies to the headway distribution as well, where a relationship between the probability distribution parameters in the microcosm and prototype environments can be defined by $E[h^m] = \frac{1}{r} E[h^p]$ and $V[h^m] = \frac{1}{r^2} V[h^p]$ (where, r is the downsampling ratio). This shows that the statistical parameters, mean and variance, of the headway distribution must be scaled by the ratio $\frac{1}{r}$ and $\frac{1}{r^2}$, respectively, in the downsampling process; i.e. both the mean and the variance will increase in the microcosm environment. This establishes a relationship between the microcosm and the prototype environments under stochastic conditions as well. However, this study is limited only to deterministic environment and only a brief discussion of stochastic behavior is provided later on. The implications of stochasticity on downsampling are beyond the current scope of this study.

3.3.SCALABILITY OF BEHAVIORAL MODELS

In order to preserve the overall system operating characteristics in prototype and microcosm environments, adjustments to the car-following model are essential as microscopic simulation models are built upon car-following and lane-changing models that define the behavior and interaction of individual travelers in the network. Without such an adjustment to the car-following models, the remaining vehicles would follow their prototype behavior by attempting to compensate for the vehicle omissions, thereby skewing the simulation results. Therefore, it is essential that the car-following model be modified so that the remaining vehicles would retain their trajectories even after downscaling the network. An error function, formulated to estimate the difference between corresponding trajectories of vehicles in prototype and microcosm environments, is described in subsequent sections. Lane-changing models describe the level of opportunities available to the drivers for lane-changing. Lane changing opportunities become available under light traffic conditions where the headway distribution is random. The objective then is to preserve the level of lane-changing opportunities available to the drivers in the prototype and microcosm environments. However, a discussion on lane-changing models is beyond the scope of this study.

3.4. PROBLEM FORMULATION

The primary objective of the mathematical formulation of the downsampling process is to optimize the critical behavioral parameters in the adopted car-following model such that the errors associated with the downsampling procedure are minimized. The error function is formulated to optimize the behavioral parameters in a homogenous freeway section, where vehicles are restricted from lane-changing, entering or exiting the stream.

Descriptively, the main objective here is to optimize the parameters of the car-following model such that the behavior of vehicles in the microcosm environment closely matches that in the prototype environment. Quantitatively, optimization can be achieved by formulating a downsampling error function that is derived from the sum of squared (SSE) differences between the vehicular trajectories in the prototype environment and their corresponding representatives in the microcosm environment. Mathematically,

$$SSE = \sum_{j=0}^{rN} \sum_{i=0}^{T/\delta t} \left[X^{p}_{j/r+1}(i\delta t) - 1/r\, X^{m}_{j+1}(i\delta t) \right]^2 \tag{4}$$

Where

SSE = sum of squared (SSE) differences between the prototype and microcosm trajectories during time period *T*

r = the desired downsampling ratio (e.g. $1/2$, $1/3$, etc.)

$X^{p}_{j/r+1}(i\delta t)$ = the position of vehicle $j/r+1$ in the prototype environment at time $i\delta t$

$X^{m}_{j+1}(i\delta t)$ = the position of vehicle $j+1$ in the microcosm environment at time $i\delta t$

$1/r\, X^{m}_{j+1}(i\delta t)$ = the adjusted position of vehicle $j+1$ in the microcosm environment at time $i\delta t$ with reference to the prototype's reference system

δt = the time increment used to update the simulation system (e.g. 0.1 second)

T = the overall simulation time period

N = the total number of simulated vehicles in the prototype environment

rN = the total number of simulated vehicles in the microcosm environment

Equation (4) shows that the objective function *SSE* is derived from the deviation in vehicular trajectories. For instance, if the downsampling ratio *r* is assumed 50% (*r* = 0.5), then the sequence of trajectories in the microcosm environment will be indexed as 1, 2, … *rN*, where *rN* is the total number of vehicles simulated in the microcosm environment. That sequence, however, corresponds to a sequence of trajectories in the prototype environment that is indexed by j/r +1 or (1, 3, 5, 7,…), where even-numbered trajectories are skipped in the prototype environment. As such, the matching procedure will apply to the following trajectory pairs in the microcosm and prototype environments, respectively: (1, 1), (2, 3), (3, 5), (4, 7)… (j +1, j/r +1).

3.4.1. CONSTRAINTS

Constraints are required to control the behavior of vehicles and maintain safe operation within practical boundaries, as often applied in simulation models. In this exploratory study, the first GM car-following model was adopted and deterministic driving characteristics were assumed in the formulation of constraints. However, the methodology applies to any car-following model by simple substitution of the car-following constraints in the microcosm and prototype environments. Essentially, these models are fundamentally similar and are described by a relationship, where a stimulus caused by the lead vehicle triggers a response by the following vehicle in the traffic stream. Although the models are similar, the stimulus-response relationship is often stochastic in nature since the response to the same stimulus varies from one driver to another, depending on the driver characteristics.

In a deterministic context the solution of the mathematical problem leads to the "expected" optimal values of two behavioral parameters in the microcosm environment: (1) the sensitivity parameter α^m and (2) the adaptation time ΔT^m. The following is the set of constraints to be imposed in both *prototype* and *microcosm* environments.

$$a^m_{i+1}(t) = \alpha^m[S^m_i(t-\Delta T^m) - S^m_{i+1}(t-\Delta T^m)] \qquad \forall i \in 1,2,...rN \qquad \forall t \in 0,\delta t,...T \tag{6}$$

$$a^p_{i+1}(t) = \alpha^p[S^p_i(t-\Delta T^p) - S^p_{i+1}(t-\Delta T^p)] \qquad \forall i \in 1,2,...N \qquad \forall t \in 0,\delta t,...T \tag{7}$$

$$X^p_i(t) - X^p_{i+1}(t) \ge X^p_{i+1,\min}(t) \qquad \forall i \in 1,2,...N \qquad \forall t \in 0,\delta t,...T \tag{8}$$

$$X^m_i(t) - X^m_{i+1}(t) \ge X^m_{i+1,\min}(t) \qquad \forall i \in 1,2,...rN \qquad \forall t \in 0,\delta t,...T \tag{9}$$

$$S^p_{\max} \ge S^p_i(t) \ge 0 \qquad \forall i \in 1,2,...N \qquad \forall t \in 0,\delta t,...T \tag{10}$$

$$rS^p_{\max} \ge S^m_i(t) \ge 0 \qquad \forall i \in 1,2,...rN \qquad \forall t \in 0,\delta t,...T \tag{11}$$

$$a^p_{\max} \geq a^p_i(t) \geq a^p_{\min} \qquad \forall i \in 1,2,...N \qquad \forall t \in 0, \delta t, ...T \quad (12)$$

$$ra^p_{\max} \geq a^m_i(t) \geq ra^p_{\min} \qquad \forall i \in 1,2,...rN \qquad \forall t \in 0, \delta t, ...T \quad (13)$$

$$\alpha^m \geq 0 \qquad and \qquad \Delta T^m \geq 0 \quad (14)$$

Where

$a^p_{i+1}(t)$ and $a^m_{i+1}(t)$ = the acceleration of the $i+1$ th vehicle at time t in the prototype and the microcosm environments, respectively.

ΔT^p and ΔT^m = the adaptation times in the prototype and the microcosm environments, respectively.

α^p and α^m = value of sensitivity parameter in the prototype and the microcosm environments, respectively.

$S^p_i(t-\Delta T^p)$ and $S^p_{i+1}(t-\Delta T^p)$ = the speed of the i th and $i+1$ th vehicle at time $t-\Delta T^p$ in the prototype environment

$S^m_i(t-\Delta T^m)$ and $S^m_{i+1}(t-\Delta T^m)$ = the speed of the i th and $i+1$ th vehicle at time $t-\Delta T^m$ in the microcosm environment

$X^p_i(t)$ and $X^p_{i+1}(t)$ = the position of the i th and $i+1$ th vehicle at time t in the prototype environment

$X^m_i(t)$ and $X^m_{i+1}(t)$ = the position of the i th and $i+1$ th vehicle at time t in the microcosm environment

$S^p_i(t)$ and $S^m_i(t)$ = speed of i th vehicle at time t in the prototype and the microcosm environments, respectively.

$a_i^p(t)$ and $a_i^m(t)$ = acceleration of i^{th} vehicle at time t in the prototype and the microcosm environments, respectively.

$S_{\max}^p$ = maximum vehicular speed in the prototype environment

$a_{\min}^p$ and $a_{\max}^p$ = minimum and maximum vehicular acceleration in the prototype environment

$X_{i+1,\min}^p(t)$ and $X_{i+1,\min}^m(t)$ = minimum spacing between vehicle $i+1$ and vehicle i at time t in the prototype and the microcosm environments, respectively, given by

$$X_{i+1,\min}^p(t) = L^p + h_{\min}^p S_{i+1}^p(t) \text{ and } X_{i+1,\min}^m(t) = L^m + h_{\min}^m S_{i+1}^m(t) \qquad (15)$$

While adaptation time ($\Delta T^p, \Delta T^m$) of all drivers is considered to be same in prototype and microcosm environments, respectively in deterministic modeling, ΔT^p and ΔT^m varies from 0 to $\Delta T_{\max}$ seconds in stochastic modeling and is chosen from a probabilistic adaptation time distribution.

To set the minimum spacing between vehicles, we assume a linear relationship in the form $X_{i+1,\min}(t) = L + h_{\min} S_{i+1}(t)$, where L is the effective vehicle length (vehicle length plus gap), $h_{\min}$ is the minimum headway, and $S_{i+1}(t)$ is the current speed of vehicle. Applying this relationship to both prototype and microcosm environments will set $X_{i+1,\min}^p(t) = L^p + h_{\min}^p S_{i+1}^p(t)$ and $X_{i+1,\min}^m(t) = L^m + h_{\min}^m S_{i+1}^m(t)$. Since minimum spacing between vehicles and vehicle length variables are not to be scaled (to preserve density), then we set $h_{\min}^m S_{i+1}^m(t) = h_{\min}^p S_{i+1}^p(t)$, or $\dfrac{S_{i+1}^m(t)}{S_{i+1}^p(t)} = \dfrac{h_{\min}^p}{h_{\min}^m} = r$.

This shows that the minimum headway in the microcosm environment should be scaled by the reciprocal of *r*. Since capacity can be defined by the reciprocal of minimum headway, then we deduce that the capacity in the microcosm environment will consequently be scaled by the ratio r, or $\frac{h_{\min}^{p}}{h_{\min}^{m}} = \frac{\mu^{m}}{\mu^{p}} = r$. Since both flow rate and capacity were downscaled by the same ratio r, then the flow-to-capacity ratio (q/μ) in the downsampling process remains unchanged. Clearly, this is a critical system operating characteristic that is preserved in the downsampling process.

3.4.2. INTERPRETATION OF CONSTRAINTS

The constraints imposed by Equation (6) and (7) are derived from the adopted car-following model (GM-1) in the prototype and the microcosm environments, respectively. However, in Equation (6) the values of α^{m} and ΔT^{m} are not known and need be optimized by minimizing the cost function SSE. Equations (8) and (9) impose the constraint on the spacing between two consecutive vehicles in the stream with reference to the prototype and the microcosm environments, respectively. The spacing must remain larger than or equal to a minimum practical value to maintain safe distance and prevent collision at different vehicle speeds. The minimum spacing between vehicles can be assumed a function of the speed of the following vehicle $i+1$, where small spacing is associated with low speeds and large spacing is required at high speeds.

In Equations (10) and (11) constraints are imposed on the speed value of vehicles in the prototype and the microcosm environments, respectively. The speed value must be non-negative and should not exceed the maximum driver's free-flow speed. Note that the maximum free-flow speed in the microcosm environment is determined by downscaling its value in the prototype environment with the same ratio *r*. This is necessary to retain the kinematic similarity as explained earlier. Equations (12) and (13) impose constraints on the acceleration/deceleration rates of vehicles in the prototype and the microcosm environments, respectively. Acceleration rates should also fall within a practical range of values, often determined by driver and vehicle characteristics. Similarly, the maximum acceleration/deceleration rates in the microcosm environment are downscaled by the same ratio to maintain kinematic similitude. The last two constraints exhibited by Equations (14) are simply non-negativity constraints for the two unknown behavioral parameters.

3.5.PERFORMANCE MEASURES

Performance measures indicate the level of achievement of the desired or intended objective. In the context of this study, Performance measures are necessary to assess the efficiency of the proposed methodology and to verify whether the desired objective has been achieved. The performance measures used in this study are Root Mean Square Error (RMSE) and Average Vehicular Delay.

3.5.1. RMSE

The trajectory of each vehicle in the microcosm environment after simulation was compared with corresponding trajectory of that vehicle in the prototype environment and the root mean of sum of squared (RMSE) difference for all the vehicles is computed as error. RMSE is derived from the sum of squared errors (*SSE*) expressed in Equation (4). Mathematically,

$$RMSE = \sqrt{\frac{SSE}{(rN)(T/\delta t)}} \tag{16}$$

Where all the variables are as described before.

This cost function defines the magnitude of the error between the trajectories prior to and after downsampling. The objective is to obtain a minimum value of error for all the vehicles. A low RMSE value indicates minimum loss of information in the downsampling process. Ideally, zero RMSE value is desired but is not possible as RMSE is evaluated based on microscopic driver behavior; the errors are cumulative and do not cancel out. Based on the computed error, α^m and ΔT^m values (sensitivity and adaptation time, respectively) are adjusted accordingly to further minimize the error in the next iteration.

3.5.2. AVERAGE VEHICULAR DELAY

While optimization of the downsampling procedure is critically important to the overall simulation performance in the microcosm, it is equally important that the process be reversible so that we can upsample the system characteristics back to the prototype environment. To test the reversibility of the downsampling process, average vehicular delay is used as the other performance measure. Average vehicular delay at the end of the simulation period is often considered as a significant macroscopic measure collected from microscopic simulation models. The main goal is to examine the relationship between the average vehicular delay in the prototype and the microcosm environments. Ideally, an important system operating characteristic will be retained if the average vehicular delay in the microcosm and prototype environments is the same. Further discussion on delay can be found in the Results and Discussion chapter.

3.6.SUMMARY

A mathematical approach is developed to improve the computational efficiency of microscopic simulation systems. Performance measures are defined to assess the performance of the downsampling process. The developed methodology is now used to conduct the experimental analysis.

4. EXPERIMENTAL ANALYSIS

4.1. INTRODUCTION

The methodology discussed in the previous chapter is used to microscopically simulate a hypothetical traffic network described in the next section. This chapter also explains the experimental analysis that is conducted in two stages on this hypothetical network and is described in the subsequent sections. Finally, the concept of local stability in the prototype and microcosm environments is introduced. Local stability is an important system operating characteristic that should be retained in the microcosm environment.

4.2. STUDY SECTION

A basic homogenous single lane freeway segment of length L is considered. All the ideal geometric and operating conditions (12 ft width lane, 6ft lateral clearance, level terrain, 0% trucks, and commuting traffic) are assumed to prevail. The freeway section under study is also assumed to be free from on-ramps and off-ramps and hence there was only one entry and one exit point for the entire section. Such an arrangement is setup to ensure that the behavior of entities in the network is primarily determined by the car-following model and to negate the effect of other behavioral factors.

4.3. EXPERIMENTAL WORK

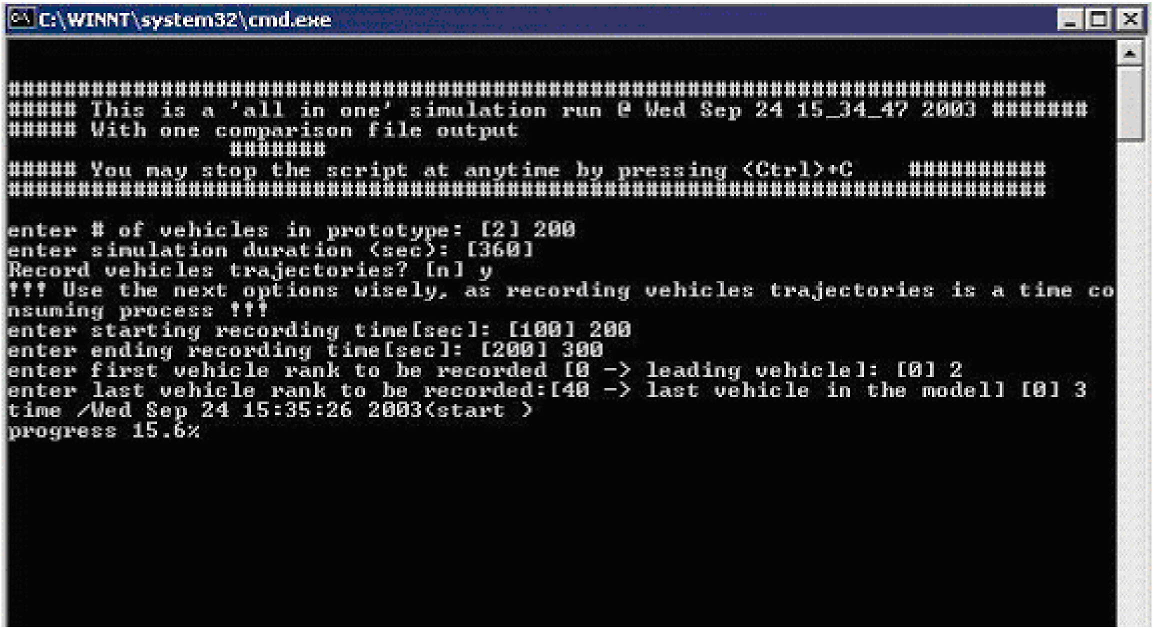
Experimental analysis was performed using a pilot simulation module, developed in Practical Extraction and Report Language (PERL), to perform the entire operation of downsampling, simulating in microcosm environment and upsampling the results back to the prototype environment. PERL is derived from the C programming language. The development of a new simulation module to perform the downsampling-upsampling operation was necessary as existing microscopic simulation programs cannot account for the behavioral scalability of system entities. In other words, the car-following and other behavioral models on which existing microscopic simulation programs are built on cannot be modified to account for the downsampling process as the program code of these simulation programs has a closed architecture and cannot be accessed by the user. Some simulation programs, like PARAMICS and AIMSUN2, have an open architecture but these modules fail to provide total access/control to modify the source code.

The developed program, when executed in batch mode, gives us the option to enter some of the traffic and vehicle operating characteristics like number of vehicles, duration of simulation, vehicle position updating time. The downsampling ratio, flow rate (rate at which vehicles are released into the system), the car-following parameters in the prototype and microcosm environments are hard-coded in the program. However, they can be modified by accessing the program code directly. A range of sensitivity and adaptation time values for the microcosm environment can be specified to produce a set of RMSE values as output. Other forms of output that are generated by the program include Mean Square Error (MSE), Absolute Average Relative Error (AARE) and Average Vehicular Delay. However, only RMSE and Average Vehicular Delay are used as performance measures in this study for further analysis. The generated output is written into a text file (one text file each for a combination of flow rate, downsampling ratio, number of vehicles in the prototype environment and a range of car-following parameters for the microcosm environment). Table 4-1 shows an example of a text file produced from each combination. The program code is accessed to modify the hard-coded variables (flow rate, downsampling ratio and range of car-following parameters in microcosm environment) to generate different outputs in the form of a text file.

Table 4-1 : RMSE values generated by the Program

Scenario 1	Vehicles in Prototype - 100					Prototype Flow rate - 500 (veh/hour)				
	Adaptation Time (sec)									
Sensitivity	1.6	1.8	2	2.2	2.4	2.6	2.8	3	3.2	3.4
0.23	334.90	333.99	335.50	342.23	354.04	384.48	568.05	832.93	1112.11	1321.26
0.24	179.52	173.16	171.80	176.95	187.91	283.63	537.99	828.92	1038.35	1231.08
0.25	126.79	112.31	100.71	91.34	89.19	254.90	548.58	761.24	952.83	1141.83
0.26	225.62	214.93	202.34	188.33	129.32	277.38	493.93	682.14	868.76	1057.49
0.27	355.39	346.20	333.51	311.80	179.27	255.80	423.80	603.54	789.58	979.35
0.28	483.45	474.16	460.98	406.05	213.59	211.70	354.09	529.68	715.41	906.73

These text files can be exported to any database (Microsoft Excel® or Microsoft Access®) for further analysis. A snapshot of the simulation module developed and used in this study is shown in Figure 4-1. The trajectories of all the vehicles in the microcosm and prototype at each updating time interval can also be obtained but this option is used cautiously as the simulation program requires heavy computational resources for this operation.



```
############################################################################
##### This is a 'all in one' simulation run @ Wed Sep 24 15_34_47 2003 ########
##### With one comparison file output
               #######
##### You may stop the script at anytime by pressing <Ctrl>+C    ##########
############################################################################

enter # of vehicles in prototype: [2] 200
enter simulation duration (sec): [360]
Record vehicles trajectories? [n] y
!!! Use the next options wisely, as recording vehicles trajectories is a time co
nsuming process !!!
enter starting recording time[sec]: [100] 200
enter ending recording time[sec]: [200] 300
enter first vehicle rank to be recorded [0 -> leading vehicle]: [0] 2
enter last vehicle rank to be recorded:[40 -> last vehicle in the model] [0] 3
time /Wed Sep 24 15:35:26 2003(start )
progress 15.6%
```

Figure 4-1: Snapshot of simulation module window showing different parameters

The constraints on kinematic characteristics of the vehicle like speed and acceleration were hard-coded and were preset to a minimum and maximum values of 0 and 110 fps, respectively for speed and -15 to +15 fps^2, respectively for acceleration. The minimum spacing between two vehicles is calculated from Equation(15). In the equation, the vehicle length and minimum headway were assumed to be 20 ft and 1.125 seconds, respectively. The vehicle updating time was fixed at 0.1 seconds. This study was restricted to using the first car following model developed by General Motors Research group in late 1950's. The experimental work was also restricted to the case of one-lane freeway segment and assuming deterministic driver characteristics; i.e. the sensitivity and adaptation parameters in the prototype were assumed the same for all drivers ($\alpha^p = 0.5$ sec^{-1} and $\Delta T^p = 1.0$ sec). This is said to be a deterministic approach which is contrary to the behavior in existing microscopic simulation models, which are essentially stochastic.

4.3.1. STAGE ONE

The main goal of this experimental analysis is to derive an optimal solution for sensitivity and adaptation time in the microcosm environment by varying the flow rate, number of vehicles and operating conditions in the prototype with 50% downsampling ratio. The experimental work did not account for stochastic driving behavior; i.e. the sensitivity and adaptation parameters are assumed the same for all drivers ($\alpha^p = 0.5$ sec^{-1} and $\Delta T^p = 1.0$ sec) in the prototype. The three variables which were controlled in this experiment: (1) average flow rate used to generate vehicles at the entry point; (2) number of simulated vehicles in the prototype environment; and (3) operating conditions along the freeway segment, generated 36 cases that we seek optimal solution for and the analysis was conducted for all possible combinations of the three variables. For each case, the search for α_0^m and ΔT_0^m was carried out using combinations of a wide range of values that was later narrowed down to locate the minimum RMSE values more precisely.

In this stage, we restrict the downsampling ratio to 50% (r = 0.5) and set the simulation updating period to 0.1 seconds. Four levels of traffic flow rates are considered (500, 1000, 1500, and 2000 vph). Different numbers of simulated vehicles are considered (100, 200, and 300). The operating conditions along the segment are set by a predefined trajectory of a lead vehicle in the stream. Trajectory of leading vehicle is obtained by assigning typical (practical) acceleration and deceleration values for over a period of time. The three different operating scenarios considered are as shown in Figure 4-2. Each scenario is obtained by increasing the number of stops the lead vehicle makes throughout the simulation period (one hour). Varying the operating scenarios is necessary to introduce mixed free-flow and forced-flow conditions that activate the car-following behavior. Each stop lasted for 5 minutes and was followed by acceleration and a cruising period at free-flow speed.

4.3.2. STAGE TWO

At this stage, the concept of microscopic simulation systems scalability is further explored to examine the effect of different downsampling ratios on the overall performance of the procedure. The following downsampling ratios are considered: $r = \frac{1}{3}, \frac{1}{4}, and\ \frac{1}{5}$. Here, the operating conditions along the freeway segment were defined by the trajectory of the lead vehicle that follows scenario three with three complete stops.

The downsampling performance was evaluated under different flow rates (500, 1000, 1500, and 2000 vph), different number of simulated vehicles (100, 200 and 300) in

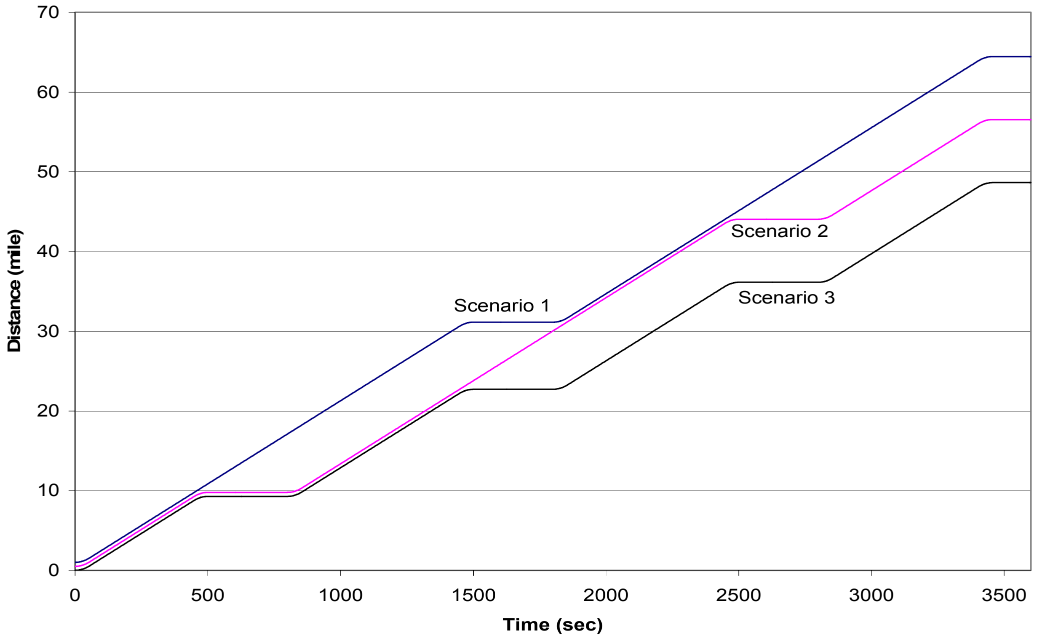


FIGURE 4-2: SCENARIOS OF LEAD VEHICLE TRAJECTORIES.

the prototype environment, and different downsampling ratios as mentioned before. The different combinations of all factors considered generated a total of 36 cases. The optimal values of the sensitivity parameter (α_0^m) and adaptation time (ΔT_0^m) were searched numerically for each case. The downsampling performance was measured in terms of the root mean square error (RMSE), which is derived from the objective error function *E* and reflects the amount of information loss caused by downsampling.

4.3.3. STAGE THREE – EXAMINING LOCAL STABILITY

Local stability addresses the issue of stability in the car-following model. It characterizes the response of the following vehicle to the fluctuations in the motion of the lead vehicle. For a given range of model parameters (sensitivity and adaptation time) in a car-following model, the traffic stream can be characterized as locally stable based on a set of criteria that is discussed later on (Revised monograph, 2000). The formulation proposed by Herman et al. (1959) (Equation (17)) determines these set of criteria to verify the existence of local stability between pairs of vehicles. The formulation is derived exclusively for the first GM car-following model described in Equation (1).

From Equation (1), re-scaling the time *t* in units of adaptation time, Δt, using the transformation, $t = \tau.\Delta t$, we obtain:

$$a_{n+1}(\tau + 1) = C(\dot{x}_n(\tau) - \dot{x}_{n+1}(\tau)) \tag{17}$$

Where,

$C = \alpha.\Delta t$ and

τ is the scaling parameter

Equation (17) is solved using Laplace transforms for *C* (given by sensitivity multiplied by the adaptation time). An extremely low value of *C* implies that the change in spacing over time between the lead and the following vehicle is non-oscillatory and will be exponentially damped. The change in spacing over time is caused due to the disturbance created in the lead vehicle trajectory for a short period of time. Table 4-2 gives the interpretation of different values of *C* and establishes the set of criteria to examine local stability in a traffic stream.

Table 4-2 :Different values of *C* and their interpretation

Case I: $C \le e^{-1} (\approx 0.368)$	The change in spacing is non-oscillatory and exponentially damped
Case II: $e^{-1} \le C \le \pi/2$	The change in spacing is oscillatory and exponentially damped
Case III: $C = \pi/2$	The change in spacing is oscillatory with a constant amplitude
Case IV: $C \ge \pi/2$	The change in spacing is oscillatory with increasing amplitude

The concept of local stability is investigated in the third stage of the experimental analysis. This stage was dependent on the first two stages, as we examine the local stability in microcosm and prototype environments under optimal conditions for α and ΔT. Therefore, the car-following parameters used to obtain the response of the following vehicles in the microcosm environment are the optimized sensitivity and adaptation time values for different downsampling ratios obtained from the experimental work in stages one and two.

The trajectory of the lead vehicle is similar to the scenario 1 shown in Figure 4-2 but, in addition, a small disturbance is introduced. This disturbance is created by changing the speed profile of the lead vehicle in the prototype and microcosm environments at the rate of -5 ft/sec^2 and 5 ft/sec^2 for five seconds each, respectively. The disturbance in the lead vehicle trajectory is shown in Figure 4-3. The response of the following vehicles in the prototype and also in the microcosm is recorded using the

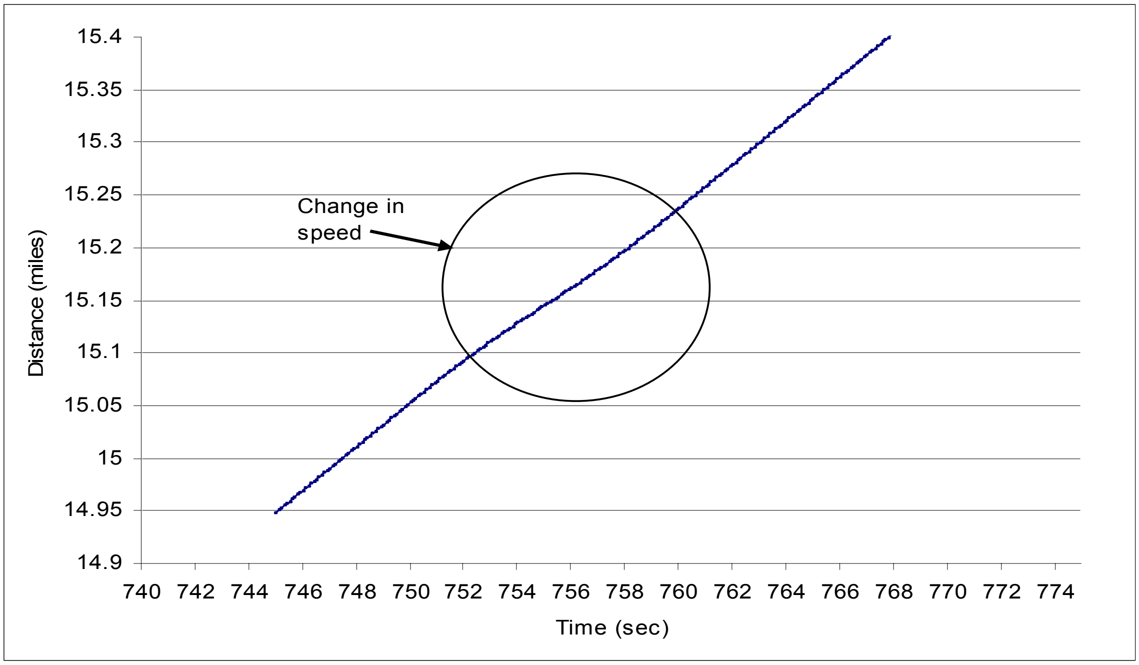


Figure 4-3 Disturbance created in lead vehicle trajectory to test local stability

“Record Trajectories” option in the simulation program.

An important system operating characteristic is retained in the downsampling process if the responses of the following vehicle in the prototype and microcosm environments, to the fluctuations in the lead vehicle, are similar. It follows that the following vehicle in a lead-following pair in the microcosm environment demonstrates similar behavior to its corresponding lead-following pair in the prototype environment.

4.4.SUMMARY

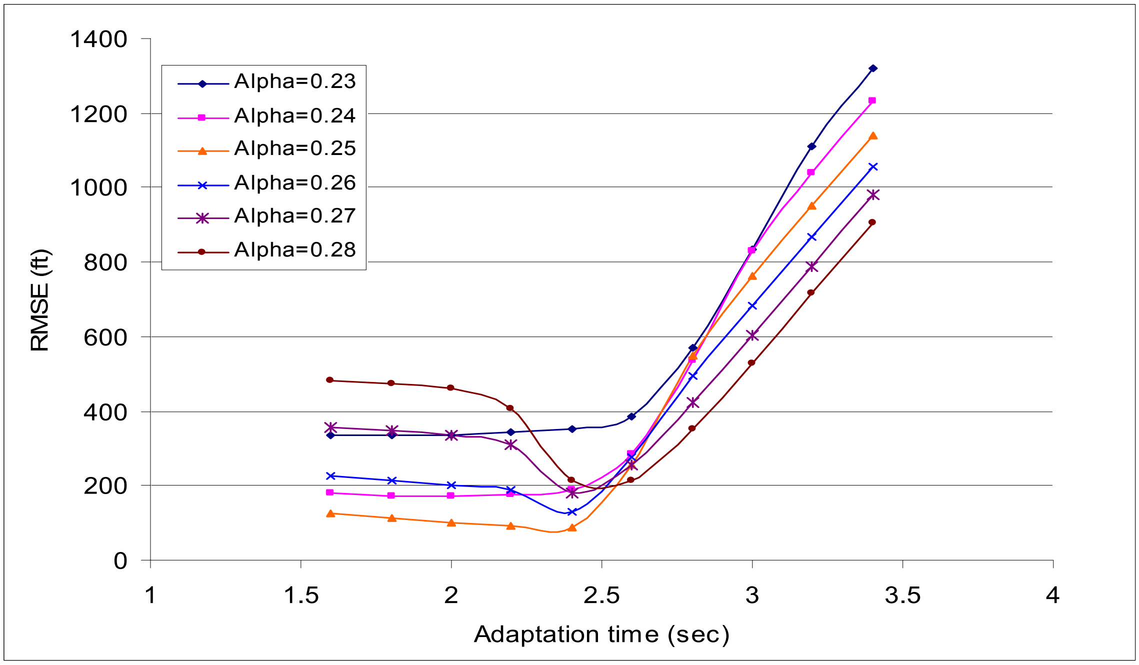
This chapter discussed the procedure used to conduct the experimental analysis. The three stages of the experimental analysis were described, where each stage forms the basis to the next stage. The first stage of the experimental work involves testing the methodology under different flow and operating conditions for 50% downsampling ratio. The second stage performs the experiment for different downsampling ratios under different flow conditions. The final stage examines the local stability in the prototype and microcosm environments. The results of the experimental analysis are discussed in the next chapter.

5. RESULTS AND ANALYSIS

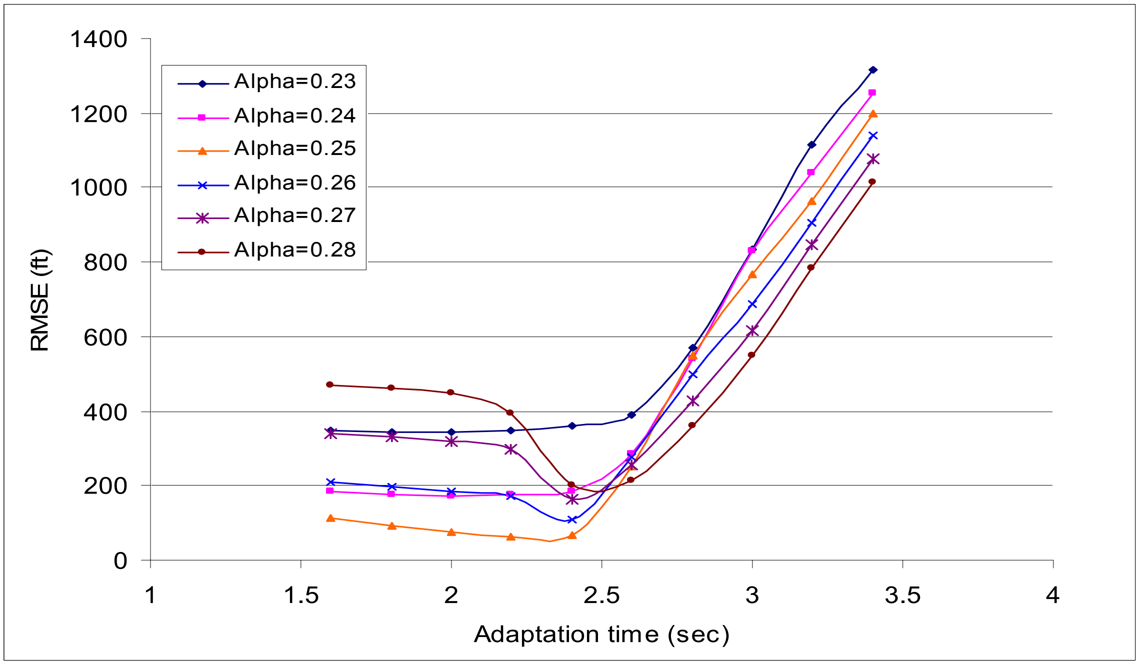
This chapter presents the results and discussion of the experimental work conducted to evaluate the downsampling process. The experimental work produced results for the two performance measures: RMSE and average delay. For the first stage of the experiment, the performance measures were derived for each of the 36 cases generated from all the possible combinations of the three variables (flow rate, number of vehicles in the prototype environment and operating conditions for a 50% downsampling ratio). Similarly, performance measures for the 36 cases (obtained by different combinations of downsampling ratio, flow rate and number of vehicles operating in scenario 3) were derived in the second stage of the experiment. The following sections describe and discuss the results in the order in which they were obtained.

5.1.EXPERIMENTAL RESULTS OF STAGE ONE

In this section, we present the results obtained from the experimental stage one. The RMSE and average vehicular delay values were obtained for each of the 36 cases. For each case, a range of sensitivity (α^m) and adaptation time (ΔT^m) values were tested to trap the optimal values of α^m and ΔT^m in the microcosm environment in terms of minimum RMSE. Figure 5-1 shows the change in RMSE values with α^m and ΔT^m for scenario 1 and 100 vehicles in the prototype environment. Each individual figure in Figure 5-1 represents a separate case and corresponds to a different flow rate. A range of 0.23 to 0.28 sec^{-1} for α^m and 1 to 4 seconds for ΔT^m was tested to generate the curves for each flow rate. A wider range of α^m and ΔT^m were used initially (α^m = 0.10 to 0.50 sec^{-1} with an interval of 0.05 and ΔT^m=1.0 to 6.0 seconds with an interval of 1.0 seconds), but the range was narrowed down to use smaller intervals to locate the exact

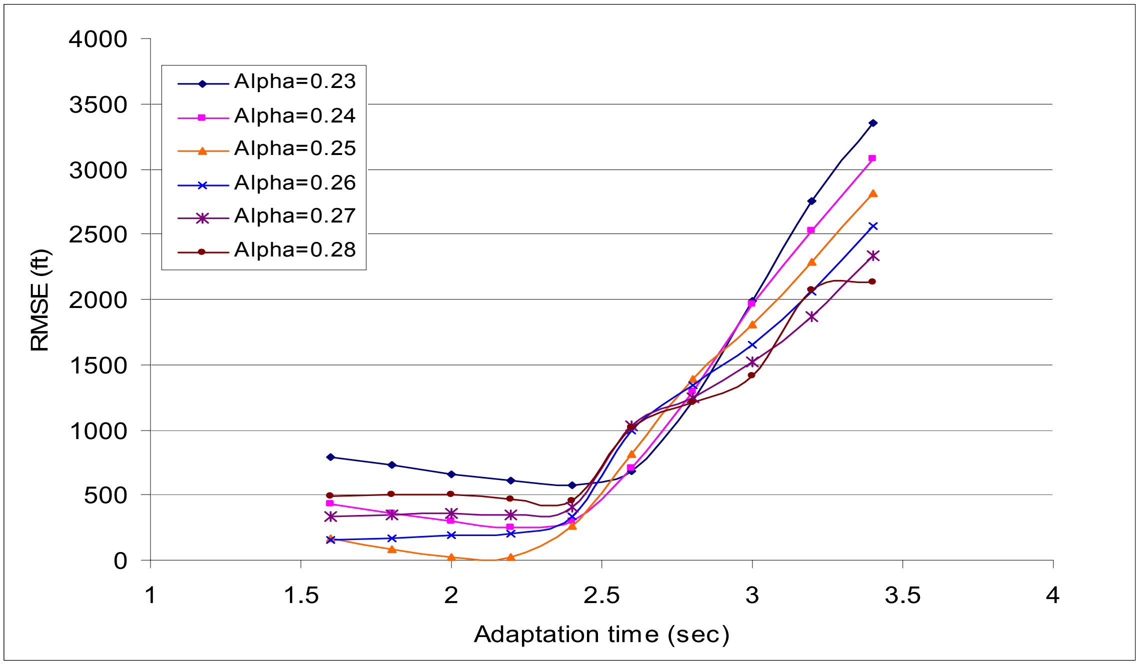


(a) $q^p = 500\ vph$

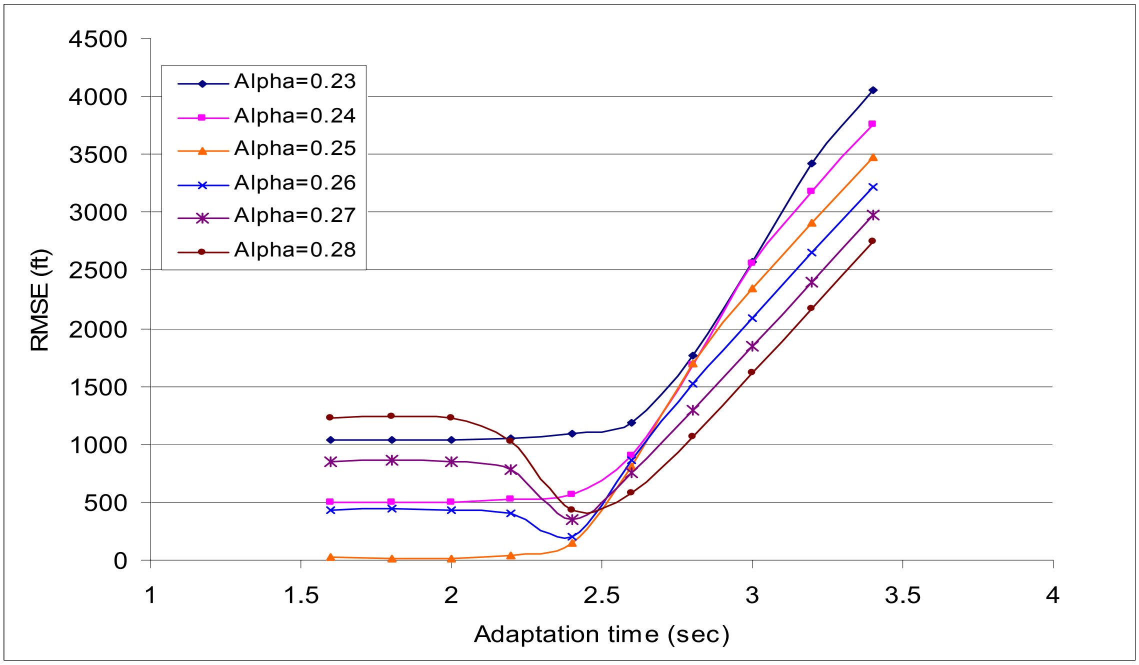


(b) $q^p = 1000\ vph$

Figure 5-1 : RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - Scenario 1 (N=100) ($q^p = 500{,}1000 vph$).



(c) $q^p = 1500\ vph$



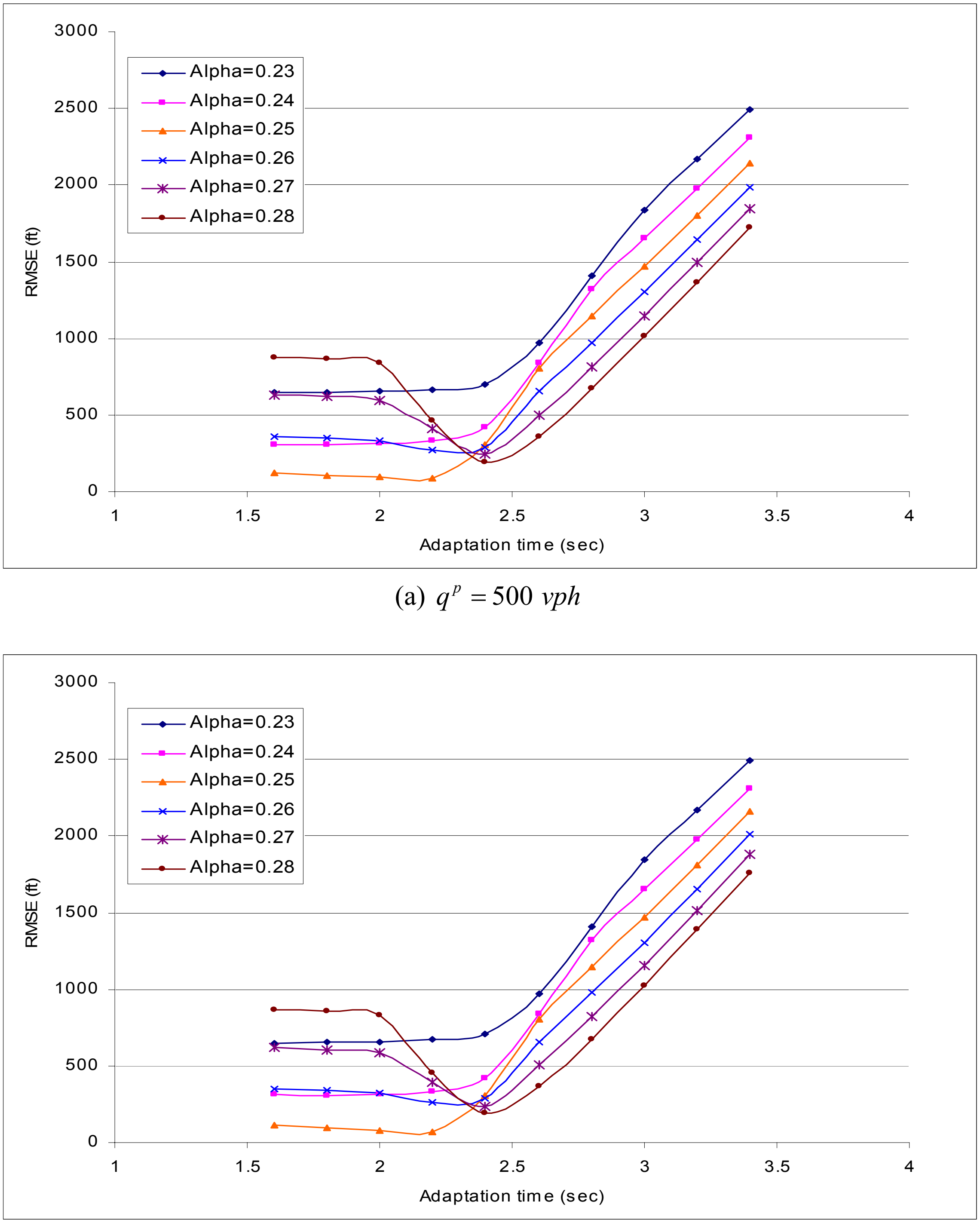
(d) $q^p = 2000\ vph$

Figure 5.1 (Continued):($q^p = 1500, 2000 vph$)

minimal RMSE values. The curves for different α^m converge to a minimum RMSE in the neighborhood of ΔT^m=2.0 seconds for all the flow rates. Similarly, RMSE curves are generated (using the simulation program) for cases 5 to 8 with 200 vehicles in the prototype environment for scenario 1. The same range of values for α^m and ΔT^m are used, as in cases 1 to 4, to obtain the optimal parameters that correspond to minimum RMSE. Figure 5-2 shows the RMSE curves for cases 5 to 8. Each figure in the Figure 5-2 represents a separate case and corresponds to a different flow rate. Figure 5-3 represents cases 9 to 12 with different flow rates that correspond to 300 vehicles in the prototype environment for scenario 1. For each case, a global minimum for RMSE corresponded to α_0^m=0.25 and ΔT_0^m in the neighborhood of 2.0 seconds. This global minimum reflects the minimum information loss that will result from applying the downsampling procedure. Although the global RMSE minima change with the flow rate, the corresponding optimal solution (optimal sensitivity and adaptation time in the microcosm environment) in each case remains almost the same.

The experimental work was performed in a similar manner for the remaining cases (cases 13 to 36) that correspond to scenarios 2 and 3, with each scenario generating 12 cases. The RMSE curves were generated for different flow rates and different number of vehicles in the prototype as shown in FIGURE 5-4 through FIGURE 5-9. The range of values for α^m and ΔT^m used to generate these curves were similar to cases 1 to 12. For cases 1 to 12, the minimum RMSE values corresponded to α_0^m=0.25 and ΔT_0^m in the neighborhood of 2.0 seconds.

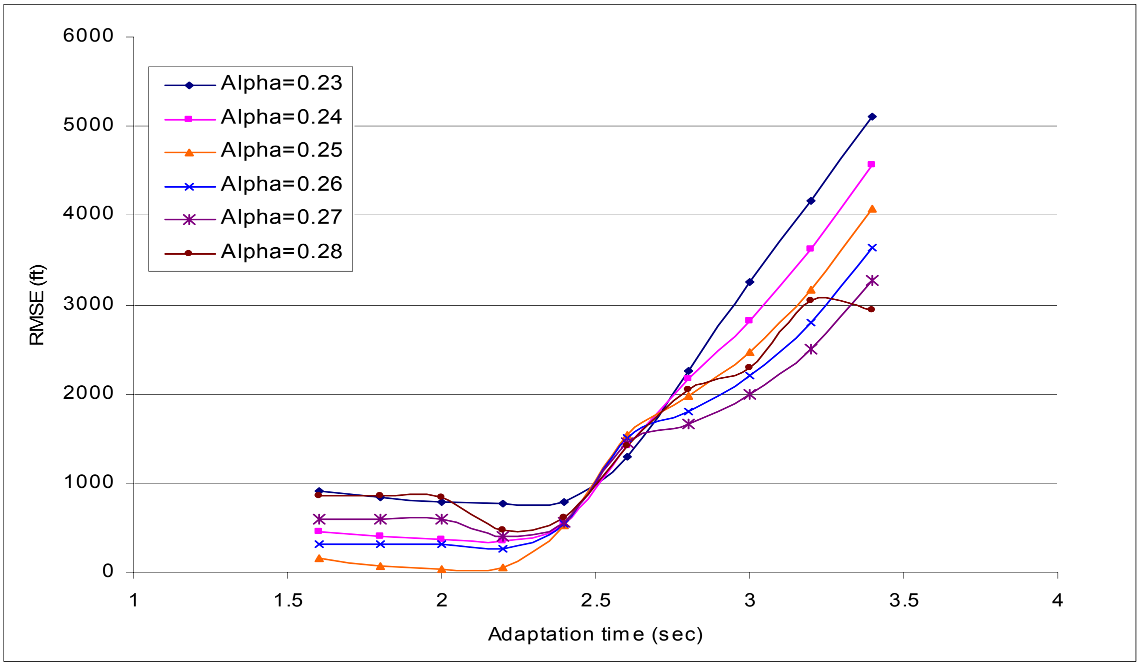
This observation appears to be consistent for scenarios 2 and 3 as well. This consistency in the optimal solutions is critically important to ensure that the equivalent



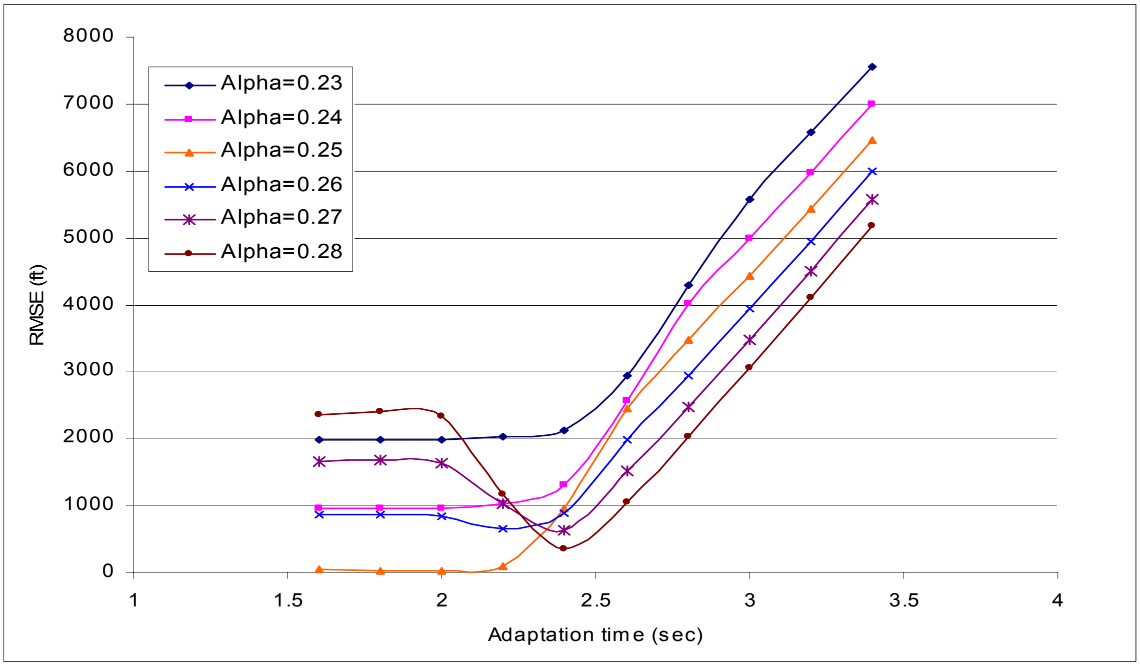
(a) $q^{p} = 500\ vph$

(b) $q^{p} = 1000\ vph$

Figure 5-2 : RMSE for different values of $\alpha^{m}, \Delta T^{m}$ and flow rates - Scenario 1 (N=200) ($q^{p} = 500{,}1000vph$).

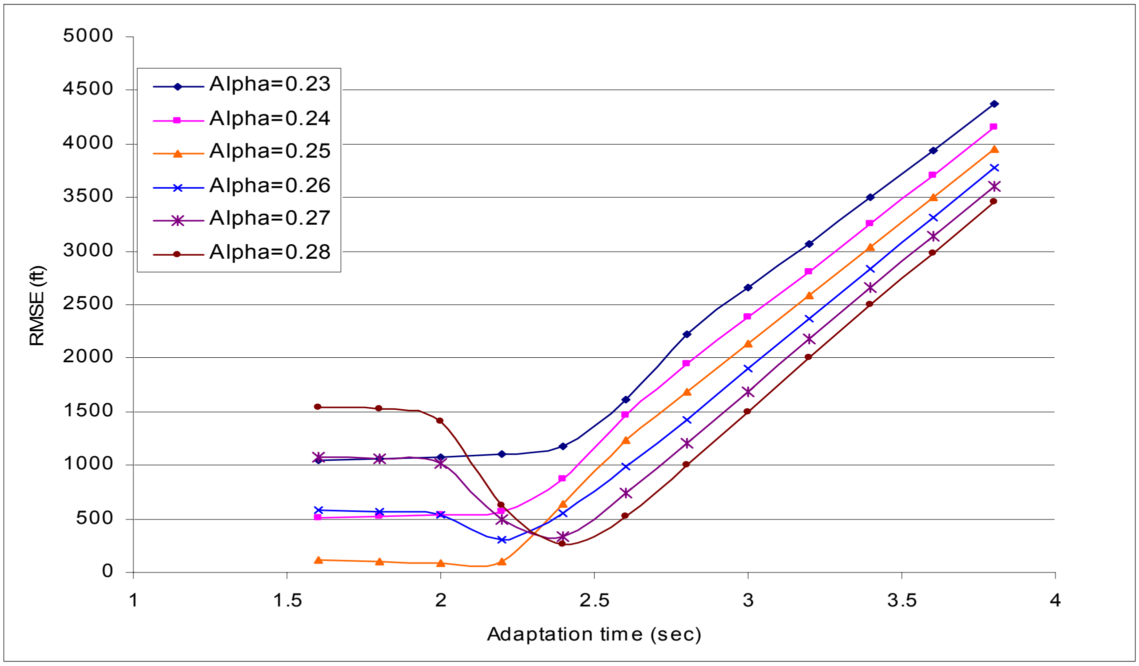


(c) $q^p = 1500\ vph$

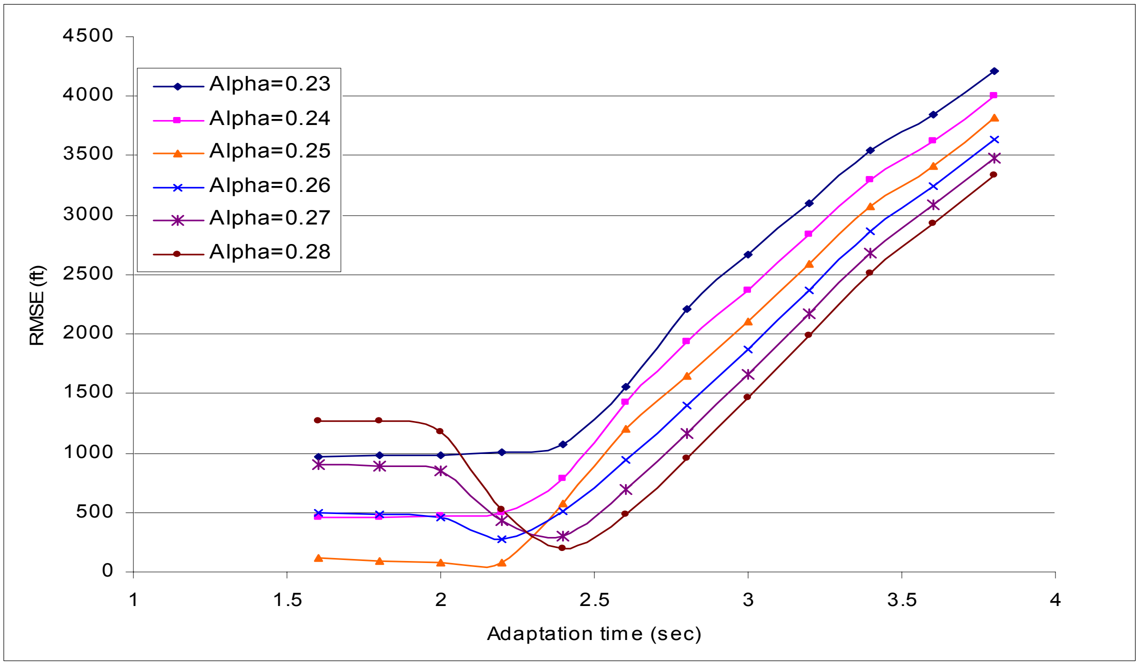


(d) $q^p = 2000\ vph$

Figure 5.2 (Continued):($q^p = 1500, 2000 vph$ **)**

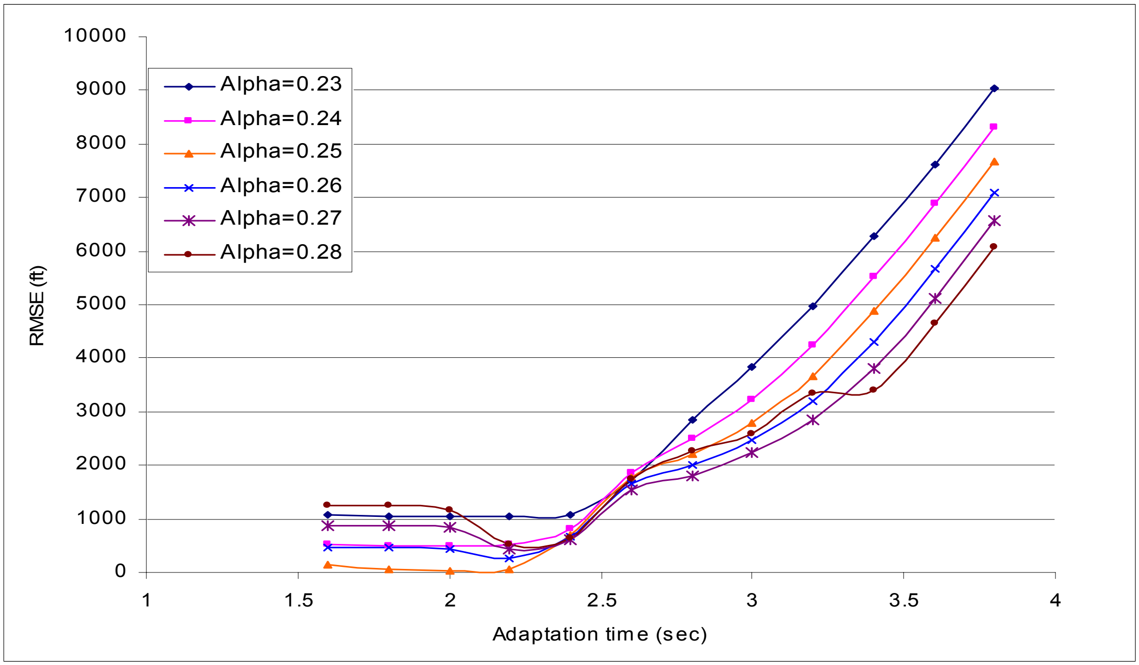


(a) $q^p = 500\ vph$

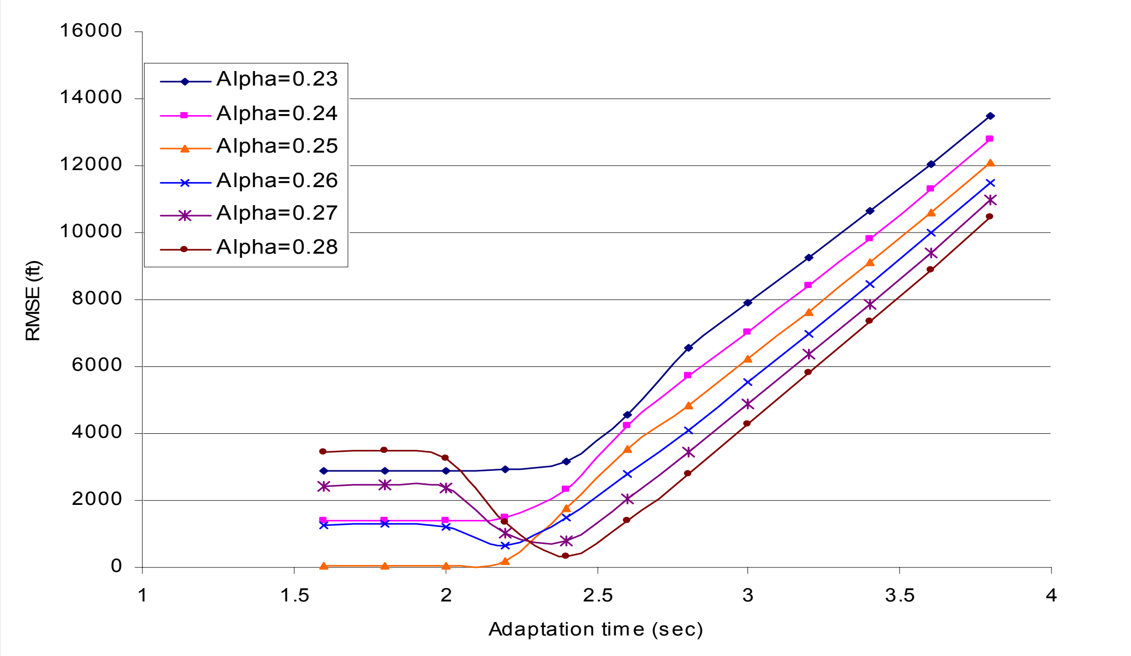


(b) $q^p = 1000\ vph$

Figure 5-3 : RMSE for different values of α^m, ΔT^m and flow rates - Scenario 1 (N=300) ($q^p = 500{,}1000 vph$).



(c) $q^p = 1500\ vph$



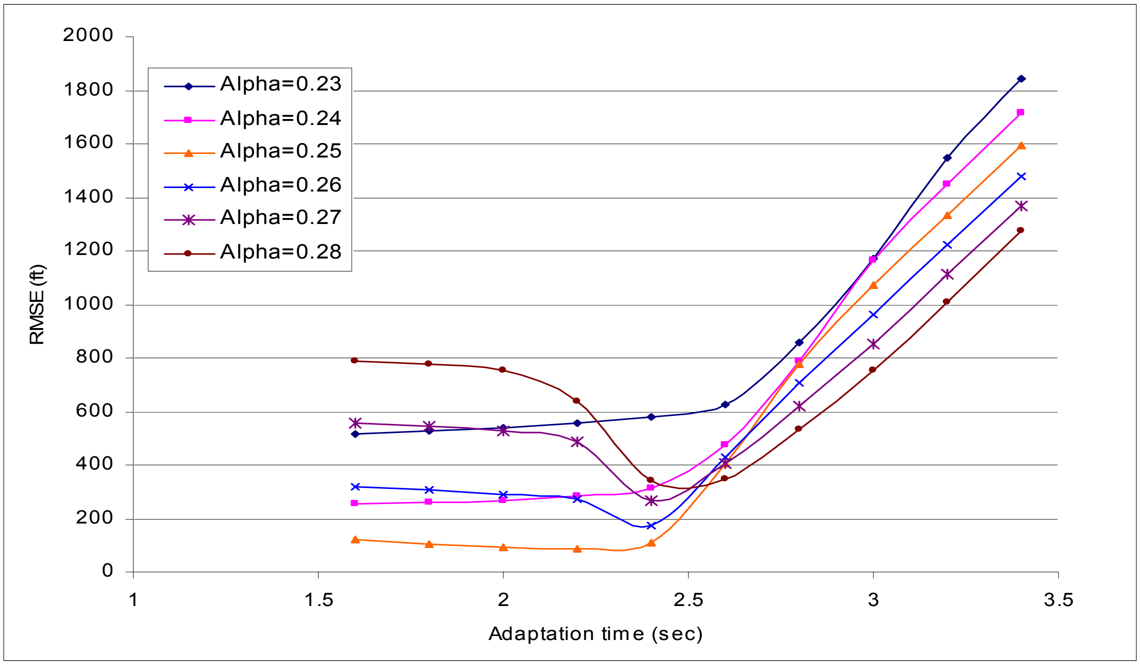
(d) $q^p = 2000\ vph$

Figure 5-3 (Continued): ($q^p = 1500, 2000 vph$**)**

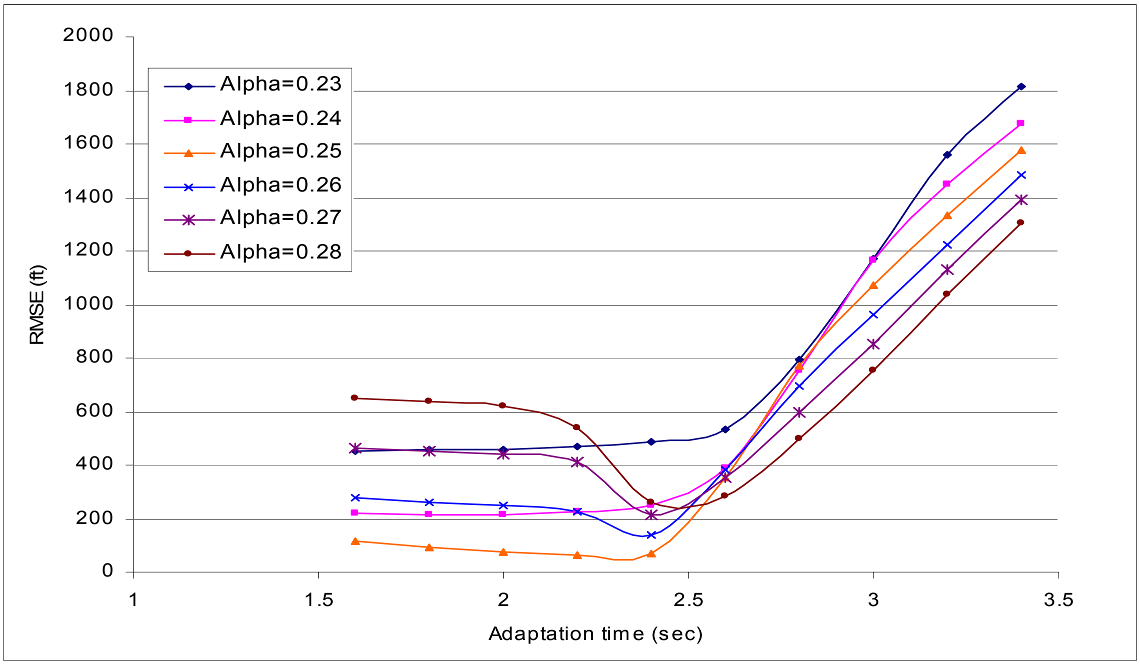
driving behavior in the microcosm is independent of the flow rate and driving conditions.

For all cases considered, the value of α_0^m was 0.25, which is half the assumed value of α^p. This suggests that the optimal sensitivity ratio ${}^{\alpha_0^m}\!/\!{}_{\alpha^p}$ is equal to the downsampling ratio *r*. The suggested linear relationship, however, must be further verified for different downsampling ratios before reaching a final conclusion. Similarly, the values of ΔT_0^m were in the neighborhood of 2.0 seconds, which is twice the assumed prototype value of ΔT^p. In some cases, however, slight deviations from 2.0 seconds were observed in the order of nearly 0.1 seconds, which could be attributed to the rounding effect of the simulation updating period (0.1 seconds). This observation suggests that the optimal adaptation ratio ${}^{\Delta T_0^m}\!/\!{}_{\Delta T^p}$ approaches the reciprocal of the downsampling ratio (1/*r*). However, further investigation is required to verify such relationship with different downsampling ratios which is explained in the next stage of the experiment.

Another important characteristic of the downsampling procedure that must be addressed here is how the optimal performance or minimum information loss varies from one case to another. This is necessary to examine the stability and robustness of the procedure. Using α_0^m=0.25 (optimal) and ΔT_0^m=2.0 (near-optimal), the system performance was examined, as shown in Figure 5-10, under different flow rates, operating conditions, and number of simulated vehicles. The figure clearly shows that the information loss in terms of RMSE for near-optimal conditions decreases with the increase in flow rate, regardless of the number of simulated vehicles or operating conditions. There is also a slight tendency for the errors to decrease (increase) at low

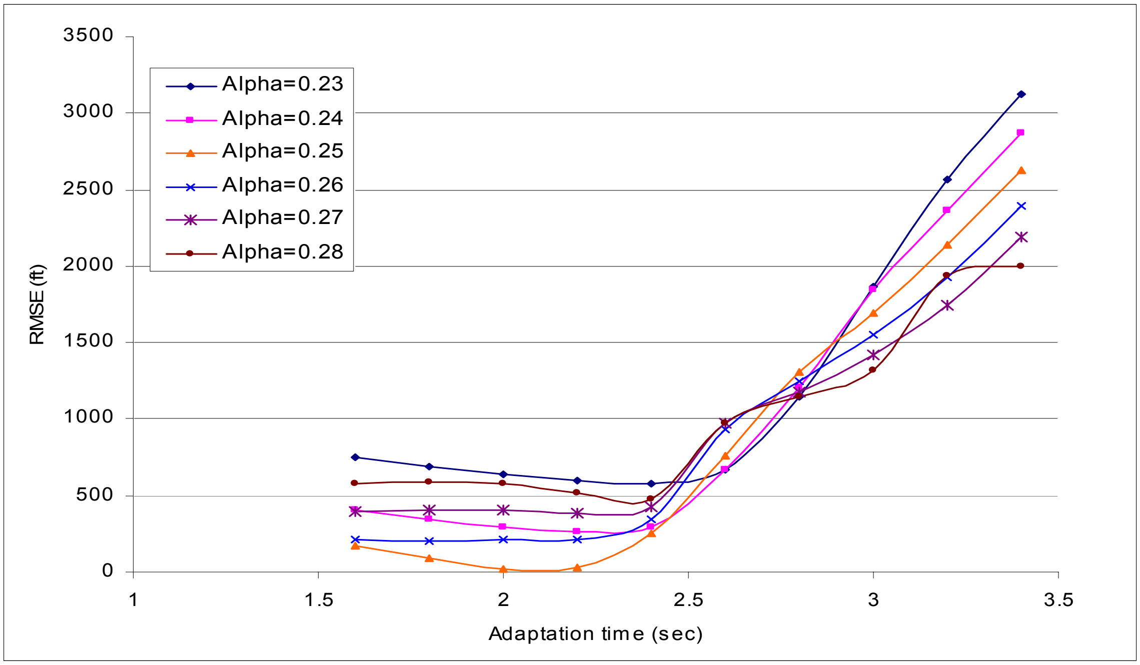


(a) $q^p = 500\ vph$

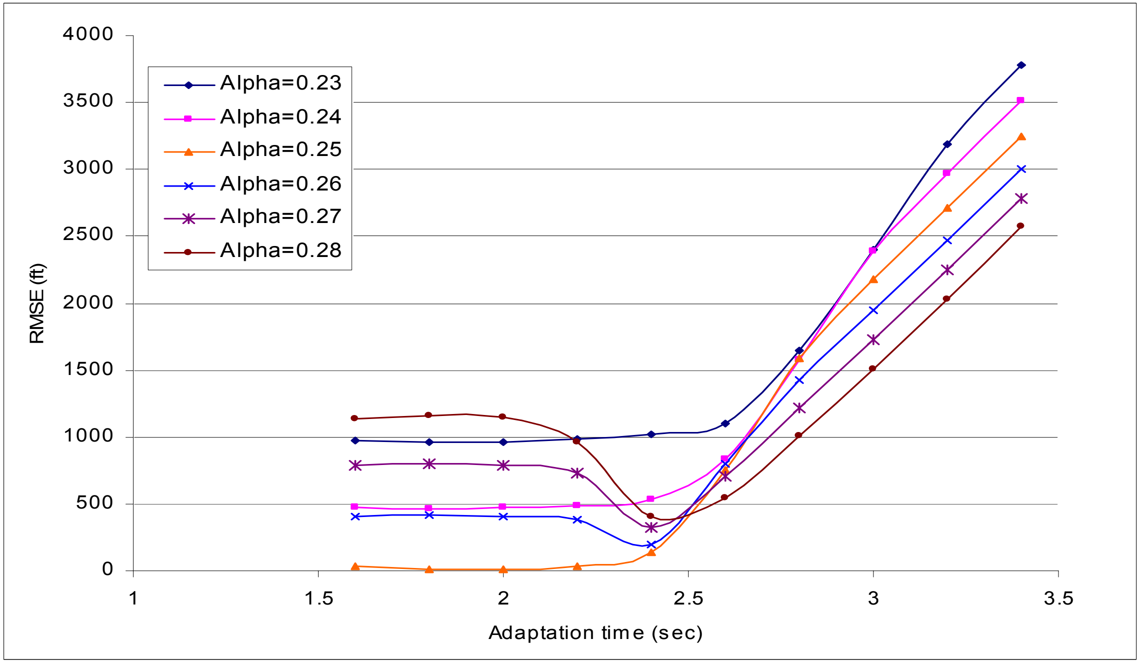


(b) $q^p = 1000\ vph$

Figure 5-4: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - Scenario 2 (N=100) ($q^p = 500, 1000 vph$)

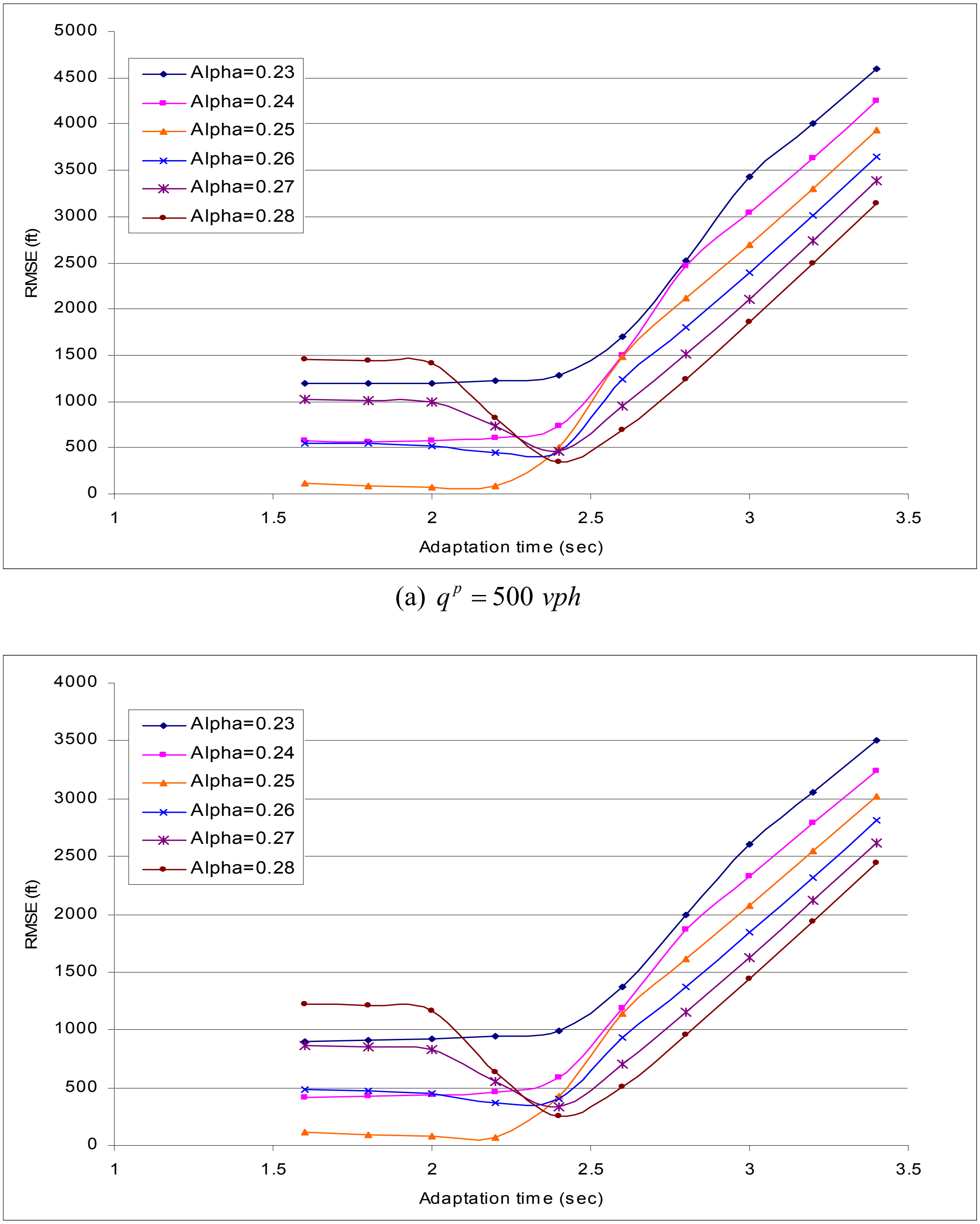


(c) $q^p = 1500\ vph$



(d) $q^p = 2000\ vph$

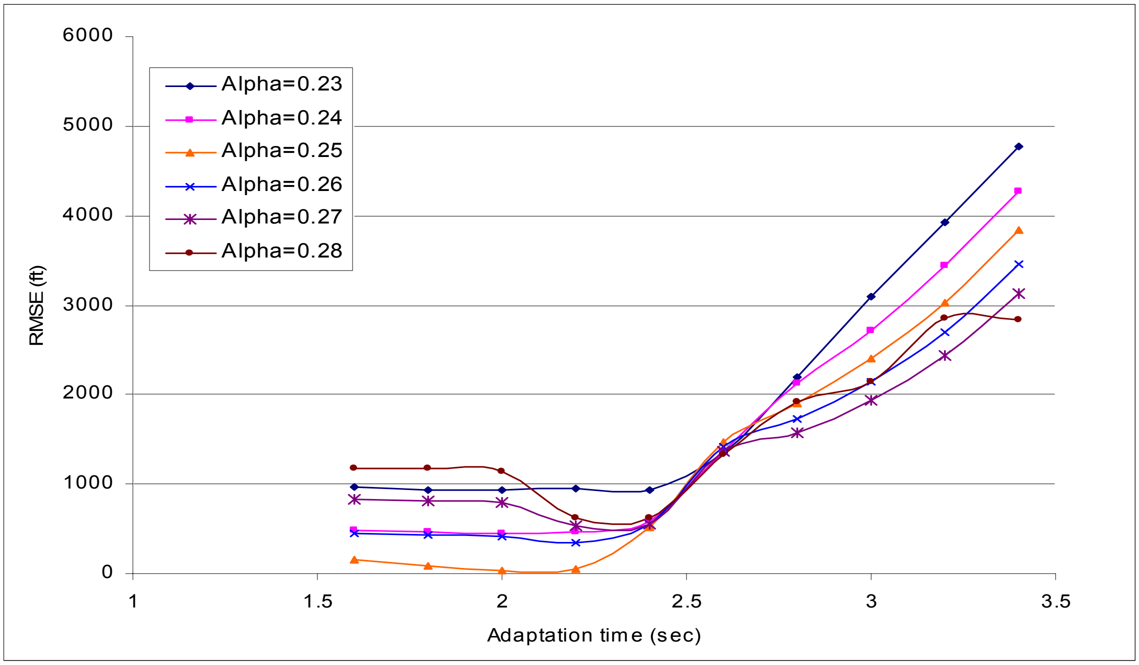
Figure 5-4 (Continued): ($q^p = 1500, 2000 vph$)



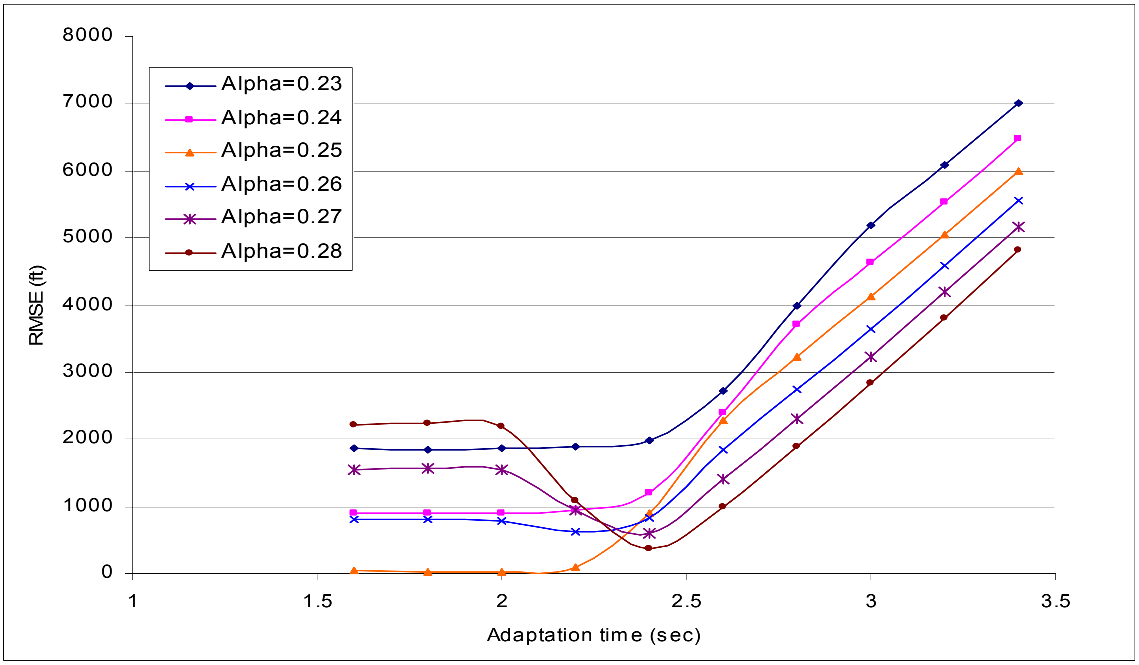
(a) $q^p = 500\ vph$

(b) $q^p = 1000\ vph$

Figure 5-5: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - Scenario 2 (N=200) ($q^p = 500, 1000 vph$)

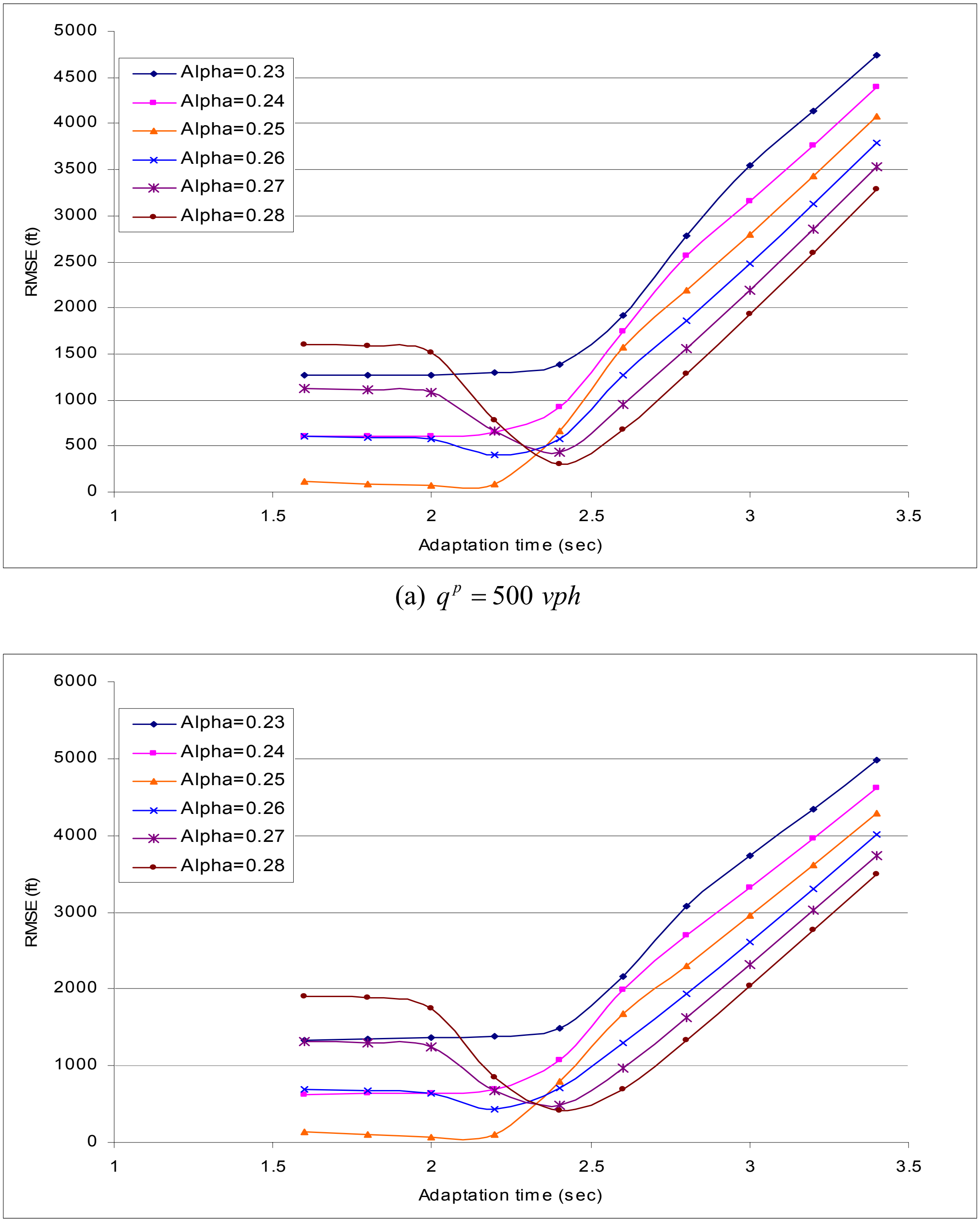


(c) $q^p = 1500\ vph$



(d) $q^p = 2000\ vph$

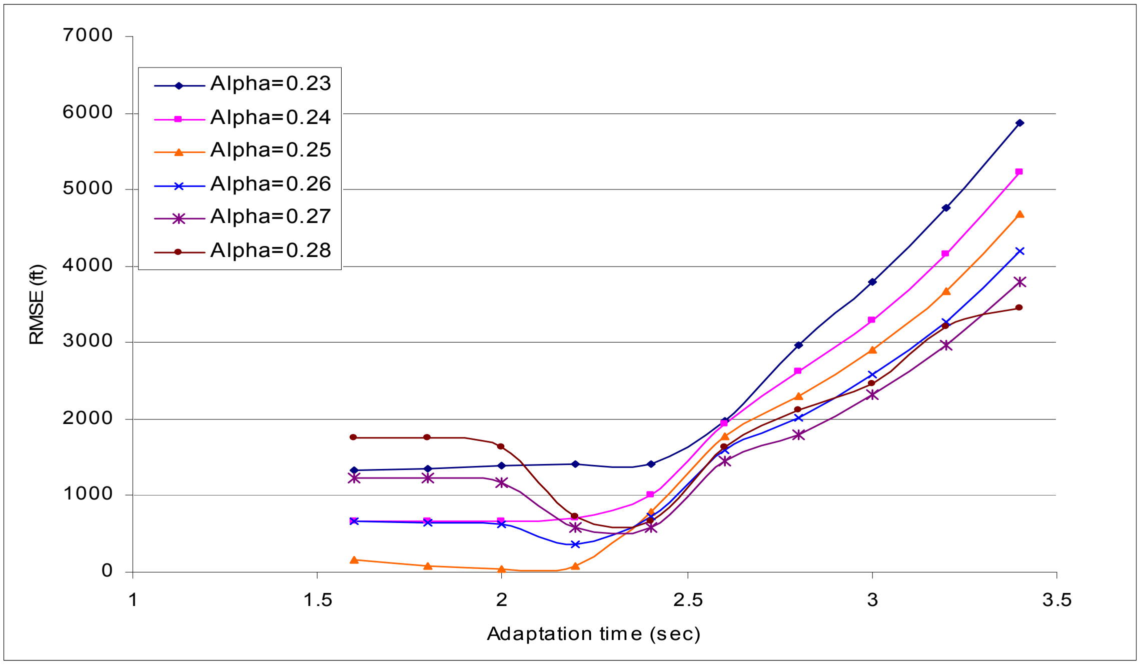
Figure 5-5 (Continued): ($q^p = 1500, 2000 vph$)



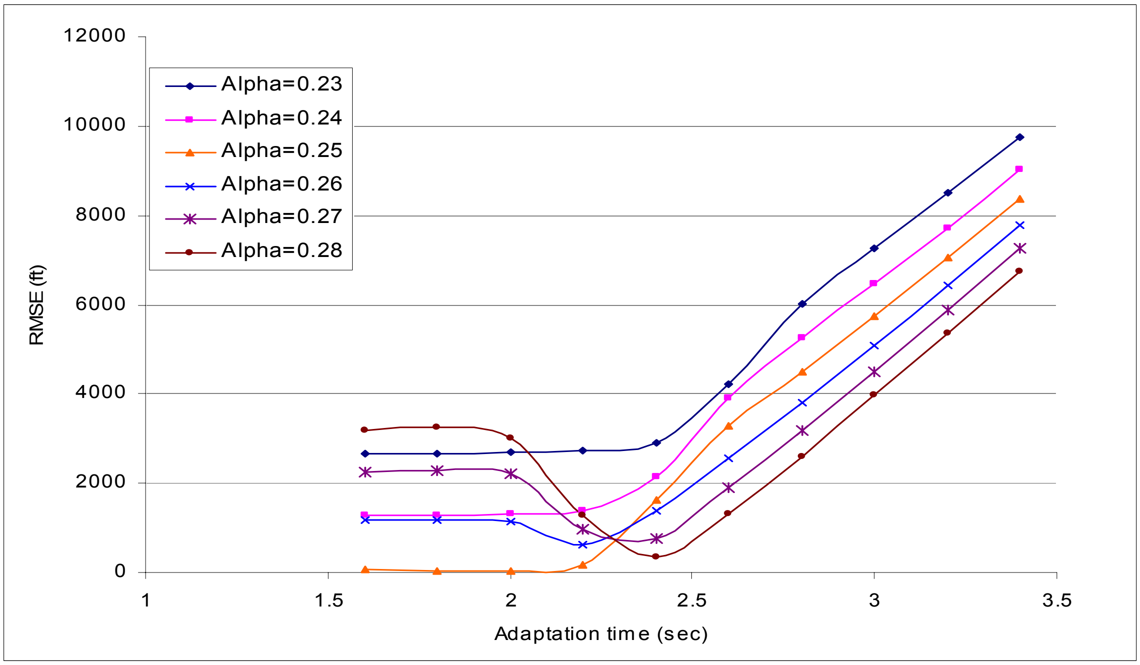
(a) $q^p = 500\ vph$

(b) $q^p = 1000\ vph$

Figure 5-6: RMSE for different values of α^m, ΔT^m and flow rates - Scenario 2 (N=300) ($q^p = 500, 1000 vph$)

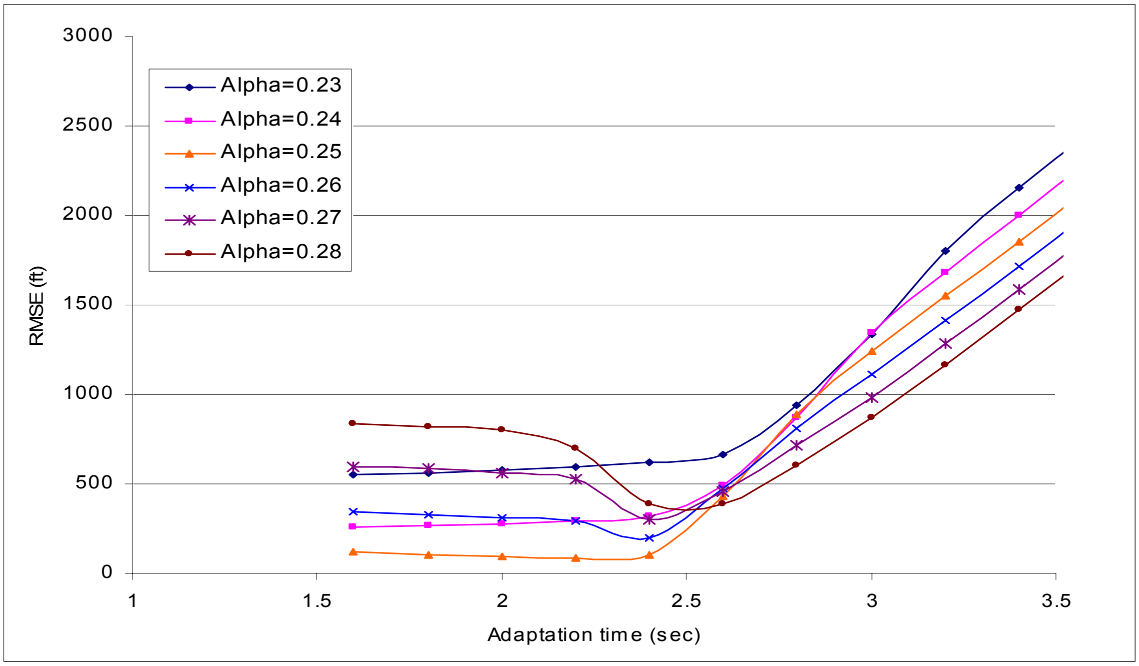


(c) $q^p = 1500\ vph$

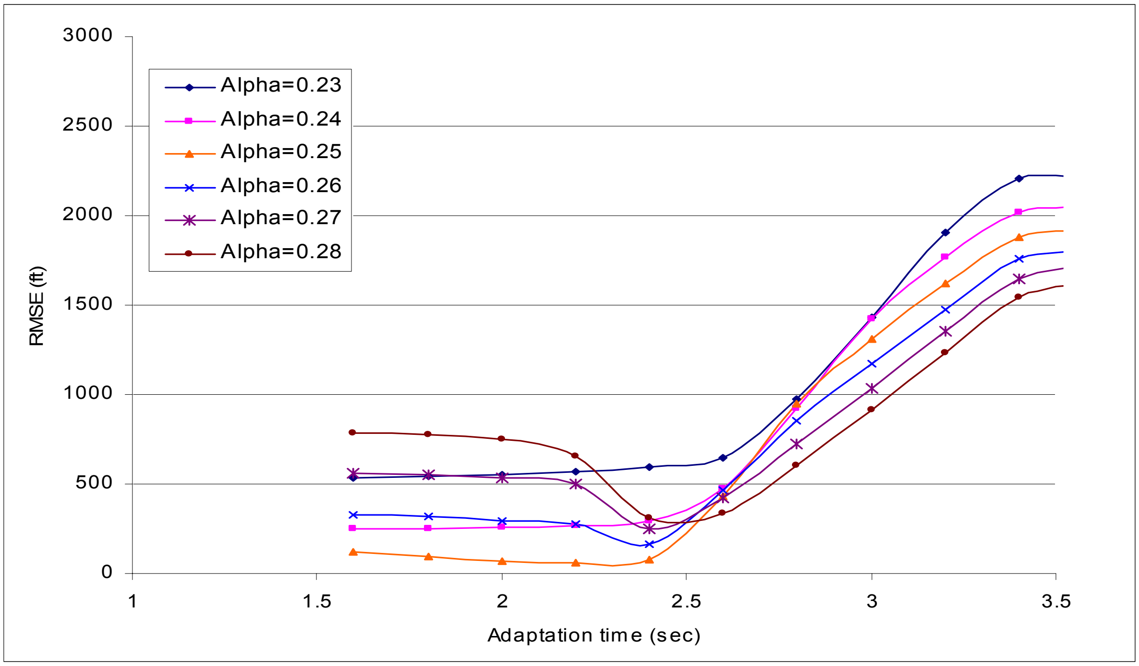


(d) $q^p = 2000\ vph$

Figure 5-6 (Continued): ($q^p = 1500, 2000 vph$**)**

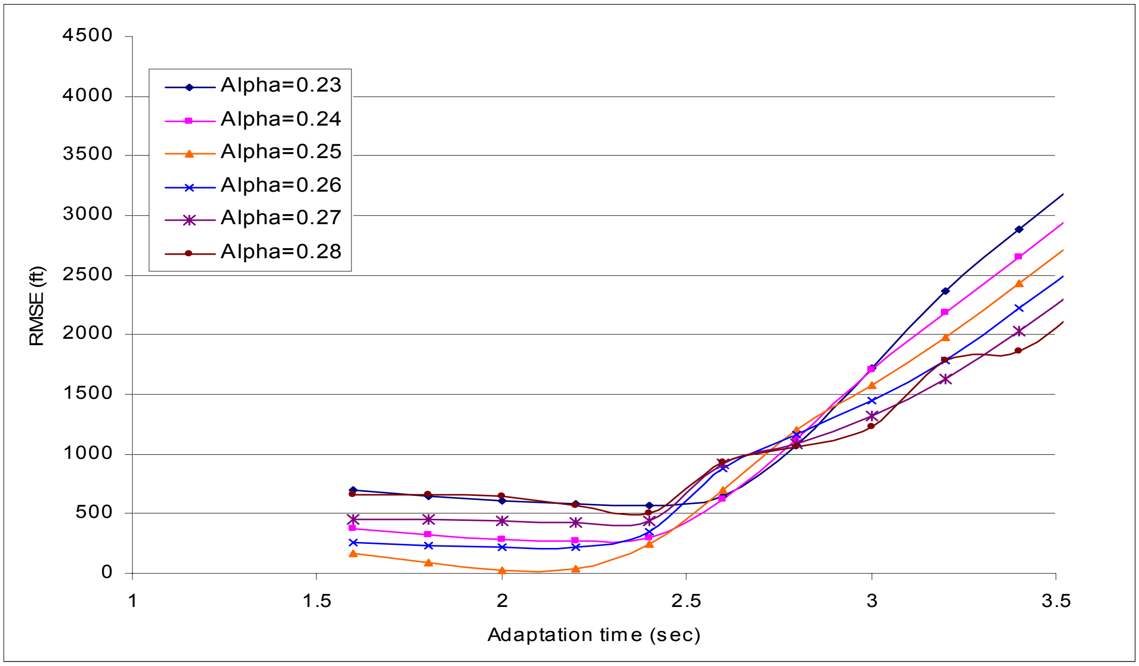


(a) $q^p = 500\ vph$

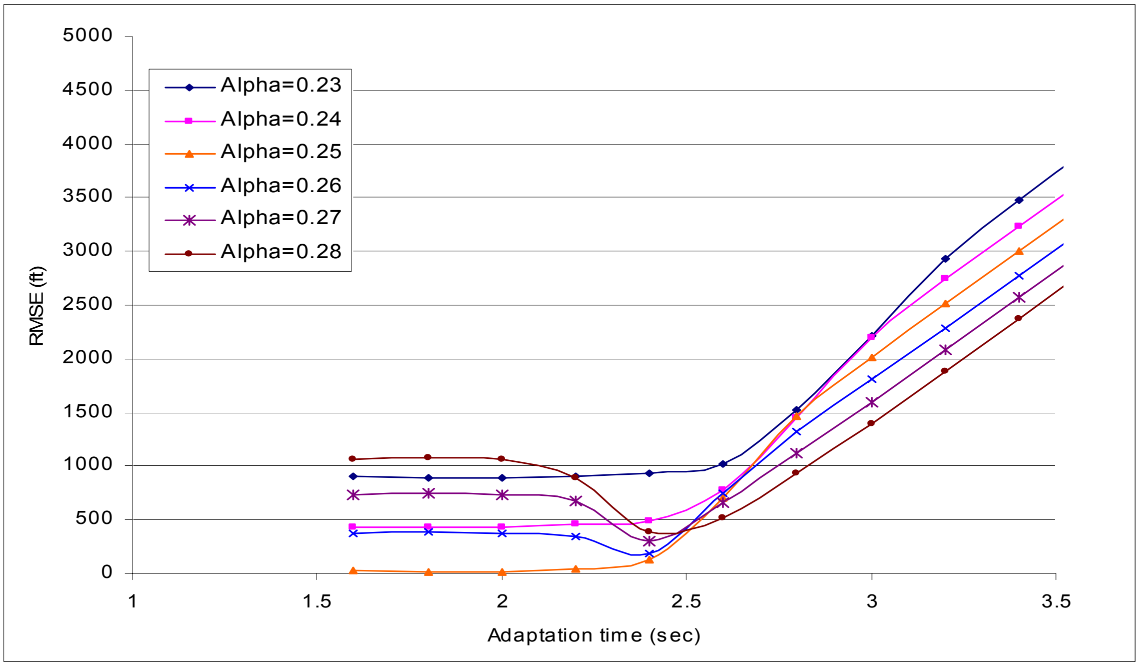


(b) $q^p = 1000\ vph$

Figure 5-7: RMSE for different values of α^m, ΔT^m and flow rates - Scenario 3 (N=100) ($q^p = 500, 1000 vph$)

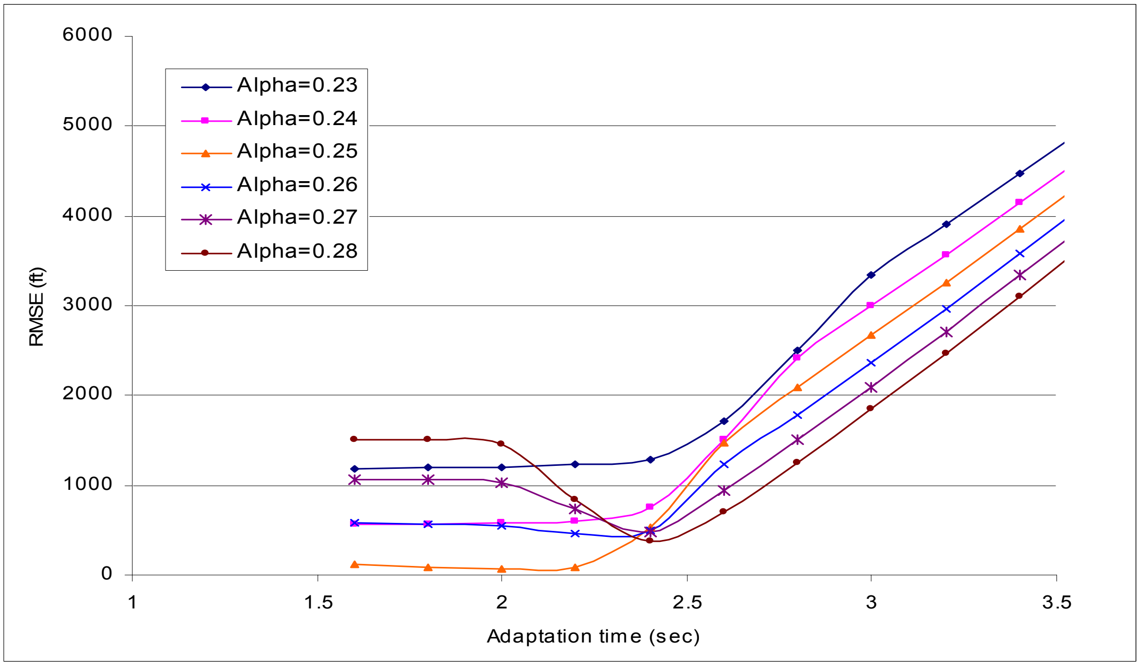


(c) $q^p = 1500\ vph$

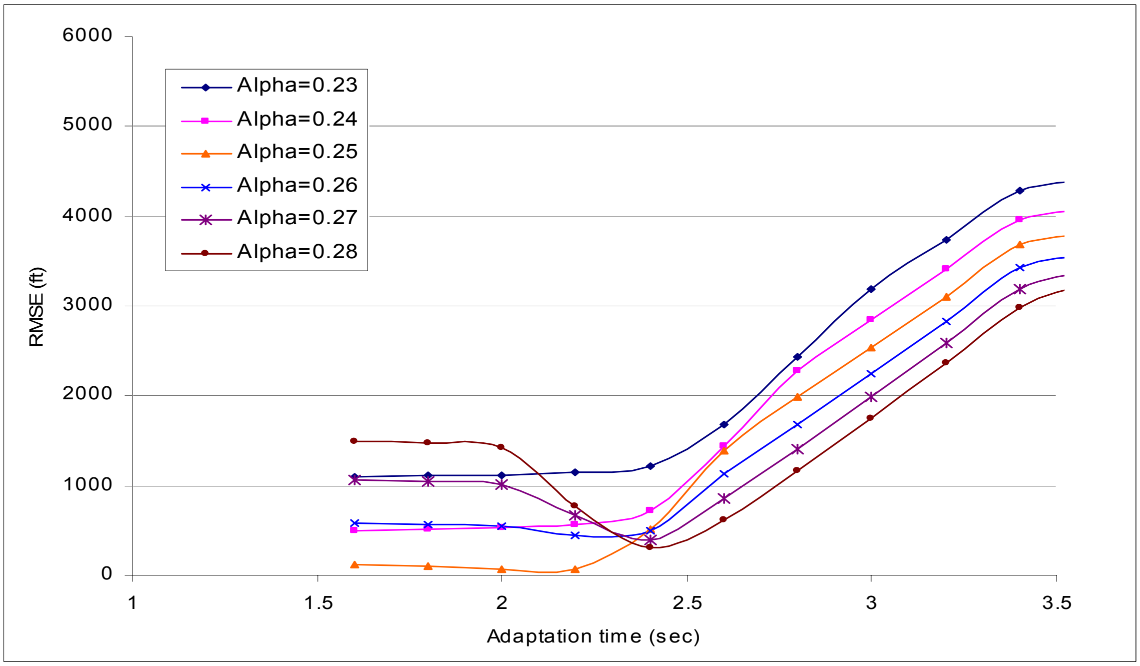


(d) $q^p = 2000\ vph$

Figure 5-7 (Continued): ($q^p = 1500, 2000 vph$)

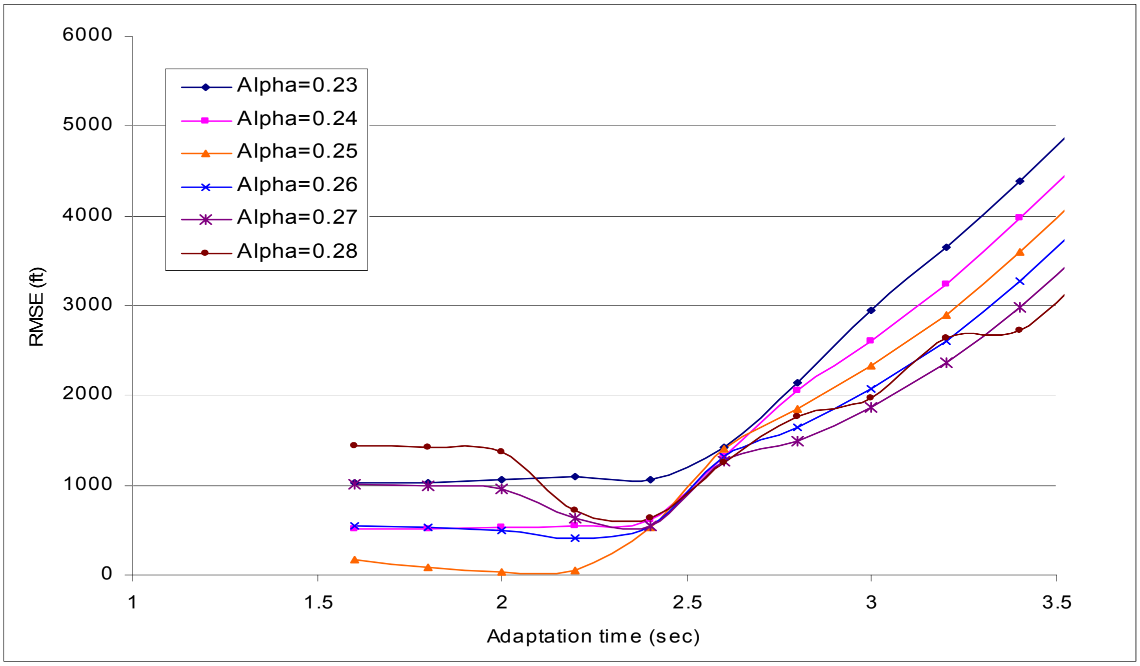


(a) $q^p = 500\ vph$

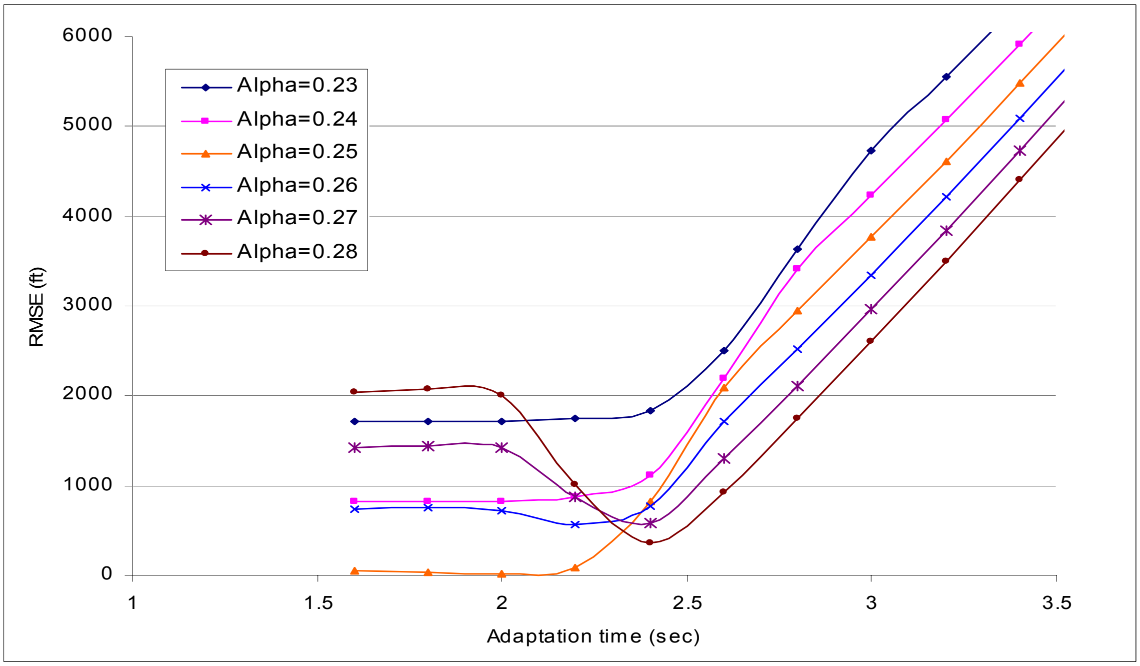


(b) $q^p = 1000\ vph$

Figure 5-8: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - Scenario 3 (N=200) ($q^p = 500, 1000 vph$)

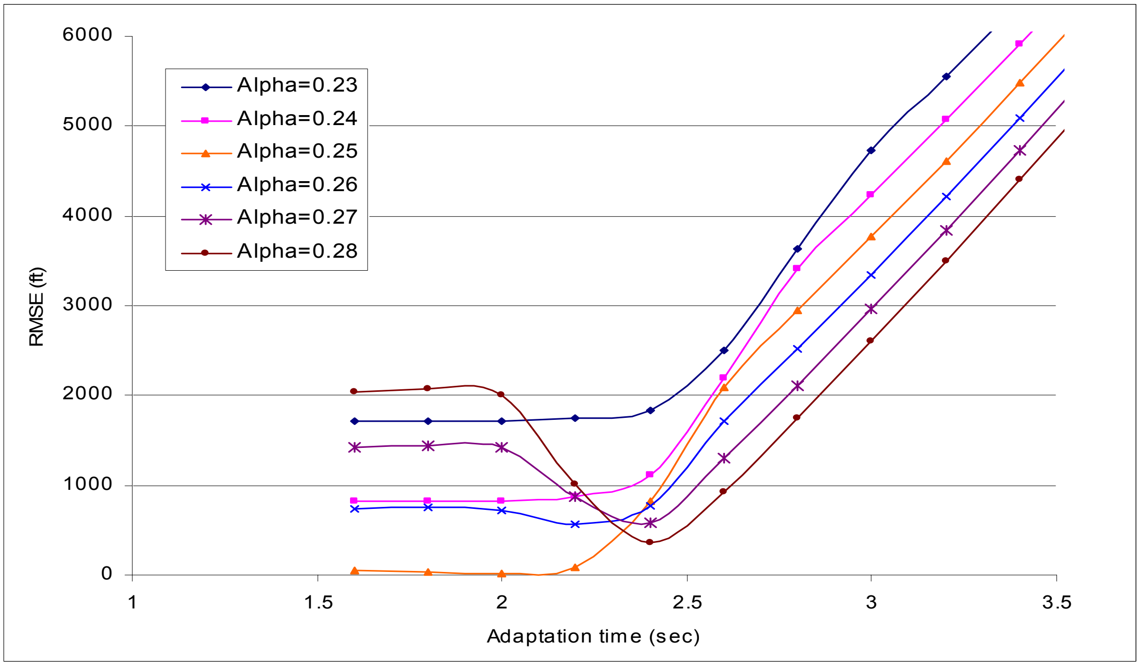


(c) $q^p = 1500\ vph$

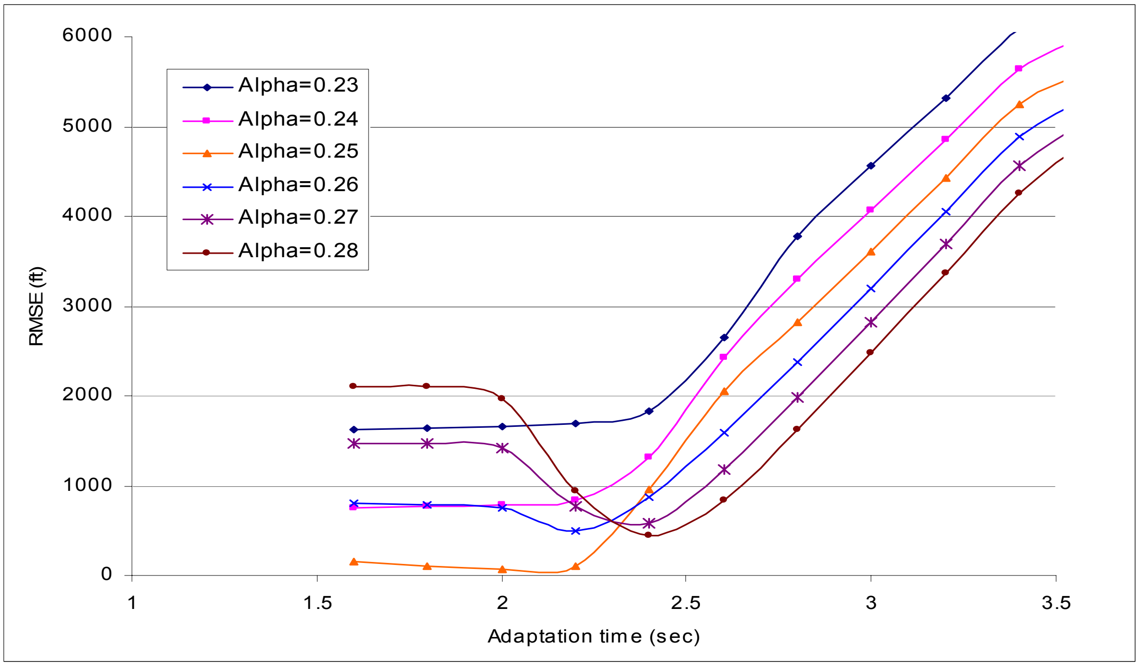


(d) $q^p = 2000\ vph$

Figure 5-8 (Continued): ($q^p = 1500, 2000 vph$)

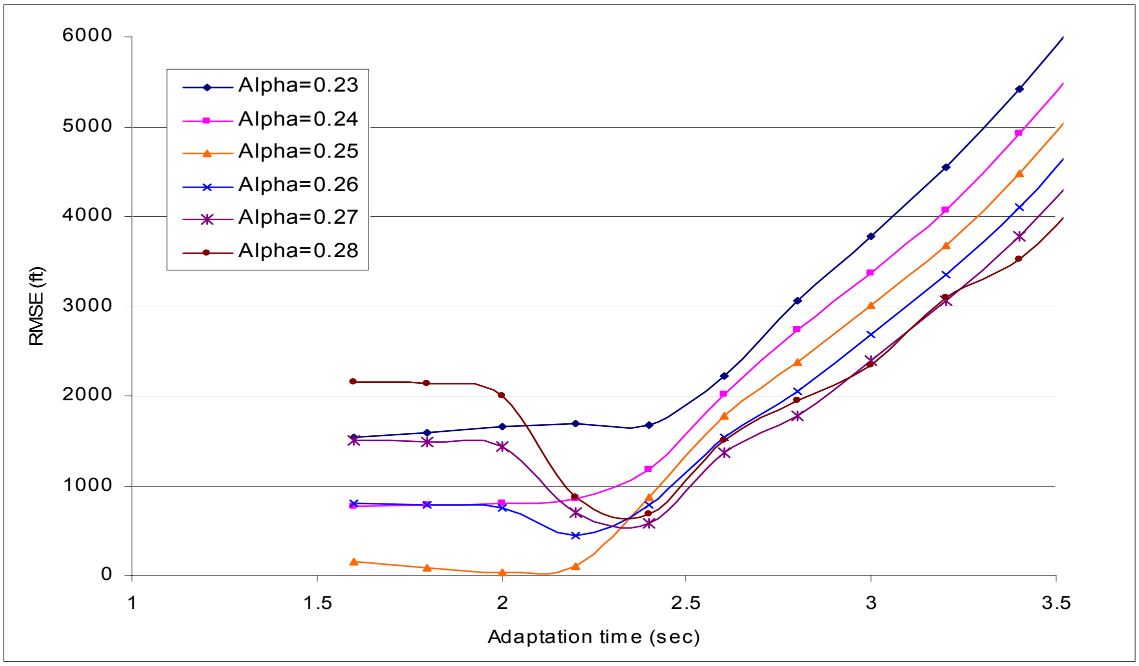


(a) $q^p = 500\ vph$

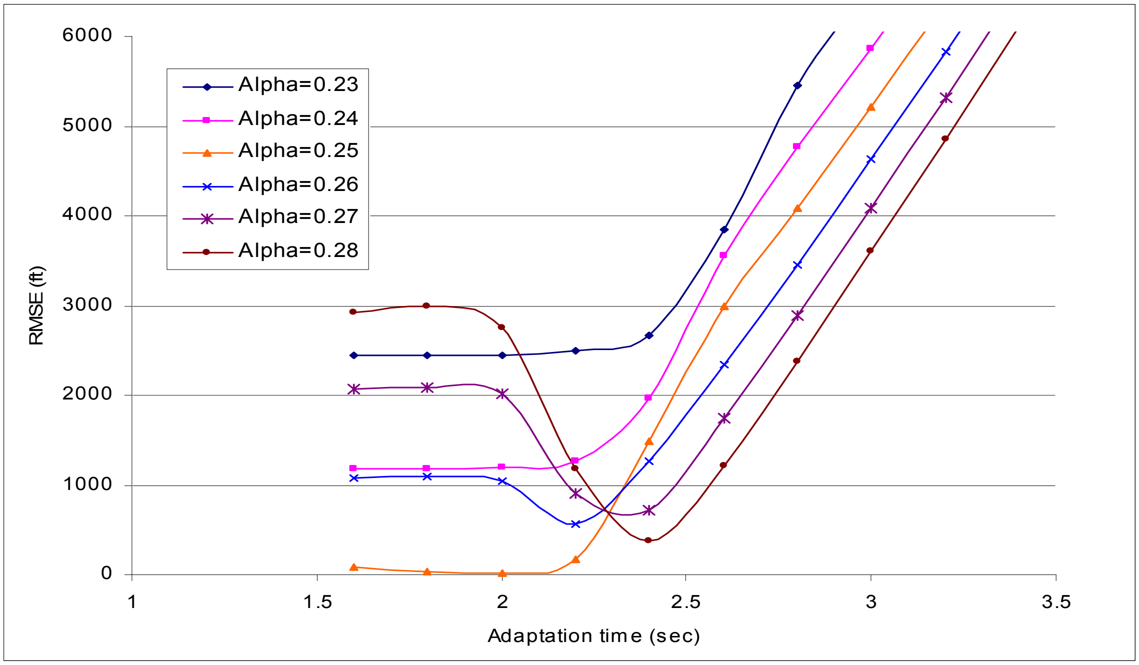


(b) $q^p = 1000\ vph$

Figure 5-9: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - Scenario 3 (N=300) ($q^p = 500, 1000 vph$)



(c) $q^p = 1500\ vph$



(d) $q^p = 2000\ vph$

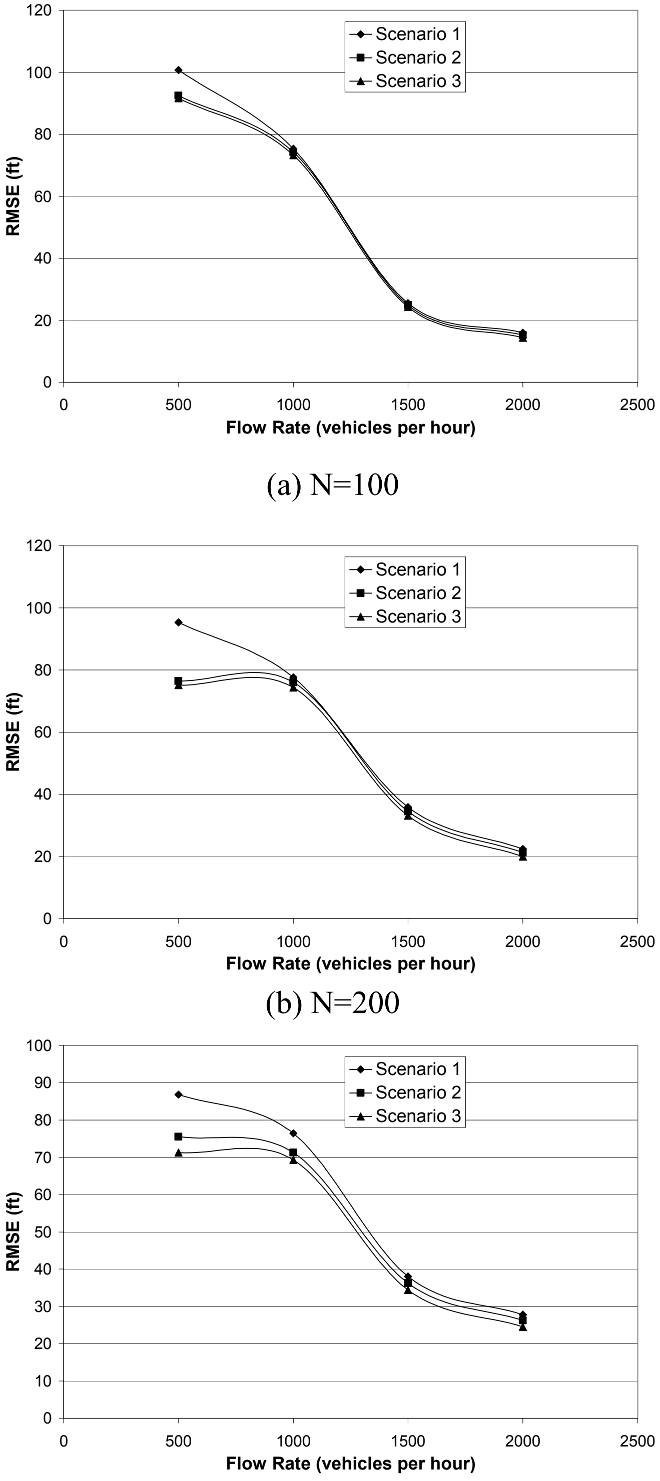
Figure 5-9 (Continued): ($q^p = 1500, 2000 vph$)

(high) flow rates with the increase in number of simulated vehicles. A slightly higher variation in RMSE was also observed between the three scenarios at low flow rates.

Interestingly, the information loss tends to decrease from scenario 1 (one-stop trajectory) to scenario 3 (three-stop trajectory). This suggests that the information loss generally tends to decrease as traffic moves further into forced-flow, as opposed to, free-flow conditions. Therefore, the performance of the downsampling process appears to improve under heavy traffic conditions. This may seem intuitive since forced-flow driving conditions are more likely to contain more information redundancy. Further research is required, however, to verify such observation.

5.2.EXPERIMENTAL RESULTS OF STAGE TWO

The experimental analysis in this stage builds on the procedure applied in stage one of the experiment. The RMSE and average vehicular delay values were obtained for the 36 cases generated by all the possible combinations of different flow rates, number of vehicles in the prototype environment (100,200 and 300) and downsampling ratios. Downsamping ratios of $r = \frac{1}{3}, \frac{1}{4}, and \frac{1}{5}$ (i.e., 33%, 25% and 20%) were used in this stage. The operating conditions in the prototype environment were represented by scenario 3 to simulate under mixed traffic flow conditions. For each case, a range of sensitivity (α^m) and adaptation time (ΔT^m) values were tested to trap the optimal values of α^m and ΔT^m in the microcosm environment in terms of minimum RMSE. Figure 5-11 shows the information loss for different α^m and ΔT^m for a downsampling ratio of *r* = 1/3 and 100 vehicles in the prototype environment. Each individual figure in Figure 5-11 represents a separate case and corresponds to a different flow rate. A range of 0.13 to 0.27 sec^{-1} for α^m and 2 to 5.5 seconds for ΔT^m was tested to generate the curves for



(a) N=100

(b) N=200

(c) N=300

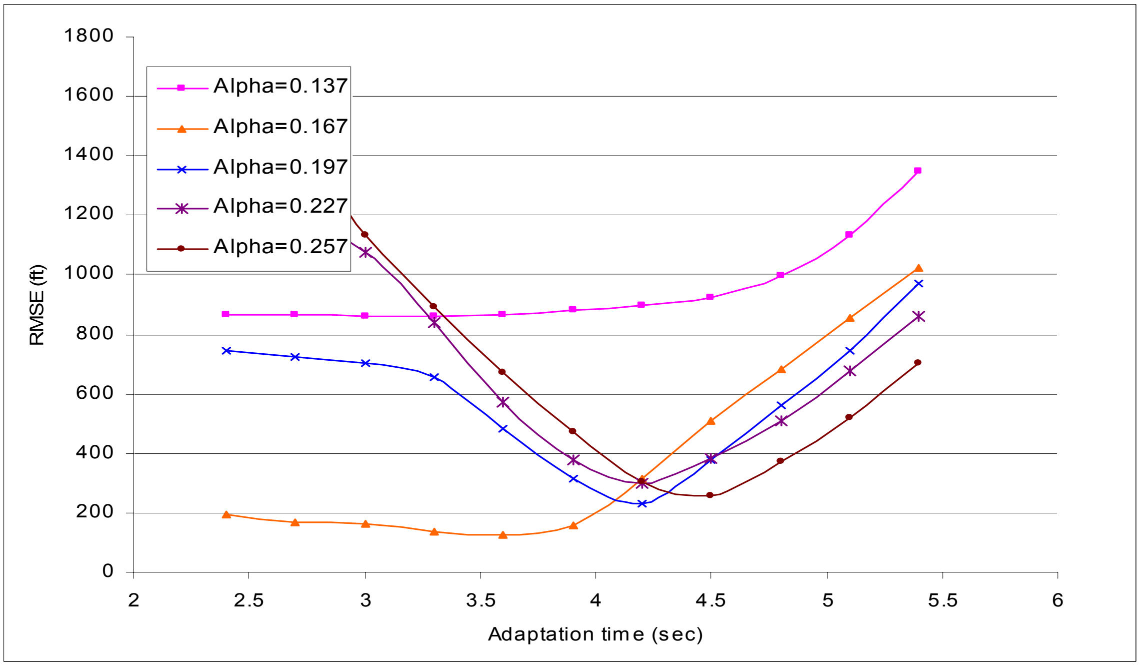
Figure 5-10: Effect of flow rate and number of simulated vehicles on near-optimal downsampling performance.

each flow rate. A wider range of α^m and ΔT^m were used initially (α^m= 0.10 to 0.50 sec^{-1} with an interval of 0.05 and ΔT^m=1.0 to 6.0 seconds with an interval of 1.0 seconds), but the range was narrowed down to use smaller intervals to locate the exact minimal RMSE values. Similarly, 8 more cases were studied with different number of vehicles (*N*=200 and 300) keeping the downsampling ratio, *r* = 1/3, same. Figure 5-12 and Figure 5-13 show the RMSE values for N=200 and 300, respectively. In all the figures, the minimum RMSE values corresponded to $\alpha_o^m = 0.167$ and $\Delta T_o^m \approx 3.0$.

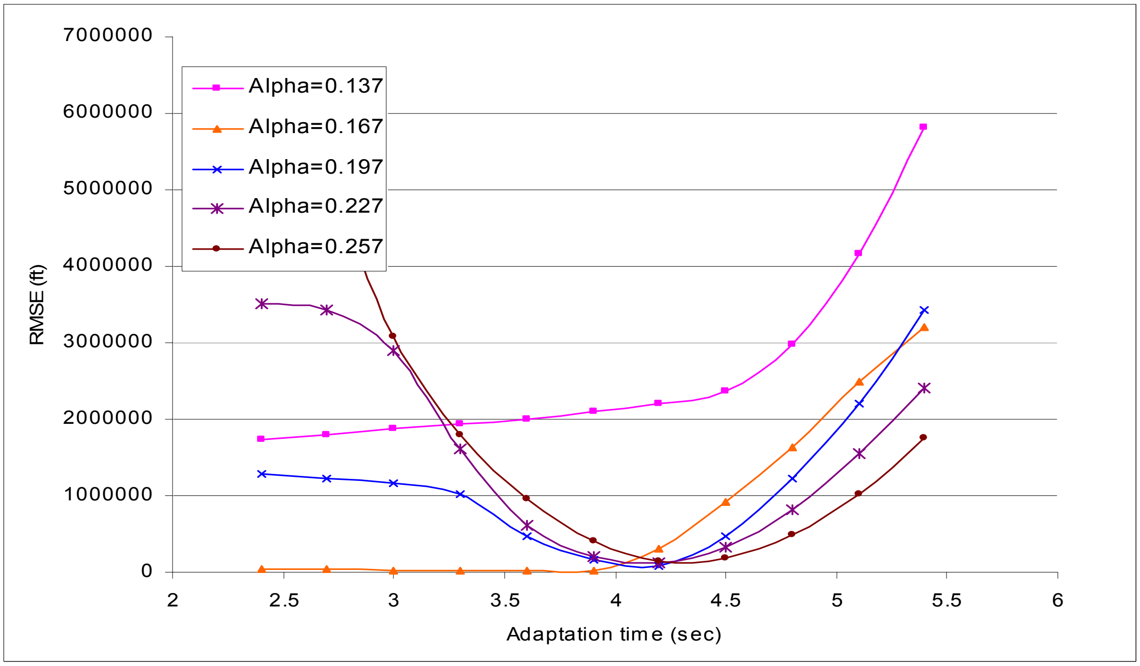
Slight deviation from the 3.0-second near-optimal adaptation value was observed for cases where the flow rate was less than 2000 vph. However, the additional information loss (RMSE) due to this deviation is marginal, and therefore, the exact optimal adaptation time can be substituted with the near-optimal value of 3.0 seconds.

The experimental work was performed in a similar manner for the remaining cases (cases 13 to 36) that correspond to downsampling ratios of *r* = 1/4 and *r* = 1/5, with each ratio generating 12 cases. The RMSE curves were generated for different flow rates and different number of vehicles in the prototype as shown in Figure 5-14 through Figure 5-16 for a downsampling ratio of *r* = 1/4. Figure 5-17 through Figure 5-19 correspond to a downsampling ratio of *r* = 1/5.

A range of 0.10 to 0.25 sec^{-1} for α^m and 3 to 8.5 seconds for ΔT^m was tested to generate the curves for each flow rate for the downsampling ratio of *r* = 1/4 and *r* = 1/5. A wider range of α^m and ΔT^m were used initially (α^m= 0.10 to 0.50 sec^{-1} with an interval of 0.05 and ΔT^m=3.0 to 10.0 seconds with an interval of 1.0 seconds), but the range was narrowed down to use smaller intervals to locate the exact minimal RMSE



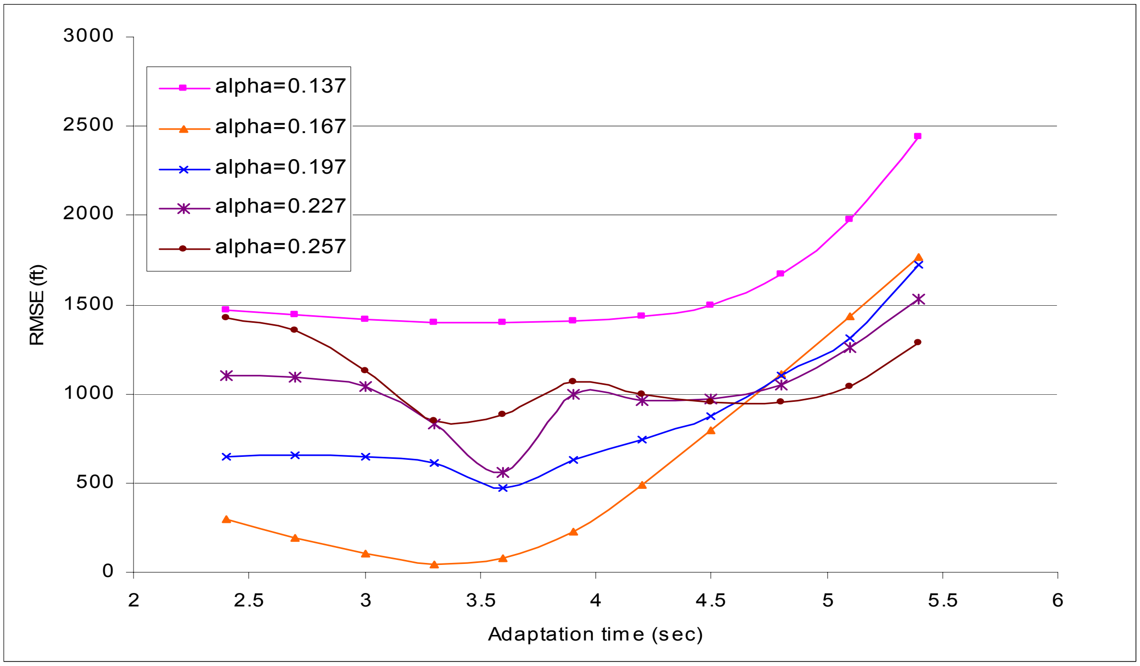
(a) $q^p = 500\ vph$



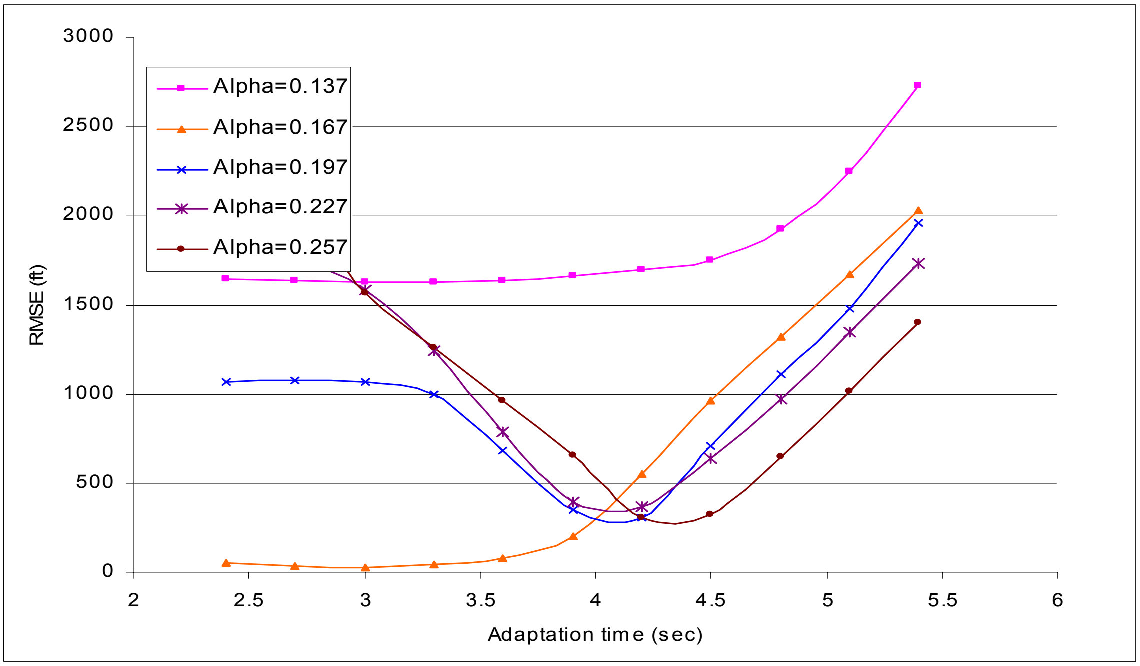
(b) $q^p = 1000\ vph$

Figure 5-11: RMSE for different values of α^m, ΔT^m and flow rates - (r = 1/3, N=100)

($q^p = 500{,}1000 vph$)

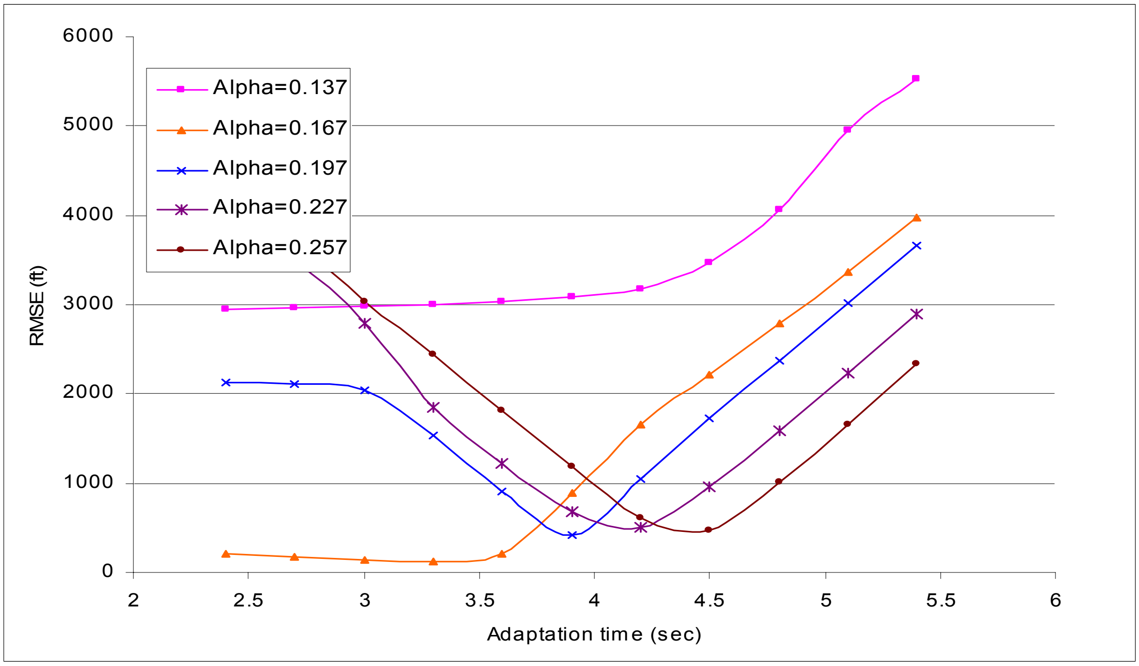


(c) $q^p = 1500\ vph$

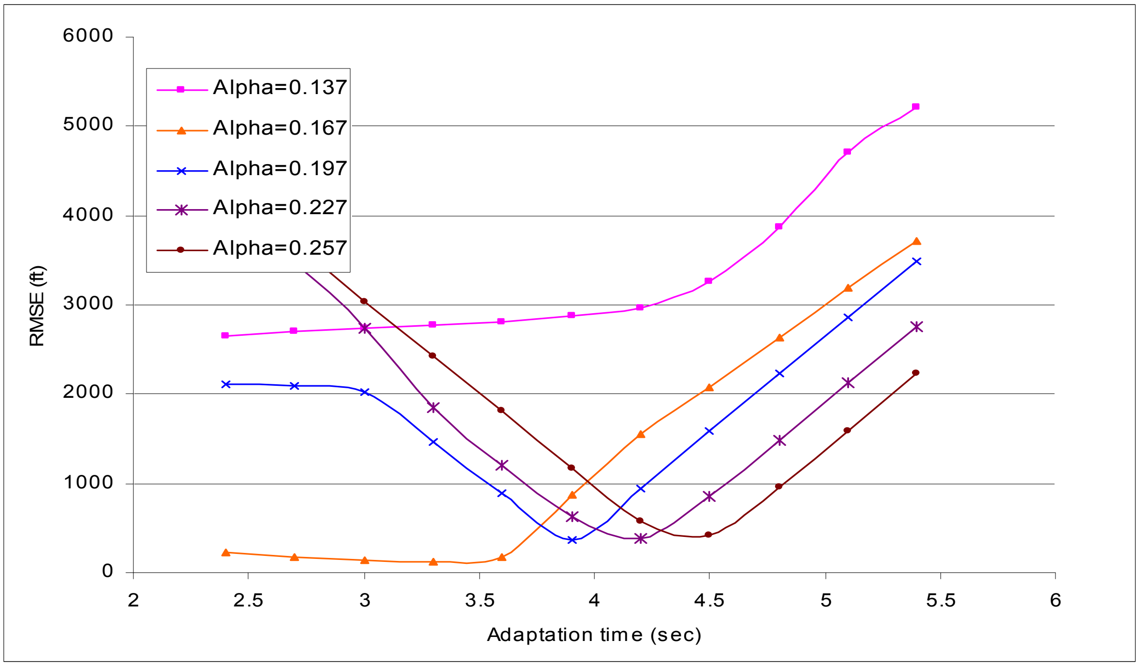


(d) $q^p = 2000\ vph$

Figure 5-11 (Continued): ($q^p = 1500, 2000 vph$ **)**

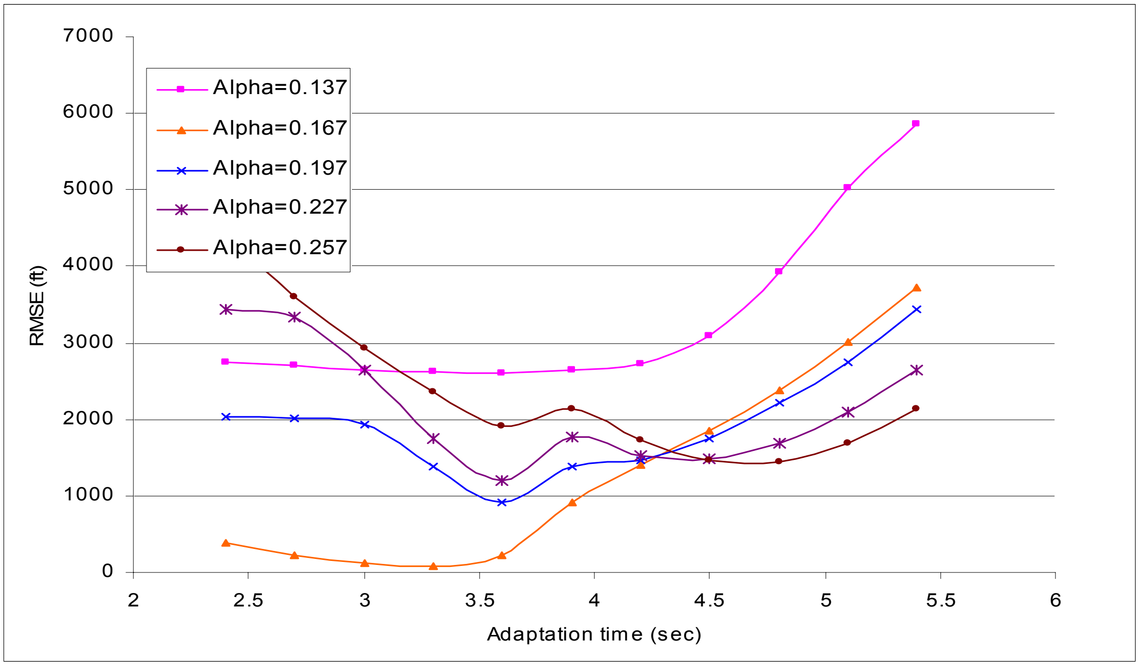


(a) $q^p = 500\ vph$

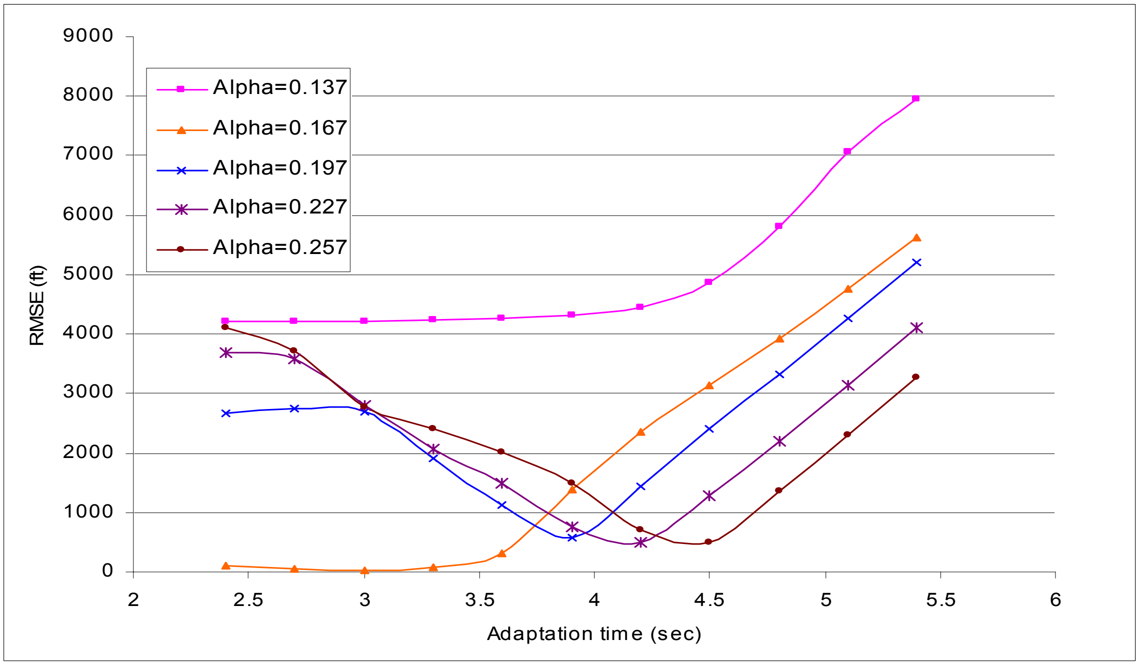


(b) $q^p = 1000\ vph$

Figure 5-12: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - (r = 1/3, N=200) ($q^p = 500, 1000vph$).

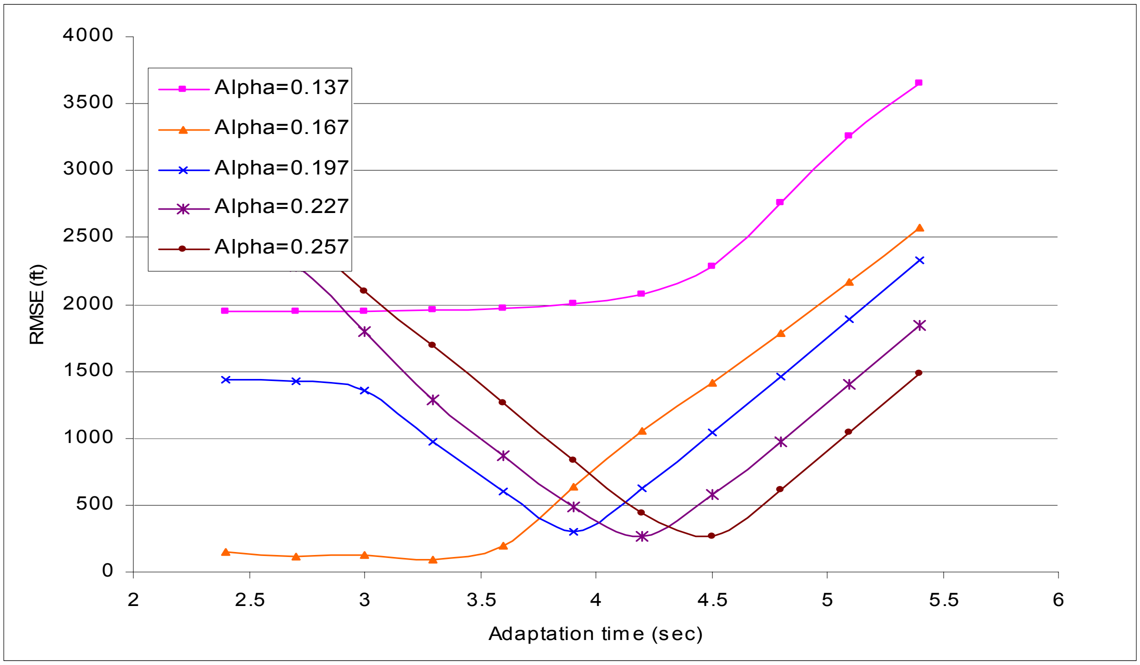


(c) $q^p = 1500\ vph$

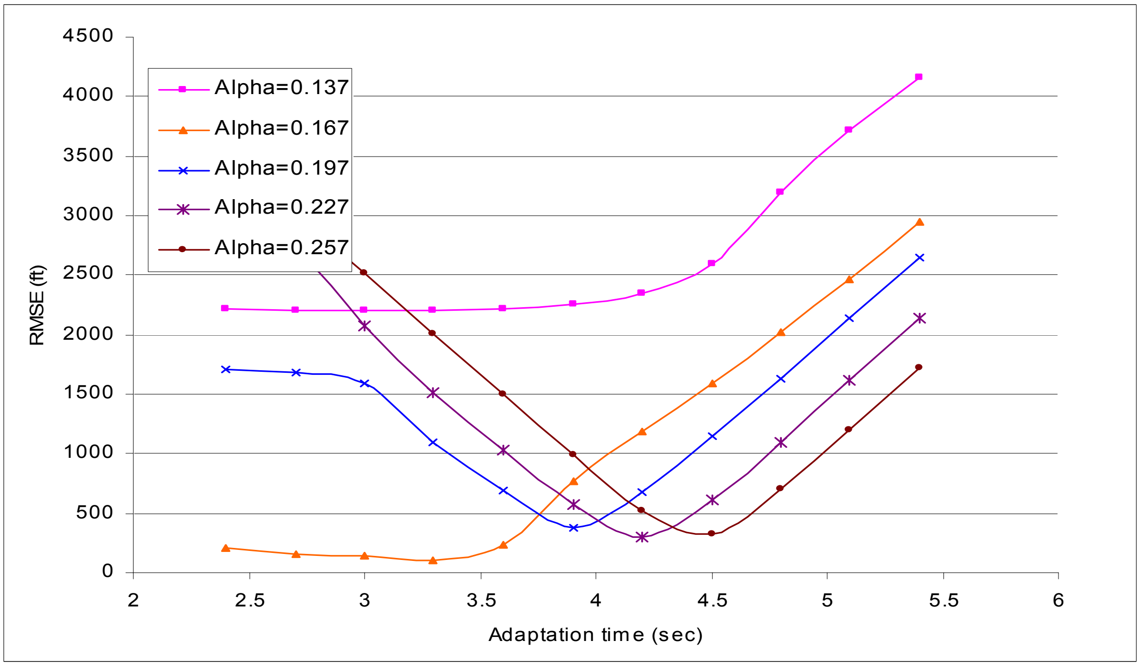


(d) $q^p = 2000\ vph$

Figure 5-12 (Continued): ($q^p = 1500, 2000 vph$)

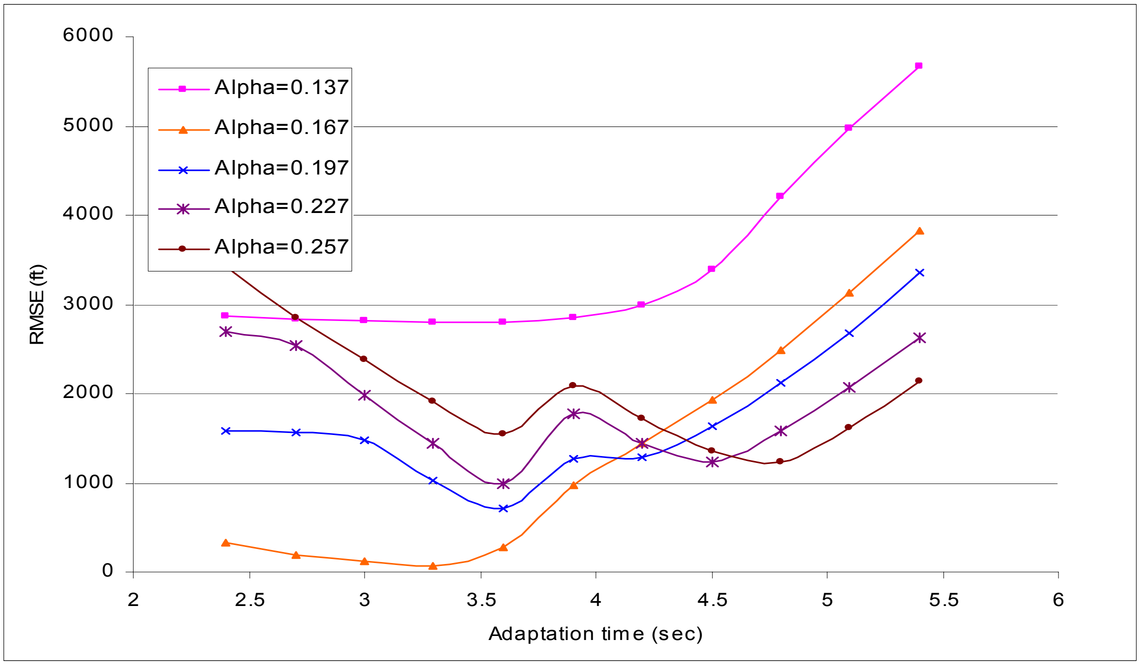


(a) $q^p = 500\ vph$

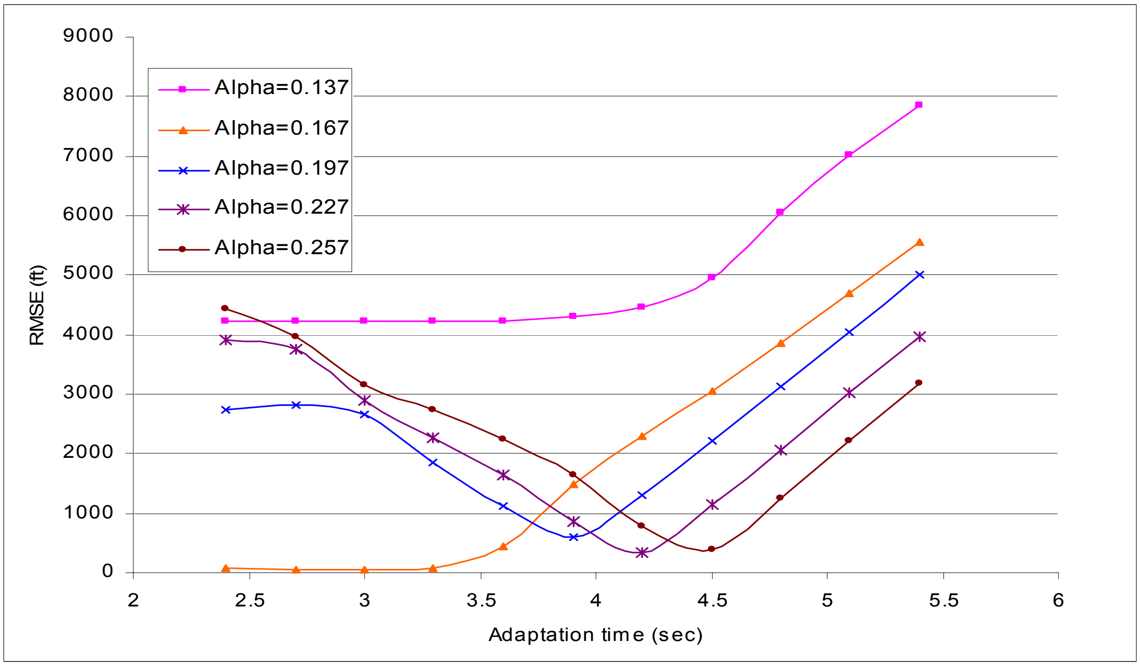


(b) $q^p = 1000\ vph$

Figure 5-13: RMSE for different values of α^m, ΔT^m and flow rates - (r = 1/3, N=300) ($q^p = 500,1000 vph$).



(c) $q^p = 1500\ vph$



(d) $q^p = 2000\ vph$

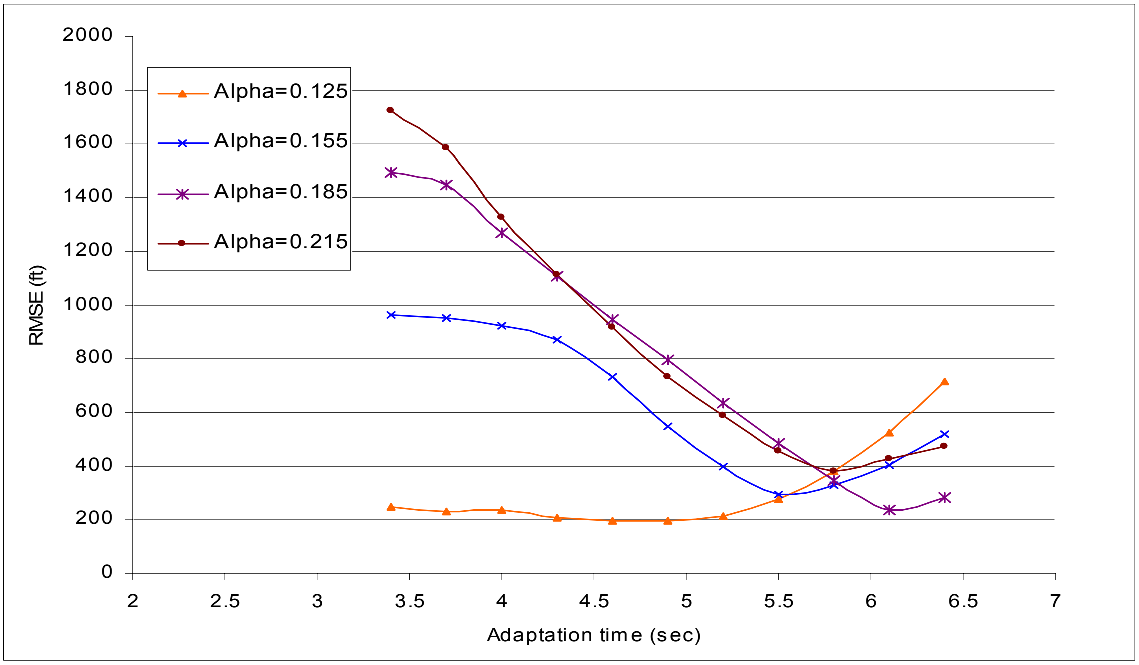
Figure 5.13 (Continued): ($q^p = 1500, 2000 vph$**)**

values. For *r*=1/4, the minimum information loss was found where $\alpha_o{}^m = 0.125$ and $\Delta T_o{}^m \approx 4.0$. For *r*=1/5, $\alpha_o{}^m = 0.1$ and $\Delta T_o{}^m \approx 5.0$.

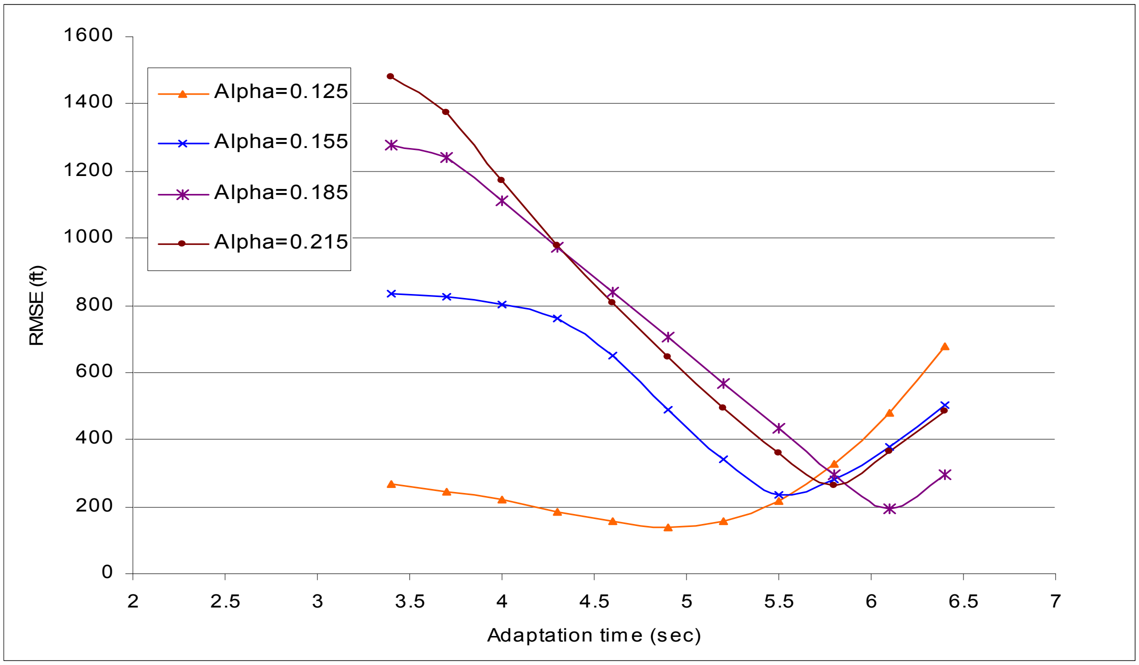
For all cases considered, the value of α_0^m in the microcosm environment was proportionately scaled down by the downsampling ratio *r* to the prototype sensitivity value α^p. This confirms that the optimal sensitivity ratio ${}^{\alpha_0^m}\!/\!_{\alpha^p}$ is equal to the downsampling ratio *r*. Similarly, the values of ΔT_0^m were proportionately scaled up by the inverse of the downsampling ratio *r* to the prototype value of ΔT^p (Ishak, et al. 2003). In some cases, however, slight deviations from this trend were observed in the order of nearly 0.1 seconds, which could be attributed to the rounding effect of the simulation updating period (0.1 seconds). This observation confirms that the optimal adaptation ratio ${}^{\Delta T_0^m}\!/\!_{\Delta T^p}$ approaches the reciprocal of the downsampling ratio (1/*r*). This consistency in the optimal solutions observed for each downsampling ratio is critically important to ensure that the equivalent driving behavior in the microcosm environment is independent of the flow rate and driving conditions.

5.3.OPTIMAL AND NEAR-OPTIMAL ADAPTATION RATIOS

Examining the relationship between the downsampling ratio and the optimal behavioral parameters from the experimental work shows consistently that the optimal sensitivity parameter in the microcosm environment, $\alpha_o{}^m$, can be derived by scaling the prototype value α^p with the downsampling ratio *r*; i.e. $\alpha_o{}^m = r\alpha^p$, and that the optimal microcosm adaptation time, ΔT_0^m, can be derived by scaling the prototype value ΔT^p with the inverse of *r*; i.e. $\Delta T_0^m \approx {}^1\!/\!_r \Delta T^p$.

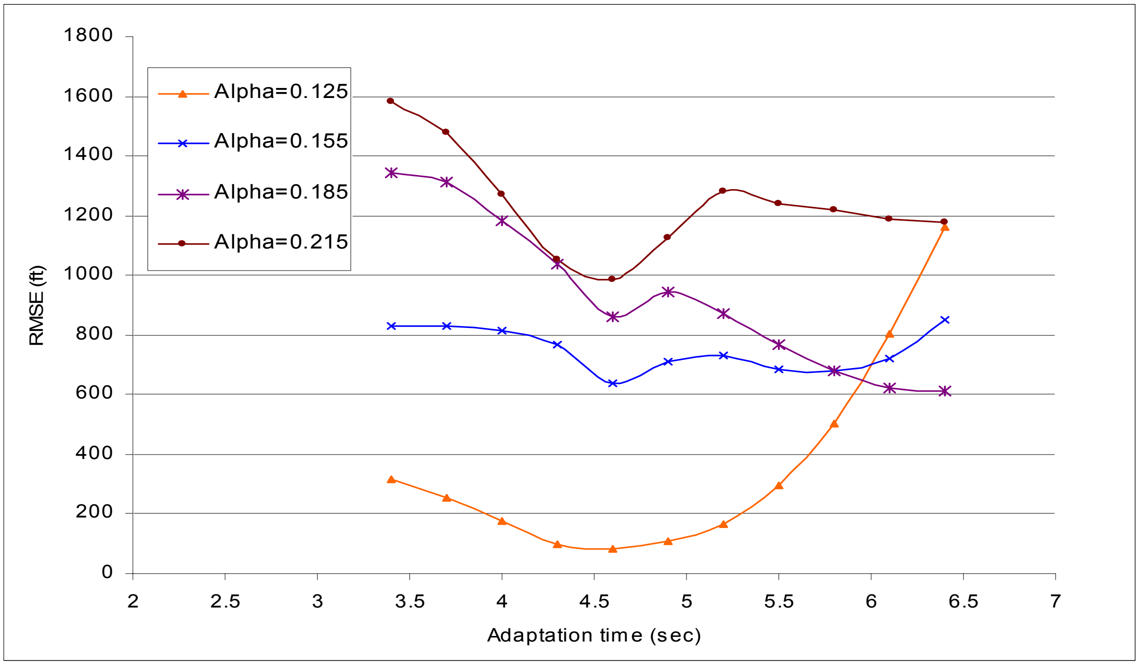


(a) $q^p = 500\ vph$

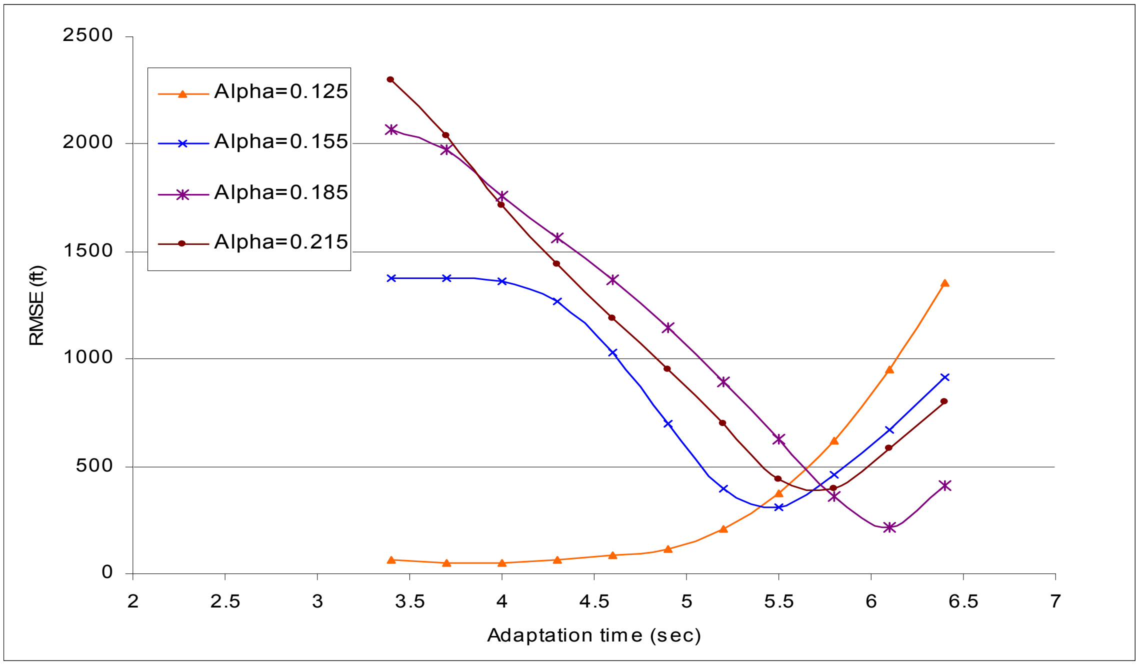


(b) $q^p = 1000\ vph$

Figure 5-14: RMSE for different values of α^m, ΔT^m and flow rates - (r = 1/4, N=100) ($q^p = 500, 1000 vph$).

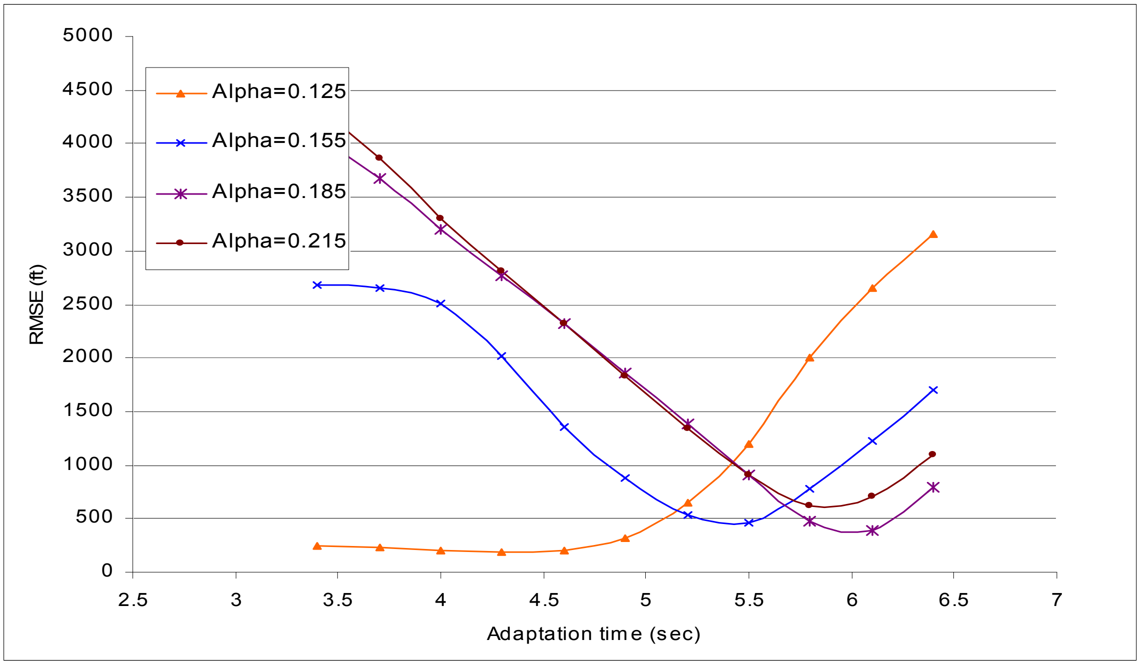


(c) $q^p = 1500\ vph$

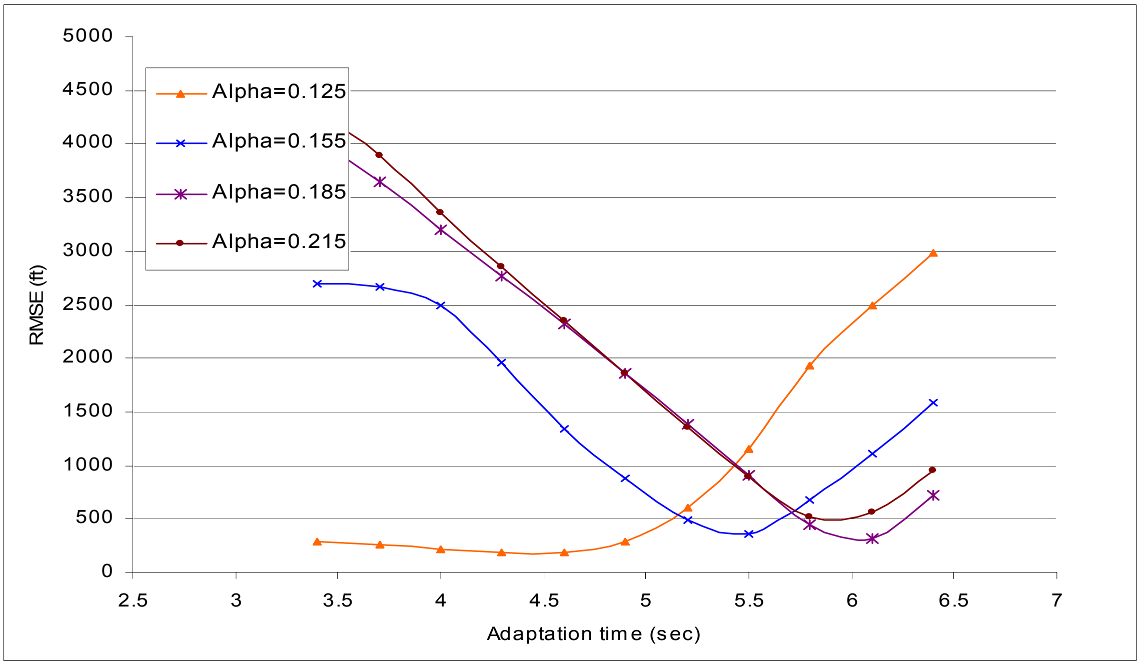


(d) $q^p = 2000\ vph$

Figure 5-14 (Continued): ($q^p = 1500, 2000vph$ **)**

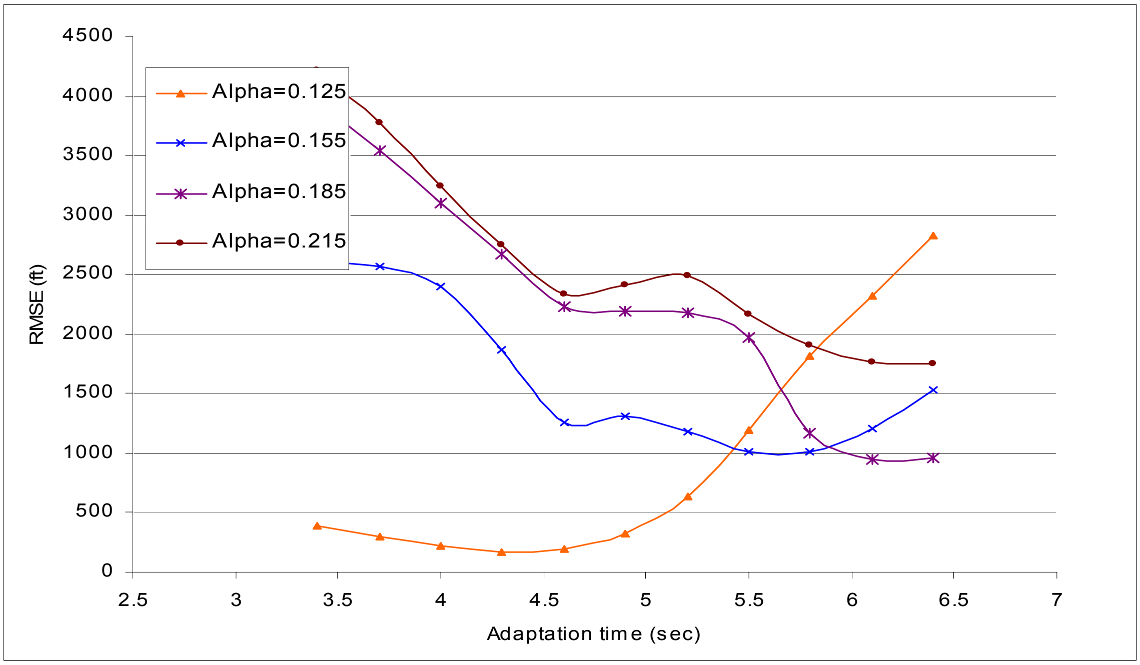


(a) $q^p = 500\ vph$

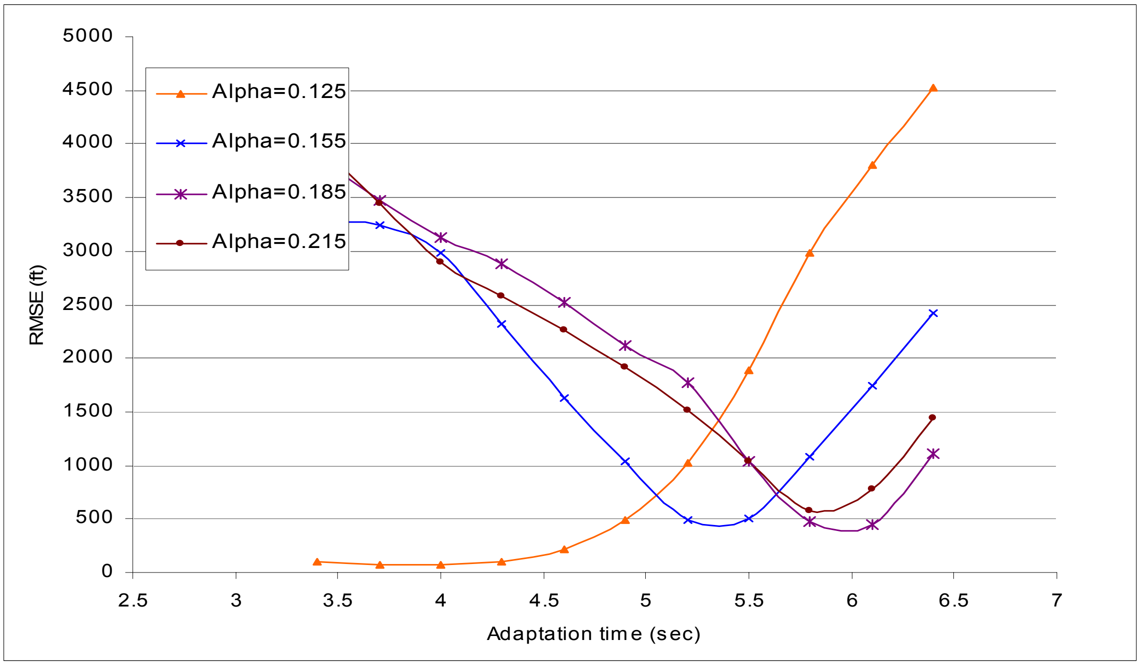


(b) $q^p = 1000\ vph$

Figure 5-15: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - (*r* = 1/4, *N*=200) ($q^p = 500{,}1000 vph$).

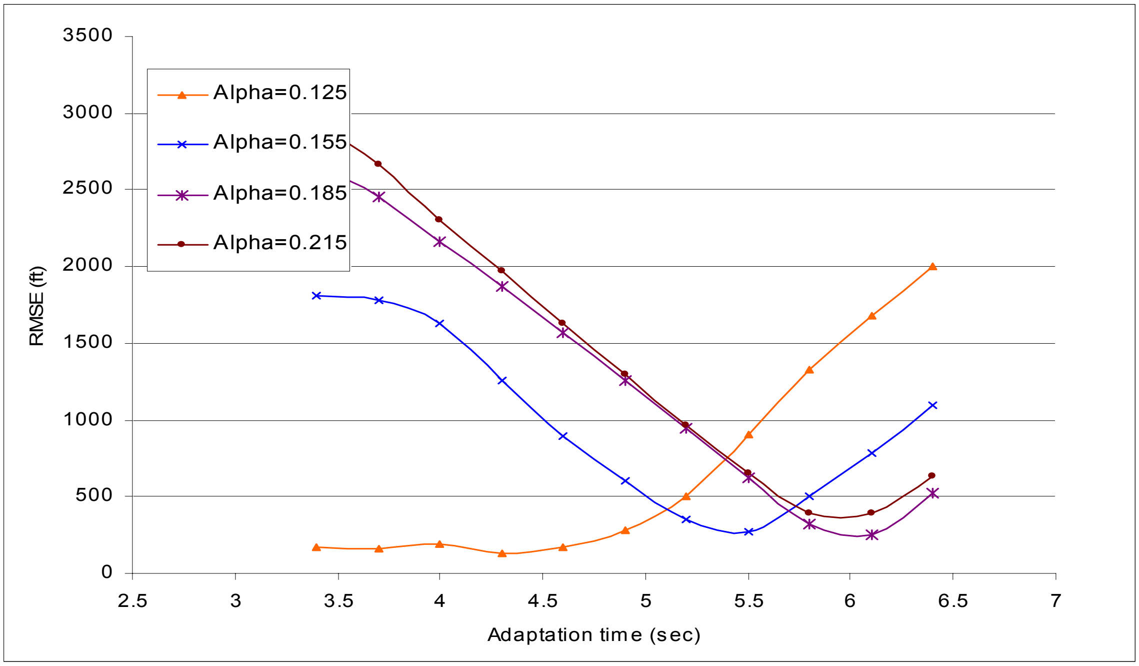


(c) $q^p = 1500\ vph$

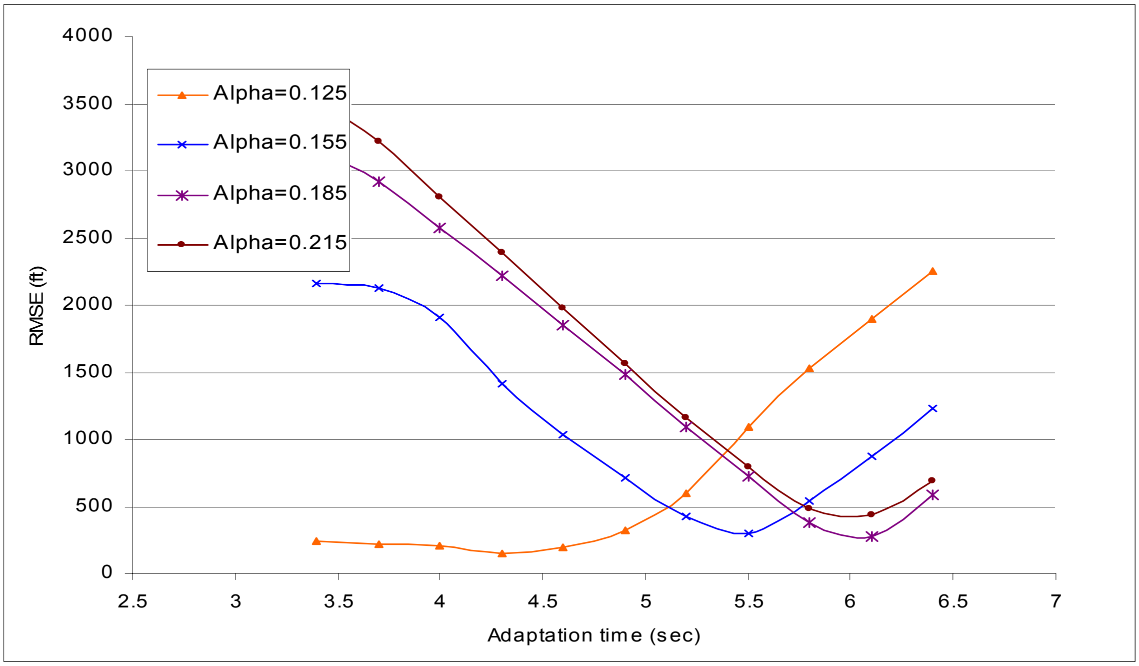


(d) $q^p = 2000\ vph$

Figure 5-15 (Continued): ($q^p = 1500, 2000 vph$)

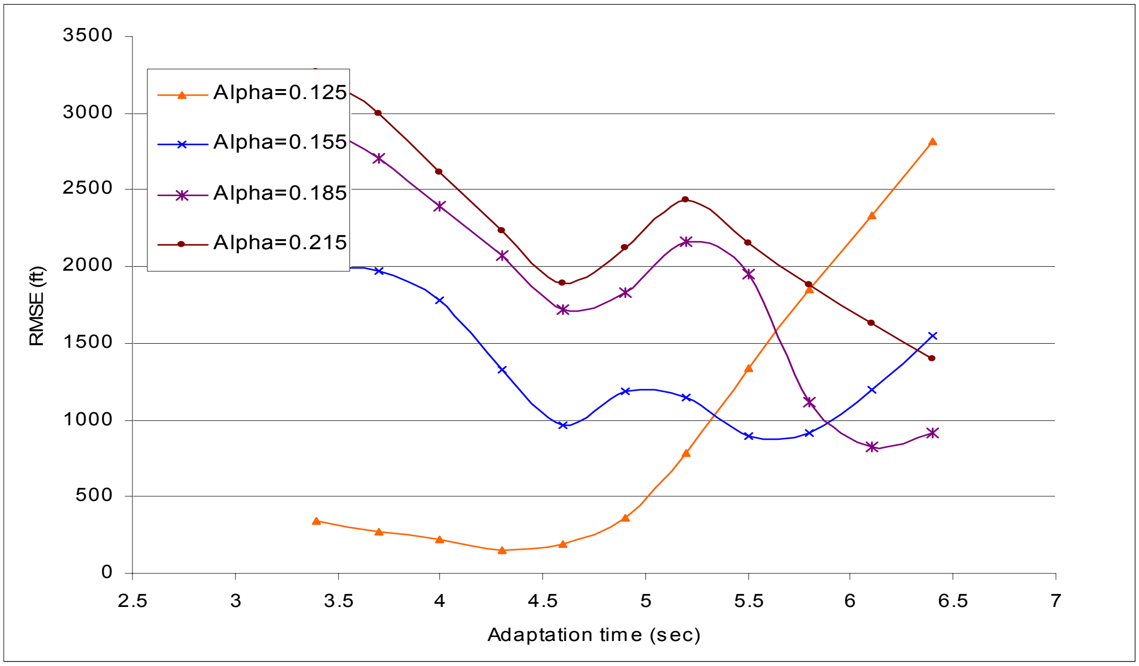


(a) $q^p = 500\ vph$

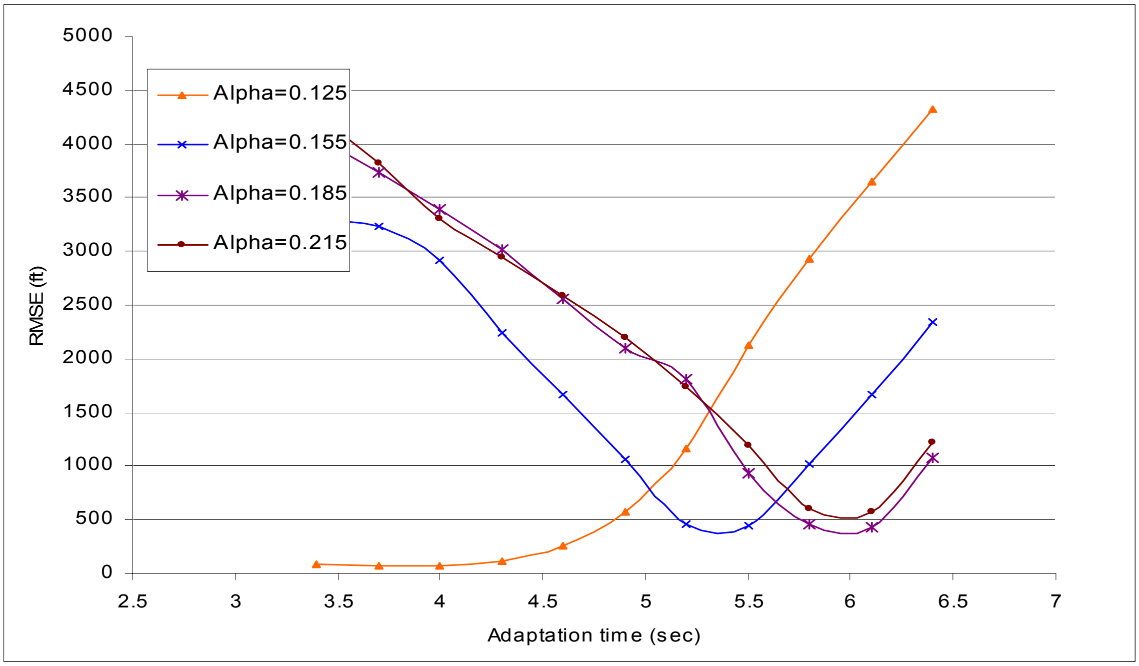


(b) $q^p = 1000\ vph$

Figure 5-16: RMSE for different values of α^m, ΔT^m and flow rates - (*r* = 1/4, *N*=300) ($q^p = 500{,}1000 vph$).

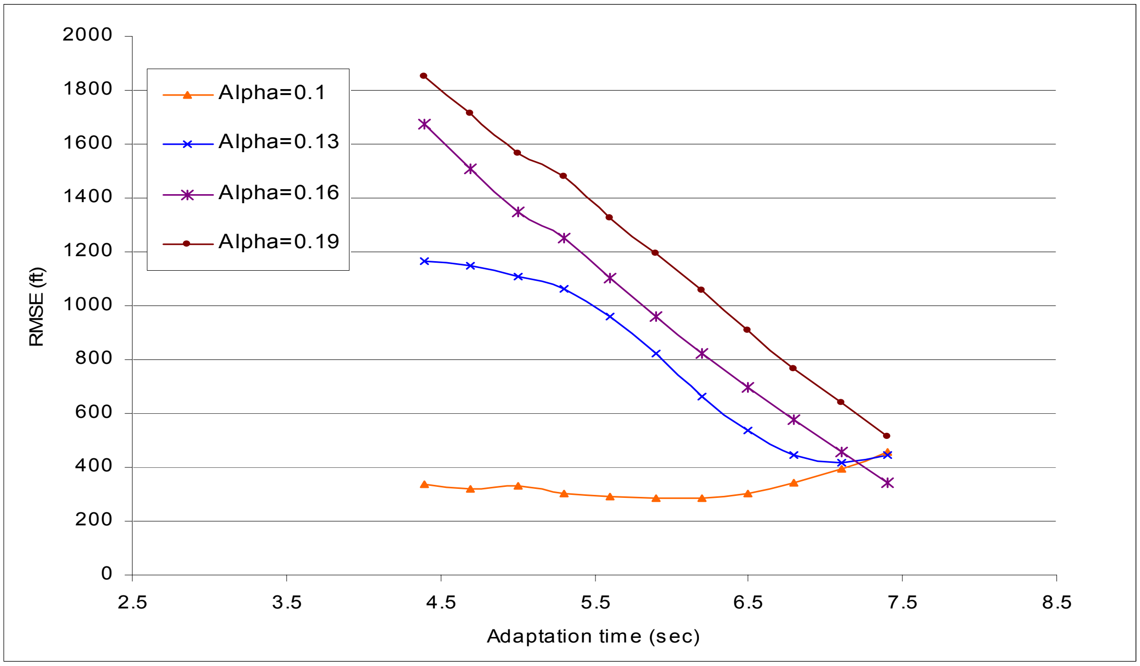


(c) $q^p = 1500\ vph$

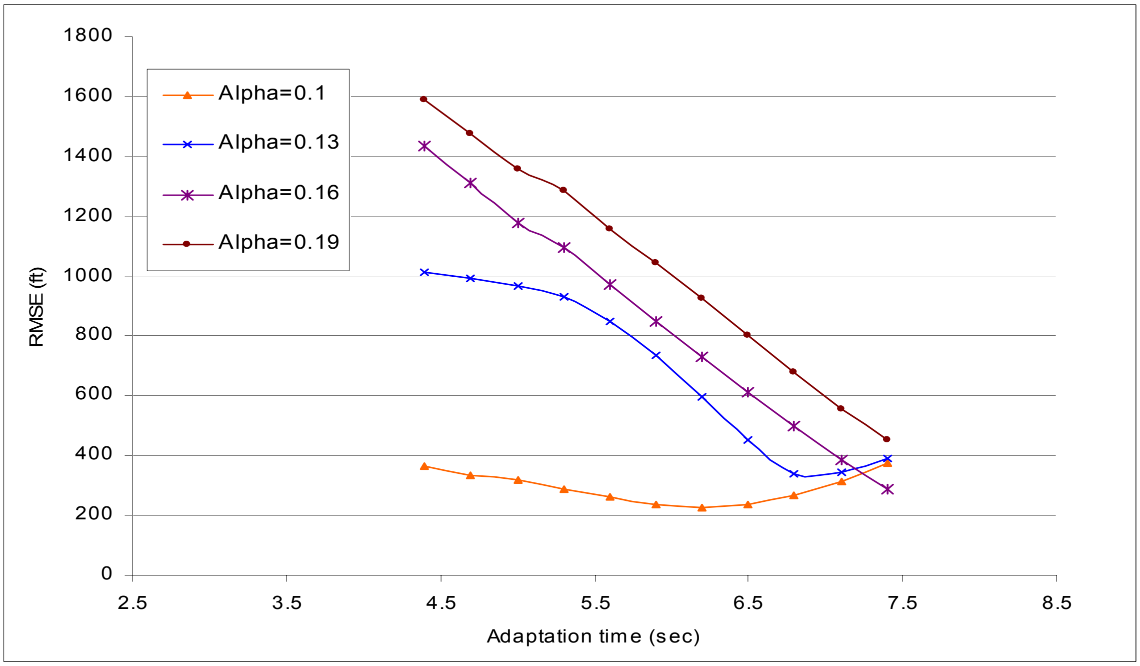


(d) $q^p = 2000\ vph$

Figure 5-16 (Continued): ($q^p = 1500, 2000 vph$ **)**

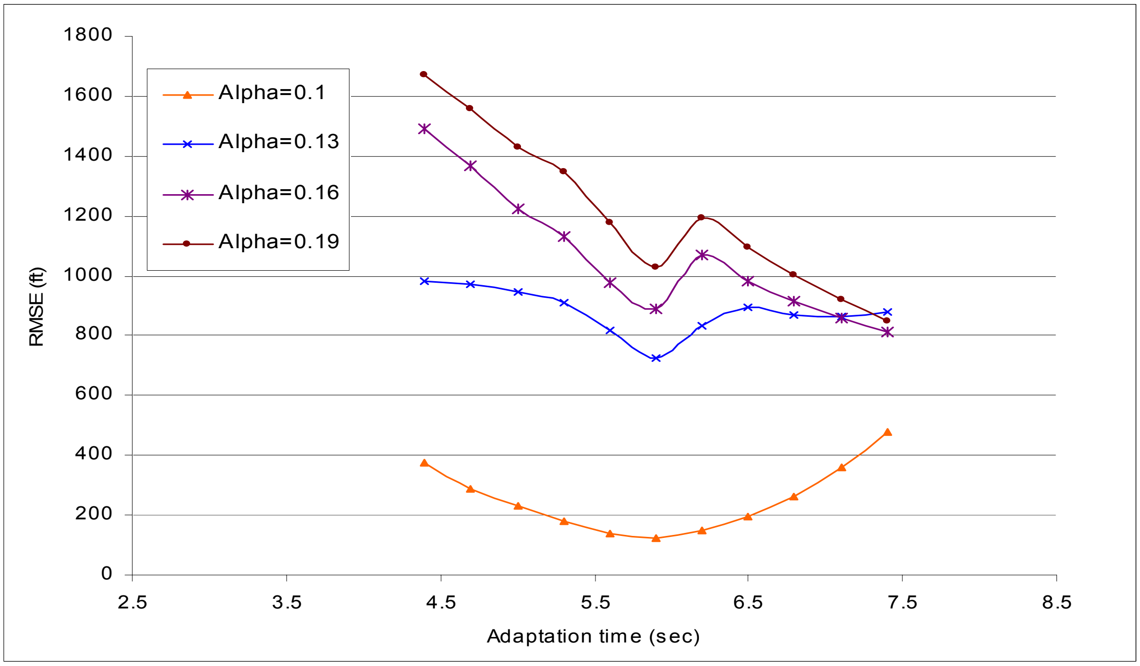


(a) $q^p = 500\ vph$

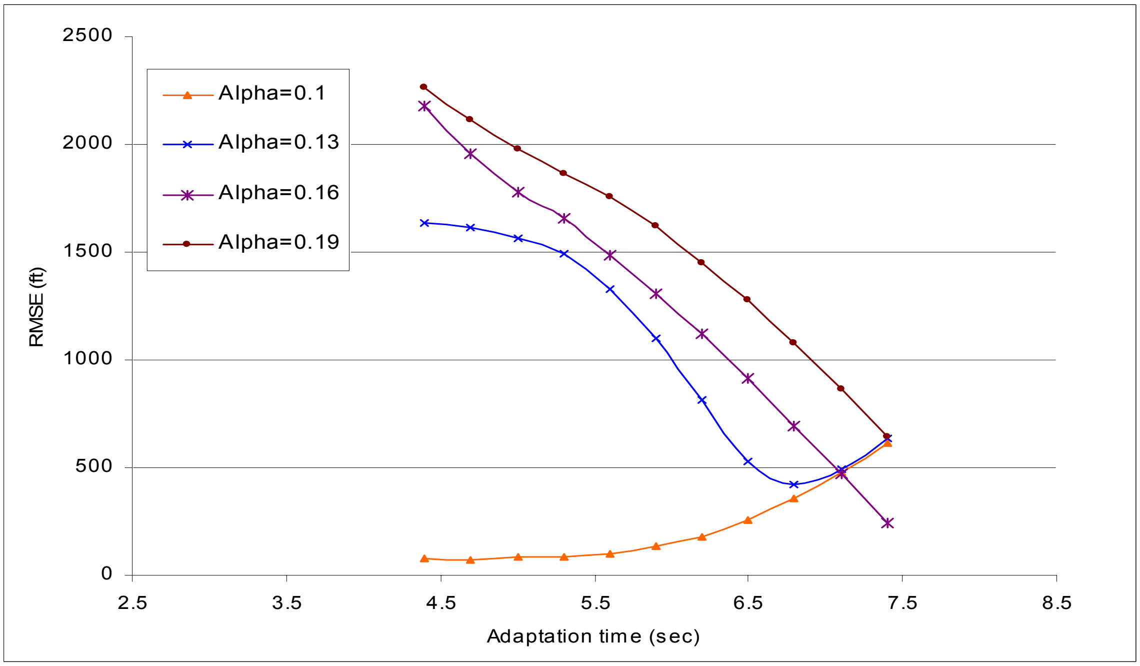


(b) $q^p = 1000\ vph$

Figure 5-17: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - (*r* = 1/5, *N*=100) ($q^p = 500{,}1000 vph$).

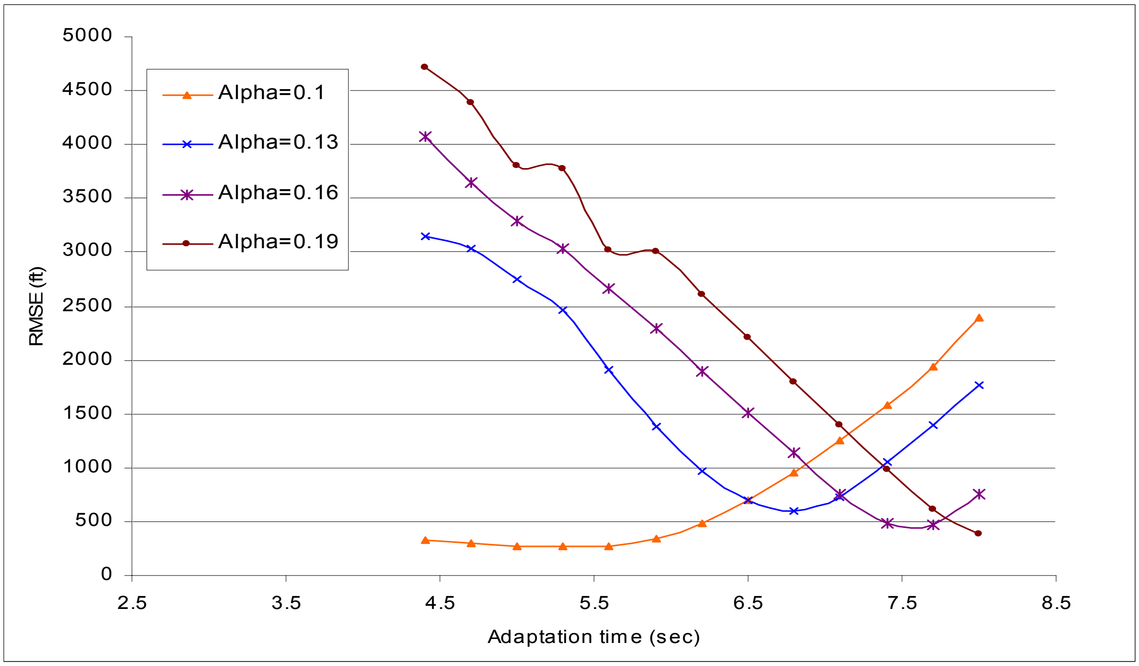


(c) $q^p = 1500\ vph$

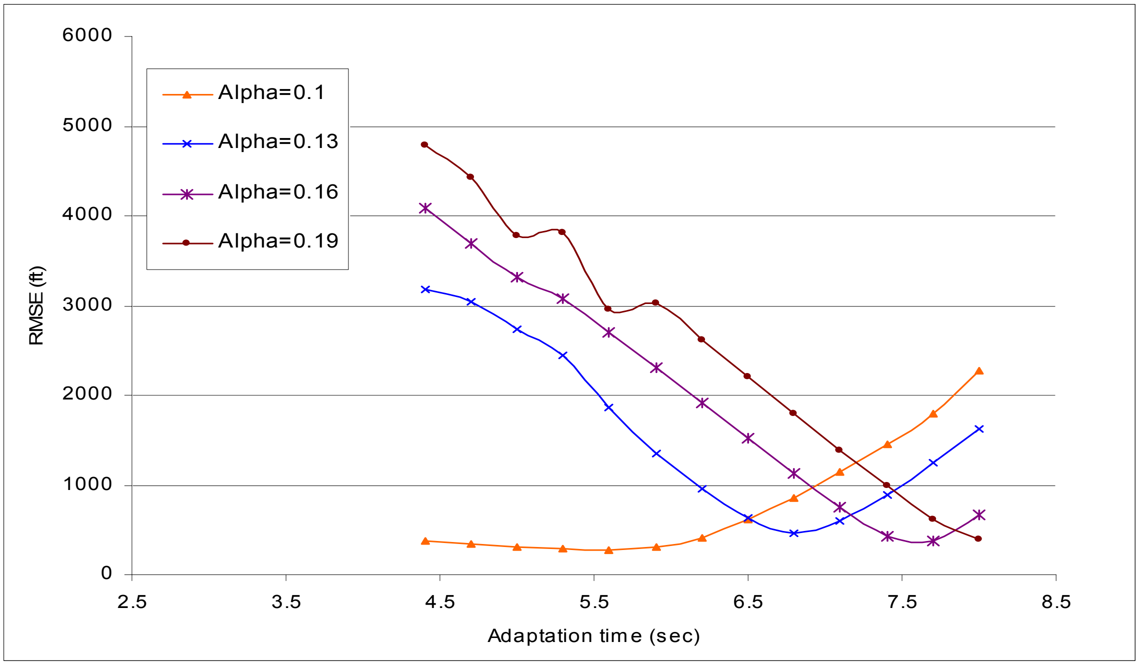


(d) $q^p = 2000\ vph$

Figure 5-17 (Continued): ($q^p = 1500, 2000 vph$)

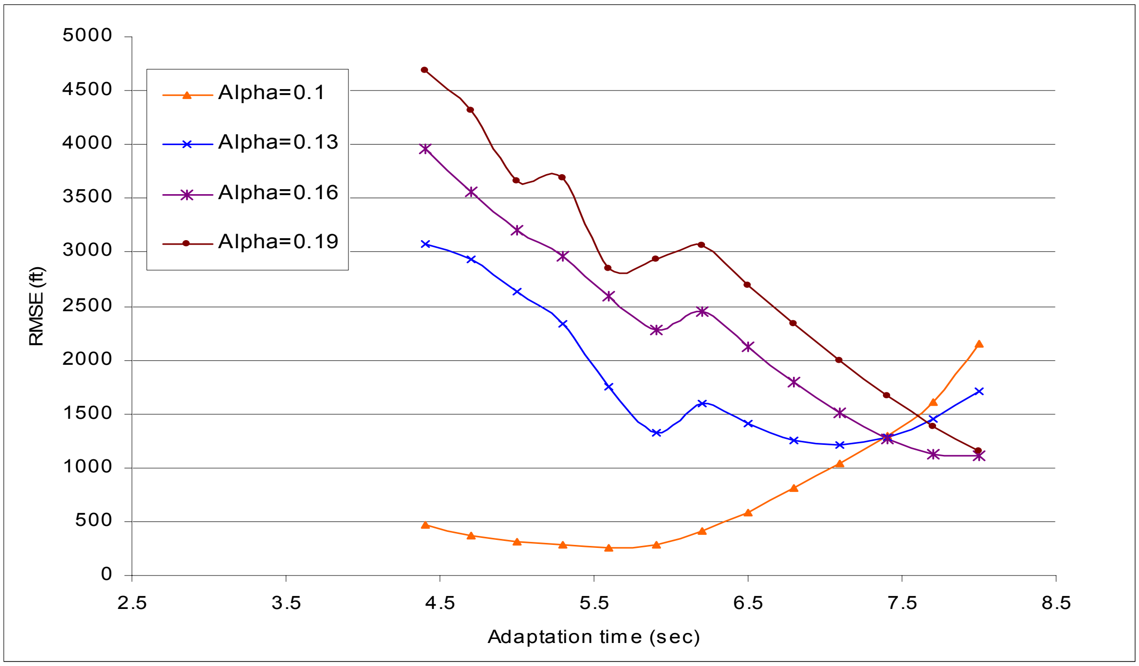


(a) $q^p = 500\ vph$

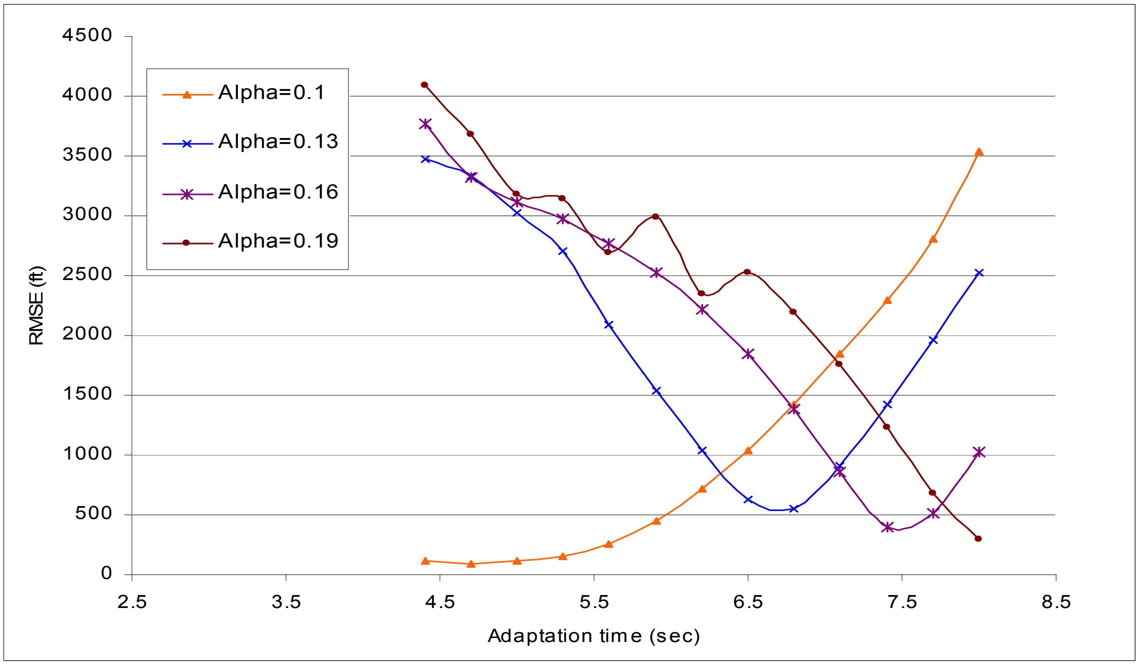


(b) $q^p = 1000\ vph$

Figure 5-18: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - (r = 1/5, N=200) ($q^p = 500,1000 vph$).

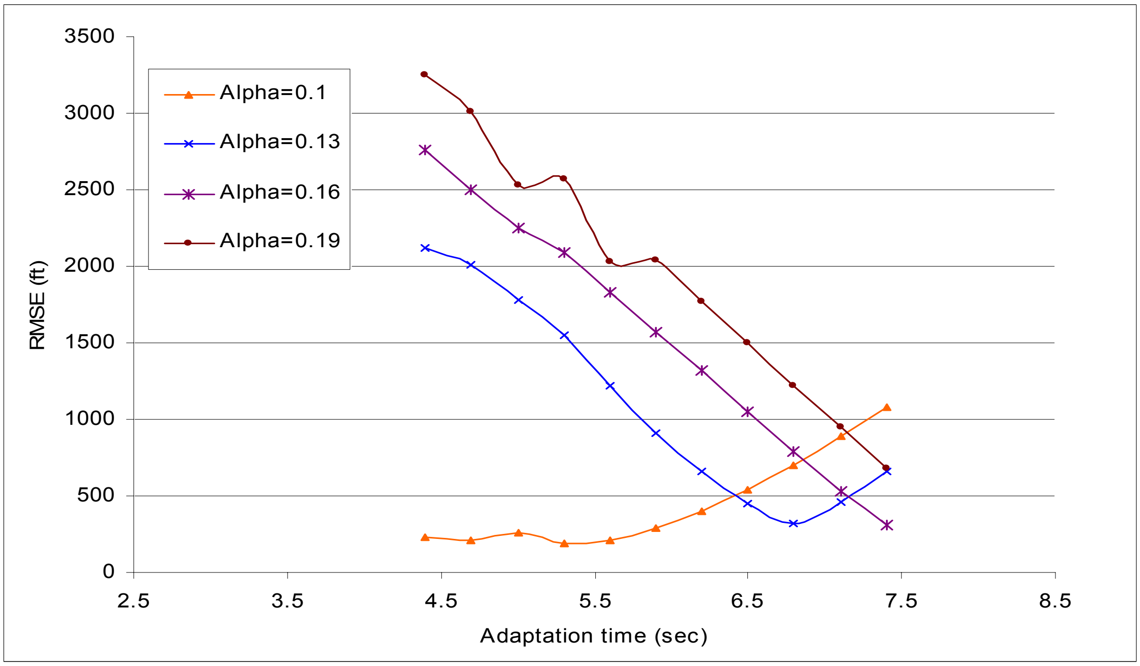


(c) $q^p = 1500\ vph$

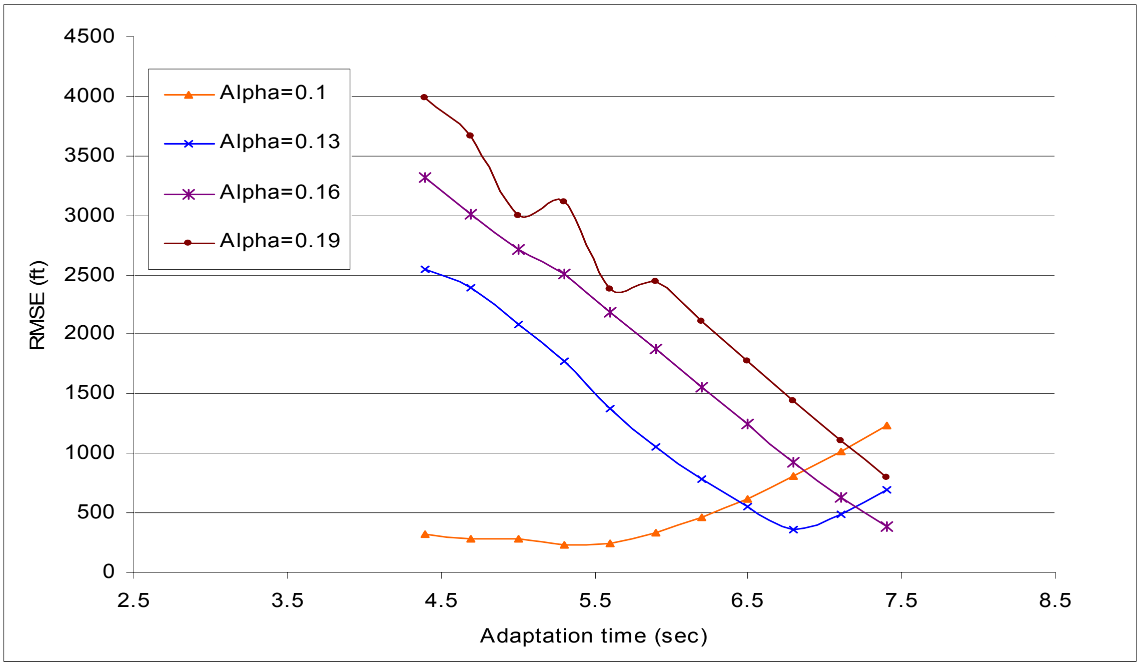


(d) $q^p = 2000\ vph$

Figure 5-18 (Continued): ($q^p = 1500, 2000 vph$ **)**

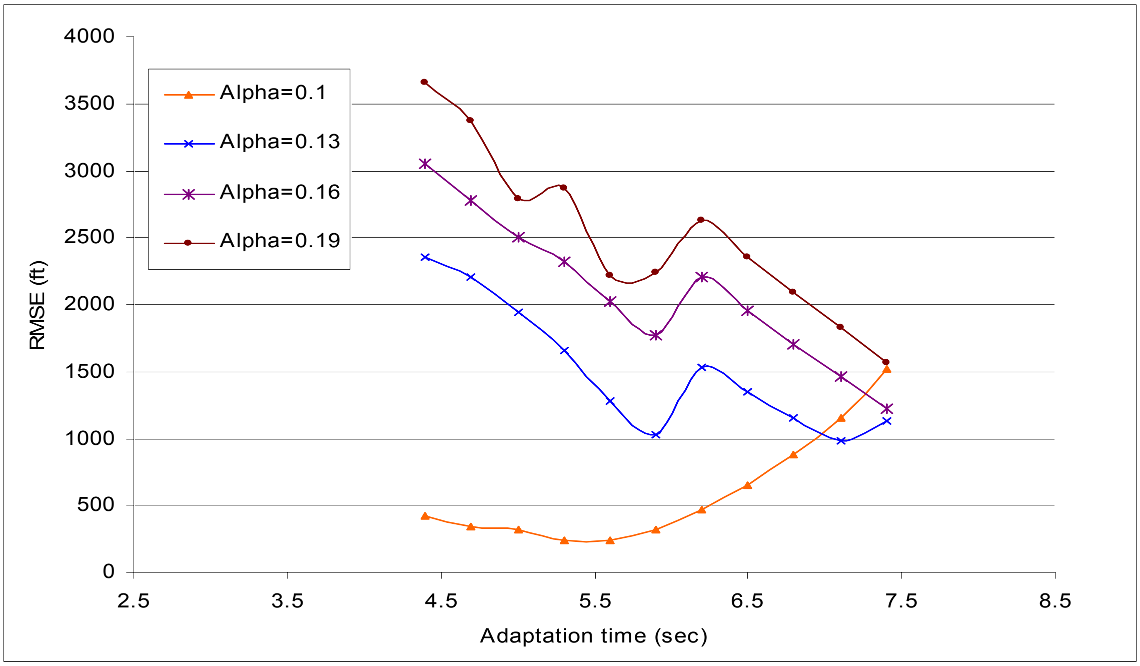


(a) $q^p = 500\ vph$

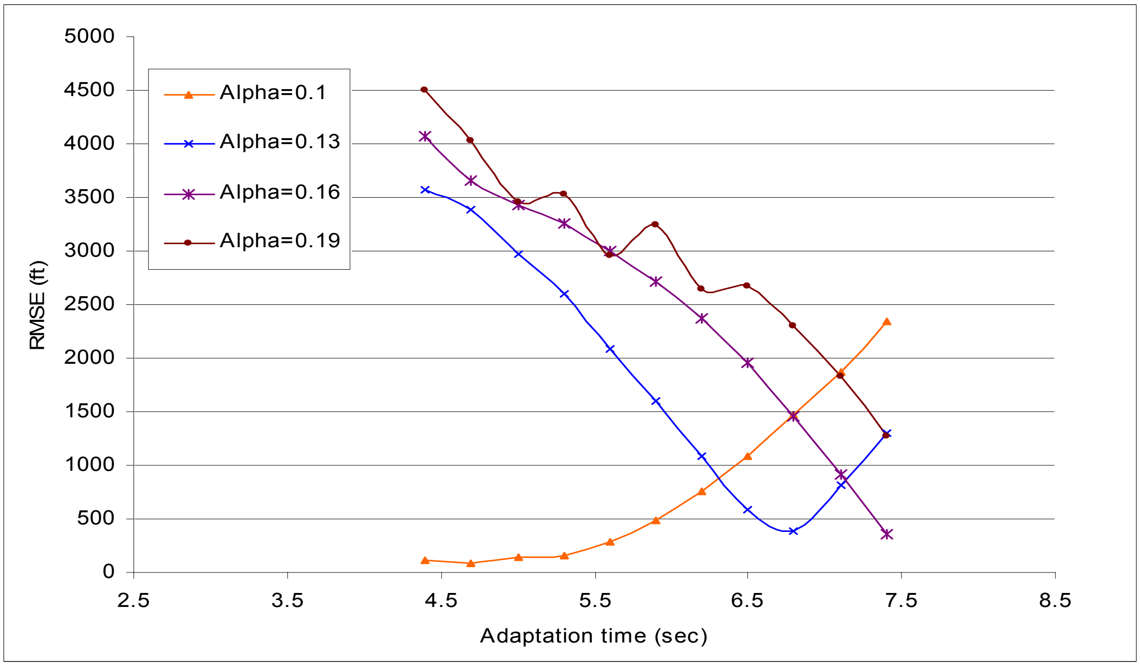


(b) $q^p = 1000\ vph$

Figure 5-19: RMSE for different values of $\alpha^m, \Delta T^m$ and flow rates - (r = 1/5, N=300) ($q^p = 500{,}1000 vph$).



(c) $q^p = 1500\ vph$



(d) $q^p = 2000\ vph$

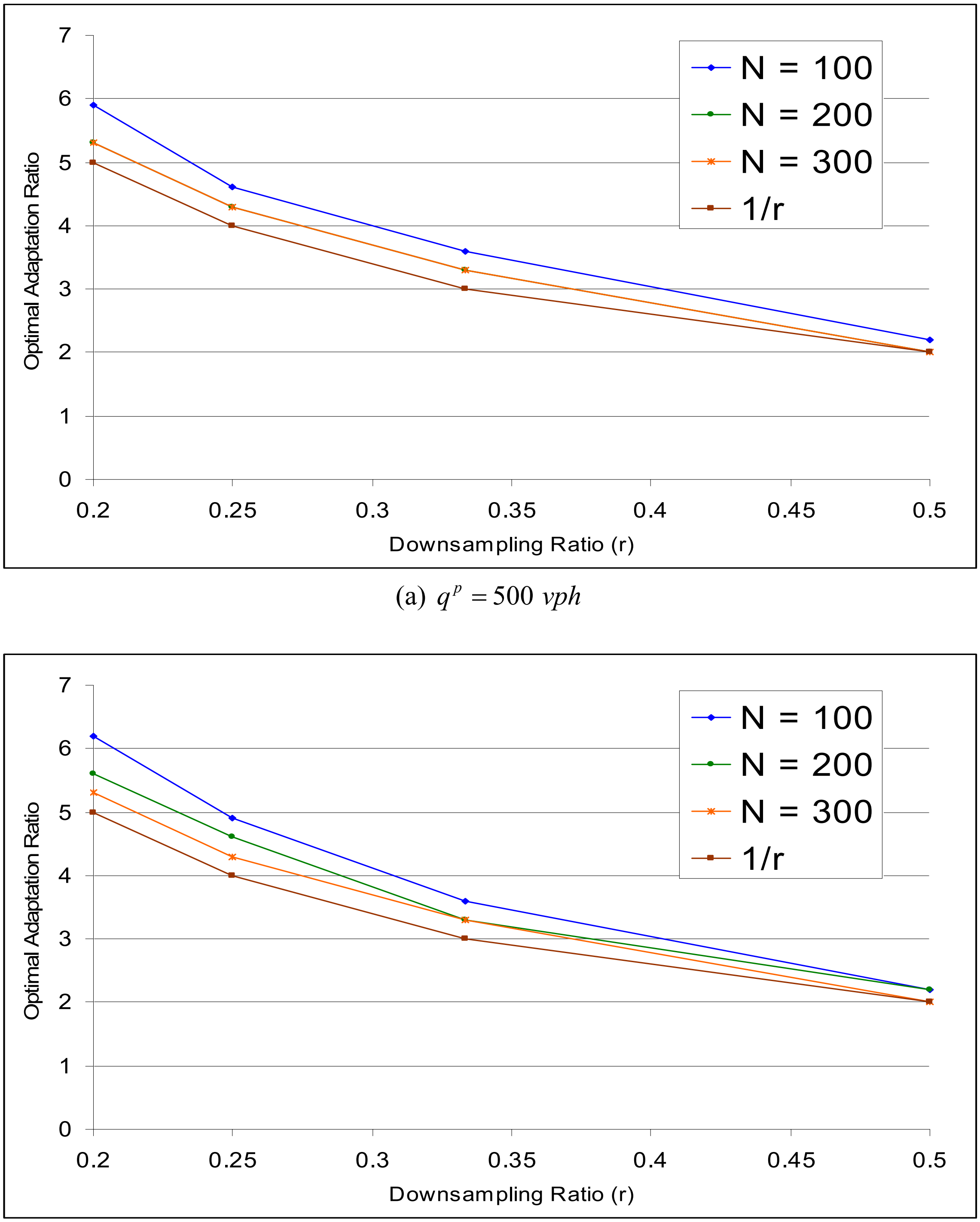
Figure 5-19 (Continued): ($q^p = 1500, 2000vph$**)**

Figure 5-20 shows the effect of downsampling ratio on the exact optimal adaptation ratio $\frac{\Delta T_0^m}{\Delta T^p}$. For each flow rate the curve representing the inverse of r in the figure (suggested near-optimal adaptation ratio) appears to set the lower boundary for all optimal adaptation ratios. The figure suggests that further deviation from $1/r$ value is observed at low downsampling ratios (e.g. r = 1/5) as opposed to high downsampling ratios. This observation is intuitive because lower downsampling ratios lead to more information loss and hence larger errors. However, such deviation appears to decrease with higher flow rate and diminishes for $q^p = 2000$ vph.

Another interesting observation is that the exact optimal adaptation ratios approach the $1/r$ boundary as the number of simulated vehicles increases from 100 to 300. This shows that $\frac{\Delta T_0^m}{\Delta T^p} \to \frac{1}{r}$ as N increases or $\lim_{N\to\infty} \frac{\Delta T_0^m}{\Delta T^p} = \frac{1}{r}$. Even for small N, as previously noted, the increase in information loss from optimal to near-optimal conditions was only marginal. This clearly indicates that near-optimal adaptation values may be adopted without compromising the overall optimal performance of the downsampling procedure.

5.4.OVERALL NEAR-OPTIMAL DOWNSAMPLING PERFORMANCE

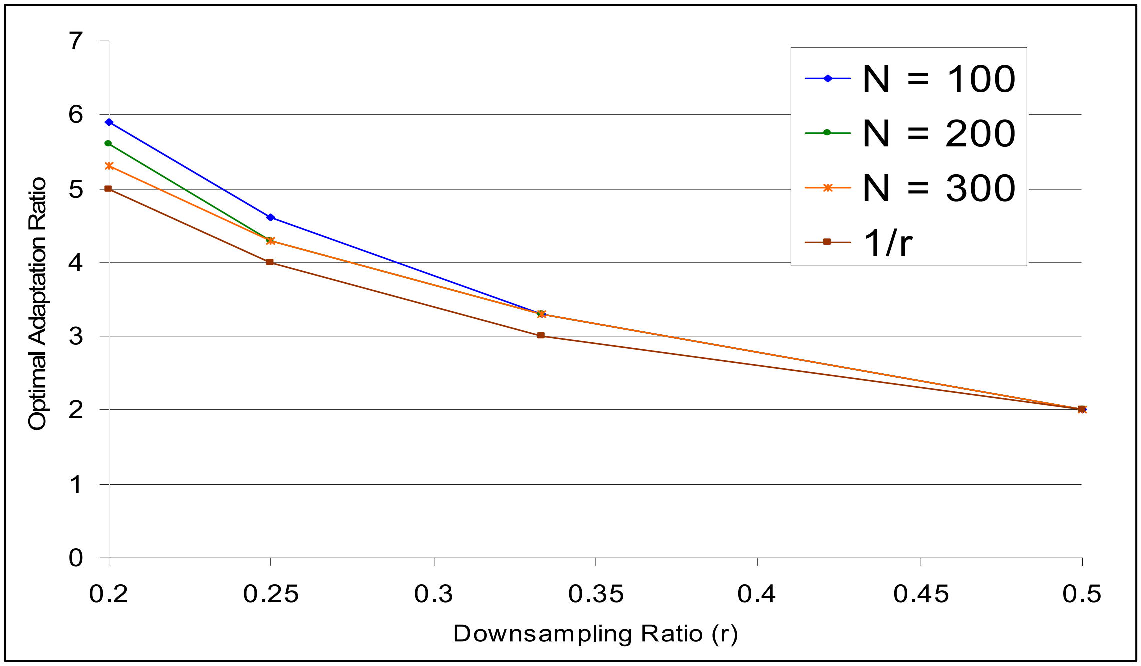
In this section, the effect of different factors (such as the flow rate, number of vehicles in the prototype environment and the downsampling ratio) on the near-optimal performance of the downsampling process is discussed. Figure 5-21 shows that the optimal downsampling performance generally improves with the increase in flow rate, for all downsampling ratios. This trend slightly changes as the number of simulated vehicles



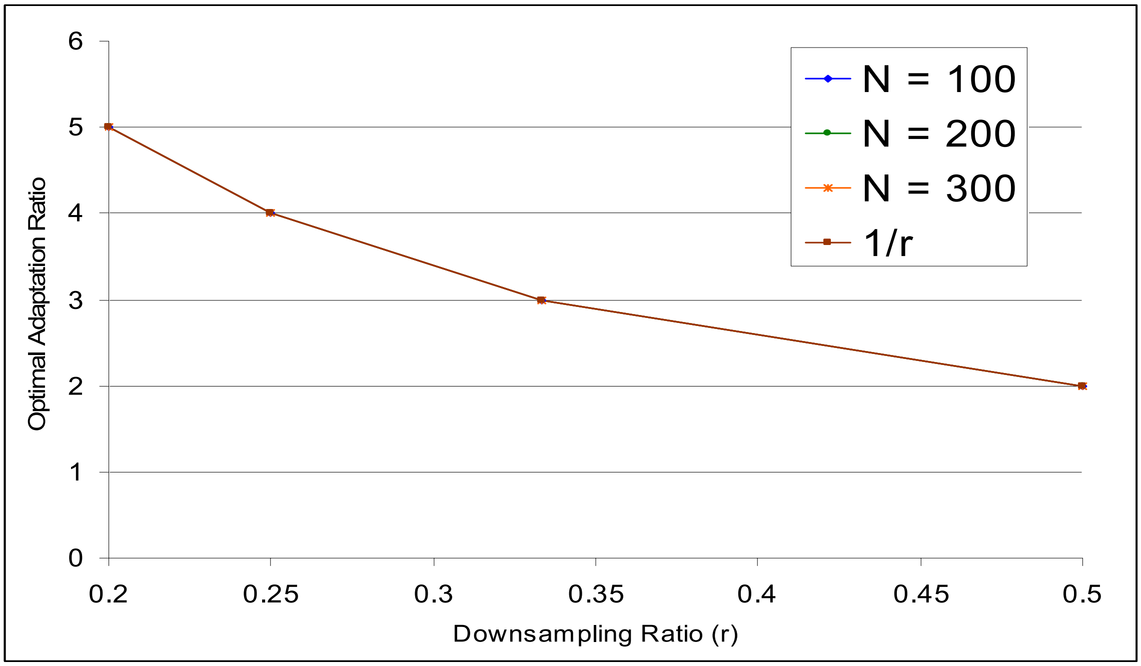
(a) $q^p = 500\ vph$

(b) $q^p = 1000\ vph$

Figure 5-20: Effect of flow rate and downsampling ratio on optimal adaptation time ratios $\frac{\Delta T_0^m}{\Delta T^p}$ ($q^p = 500{,}1000 vph$)

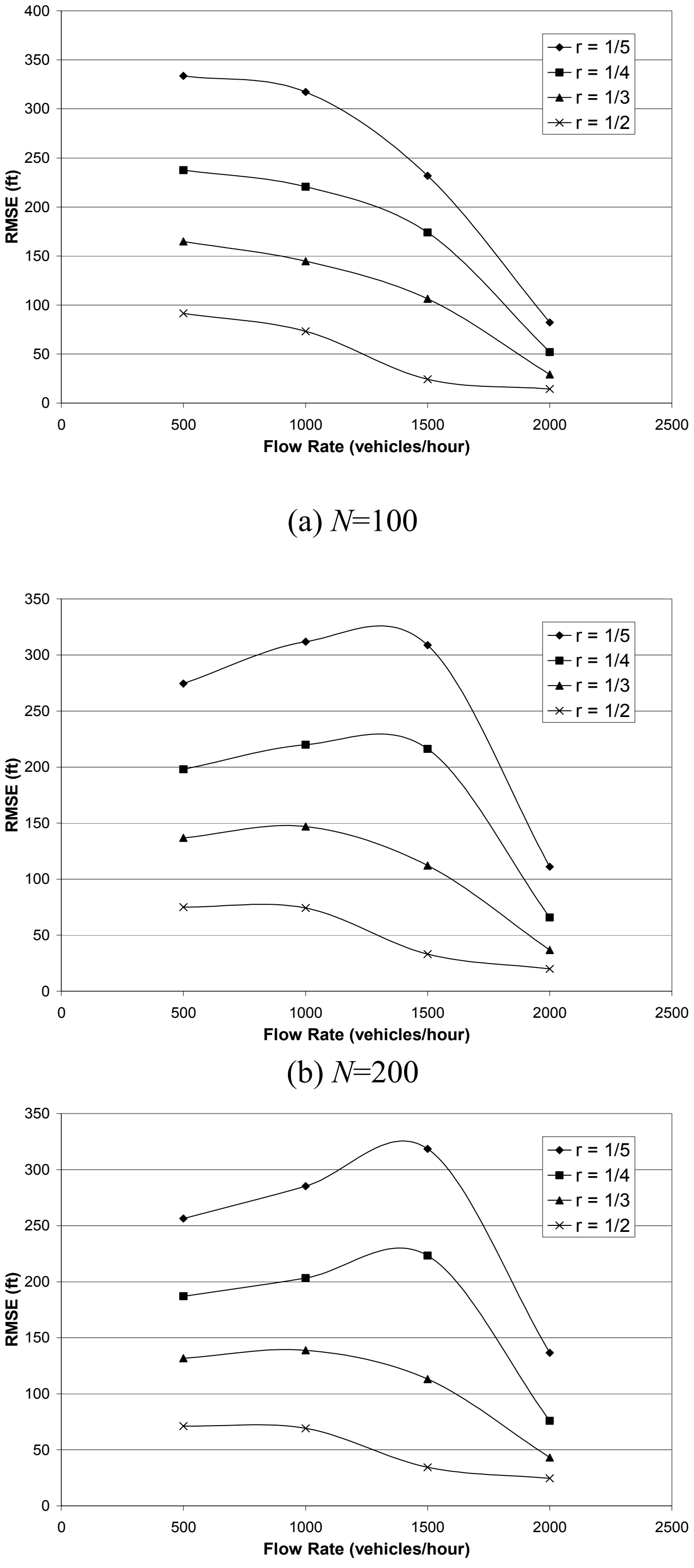


(c) $q^p = 1500\ vph$



(d) $q^p = 2000\ vph$

Figure 5-20 (Continued): ($q^p = 1500, 2000 vph$**)**



(a) N=100

(b) N=200

(c) N=300

Figure 5-21: RMSE for near-optimal values of α^m, ΔT^m and different downsampling ratios

increases from 100 to 300. For *N*=200 and *N*=300, slightly higher errors for the same downsampling ratio are observed in the mid-range of flow rates (from 500 to 2000).

The flow rate, corresponding to the highest errors, appears to increase from nearly 800 to 1400 vph as the downsampling ratio decreases from 1/2 to 1/5. The three figures clearly show that the optimal downsampling performance deteriorates with the decrease in downsampling ratio. This observation is not surprising since, intuitively, lower downsampling ratios will evidently lead to more information loss.

5.5.EFFECT OF DOWNSAMPLING PROCESS ON AVERAGE VEHICULAR DELAY

Another important measure of performance is the average delay per vehicle in the prototype and microcosm environments. Average vehicular delay is one of the performance measures in this experimental study. The experiments discussed earlier also produced results on average vehicular delay in microcosm and prototype. However, only the results of the second stage of the experimental work that generated 36 cases along with the last 12 cases of the first stage (scenario 3) were used because only the mixed traffic flow conditions (generated by scenario 3) has a tremendous impact on the delay.

Since the main objective of the downsampling procedure is to execute the simulation process in the microcosm environment, the microcosm simulation results must be upsampled back to the prototype environment. One of the most common macroscopic measures of performance in traffic simulation models is the average vehicle delay on each link and for the overall network. The ratio of average vehicular delay in the microcosm to the prototype (${}^{d^m}\!/_{d^p}$) was computed for each of the 48 cases, as shown in Table 5-1. The table shows that the ratio was very close to 1.0 in all cases, even for low downsampling ratios. This suggests that the average delay measured from the microcosm

Table 5-1 Ratio of Average Delay per Vehicle in *Microcosm* to *Prototype* ($^{d^m}\!/_{d^p}$) under Optimal Conditions

Downsampling Ratio	Flow Rate (q^p)	$N^p = 100$	$N^p = 200$	$N^p = 300$
$r = \frac{1}{2}$	$q^p = 500\ vph$	1.000701	1.00072	1.001132
	$q^p = 1000\ vph$	1.000698	1.000903	1.000823
	$q^p = 1500\ vph$	1.00026	1.00054	1.000547
	$q^p = 2000\ vph$	1.000255	1.000434	1.000435
$r = \frac{1}{3}$	$q^p = 500\ vph$	1.002366	1.004114	1.001925
	$q^p = 1000\ vph$	1.002181	1.002167	1.001554
	$q^p = 1500\ vph$	1.001819	1.001888	1.001457
	$q^p = 2000\ vph$	1.001018	1.000955	1.000696
$r = \frac{1}{4}$	$q^p = 500\ vph$	1.001840	1.002057	1.002831
	$q^p = 1000\ vph$	1.002269	1.002709	1.002377
	$q^p = 1500\ vph$	1.001733	1.002518	1.002642
	$q^p = 2000\ vph$	1.000764	1.001216	1.001219
$r = \frac{1}{5}$	$q^p = 500\ vph$	1.002454	1.002983	1.003963
	$q^p = 1000\ vph$	1.003054	1.003702	1.003382
	$q^p = 1500\ vph$	1.002079	1.003507	1.003735
	$q^p = 2000\ vph$	1.000933	1.001737	1.001915

reflects that in the prototype, and therefore, the downsampling procedure has successfully retained most of the delay and travel time information in the prototype.

Average delay per vehicle measured the downsampling performance much better than RMSE. This is because in RMSE calculations the error terms are squared and are cumulative throughout the simulation period. This is clearly not the case in average delay calculations, where trajectory errors may have the tendency to cancel out. This shows that although individual vehicles may experience slightly different delays from that in the prototype environment, the overall average delay error remains relatively insignificant. This is an important system operating characteristic that was preserved in the microcosm.

5.6.LOCAL STABILITY

Preservation of local stability in the microcosm and prototype environments is also considered one of the critical characteristics in this study. The third stage of the experimental work was conducted to assess the performance of the downsampling process in terms of local stability in the microcosm and prototype environments.

Near-optimal sensitivity and adaptation time values, obtained from the first two stages of the experimental work, were used to describe the car-following process in the microcosm. The generalized form of near optimal parameters obtained from the first two stages of experimental work is given by:

$$\alpha_o{}^m = r\alpha^p$$

$$\Delta T_0^m \approx {}^1\!/_r\, \Delta T^p$$

From the relation, $C = \alpha.\Delta t$,

$$C^m = \alpha_o{}^m.\Delta T_o^m$$
$$\Rightarrow r\alpha^p.\frac{1}{r}\Delta T^p \Rightarrow \alpha^p.\Delta T^p$$
$$= C^p$$

From these relationships, it is observed that the *C* value for all the study cases (one case correspond to the lead-following pair in the prototype environment and four cases correspond to the lead-following pair in the microcosm with different downsampling ratios) is equal to 0.5. From Table 4-2, *C* value of 0.5 corresponds to Case II, which states that the fluctuation in spacing in the traffic stream will be oscillatory but is exponentially damped. This phenomenon is observed in Figure 5-22. The figure plots the change in spacing between the lead and following vehicle, for all the five cases, when the lead vehicle is subjected to a slight disturbance. The disturbance is created by introducing a change in speed for 5 seconds. The change in spacing decreases initially but the fluctuation dampens out restoring back the stability in the traffic stream.

The discussion concludes that local stability is preserved in the microcosm environment as well. This is a critical system operating characteristic to be preserved in the downsampling process.

5.7.SUMMARY

The results of trajectory based optimization procedure in terms of RMSE and average delay were presented in this chapter. The results establish a relation between the near-optimal parameters in the microcosm and parameters in the prototype environment in a deterministic context. Further testing in a stochastic environment is required to confirm such a relation. The next chapter gives an introduction to an optimization procedure that can be used in a stochastic environment.

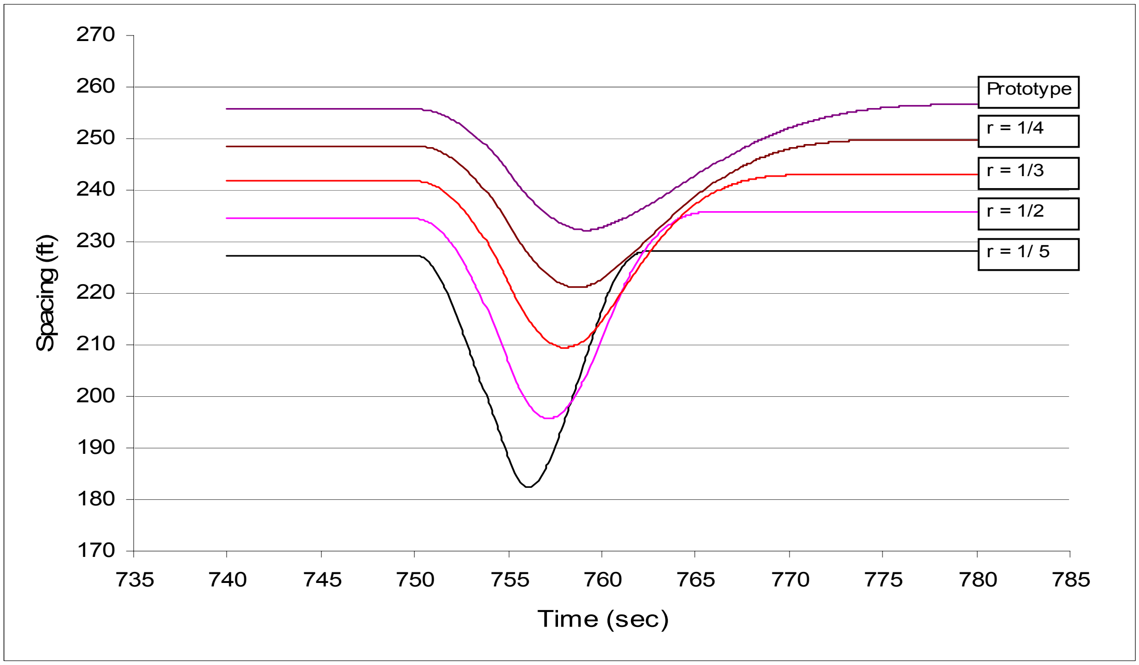


Figure 5-22: Change in spacing between a lead-following pair in prototype and microcosm

6. STOCHASTIC IMPLICATIONS AND OPTIMIZATION PROCEDURE

6.1.INTRODUTION

The results presented so far are based on an optimization process to minimize the deviation of vehicular trajectories in the microcosm environment from those in the prototype environment. While the trajectory matching procedure applies to deterministic conditions, where vehicles do not join or exit the stream, this formulation will no longer apply if stochastic variations are introduced in the system in future research. This is because of the stochastic driving behavior in the microcosm and prototype environments. In such a case, the objective function, *SSE,* must be redefined in terms of macroscopic parameters such as average delay or density. The following sections describe the methodology developed and the results obtained from the experimental work.

6.2.METHODOLOGY

A new methodology was developed to apply the optimization procedure under stochastic conditions. A new procedure is necessary because under stochastic conditions microscopic characteristic matching (e.g., vehicle trajectory matching) cannot be applied due to the stochastic behavioral environment in the microcosm and prototype. The objective function, *SSE*, to be optimized is defined in terms of a macroscopic characteristic such as density. Mathematically, the objective function *SSE* is defined as:

$$SSE = \sum_{t=1}^{T} \sum_{j=1}^{J} \left[\frac{N^p(t,j)}{L^p(j)} - \frac{N^m(t,j)}{L^m(j)} \right]^2 \tag{18}$$

Where,

$N^p(t,j)$ and $N^m(t,j)$ = number of vehicles observed in segment *j* at time *t* in the prototype and microcosm environments, respectively.

$L^p(j)$ and $L^m(j)$ = the length of segment j in the prototype and microcosm environments, respectively.

$\frac{N^p(t,j)}{L^p(j)}$ and $\frac{N^m(t,j)}{L^m(j)}$ = the density for segment j at time t in the prototype and the microcosm environments, respectively.

J = the total number of segments

T = the number of times the density is sampled over the simulation period (the simulation period divided by the sampling interval).

Note that the number of segments in each link in the prototype and the microcosm environments should be equal. As a result of geometric downscaling, the ratio $\frac{L^m(j)}{L^p(j)}$ will remain equal to the downsampling ratio *r*. Minimization of the function *SSE* ensures that most of the characteristics of the prototype have been preserved in the microcosm environment. Essentially all previously defined constraints still apply to the mathematical problem and the goal of the optimization process is to seek the optimal values of the two behavioral parameters α^m and ΔT^m.

6.3.EXPERIMENTAL WORK

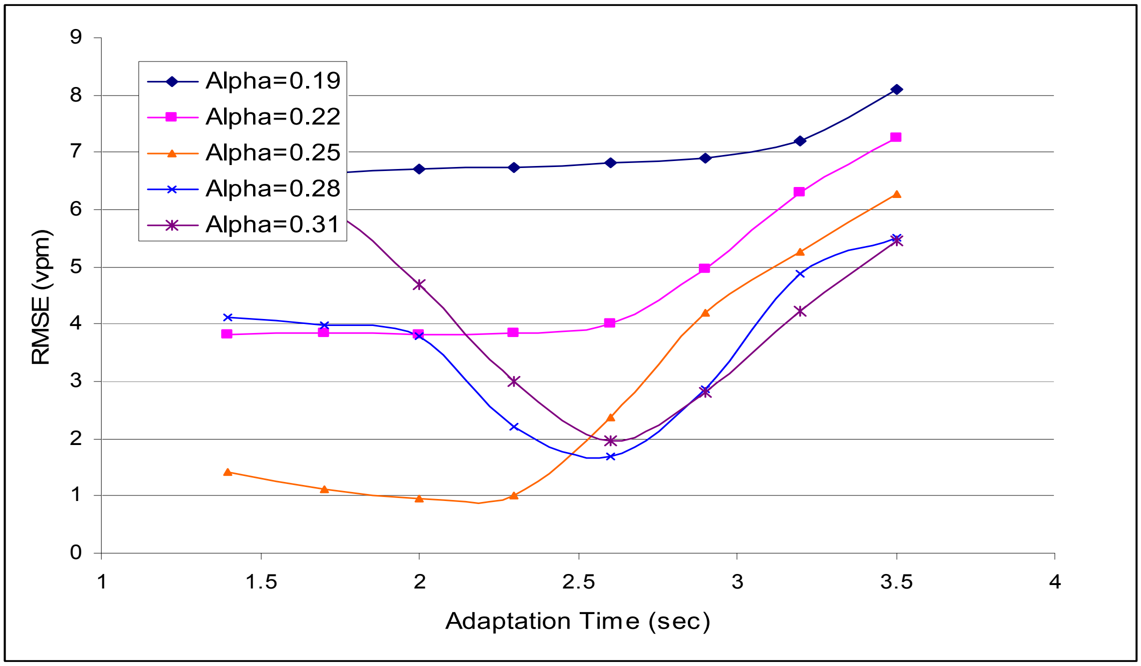
The density-based optimization process was conducted by dividing the freeway corridor into one-mile segments and sampling the density in each system every one minute throughout the one-hour simulation period. RMSE was used as the performance measure to test the efficiency of the density based optimization process. RMSE is derived from SSE, defined in Equation (18), as follows:

$$RMSE = \sqrt{\frac{SSE}{(T/\delta t)(L/\delta l)}} \tag{19}$$

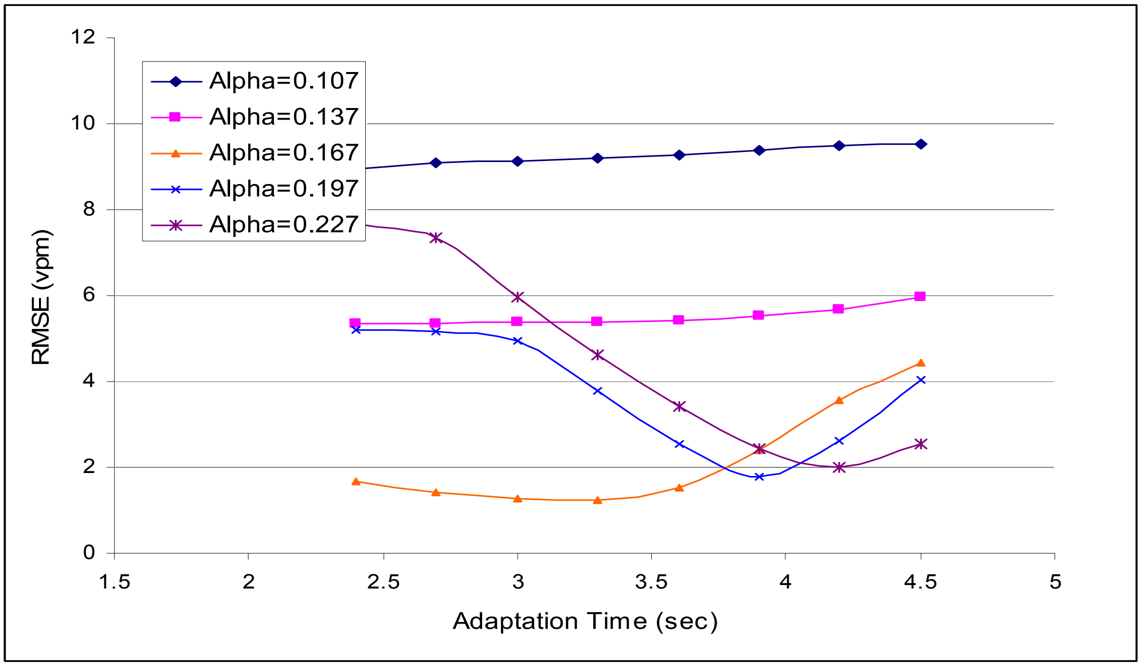
Where, δl is the length of the unit segment (1 mile, in this case) and all the variables are as previously defined.

The simulation module was modified to include the density based optimization procedure. The optimization procedure was carried out for different downsampling ratios $\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right)$ and flow rates (500 and 2000 vph) resulting in 8 cases. Number of vehicles in the prototype environment was fixed at 300 and the lead vehicle was forced to stop 3 times during the simulation period (scenario 3). RMSE was calculated for the 8 selected cases.

For each case, a range of sensitivity (α^m) and adaptation time (ΔT^m) values were tested to trap the optimal values of α^m and ΔT^m in the microcosm environment in terms of minimum RMSE. Figure 6-1 shows the information loss for different α^m and ΔT^m for a flow rate of 500 vph. Each individual figure in Figure 6-1 represents a separate case and corresponds to a different downsampling ratio. A range of 0.04 to 0.31 sec^{-1} for α^m and 1 to 8.5 seconds for ΔT^m was tested to generate the curves for each flow rate. A wider range of α^m and ΔT^m were used initially (α^m= 0.02 to 0.50 sec^{-1} with an interval of 0.05 and ΔT^m=1.0 to 10.0 seconds with an interval of 1.0 seconds), but the range was narrowed down to use smaller intervals to locate the exact minimal RMSE values. Using the same range of sensitivity and adaptation values, performance curves were similarly generated for the remaining cases for a flow rate of 2000 vph as shown in Figure 6-2. Both the figures (Figure 6-1 and Figure 6-2) show that the minimum RMSE values, derived from the function *SSE*, are observed at the same corresponding optimal values of sensitivity and adaptation parameters that were determined earlier from the trajectory error function in the stage 1 of the experimental analysis. In other words, the figures

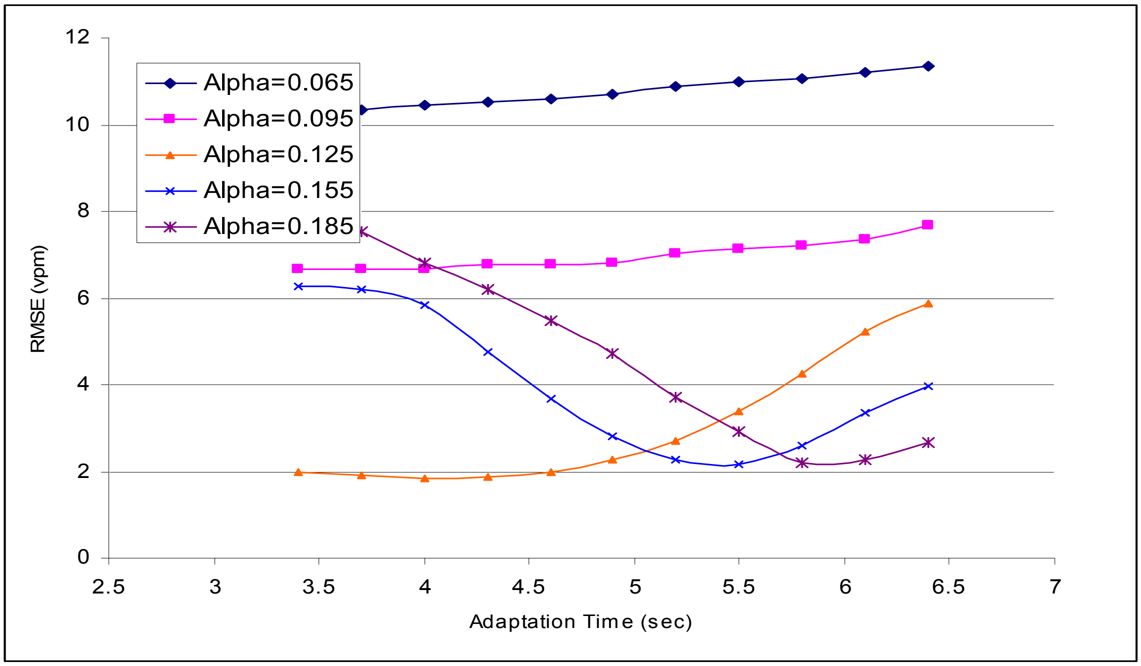


(a) *r* = 1/2

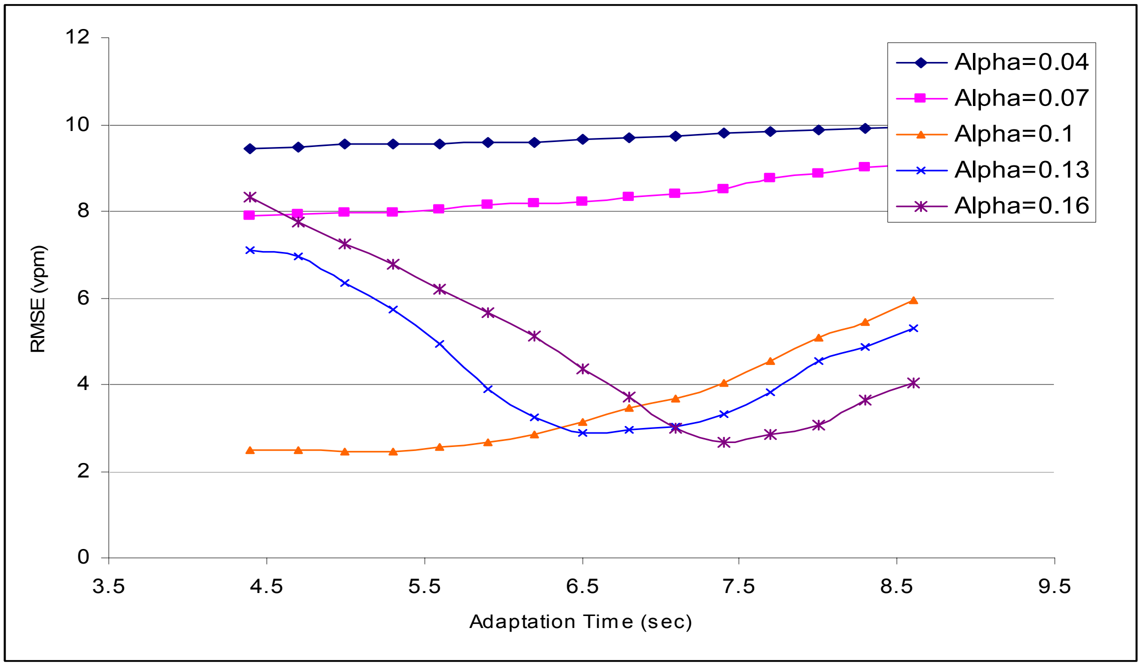


(b) *r* = 1/3

Figure 6-1: RMSE for density *k* and different values of α^m, ΔT^m and downsampling ratios *r* - (q^p = 500, *N*=300)



(c) $r = 1/4$



(d) $r = 1/5$

Figure 6-1 (Continued): ($Q^P = 500$, N=300)

confirm that the same relationship exists between the behavioral parameters in the prototype and the microcosm environments; i.e. $\alpha_o{}^m = r\alpha^p$ and $\Delta T_0^m \approx {}^1\!/_r \Delta T^p$.

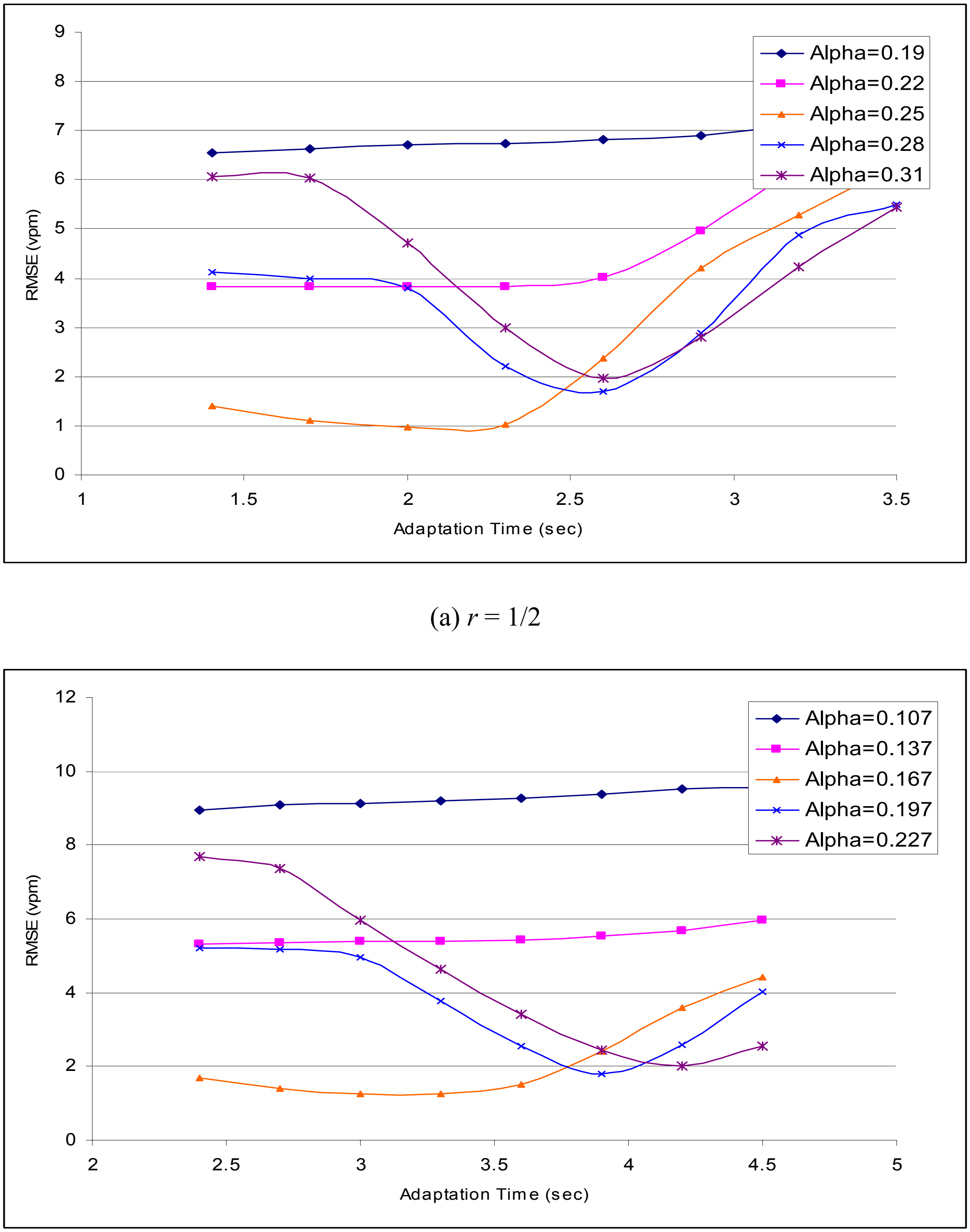
Intuitively, close trajectory matching leads to close density matching between the two environments. However, the opposite, is not necessarily true since density is a macroscopic, aggregate measure that does not retain vehicle identities. Obviously, optimization based on density is less stringent. This can be seen in both figures when comparing the difference between the minimum RMSE values for *r*=1/2 and *r*=1/5.

6.4.STOCHASTIC CONSIDERATIONS

The optimization procedure discussed so far in this chapter presents an approach to perform the scaling process in a stochastic environment but does not consider the scalability of probability distribution functions that describe the random variables in simulation processes. In stochastic environments, spacing, headway and other driver behavioral characteristics are treated as random variables with some known probability distribution function defined by the mean and variance n the prototype environment. To establish a relationship between the random variables in both environments, let and X_m and X_p be the random variables representing vehicle arrivals in the microcosm and prototype environments, respectively. Downscaling the random variable linearly by the downsampling ratio *r*,

$$X_m = rX_p \tag{20}$$

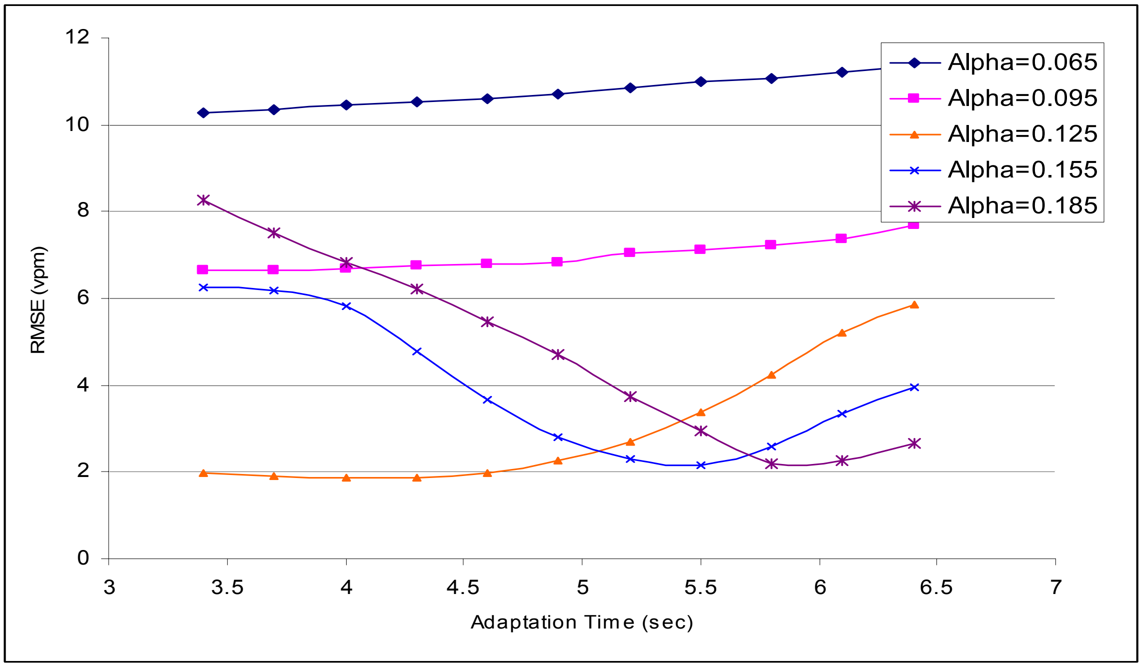
The linear downscaling of the random variable will result in downscaling the corresponding mean and variance of the probability distribution function such that $E[X^m] = rE[X^p]$ and $V[X^m] = r^2V[X^p]$, where, $E[X^m]$ and $V[X^m]$ are the mean and variance of the distribution of X_m in the microcosm environment.



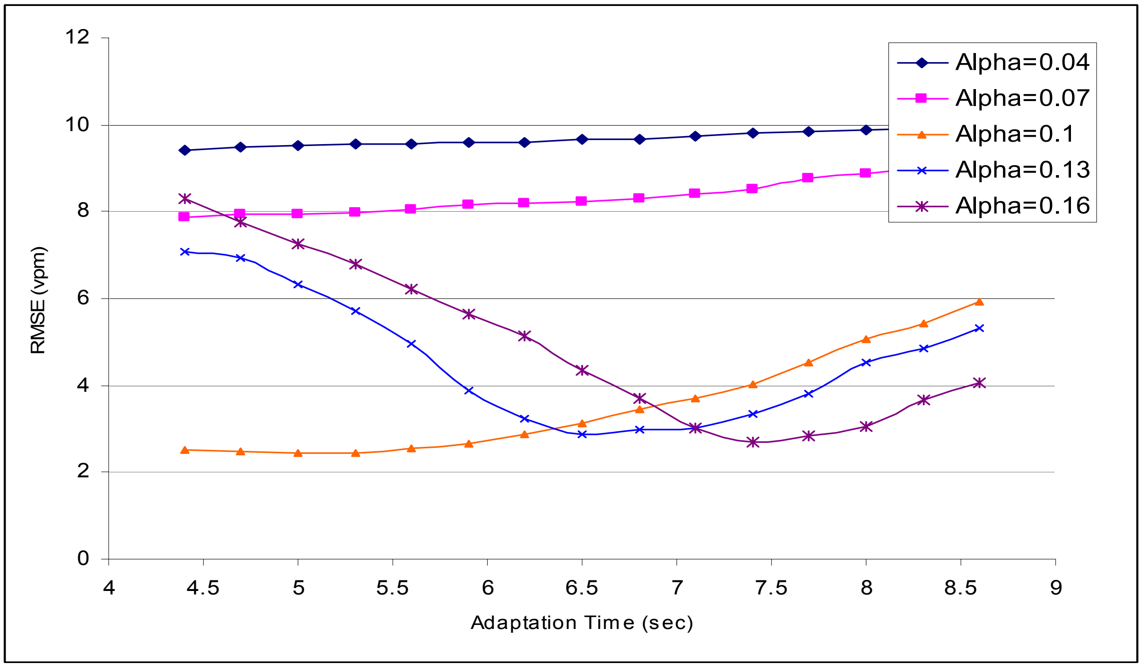
(a) $r = 1/2$

(b) $r = 1/3$

Figure 6-2: RMSE for density *k* and different values of α^m, ΔT^m and downsampling ratios *r* - (q^p = 2000, *N*=300)



(c) $r = 1/4$



(d) $r = 1/5$

Figure 6-2 (Continued): (q^p = 2000, N=300)

Another desired statistical property that we seek to preserve in the downsampling process is the scalability of the probability distribution function. This implies that the cumulative probability value for any random variable X_p in the prototype environment should be equal to the cumulative probability value for the random variable X_m in the microcosm environment (for all $x_m = rx_p$). This property should be verified for different probability distribution functions that are commonly used in microscopic traffic simulation processes.

Mathematically, let $G(x_m)$ and $F(x_p)$ be the cumulative distribution functions for the random variable X in the microcosm and prototype environments, respectively. To retain the scaling property, we set

$$G(x_m) = F(x_p) \tag{21}$$

Differentiating both sides w.r.t x_p,

$$\frac{dG(x_m)}{dx_p} = \frac{dG(x_m)}{dx_m}.\frac{dx_m}{dx_p} = \frac{dF(x_p)}{dx_p} \tag{22}$$

Since $x_m = rx_p$, then $\frac{dx_m}{dx_p} = r$ and

$$r.\frac{dG(x_m)}{dx_p} = \frac{dF(x_p)}{dx_p} \tag{23}$$

The derivative of CDF gives its corresponding probability density function. Therefore:

$$g(x_m) = \frac{1}{r}.f(x_p) \tag{24}$$

For the downsampling process to be successful in stochastic traffic conditions, this property should be preserved in the microcosm and prototype environments for any random variable; vehicle arrivals, for instance. The negative exponential distribution and normal distribution have been investigated to check if they preserve this property.

6.4.1. NEGATIVE EXPONENTIAL DISTRIBUTION

The negative exponential distribution is commonly used to represent random time headways between vehicles in a system. The cumulative distribution function of a negative exponential distribution is given by,

$$F(t) = P(h \le t) = 1 - e^{-\frac{t}{T}} \tag{25}$$

Differentiating w.r.t. 't' to obtain the probability density function,

$$\Rightarrow f(t) = \frac{1}{T} e^{-\frac{t}{T}} \tag{26}$$

The mean headways, in the prototype and microcosm environments, are T_p and T_m, respectively. Let t_m and t_p be the random variables representing time headways in the microcosm and prototype environments, respectively. Vehicle elimination from the prototype environment leads to:

$$t_m = \frac{1}{r} t_p$$

And

$$T_m = \frac{1}{r} T_p \tag{27}$$

In the prototype and microcosm environments, the headway density function is defined as:

$$f(t_p) = \frac{1}{T_p} e^{\frac{-t_p}{T_p}} \tag{28}$$

and

$$g(t_m) = \frac{1}{T_m} e^{\frac{-t_m}{T_m}} \tag{29}$$

Substituting for $T_m = \frac{1}{r} T_p$ and $t_m = \frac{1}{r} t_p$ in equation (29),

$$g(t_m) = \frac{1}{\frac{1}{r} T_p} e^{\frac{-t_p}{r} \cdot \frac{r}{T_p}} \tag{30}$$

$$\Rightarrow g(t_m) = \frac{r}{T_p} e^{\frac{-t_p}{T_p}}$$

$$\Rightarrow r.f(t_p)$$

Therefore, it is confirmed that the downscaling property is preserved in negative exponential distribution.

6.4.2. NORMAL DISTRIBUTION

Normal distribution is often used to represent the desired free-flow speed and some other driver behavioral parameters. The probability density function of a normal random variable s_p (in the prototype environment) is given by:

$$f(s_p) = \frac{1}{\sigma_p \sqrt{2\pi}} e^{\frac{-(s_p - \mu_p)^2}{2\sigma_p^2}} \quad where, -\infty < s_p < \infty \tag{31}$$

Where μ_p and $\sigma^2{}_p$ are the mean and variance of a normal distribution in the prototype environment. Let $g(s_m)$ be the probability density function in the microcosm environment:

$$g(s_m) = \frac{1}{\sigma_m \sqrt{2\pi}} e^{\frac{-(s_m - \mu_m)^2}{2\sigma_m^2}} \quad where, -\infty < s_m < \infty \tag{32}$$

Assuming the linear scalability of the random variable ($s_m = rs_p$) will result in downscaling the mean speed in the microcosm (μ_m) by r and the variance (σ_m^2) by r^2 to their corresponding values in prototype environment (μ_p, σ_p^2) in a normal distribution.

Substituting in Equation (32)

$$g(s_m) = \frac{1}{r\mu_p\sqrt{2\pi}} e^{\frac{-\left(rs_p - r\mu_p\right)^2}{2\sigma_p{}^2 . r^2}} \qquad (33)$$

$$\Rightarrow g(s_m) = \frac{1}{r} . f(s_p)$$

The downscaling property is also preserved if normal distribution is considered for the speed and other driver behavioral parameters.

6.5.SUMMARY

A density based optimization procedure was developed to optimize the downscaling process under stochastic conditions. However, the behavioral parameters were assumed deterministic to test the procedure. To introduce stochastic behavioral parameters in the traffic stream, two probability distributions for time headway were tested to investigate their downscaling properties.

7. SUMMARY AND CONCLUSIONS

7.1.STUDY SUMMARY

This study presented an approach for reducing the computational requirements of microscopic traffic simulation systems. To achieve this objective, a downsampling procedure was developed to create a geometrically, kinematically, and behaviorally representative reduced-scale system (microcosm). A methodology was developed to optimize the behavior of vehicles in the microcosm by minimizing the trajectory errors. The mathematical formulation was derived exclusively for one-lane operation and deterministic driving conditions to seek optimal solutions that would facilitate optimization under stochastic conditions in subsequent research studies.

Experimental work was conducted to examine the behavioral scalability for one-lane freeway segment under different operating conditions. A simulation module was developed in PERL to perform the experimental work. The experimental work was carried out in stages under different traffic flow conditions and downsampling ratios for the GM-I car following model. RMSE and average delay per vehicle were used as the performance measures to assess the efficiency of the downsampling process. The first stage of the experimental work investigated 36 cases to determine an optimal solution for the two behavioral parameters (sensitivity and adaptation time).

A second stage of the experiment was conducted using different downsampling ratios (1/3, 1/4 and 1/5). A total of 36 cases were investigated using different flow conditions. RMSE and average vehicular delay were used as performance measures. Finally, a third stage of the experiment was performed to investigate the preservation of local stability in the microcosm environment.

To test the downscaling process under stochastic conditions, a density based optimization procedure was developed since individual trajectory matching will no longer be feasible under stochastic conditions. This approach was tested for 8 different cases (two flow rates and 4 downsampling ratios). RMSE was used as the performance measure to evaluate the density based optimization. However, stochastic distributions for the traffic and driver characteristics were not considered. The study also introduced an initial discussion on the stochastic implications of the downsampling procedure by examining the scaling properties of two commonly used probability distributions; negative exponential distribution and normal distribution. The two distributions are commonly used to describe headway and speed distributions, respectively.

7.2.CONCLUSIONS

The study investigated (as a part of the literature review) the role and limitations of some of the state-of-the-art traffic microscopic simulation models. It was found that while CORSIM has a limitation on the network size and the number of simulated entities, PARAMICS and AIMSUN2 does not have any such limitations. However, the simulation efficiency depends on the performance of the machine on which they are run.

A mathematical approach is developed to simulate a reduced scale system with fewer entities to improve the computational efficiency of microscopic simulation of large transportation networks. An objective function with a set of constraints was defined that explains the behavioral scalability of traffic simulation processes.

Experimental analysis was conducted in three stages to test the approach performance under different traffic conditions. The results of the first stage of the experimental work show that for a 50% downsampling ratio, the optimal values of the sensitivity parameter, in terms of RMSE, are 50% ($\alpha_0^m / \alpha^p = r$) of their prototype value.

The optimal values of adaptation time , however, increased by nearly 100% of their corresponding prototype values ($\Delta T_0^m / \Delta T^p \approx 1/r$). Another important observation was that the ratio of the optimal adaptation time in the microcosm to that in the prototype environments approaches the inverse of the downsampling ratio as the number of simulated vehicles increased from 100 to 300. The results of the second stage also produced optimal values that are consistent with their corresponding downsampling ratios and confirmed the findings of the first stage. Also, the ratio of the average vehicular delays in the microcosm and the prototype environments for each case was very close to 1.0, which suggests that the effect of information loss caused by downsampling was relatively insignificant and also that the optimization procedure was successful in preserving one of the most important macroscopic characteristics in simulation processes. The results of the third stage of the experiment that investigated the local stability in the microcosm environment show that for the optimal values of behavioral parameters, local stability is preserved in the microcosm environment. The density based optimization procedure introduced to account for stochastic variations in the microcosm and prototype environments supplement the findings of the earlier stages of the experiment. These results establish the relationships between the behavioral parameters in the prototype and microcosm environments in deterministic conditions. These equations facilitate the simulation of the reduced scale system and then upsample the results back to the prototype environment.

Microscopic simulation of the reduced scale system ensures higher computational efficiency and quicker results that are extremely useful while evaluating transportation systems in real-time. The developed approach also finds applications in emergency evacuation procedures and in coarse analysis, such as planning. This approach, if successful, will have a tremendous impact on the capabilities of next generation traffic simulation models. Further research is necessary to test this approach in stochastic conditions along with different car-following and lane-changing models.

7.3.FUTURE RESEARCH

In this research, GM-I car-following model was used. Different car-following models can be used to test the applicability of this research. Possible alternatives to the one used in this study are the other car-following models in the GM family as well as some other recently developed ones. The other limitation of this research is the use of deterministic driving behavior in the simulated network. A more realistic stochastic representation of a transportation system is considered as the next most appropriate direction for this research.

REFERENCES

Adams, S., and L. Yu, “An Evaluation of Traffic Simulation Models for Supporting ITS Development” Report SWUTC/00/167602-1, Center for Transportation Training and Research-2000.

Bernauer, E., L. Breheret, S. Algers, M. Boero, C. D. Taranto, M. Dougherty, K. Fox and J. F. Gabard (1997), “Review of Micro-Simulation Models”, SMARTEST Project Report.

Brackstone, M. and M. McDonald. Car-following: A Historical Review. In *Transportation Research Part F*, Vol.2, Issue 4, 1999, pp. 181-196.

Chandler, R.E., R. Herman and E. W. Montroll, (1958) “Traffic Dynamics: Studies in Car-Following”, Operations Research, 6, 165-184.

Chu, L., H. X. Liu, W. Recker, and H. M. Zhang, “Development of A Simulation Laboratory for Evaluating Ramp Metering Algorithms”, Transportation Research Board, 2002.

Chundury, S., and B. Wolshon. Evaluation of the CORSIM Car-Following Model Using GPS Field Data. In *Transportation Research Record 1710*, TRB, National Research Council, Washington, D.C., 2000, pp. 114-121.

Gazis, D. C., R. Herman, and R. B. Potts, (1959) “Car Following Theory of Steady State Traffic Flow”, Operations Research, 7, 499-505.

Gazis, D. C., R. Herman, and R. W. Rothery, (1961) “Nonlinear Follow the Leader Models of Traffic Flow”, Operations Research, 9, 545-567.

Hatipkarasulu, Y., B. Wolshon, and C. Quiroga. A GPS Based Approach for the Analysis of Car Following Behavior. *Journal of Transportation Engineering,* ASCE, Vol. 126. No. 4, 2000, pp. 324-331.

Herman. R., R. W. Rothery, E. W. Montroll and R. B. Potts, (1959) “Traffic Dynamics: Analysis of Stability in Car Following”, Operations Research, 7, 86-106.

Highway Capacity Manual (2000), Transportation Research Board – Committee on Highway Capacity and Quality of Service, pp 31-3—31-6.

Hoefs, D. H. (1972) “*Entwicklung einer Messmethode uber den Bewegungsablauf des Kolonnenverrkehrs*”, Universitat (TH) Karlsruhe, Germany.

Hourdakis, J., M. Koka and P. G. Michalopoulos, “Computer Aided Testing and Evaluation of Adaptive Ramp Control Strategies”, Transportation Research Board, 1999.

ITT systems and Science Corporation, CORSIM User’s Guide, Contract No. DTFH61-95-C-00125, FHWA, Turner-Fairbank Highway Research Center, March 2001.

Ishak, S., S. Chakravarthy, C. Alecsandru, “Performance Versus Scalability of Microscopic Traffic Simulation Systems”, Accepted for Presentation at the 83rd TRB Annual Meeting (Publication Recommendation Pending), 2003.

Grama, A.,V. Kumar, A. Gupta and G. Karypis, “Introduction to Parallel Computing”, University of Minnesota, 1994.

Kwon, E., B. Choi and H. Park,, “A Personal Computer based Parallel Simulation System for Online Assessment of Freeway Operational Strategies”, Transportation Research Board, 1998.

Leonard II, J., “An Overview of Simulation Models in Transportation”, Simulation Technology, Vol. 5, No. 1b, September 2001.

Levison, W. Interactive Highway Safety Design Model Issues Related to Driver Modeling. In *Transportation Research Record 1631*, TRB, National Research Council, Washington, D.C., 1998, pp. 20-27.

May, A.D. (1990) “Traffic Flow Fundamentals”, Prentice Hall, Englewood Cliffs, New Jersey.

Nakayama, S., R. Kitamura, and S Fujii. Drivers’ Learning and Network Behavior: Dynamic Analysis of the Driver-Network System as a Complex System. In *Transportation Research Record 1676*, TRB, National Research Council, Washington, D.C., 1999, pp. 30-36.

Ozaki, H. (1993). “Reaction and Anticipation in the Car Following Behavior”, In Proceedings of the Thirteenth International Symposium on Transportation and Traffic Flow Theory”, 349-366.

PARAMICS Online V 3.0 User Guide, Quadstone Ltd., Edinburgh, October 2000.

Pritsker, A. A. (1986). “Introduction to Simulation and SLAM II.” *A Halted Press Book*, Wiley, New York.

Revised monograph on Traffic Flow Theory (2000) Special Report by Transportation Research Board.

Treiterer, J. and J. A. Myers, (1974) “The Hysteresis Phenomenon in Traffic Flow”, In Proceedings of the Sixth International Symposium on Transportation and Traffic Flow Theory”, Sydney, 13-38.

Velan, S. M. and M. Van Aerde, (1996) “Gap Acceptance and Approach Capacity at Unsignalized Intersections”, ITE Journal, Vol.66, No.3, pp.40-45.

APPENDIX – SIMULATION PROGRAM MODULE

```
################################################################################
#       prototype.pl
                                        #
#       The Prototype simulation module
                                #
#       Created:                Mon, Apr 14 2003
                                #
#       Modified:               Tue, May 5 2003
                                #
#
# benchmark 16/04/03 8:21PM -> approx. 10 sec simulation
# time for 200,000 vehicles_p at dt#
################################################################################
###

use strict;
require ".\\perl\\globvar1.ph";

### a big loop to run a set of predefined  scenarious
my @scenRatio = (1/3); # scenario leading veh trajectory filename extension
my @scenVehNum = (50); # scenario vehicle number
my @scenFlowRate = (500, 2000); # scenario simulation flow rate
my $totalScenarious  = ($#scenRatio+1)*($#scenVehNum+1)*($#scenFlowRate+1);
my $scenario = 0;
my @scenarioName = ();
my $recFileExtension;
$::TMax = 3600;
my $leadingVehTrajectoryFileName = "data\\trajectories\\lead_veh_3.txt";
print "time /".localtime()."(start )\n";
for (my $i_loop1=0; $i_loop1<=$#scenRatio; $i_loop1++) {                        # load four
different scaling ratio
        $::ratio = $scenRatio[$i_loop1];
        $::DTStepSize_m = 0.3;          #[sec] step increment for driver sensitivity, must be
multiple of $::dt_m/$::ratio
        $::DTMax_m = $::DT_p/$::ratio + 5*$::DTStepSize_m;      #[sec]
        $::DTMin_m = $::DT_p/$::ratio - 2*$::DTStepSize_m;      #[sec]
        $::alphaStepSize_m = 0.03;    #step increment for driver sensitivity
        $::alphaMax_m = $::alpha_p*$::ratio + 2*$::alphaStepSize_m;       #max threshold for
driver sensitivity
        $::alphaMin_m = $::alpha_p*$::ratio - 2*$::alphaStepSize_m;       #min threshold for
driver sensitivity
```

```
    $::spdMax_m = 75*5280/3600*$::ratio; #[ft/sec] - max speed constraint
    $::accMax_m = 15*$::ratio;   #[ft/sec2] - max. acceleration constraint
    $::accMin_m = -15*$::ratio;  #[ft/sec2] - max. deceleration constraint
    ##$::j_m = ($::DT_m/$::dt_m); #integer ratio DT/dt
    $::j_m = ($::DT_m/$::dt_m/$::ratio); #ratio DT/dt


    for (my $i_loop2=0; $i_loop2<=$#scenVehNum; $i_loop2++) {            # use three
different simulation periods
        $::vehMax_p = $scenVehNum[$i_loop2];
        $::vehMax_m = $::vehMax_p*$::ratio;
        for (my $i_loop3=0; $i_loop3<=$#scenFlowRate; $i_loop3++) {   # use six
different flow rates ( total 4*3*6 = 72 scenarious)
            $::headway = 3600/$scenFlowRate[$i_loop3];
            $::flow_p=$scenFlowRate[$i_loop3];



$::simulationTimeStamp = localtime();
$::simulationTimeStamp =~ s/:/_/g;

(@::k_m, @::k_p) = ();
my ($trajectories, $startRecTime, $endRecTime, $startRecVehRank, $endRecVehRank);
$recFileExtension =
sprintf("%.2f",$scenRatio[$i_loop1])."_".$scenVehNum[$i_loop2]."_".$scenFlowRate[$i_loop3
];#$::simulationTimeStamp;
if ($trajectories) {
    print "!!! Use the next options wisely, as recording vehicles trajectories is a time
consuming process !!!\n";
    $startRecTime = int(ask_for_it("enter starting recording time[sec]:", 100));
    $endRecTime = int(ask_for_it("enter ending recording time[sec]:", 200));
    $startRecVehRank = int(ask_for_it("enter first vehicle rank to be recorded [0 -> leading
vehicle]:", 0));
    $endRecVehRank = int(ask_for_it("enter last vehicle rank to be
recorded:[".$::vehMax_m." -> last vehicle in the model]", 0));
    open FH_p, ">runs\\rec_p_".$recFileExtension.".txt";
    print FH_p "time";
    for (my $k=$startRecVehRank; $k<=$endRecVehRank; $k++) {
        my $vid = ($k/$::ratio - 1); $vid = ($vid<0)?0:$vid;
        print FH_p join("_".$vid,(",X",",S",",A",""));
    }
    print FH_p "\n";
    for(my $i=0, $::DT_m=$::DTMin_m; $::DT_m<=$::DTMax_m+0.0000001; $::DT_m
+= $::DTStepSize_m, $i++) {
        for (my $j=0, $::alpha_m=$::alphaMin_m;
$::alpha_m<=$::alphaMax_m+0.0000001; $::alpha_m += $::alphaStepSize_m ,$j++) {
```

```
                        eval("open FH_m_".$i."_".$j.",
\">runs\\\\rec_m_".$j."_".$i."_".$recFileExtension.".txt\";");
                        eval("print FH_m_".$i."_".$j." \"DT =
".($::DTMin_m+$::DTStepSize_m*$i).", \";");
                        eval("print FH_m_".$i."_".$j." \"alpha =
".($::alphaMin_m+$::alphaStepSize_m*$j)."\\n\\ntime\";");
                        for (my $k=$startRecVehRank; $k<=$endRecVehRank; $k++) {

                                eval("print FH_m_".$i."_".$j."
join(\"_\".$k,(\",X\",\",S\",\",A\",\"\"));");
                        }
                        eval("print FH_m_".$i."_".$j." \"\\n\";");
                        #eval("close FH_m_".$i."_".$j.";");
                }
        }
}

my @vehicles_p = ();
my @vehicles_m = ();
my @lead_veh = ();
my @lead_veh_m = ();
my $endRunTime = $::TMax/$::dt_p;

        #all the vehicle-like arrays have 3xN dimension,
        #where each column represents position, speed and acceleration respectively
        #and represents the number of iterations the vehicle was updated
        #the lead vehicle is given (calculated in excel and exported in tab-delimeted format file)
        #@vehicles_p stores references to arrays like lead_vehicle

# main loop starts here
eval {
        # reads the file data for the lead vehicle and stores it into vehicles_p array
        &readLeadVeh(\@lead_veh, \@lead_veh_m, $leadingVehTrajectoryFileName);
        my (%traErrRMSE, %traErrMSE, %traErrAARE, %delay) = ();
        my (@delay_p, %delay_m, %delayAARE_m, %delayRMSE_m) = ();
        # creates simulation distinct table to store simulation results for every simulated vehicle
        #open FH_LOG, ">log.txt";
        my $k=0;
        for($::DT_m=$::DTMin_m; $::DT_m<=$::DTMax_m+0.0000001; $::DT_m +=
$::DTStepSize_m) {
                for ($::alpha_m=$::alphaMin_m; $::alpha_m<=$::alphaMax_m+0.0000001;
$::alpha_m += $::alphaStepSize_m) {
                        $traErrMSE{$::DT_m."-".$::alpha_m}=0;
                        $traErrRMSE{$::DT_m."-".$::alpha_m}=0;
                        $traErrAARE{$::DT_m."-".$::alpha_m}=0;
                        my @d_m = ();
                        $delay_m{$::DT_m."-".$::alpha_m}=\@d_m;
                        $scenarioName[$k++] = $::DT_m."-".$::alpha_m;
                }
        }
        for(my $i=0, $::DT_m=$::DTMin_m; $::DT_m<=$::DTMax_m+0.0000001; $::DT_m
+= $::DTStepSize_m, $i++) {
                for (my $j=0, $::alpha_m=$::alphaMin_m;
$::alpha_m<=$::alphaMax_m+0.0000001; $::alpha_m += $::alphaStepSize_m ,$j++) {
                        my @a = ();
                        $vehicles_m[$i][$j] = \@a;
                }
        }
        undef $k;
        for (my $t=0; $t<=$endRunTime; $t++) {#loop the simulation period at every update
interval

                &initLeadingVeh(\@lead_veh,\@vehicles_p,$t, $::j_p, $::dt_p, $::spdMax_p,
$::spdMin_p);
                #compute the following vehicles_p for moment $t = j*dt (prototype)
                for(my $n=1; $n<=$::vehMax_p; $n++) {#loop to compute each vehicle's
acceleration, speed and position
                        &computeFollowerVeh_GM1(\@vehicles_p, $n, $t, $::headway,
$::MinHeadway,
                                                        $::dt_p, $::j_p, $::alpha_p, $::spdMax_p,
$::spdMin_p, $::accMax_p,
                                                        $::accMin_p, 0,$scenario);
                }
                if ($trajectories && $t>=$startRecTime/$::dt_p && $t<=$endRecTime/$::dt_p) {
                        print FH_p $t*$::dt_p.",";
                        my $vehId = $startRecVehRank;
                        while ($vehId le (($endRecVehRank/$::ratio)-1)) {
                                my $vid = 3*($vehId/$::ratio - 1);
                                $vid = ($vid<0)?0:$vid;
                                print FH_p $vehicles_p[$#vehicles_p][$vid];
                                print FH_p $vehicles_p[$#vehicles_p][$vid+1];
                                print FH_p $vehicles_p[$#vehicles_p][$vid+2];
                                $vehId++;
                        }
                        print FH_p "\n";
                }
                my $k=0;
                for(my $i=0, $::DT_m=$::DTMin_m; $::DT_m<=$::DTMax_m+0.0000001;
$::DT_m += $::DTStepSize_m, $i++) {
                   $::j_m =$::DT_m/$::dt_m;
                        for (my $j=0, $::alpha_m=$::alphaMin_m;
$::alpha_m<=$::alphaMax_m+0.0000001; $::alpha_m += $::alphaStepSize_m, $j++) {
                                &initLeadingVeh(\@lead_veh_m,$vehicles_m[$i][$j],$t, $::j_m,
$::dt_m,
                                        $::spdMax_m, $::spdMin_m);

                    #compute the following vehicles_p for moment $t = j_m*dt
(model)
                    for(my $n=1; $n<=$::vehMax_m; $n++) {#loop to compute each
vehicle's acceleration, speed and position
                        &computeFollowerVeh_GM1($vehicles_m[$i][$j], $n, $t,
$::headway/$::ratio, $::MinHeadway/$::ratio,
                                $::dt_m, $::j_m, $::alpha_m, $::spdMax_m,
$::spdMin_m,
                                $::accMax_m, $::accMin_m, $k, $scenario);
                    }
                    $k++;
                    if ($t>=$::j_m) {
                        &computeErrors($vehicles_p[$::j_p],
$vehicles_m[$i][$j][$::j_m],
                                \$traErrMSE{$::DT_m."-".$::alpha_m},
\$traErrRMSE{$::DT_m."-".$::alpha_m}, \$traErrAARE{$::DT_m."-".$::alpha_m});
                    }
                    if ($trajectories && $t>=$startRecTime/$::dt_m &&
$t<=$endRecTime/$::dt_m) {
                        eval("print FH_m_".$i."_".$j." \"".($t*$::dt_m).",\";");
                        my $vehId = $startRecVehRank;
                        my $a = $vehicles_m[$i][$j];
                        while ($vehId<=$endRecVehRank) {
                            eval("print FH_m_".$i."_".$j."
\"".$vehicles_m[$i][$j][$#$a][3*$vehId]."\";");
                            eval("print FH_m_".$i."_".$j."
\"".$vehicles_m[$i][$j][$#$a][3*$vehId+1]."\";");
                            eval("print FH_m_".$i."_".$j."
\"".$vehicles_m[$i][$j][$#$a][3*$vehId+2]."\";");
                            $vehId++;
                        }
                        eval("print FH_m_".$i."_".$j."\"\\n\";");
                    }
                }
            }
            #compare the prototype and the models
            print "\r".sprintf("scenario %d of %d => progress
%.1f%",($scenario+1),$totalScenarious,$t/$endRunTime*100);
        } ##  end of the for loop with $t counter (99.9% of the script duration ends here)
        $scenario++;
        &computeFinalDelay($vehicles_p[$::j_p], \@delay_p, $::TMax, $::ratio, $::spdMax_p,
$::headway, $::vehMax_p);
eval{
        for(my $i=0, $::DT_m=$::DTMin_m; $::DT_m<=$::DTMax_m+0.0000001; $::DT_m
+= $::DTStepSize_m, $i++) {
      $::j_m =$::DT_m/$::dt_m;
        for (my $j=0, $::alpha_m=$::alphaMin_m;
$::alpha_m<=$::alphaMax_m+0.0000001; $::alpha_m += $::alphaStepSize_m ,$j++) {
            &computeFinalDelay($vehicles_m[$i][$j][$::j_m], $delay_m{$::DT_m."-
".$::alpha_m},
                                $::TMax, 1, $::spdMax_m,
$::headway/$::ratio, $::vehMax_m);
            my $aaaa=$delay_m{$::DT_m."-".$::alpha_m};
            &computeFinalDelayErrors(\@delay_p, $delay_m{$::DT_m."-
".$::alpha_m},
                \$delayAARE_m{$::DT_m."-".$::alpha_m},
\$delayRMSE_m{$::DT_m."-".$::alpha_m});
        }
    }
} || print $@;

    # process DELAY related display
    open FH_delay, ">runs\\scenarios\\delay_".$recFileExtension.".txt";
    print FH_delay "scaling ratio, $scenRatio[$i_loop1]\n";
    print FH_delay "no of vehicles in prototype, $scenVehNum[$i_loop2]\n";
    print FH_delay "simulation flow rate, $scenFlowRate[$i_loop3] [veh/hour]\n\n";
    print FH_delay "AVERAGE DELAY\n";
    print FH_delay "Prototype:
".sprintf("%.1f",(&sumArray(\@delay_p)/($#delay_p+1)))."\n";
    print FH_delay "Model[alpha\\dt]\n";

    for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
        print FH_delay "\t$i";
    }
    for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
        print FH_delay "\n".$i;
        for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
            my $size = $delay_m{$j."-".$i};
            print FH_delay "\t".sprintf("%.1f",sumArray($delay_m{$j."-
".$i})/($#$size+1));
        }
    }

    print FH_delay "\n\nAARE DELAY\n\n";
    print FH_delay "Model[alpha\\dt]\n";
    for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
        print FH_delay "\t$i";
    }
    for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
        print FH_delay "\n".$i;
```

```
                for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
                        my $size = $delay_m{$j."-".$i};
                        print FH_delay "\t".sprintf("%.3f",$delayAARE_m{$j."-".$i});
                }
        }

        print FH_delay "\n\nRMSE DELAY\n\n";
        print FH_delay "Model[alpha\\dt]\n";
        for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
                print FH_delay "\t$i";
        }
        for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
                print FH_delay "\n".$i;
                for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
                        my $size = $delay_m{$j."-".$i};
                        print FH_delay "\t".sprintf("%.3f",$delayRMSE_m{$j."-".$i});
                }
        }

        my @sum_delay_m = ();
        my $sum_delay_p = &sumArray(\@delay_p);
        print FH_delay "\n\nTOTAL DELAY\n";
        print FH_delay "Prototype: ".sprintf("%.1f",$sum_delay_p)."\n";
        print FH_delay "Model[alpha\\dt]\n";

        for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
                print FH_delay "\t$i";
        }
        for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
                print FH_delay "\n".$i;
                for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
                        $sum_delay_m[$#sum_delay_m+1] = sumArray($delay_m{$j."-".$i});
                        print FH_delay "\t".sprintf("%.1f",$sum_delay_m[$#sum_delay_m]);
                }
        }

        print FH_delay "\n\nARE TOTAL DELAY\n";
        print FH_delay "Model[alpha\\dt]\n";

        for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
                print FH_delay "\t$i";
        }
        my $ind_i=0;
        for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
```

```
        print FH_delay "\n".$i;
        for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
            print FH_delay "\t".sprintf("%.3f",abs(($sum_delay_p -
$sum_delay_m[$ind_i]/$::ratio)/$sum_delay_p));
            $ind_i++;
        }
    }
    undef $ind_i;
    close FH_delay;

    # process TRAJECTORY related display
    open FH, ">runs\\scenarios\\traject_err_".$recFileExtension.".txt" || die "Error opening
file: $!";
    print FH "scaling ratio, $scenRatio[$i_loop1]\n";
    print FH "no of vehicles in prototype, $scenVehNum[$i_loop2]\n";
    print FH "simulation flow rate, $scenFlowRate[$i_loop3] [veh/hour]\n\n";
    print FH "TRAJECTORIES MSE\n\nalpha\\delta_t";
    for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
        print FH "\t$i";
    }
    for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
        print FH "\n".$i;
        for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
            print FH "\t".sprintf("%.1f",$traErrMSE{$j."-".$i}/($::vehMax_m-
1)/($endRunTime));
        }
    }

    print FH "\n\nTRAJECTORIES RMSE\n\nalpha\\delta_t";
    for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
        print FH "\t$i";
    }
    for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
        print FH "\n".$i;
        for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
            print FH "\t".sprintf("%.1f",sqrt($traErrMSE{$j."-".$i}/($::vehMax_m-
1)/($endRunTime)));
        }
    }

    print FH "\n\nTRAJECTORIES AARE\n\nalpha\\delta_t";
    for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
        print FH "\t$i";
    }
    for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
        print FH "\n".$i;
        for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
            print FH "\t".sprintf("%.2e",$traErrAARE{$j."-".$i}/($::vehMax_m-
1)/($endRunTime));
        }
    }

    # process DEnSITY related display
    print FH "\n\nDENSITY RECORDS";
    my $timeSnapShots = $endRunTime/($::densityTimeInterval/$::dt_p);
    my $spaceSnapShots = $::maxDistance/$::densitySpaceInterval;

    print FH "\nPrototype Density Matrix\nSpace[miles]\\Time[min]";
    for (my $i=1; $i <=$timeSnapShots; $i++) {
        print FH "\t".($i*$::densityTimeInterval/60)."-min";
    }
    for (my $i=1; $i <=$spaceSnapShots; $i++) {
        print FH "\n".($i*$::densitySpaceInterval/5280);
        for (my $j=0; $j <$timeSnapShots; $j++) {
            print FH "\t".($::k_p[$scenario-1][$i-1][$j]+0);
        }
    }

    print FH "\nMicrocosm Density Matrices\n";
    my %min_k = ();
    for (my $k=0; $k<=$#scenarioName; $k++) {
        print FH "\ndelta_t\\alpha,$scenarioName[$k]\nSpace[miles]\\Time[min]";
        $min_k{$scenarioName[$k]} =  0;
        for (my $i=1; $i <=$timeSnapShots; $i++) {
            print FH "\t".($i*$::densityTimeInterval/60)."-min";
        }
        my $count = 0;
        for (my $i=1; $i <= $spaceSnapShots; $i++) {
            print FH "\n".($i*$::densitySpaceInterval*$::ratio/5280);
            for (my $j=0; $j <$timeSnapShots; $j++) {
                print FH "\t".( $::k_p[$scenario-1][$i-1][$j] >0
                    ? sprintf("%.3f",$::k_m[$scenario-1][$k][$i-
1][$j]/($::ratio*$::k_p[$scenario-1][$i-1][$j]))
                    : "0");
                $min_k{$scenarioName[$k]} += power(($::k_m[$scenario-
1][$k][$i-1][$j]/($::densitySpaceInterval*$::ratio/5280) - $::k_p[$scenario-1][$i-
1][$j]/($::densitySpaceInterval/5280)), 2);
                $count += (($::k_p[$scenario-1][$i-1][$j] >0) ? 1 : 0);
            }
        }
        $min_k{$scenarioName[$k]} /= $count;
    }

    print FH "\n\nDENSITY RMSE\nalpha\\delta_t";
    for (my $i= $::DTMin_m; $i<=$::DTMax_m+0.0000001; $i+=$::DTStepSize_m ){
        print FH "\t$i";
    }
    for(my $i = $::alphaMin_m; $i<=$::alphaMax_m+0.0000001; $i+=$::alphaStepSize_m){
        print FH "\n".$i;
        for (my $j= $::DTMin_m; $j<=$::DTMax_m+0.0000001; $j+=$::DTStepSize_m
){
            print FH "\t".sprintf("%.3f",sqrt($min_k{$j."-".$i}));
        }
    }

    for(my $i=0, $::DT_m=$::DTMin_m; $::DT_m<=$::DTMax_m+0.0000001; $::DT_m
+= $::DTStepSize_m, $i++) {
        for (my $j=0, $::alpha_m=$::alphaMin_m;
$::alpha_m<=$::alphaMax_m+0.0000001; $::alpha_m += $::alphaStepSize_m ,$j++) {
            eval("close FH_m_".$i."_".$j.";");
        }
    }
    ($trajectories)
        ? close FH_p
        : "";

    open FH, ">log\\allin1_log".$::simulationTimeStamp.".txt" || die "Error opening file: $!";

        print FH "#prototype related variables\n";
        print FH "dt_p = $::dt_p #vehicles updating interval\n";
        print FH "DT_p = $::DT_p;   #[sec] - reaction time\n";
        print FH "vehMax_p = $::vehMax_p;           #[veh] - N, nr. of vehicles released in
the prototype\n";
        print FH "alpha_p = $::alpha_p;                  #driver sensitivity\n";
        print FH "spdMax_p = $::spdMax_p; #[ft/sec] - max speed constraint\n";
        print FH "spdMin_p = $::spdMin_p; #[ft/sec] - min speed constraint\n";
        print FH "accMax_p = $::accMax_p; #[ft/sec2] - max. acceleration constraint\n";
        print FH "accMin_p = $::accMin_p;  #[ft/sec2] - max. deceleration constraint\n";
        print FH "flow_p = $::flow_p; #[veh/hour] - flow rate = $::> vehicle headway
when released in the system\n";
        print FH "dist_headway = $::dist_headway; #[ft] start-up distance headway
betwen vehicles\n\n";

        print FH "ratio = $::ratio;       #-scaling ratio model vs. prototype\n";
        print FH "headway = $::headway; #[sec] time headway derived from the flow \n";
        print FH "defaultSpeed = $::defaultSpeed; #[ft/sec] default speed value for the
vehicles entering the system, position = $:: 0\n\n";
```

```
        #model related variables
        print FH "dt_m = $::dt_m;    #[sec] vehicle updating time\n";
        print FH "DTMax_m = $::DTMax_m;        #[sec]\n";
        print FH "DTMin_m = $::DTMin_m;        #[sec]\n";
        print FH "DTStepSize_m = $::DTStepSize_m;      #[sec] step increment for
driver sensitivity\n\n";

        print FH "alphaMax_m = $::alphaMax_m;  #max threshold for driver
sensitivity\n";
        print FH "alphaMin_m = $::alphaMin_m;  #min threshold for driver
sensitivity\n";
        print FH "alphaStepSize_m = $::alphaStepSize_m; #step increment for driver
sensitivity\n\n";

        print FH "spdMax_m = $::spdMax_m; #[ft/sec] - max speed constraint\n";
        print FH "spdMin_m = $::spdMin_m; #[ft/sec] - min speed constraint\n";
        print FH "accMax_m = $::accMax_m;        #[ft/sec2] - max. acceleration
constraint\n";
        print FH "accMin_m = $::accMin_m;        #[ft/sec2] - max. deceleration
constraint\n\n";

        print FH "simulation duration = $::TMax [sec]\n";

    close FH;
    #close FH_LOG;
    #if( $#::errorList<0) {
    #       &closeDBConnection();
    #}
    #else {
    #       &rollbackTransaction(); # required by MSSQL ODBC driver behaviour
    #}
} || push (@::errorList, $@);




        }       # end of use six different flow rates ( total 4*3*6 = 72 scenarious)
    }           # end of use three different simulation periods
}               # end of load four different trajectories


print "\ntime /".localtime()."(end )\n";


_error:
if ($#::errorList >=0) {
    open STDERR, ">err\\simulation_".$::simulationTimeStamp."_err.txt";
    for (my $i=0; $i<=$#::errorList; $i++) {
        print STDOUT "$::errorList[$i]";
    }
    close STDERR;
    print "check the error file!";
}

# reads the trajectory of the leading veh from an initialization file
# and puts the vehicle data in the @lead_veh array

sub readLeadVeh {
    my ($veh_p, $veh_m, $file) = @_;
    #my (@veh_acc, @veh_spd, @veh_pos);

    open FH1, $file || die "Error opening file: $!";
    my $line = <FH1>; # skip first line, that has a header text
    while (defined($line = <FH1>) && $#$veh_p <= ($::TMax/$::dt_p)) {
        chop($line);
        my @a = split(/,/,$line);
        #max and min acc and speed initialization (no vehicle should exceed the speed
and accel of the leading vehicle??!)
        #$::accMin_p = ($::accMin_p>$a[0]) ? $a[0] : $::accMin_p;
        #$::accMax_p = ($::accMax_p<$a[0]) ? $a[0] : $::accMax_p;
        #$::spdMin_p = ($::spdMin_p>$a[1]) ? $a[1] : $::spdMin_p;
        #$::spdMax_p = ($::spdMax_p<$a[1]) ? $a[1] : $::spdMax_p;
        push (@$veh_p, \@a);
        my @b = @a;
        $b[2] = $a[2]*$::ratio;
        push (@$veh_m, \@b);
    }
#       print FH "accMin_p - $::accMax_p - $::spdMin_p - $::spdMax_p \n";
    close FH1;
}

#initializes the @vehicles_p array with the leading vehilce for each $dt interval
sub initLeadingVeh {
    my ($lead, $veh, $t, $j, $dt, $spdMax, $spdMin) = @_;
    if($t>0) {
        if (!defined($$lead[$t][2])) {
                $$lead[$t][2] = 0
        }
        my $spd_p = $$lead[$t-1][1] + $dt*($$lead[$t-1][2]+$$lead[$t][2])/2;
        $spd_p = ($spd_p < $spdMin)? $spdMin : ( ($spd_p > $spdMax) ? $spdMax :
$spd_p);
        my $pos_p = $$lead[$t-1][0] + $dt*($$lead[$t-1][1] + $spd_p)/2;
        $$lead[$t][0] = $pos_p;
        $$lead[$t][1] = $spd_p;
    }
    else {
        $$lead[$t][0] = 0;
        $$lead[$t][1] = $::defaultSpeed;
    }
    if ($t<=$j) {
        $$veh[$t][0] = $$lead[$t][0].",";
        $$veh[$t][1] = $$lead[$t][1].",";
        $$veh[$t][2] = $$lead[$t][2].",";
    }
    else {
        shift(@$veh);
        $$veh[$j][0] = $$lead[$t][0].",";
        $$veh[$j][1] = $$lead[$t][1].",";
        $$veh[$j][2] = $$lead[$t][2].",";
    }
}


#calculates the vehicle accel, spd and pos according with the specified car-following model
sub computeFollowerVeh_GM1 {
    my ($veh, $n, $t, $headway, $MinHeadway, $dt, $j, $alpha, $spdMax, $spdMin,
$accMax, $accMin, $config, $scenario) = @_;
    my ($acc_p, $spd_p, $pos_p) = (0,0,0);

    if ($t>$j) {
        # force position of the calculated vehicle to x=0 if the time headway is met
        # and the vehicle is not yet in the system (i.e. $pos_p is negative)
        if ($t*$dt <= $n*$headway) {
            $pos_p = 0;
            $spd_p = $::defaultSpeed;
            $acc_p = 0;
        }
        else {
            my $sl_0 = $veh->[0][3*($n-1)+1]; chop($sl_0);
            my $sf_0 = $veh->[0][3*$n+1]; chop($sf_0);
            $acc_p = $alpha*($sl_0 - $sf_0);

            my $af_t = $veh->[$j-1][3*$n+2]; chop($af_t);
            my $sf_t = $veh->[$j-1][3*$n+1]; chop($sf_t);

            # extra constraint, not necesarily needed
            if ($sf_t == $spdMax && $acc_p > 0) {
                $acc_p = 0;
            }
            $spd_p =  $sf_t + $dt*($af_t + $acc_p)/2;
            my $pf_t = $veh->[$j-1][3*$n]; chop($pf_t);
            $pos_p =  $pf_t + 0.5*($sf_t+$spd_p)*$dt;

            #check headway spacing constraint, if is not met
            # adjust speed to get the minimum headway
            my $pos_lv = $$veh[$j][3*($n-1)]; chop($pos_lv);
            my $minHeadway = $::dist_headway + $MinHeadway*$spd_p;
            if (0 && $pos_lv < $pos_p + $minHeadway) {
                $pos_p =  $pos_lv - $minHeadway;
                if ($pos_p < $pf_t) { # the vehicle backs-up, not good!! set it to
previous position
                    $pos_p = $pf_t;
                }
                $spd_p = ($pos_p - $pf_t)/$dt;
                $acc_p = 0;
            }
            #could be redundant check for GM1
            $pos_p = $pos_p < 0 ? 0 : $pos_p;

            #could be redundant check for GM1
            $spd_p = ($spd_p < $spdMin) ? $spdMin : (($spd_p > $spdMax) ?
$spdMax: $spd_p);

            #could be redundant check for GM1
            $acc_p = ($acc_p < $accMin) ? $accMin : (($acc_p > $accMax) ?
$accMax : $acc_p);
        }

        #shift(@$veh);
        $$veh[$j][3*$n] = $pos_p.",";
        $$veh[$j][3*$n+1] = $spd_p.",";
        $$veh[$j][3*$n+2] = $acc_p.",";

        #check for a snap-shot density
        if ($t%($::densityTimeInterval/$dt) == 0) {
            if ($headway == $::headway) {
                &computeDensityP($pos_p, ($t/($::densityTimeInterval/$dt)),
$scenario);
            }
            else {
                &computeDensityM($pos_p, ($t/($::densityTimeInterval/$dt)),
$config, $scenario);
            }
        }
    }
    else {
        # force position of the calculated vehicle to x=0 if the time headway is met
        # and the vehicle is not yet in the system (i.e. $pos_p is negative)
        if ($t*$dt <= $n*$headway) {
            $pos_p = 0;
            $spd_p = $::defaultSpeed;
            $acc_p = 0;
        }
        $$veh[$t][3*$n] = $pos_p.",";
        $$veh[$t][3*$n+1] = $spd_p.",";
        $$veh[$t][3*$n+2] = $acc_p.",";
    }
}


sub computeErrors {
    my ($vehicles_p, $vehicles_m, $traErrMSE, $traErrRMSE, $traErrAARE) = @_;
    #my ($traErrMSE, $traErrRMSE) = 0;
    my $imax = $::vehMax_m;
    for (my $i=0; $i<=$imax; $i++) {
        my $m_ = $$vehicles_m[3*$i]; chop($m_);
        my $p_ = $$vehicles_p[(3/$::ratio)*$i]; chop($p_);
        $$traErrMSE += ($m_/$::ratio-$p_)*($m_/$::ratio-$p_);
        $$traErrRMSE += abs($m_/$::ratio-$p_);
        if ($p_>0.00001) {
            $$traErrAARE += abs($m_/$::ratio-$p_)/$p_;
        }
    }
}

sub computeFinalDelay {
    my ($veh, $delay, $t, $ratio, $spdMax, $headway, $vehMax) = @_;
    for (my $n=0; ($t-$n*$headway > 0) && ($n<=$vehMax); $n++) {
        my $position=$$veh[3*$n];chop($position);
        $$delay[$n] = $t-$position/$spdMax-$n*$headway;
    }
}

sub computeFinalDelayErrors {
    my ($delay_p, $delay_m, $delayAARE_m, $delayRMSE_m) = @_;
    my $aare = abs($$delay_p[0] - $$delay_m[0])/$$delay_p[0];
    my $rmse = ($$delay_p[0] - $$delay_m[0])*($$delay_p[0] - $$delay_m[0]);
    for (my $i=0;$i<$#$delay_m;$i++) {
        my $ind_p = $i/$::ratio + 1;
        $aare +=abs($$delay_p[$ind_p] - $$delay_m[$i+1])/$$delay_p[$ind_p];
        $rmse +=($$delay_p[$ind_p] - $$delay_m[$i+1])*($$delay_p[$ind_p] -
$$delay_m[$i+1]);
    }
```

```
        if (!($#$delay_m+1)) {print ($#$delay_m);exit;}
        $$delayAARE_m = $aare/($#$delay_m+1);
        $$delayRMSE_m = $rmse/($#$delay_m+1);
        $$delayRMSE_m = sqrt($rmse);
}

sub computeDensityP {
        my ($position, $t, $scenario) = @_;
        for (my $i = 0; $i<=$::maxDistance; $i += $::densitySpaceInterval) {
                if ($position > $i && $position <= $i+$::densitySpaceInterval) {
                        $::k_p[$scenario][($i/$::densitySpaceInterval)][$t-1] += 1;
                }
        }
}

sub computeDensityM {
        my ($position, $t, $config, $scenario) =  @_;
        for (my $i = 0; $i<=$::maxDistance; $i += $::densitySpaceInterval*$::ratio) {
                if ($position > $i && $position <= $i+$::densitySpaceInterval*$::ratio) {
                        $::k_m[$scenario][$config][($i/($::densitySpaceInterval*$::ratio))][$t-1]
+= 1;
                }
        }
}

sub computeDelayOld {
        my ($veh, $delay, $t, $ratio, $spdMax, $headway, $vehMax) = @_;
        my $position = $$veh[0]; chop($position);
        $$delay[0] += $t-$position/$spdMax;
        my $vin = 1;
        for (my $n=1; $n<=$vehMax; $n+=(1/$ratio)) {
                if (!defined($$delay[$vin])) {$$delay[$vin] = 0;}
                $position=$$veh[3*$n];chop($position);
                $$delay[$vin++] += $t-$position/$spdMax-$n*$headway;
        }
}

sub sumArray {
        my ($a) = @_;
        my $sum = 0;
        for (my $i=0; $i<=$#$a;$i++) {
                $sum += $$a[$i];
        }
        return $sum;
}
```

```
sub ask_for_it {
   my $m = shift || "Enter value";
   my $def = shift;
   $def = "" unless defined $def;
   print "$m \[$def\] ";
   my $v = <STDIN>;
   chomp $v;
   return ($v =~ /^\s*$/ ? $def : $v);
}

sub yes_or_no {
   my $m = shift || "Which?";
   print "$m [n] ";
   return 1 if scalar(<STDIN>) =~ /^y$/i;
}

sub dumpRows2File {
        my ($rows, $t) = @_;
        for (my $j=0; $j<=$#$rows; $j++) {
                my $a = $$rows[$j];
                print FH @$a;
                print FH "\n";
        }
}

sub power {
        my ($n, $p) = @_;
        my $ret = 1;
        for (my $i=1;$i<=$p; $i++) {
                $ret *= $n;
        }
        return $ret;
}
```

VITA

Srikanth Chakravarthy was born in Kazipet, India in the year 1980. He completed his schooling from Sainik School Korukonda; an institution known for its discipline and academic standing in India. He went on to obtain his Bachelor's degree in Civil Engineering from Kakatiya University, India in the Spring of 2002. To fulfill his passion for research, he joined the Master's program in Transportation Engineering at Louisiana State University in August 2002 and expects to receive his degree in December 2003. His future plans are to pursue a doctoral program in Transportation Engineering.