

THE CATHOLIC UNIVERSITY OF AMERICA

Highway Traffic Modeling

A DISSERTATION

Submitted to the Faculty of the
Department of Civil Engineering

School of Engineering

Of The Catholic University of America

In Partial Fulfillment of the Requirements

For the Degree

Doctor of Philosophy

©

Copyright

All Rights Reserved

By

Jun Yang

Washington, D.C.

2010

Highway Traffic Modeling

Jun Yang, Ph.D.

Director: Lu Sun, Ph.D.

With the growth of the number of vehicles around the world, the amount of congestion, pollution, and accidents is increasing. To solve this problem, highway traffic modeling, as one of the key components in traffic management, is becoming more important.

In this dissertation, a methodological framework is first developed to deal with traffic-stream modeling based on data mining, steepest-ascend algorithm, and genetic algorithm. The new method is adaptive in nature and has greater flexibility and generality compared with existing methods.

Secondly, a new method is developed to estimate and predict macroscopic traffic conditions in the area where no existing traffic information is available. The new method is based on shock wave theory. Unlike widely used data-driven methods, the proposed method has a clear traffic explanation and gives an accurate estimate and prediction of traffic flow.

Based on the estimation and prediction of traffic conditions, travel time is the next information that needs to be estimated and predicted in traffic management. A piecewise truncated quadratic trajectory is proposed here to mimic the unknown speed trajectory between point detectors. The basis functions of the new method consist of quadratic and

constant functions of time. Using the actual travel time obtained from field experiments, the new method yields a more accurate travel time estimate than other trajectory-based methods.

Finally, for the microscopic level of traffic modeling, a new car-following model is proposed to solve problems in the application of existing Gipps car-following models. Gipps car-following models are based on the assumption that in car following behavior, drivers always attempt to get the maximum speed that is safe to prevent rear-end collisions in the event of an emergency stop. Since this assumption may not always be true during driving, it causes imaginary numbers due to the square root function in the model. This study introduces a new model without square root by using a nonlinear braking rate that was not adopted in the existing car-following models.

This dissertation by Jun Yang fulfills the dissertation requirement for the doctoral degree in civil engineering approved by Lu Sun, Ph.D., as Director, and by Hsien Ping Pao, Ph.D., and by Gunnar Lucko, Ph.D., as Readers.

Lu Sun, Ph.D., Director

Hsien P. Pao, Ph.D., Reader

Gunnar Lucko, Ph.D., Reader

TABLE OF CONTENTS

List of Figures	vi
List of Tables	viii
Acknowledgements	ix
1. Introduction	1
2. Literature Review	9
2.1 Traffic-stream Models	9
2.2 Shock Wave Based Traffic Modeling and Prediction	13
2.3 Travel Time Estimation	14
2.4 Car-following Models	19
3. Methodology	31
3.1 Traffic-stream Models	31
3.2 Shock Wave Based Traffic Modeling and Prediction	39
3.3 Travel Time Estimation	42
3.3.1 Notations	42
3.3.2 Quadratic Basis Functions	44
3.3.3 Constant Basis Functions	46

3.3.4 Combination of Quadratic and Constant Basis Functions49
3.4 Improved Gipps Car-following Model53
4. Data Preparation58
4.1 Traffic-stream Models and Shock Wave Based Traffic Modeling and Prediction58
4.2 Travel Time Estimation61
4.3 Improved Gipps Car-following Model63
5. Experimental Results67
5.1 Traffic-Stream Model67
5.2 Shock Wave Based Traffic Flow Modeling and Prediction86
5.3 Travel Time Estimation Based on Piecewise Truncated Quadratic Speed Trajectory94
5.4 Improved Gipps Car-following Model101
6. Conclusions and Future Research109
Appendix I Gradients for the Basic Functions in Traffic-Stream Models113
Appendix II Three-detector Shock Wave Simulation VB Program115
Appendix III Two-detector Shock Wave Simulation VB Program135
Appendix IV Travel Time Estimation VB Program151

Bibliography171

LIST OF FIGURES

Figure 2-1 Hypothetical speed trajectories between adjacent detectors	17
Figure 3-1 Single-regime traffic-stream models	35
Figure 3-2 Generation of initial shock waves	40
Figure 3-3 Generation of new shock waves	40
Figure 3-4 Shock Wave X meet Shock Wave Y	41
Figure 3-5 Travel distance conditions for adjacent detectors	47
Figure 4-1 Raw data of speed against density	60
Figure 4-2 A video frame taken from travel time experiment	62
Figure 5-1 Mating and crossover as well as mutation in GA	74
Figure 5-2 Two-regime traffic-stream models without continuity	77
Figure 5-3 Two-regime traffic-stream models with continuity	81
Figure 5-4 Speed estimation result	87
Figure 5-5 Traffic volume estimation result	88
Figure 5-6 Density estimation result	89
Figure 5-7 Comparison of shock wave simulation and linear interpolation	90
Figure 5-8 Two minutes speed prediction result	91
Figure 5-9 Ten minutes speed prediction result	92
Figure 5-10 Sixty minutes speed prediction result	92
Figure 5-11 Congestion prediction	93
Figure 5-12 Comparison of real travel time and estimated travel time using four	

methods95
Figure 5-13 Relative errors of travel time estimated using linear and truncated quadratic speed trajectories96
Figure 5-14 Performance examples of the Gipps category car-following models104

LIST OF TABLES

Table 2-1 Estimation results of the car-following model proposed by Gazis et al. (1959)	22
Table 3-1 Scenarios of truncated quadratic speed trajectory	52
Table 4-1 NGSIM trajectory data dictionary (FHWA)	64
Table 5-1 Single-regime traffic-stream models	76
Table 5-2 Mean residual squares of two-regime traffic-stream models	85
Table 5-3 Comparison between estimated travel times and real travel times	98
Table 5-4 t-test: paired two sample for means	99
Table 5-5 Average estimation results of Gipps category car-following models	103
Table 5-6 Simulation results of the Gipps category car-following models	108

ACKNOWLEDGEMENTS

I wish to thank my family. I could not have finished this dissertation without the support of my parents, my wife, my brother, and other family members.

I would like to thank my adviser, Dr Sun, other committee members, Dr Pao and Dr Lucko, and other faculty and staff. I appreciate the opportunity of studying and researching in the Catholic University of America (CUA), and the courses in the University of Maryland at College Park.

I also wish to extend my gratitude to other friends in CUA, specially Dr Luo and Dr Ardakani.

I am grateful to all the sponsors, specially the National Science Foundation (NSF), Federal Highway Administration (FHWA) and its Turner-Fairbank Highway Research Center, Dwight David Eisenhower Transportation Fellowship, and Next Generation SIMulation (NGSIM) program, Texas Department of Transportation (TxDOT) and its TransGuide system.

Chapter 1 Introduction

Roadway traffic management is becoming more important to society, and traffic modeling is the key role to traffic management. The more accurately traffic can be modeled, the more efficiently traffic management will be. The objective of this dissertation is to develop more advanced highway traffic modeling methods.

In general, highway traffic models can be categorized as macroscopic models and microscopic models. Macroscopic models deal with traffic characteristics like speed, flow, density, and travel time. Microscopic models, on the other hand, reveal driving behaviors (e.g. car following behaviors). Traffic flow is a system and its variables are related both in time and space, and it needs to be modeled on both the macroscopic level and microscopic level.

In macroscopic models, traffic flow, density, and space mean speed are three fundamental characteristics to describe the collective vehicle behavior (May 1990; Khisty and Lall 2003). Traffic flow can be described by an equation system consisting of a continuity equation $\partial q / \partial x + \partial k / \partial t = 0$, a state equation $q = ku$, and an equilibrium speed-density relation $u = u_e(k)$ (traffic-stream model), where, q , k , and u are flow, density, and space mean speed. Continuity and state equations reflect the fundamental physics of traffic dynamics, while the traffic-stream model specifies the characteristics of traffic flow at a particular site where the measurement is conducted.

The significance of traffic-stream models is not only for macroscopic traffic flow theory but also for microscopic traffic-flow theory. For instance, traffic-stream model is typically used in Dynamic Traffic Assignment (DTA) systems for simulating traffic conditions (Ben-Akiva et al. 2001; Mahmassani 2001). Furthermore, every microscopic car-following model corresponds to a traffic-stream model under steady state flow conditions (Gazis et al. 1959, 1961; May and Keller 1967; May 1990; Helbing and Treiber 1998; Zhang and Kim 2001; Sun and Zhou 2005a). It is possible that error structures may be related to microscopic driver behavior. Therefore, a better description of the traffic-stream model will be beneficial to capturing more precise driving behaviors, and will also help simulate and manage traffic.

Developing a traffic-stream model is equivalent to approximating speed as a function (often perturbed by noise) of density (the dependent variable). The goal is to model the dependence of a response variable, u , on one predictor variable, k , given realizations (data), $\{k_i, u_i\}_1^N$.

$$u = u_e(k) + \varepsilon. \quad [1-1]$$

The single-valued deterministic function, u_e , captures the predictive relationship of u on k . The additive stochastic component, ε , whose expected value is defined as zero, usually reflects the dependence of u on quantities other than k that are neither controlled nor observed. The aim of developing a traffic-stream model is to use the data to construct a function, $\hat{u}_e(k)$, that can serve as a reasonable approximation to $u_e(k)$ over the domain of interest.

The existing approaches for developing a traffic-stream model can be classified into parametric approaches and nonparametric approaches. The former pre-specifies a function form and uses it to fit the data, while the latter does not have a specific function form but uses the data by itself to fit the data. The benefit of parametric approaches is that the model is concise and can be interpreted easily, which is the price of reduced fitting to the actual data. The benefit of nonparametric approaches is that they typically fit the actual data very well. However, models using nonparametric approaches are very complex and difficult to interpret and use.

To overcome the disadvantages of both approaches, an adaptive-regression method that is based upon data mining is developed for traffic-stream modeling. It uses the concept of data mining, genetic algorithm, and gradient-based optimization algorithm to achieve high flexibility and automation in model estimation and selection. The new approach takes advantage of parametric and nonparametric estimation and yields excellent fitting while also being easy to interpret. A computer program is developed using Visual Basic programming language (VB) to implement this process with the collected macroscopic traffic dataset. Specifically, the advantages of the proposed method include: (1) knot positions and model parameters are estimated optimally and simultaneously using Genetic Algorithm (GA), and presetting of knot positions can be performed in terms of either density or speed; (2) the method is automatic and data driven, and it will always lead to the best fitting model from the basic functions using actual traffic data; and, finally, (3) the user has greater flexibility in specifying the degree-model continuity and defining and adding new basis functions that are parsimonious and better fit the traffic data.

Although a traffic-stream model helps to estimate the unknown macroscopic information at the locations where observed data is available, traffic estimation at other locations and traffic prediction cannot be solved by traffic-stream model alone. Highway traffic flow is highly nonlinear and varies with different times and locations and, therefore, another method is required to estimate traffic information where no observed data is available and also to predict traffic information. Although traffic status changes constantly, and differs from place to place, it is timely and spatially related. Traffic in one place can be studied with the consideration of traffic in the upstream or downstream. In prediction, this dissertation mainly focuses on short-term prediction, which is no more than half an hour. Long-term prediction, which involves months or years, uses quite different methods.

In this dissertation, shock wave simulation is used to solve the above problems. Shock waves have been studied with emphasis on theoretical and mathematical analyses, without applicability to traffic flow prediction. Using a hydrodynamic analogy (Lighthill and Whitham 1955), a shock wave is said to exist whenever traffic stream of varying stream conditions meet. Labeling the two conditions as, a and b , in the direction of traffic movement, the magnitude and direction of the speed of the shock wave between the two conditions is given by the following equation:

$$u_{sw} = \frac{q_b - q_a}{k_b - k_a}, \quad [1-2]$$

where u_{sw} is the speed of the shock wave. If the sign of the shock wave speed is positive, the shock wave is traveling in the direction of traffic flow; if it is negative, the shock wave moves in the upstream direction. The estimation and prediction effects are then compared to

the effects using other short-term traffic estimation and prediction methods, such as Autoregressive Integrated Moving Average (ARIMA).

With the basic traffic information estimated, travel time is another essential parameter for evaluating operational efficiency of transportation networks, for assessing the performance of traffic management strategies, and for developing real-time vehicle route guidance systems (Dailey 1993; Nam and Drew 1996; Petty et al. 1998). Travel time is defined as the time duration that a vehicle spends in traversing from an origin to a destination (O-D). The travel time between two adjacent detectors is termed "link travel time", while travel time from the origin to the destination between which there are multiple detectors is referred to as "corridor travel time". Clearly, link travel time is the building block for corridor travel time. The incorporation of travel time in Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) provides traffic management an important quantitative performance measure for highway transportation networks. Reliable travel time estimation and prediction over a road network provides useful information to travelers for rerouting and congestion mitigation.

In this dissertation, a trajectory-based method is presented for travel time estimation. It uses piecewise truncated quadratic sample paths of speeds measured at detectors to mimic actual (unknown) vehicle speed trajectories. It was found that travel time estimation using different approaches is similar during free-flow conditions but significantly different during transition flow and congestion conditions. Therefore, this dissertation does not focus on the free-flow condition. Suppose that detectors are located at two ends of a link. Let $(\mathbf{x}_o, \mathbf{x}_d)$ be the spatial location of an O-D pair. The speed of a vehicle can be measured at detectors.

Denote, t , the time of departure of the vehicle from its origin. Denote, $AT(t; \mathbf{x}_O, \mathbf{x}_D)$, the arrival time of the vehicle at destination, D . Denote, $TT(t; \mathbf{x}_O, \mathbf{x}_D)$, the time-dependent travel time with departure time, t . Denote, Δx , the length of the link and, $V(t')$, the vehicle speed trajectory. Unless otherwise necessary, in the following notations, the O-D pair will no longer be explicitly presented and arrival time, $AT(t; \mathbf{x}_O, \mathbf{x}_D)$ and travel time $TT(t; \mathbf{x}_O, \mathbf{x}_D)$, will be abbreviated as $AT(t)$ and $TT(t)$, respectively. At any moment, t' , where $t \leq t' \leq AT(t)$, the relationship among vehicle speed, $V(t')$, arrival time, $AT(t)$ and the traversed distance, Δx , is given by equation 1-3. Clearly arrival time, $AT(t)$, depends upon the time of departure, t , the distance, Δx , and vehicle speed, $V(t')$ such that the following equation can be specified:

$$\Delta x = \int_t^{AT(t)} V(t') dt'. \quad [1-3]$$

The constant functions, corresponding to upper and lower speed bounds, are determined using the maximum likelihood estimates of highest and lowest speeds that have been historically observed within a time interval. The purpose of setting a lower (upper) speed bound for simulating vehicle speed trajectory is to mimic a low (high) average speed during transition flow and congestion, and to restrict a quadratic speed trajectory to be within a realistic speed range, respectively.

Computational implementation of the new trajectory method is tractable and can be done very efficiently, making it suitable for online real-time travel time estimation.

Another category of traffic models is microscopic models. These models offer significant potential for analyzing highway traffic, the operation of the Intelligent Transportation Systems (ITS), and developing and testing traffic management systems and strategies. Other application areas, such as safety studies and capacity analysis, can also be applied with traffic simulation, in which aggregate traffic flow characteristics can be derived from the behavior of individual drivers.

If not in free-flow conditions, on most occasions, drivers follow the vehicles in front of them. Therefore, a car-following model is one of the key components of microscopic traffic modeling and simulation. There are many types of car-following models. This dissertation mainly focuses on Gipps category car-following models. This category of model assumes that in car-following behavior, the velocity that a driver tries to achieve is the maximum velocity that can avoid rear-end collision in emergency stops. Suppose the leading vehicle suddenly brakes at the rate of b_{n-1} (assuming $b_{n-1} < 0$). The driver of the subject vehicle will start to stop the vehicle after the reaction time, τ_n . If the leading vehicle's position is $x_{n-1}(t)$ at time t , the stopping/braking distance of the leading vehicle may be defined as:

$$x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}}, \quad [1-4]$$

and the stopping/braking distance of the subject vehicle is given by:

$$x_n(t) + \frac{[v_n(t) + v_n(t + \tau_n)]\tau_n}{2} - \frac{v_n(t + \tau_n)^2}{2b_n}, \quad [1-5]$$

where $x_n(t)$ is the subject vehicle's position at time t .

To avoid rear-end collision, the subject vehicle must stop behind the rear bumper of the leading vehicle, which can be displayed as follows:

$$x_n(t) + \frac{[v_n(t) + v_n(t + \tau_n)]\tau_n - v_n(t + \tau_n)^2}{2} \leq x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1}, \quad [1-6]$$

where l_{n-1} is the length of the leading vehicle. By solving the inequality, the car-following model can be written as:

$$v_n(t) = \frac{1}{2}\tau_n b_n + \sqrt{\frac{1}{4}\tau_n^2 b_n^2 + v_n(t - \tau_n)\tau_n b_n + \frac{v_{n-1}(t - \tau_n)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t - \tau_n)}. \quad [1-7]$$

If the driver wants a buffer distance, d_n , and a relaxing time, θ_n , the model can be written as follows:

$$v_n(t) = b_n \left(\frac{\tau_n}{2} + \theta_n \right) + \left[\frac{1}{4}(\tau_n + 2\theta_n)^2 b_n^2 + v_n(t - \tau_n)\tau_n b_n + \frac{v_{n-1}(t - \tau_n)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} + 2b_n d_n - 2b_n \Delta x_{n-1,n}(t - \tau_n) \right]^{\frac{1}{2}}. \quad [1-8]$$

However, drivers do not always follow the assumption of the Gipps category car-following model and, in some instances, it causes imaginary numbers from the model. To overcome this problem, the square root function in the model can be replaced by other functions. In this regard, a cube root function is finally adopted.

Chapter 2 Literature Review

This chapter presents a review of related research. It includes the existing models, data used, and the estimation results.

2.1 Traffic-stream Models

The earlier speed-density relations assumed a single regime phenomenon over the complete range of flow conditions. The first single regime model was developed by Greenshields (1935), which assumes a linear relation between density and speed:

$$u_e = u_f (1 - k / k_{jam}), \quad [2-1]$$

where the constant, u_f is free-flow speed and k_{jam} is the jam density.

Greenberg (1959) proposed the following model:

$$u_e = u_o \ln(k_{jam} / k), \quad [2-2]$$

where the constant, u_o is the optimum speed.

Greenberg's model leads to the conclusion that free-flow speed is infinite, which is unrealistic. To overcome this defect, Underwood proposed the following model (May 1990):

$$u_e = u_f \exp(-k / k_0), \quad [2-3]$$

where k_0 is the optimal traffic density. Taking similar functional form, Drake et al. (1967) developed another single regime model, which uses an S-shape curve:

$$u_e = u_f \exp\left[-\frac{1}{2(k/k_0)^2}\right]. \quad [2-4]$$

Under the assumption of identical vehicles and spacing, a generalized, steady-state, traffic-stream model can be derived from the generalized car-following model:

$$u_e = u_0 [1 - (k/k_{jam})^{\beta-1}]^{1/(1-\alpha)}, \quad [2-5]$$

where, α and β are the parameters appearing in the car-following models (Aycin and Benekohal 2001; Helbing 2001). Single regime models that fit into this generalized model include the Pipes model (Pipes 1967), and the Drew model (Drew 1968).

Van Aerde (1995) as well as Van Aerde and Rakha (1995) combined the Pipes model and Greenshields model into a single-regime model as follows:

$$k = \left(c_1 + \frac{c_2}{u_f - u_e} + c_3 u_e\right)^{-1}, \quad [2-6]$$

where, c_1 , c_2 , and c_3 , are the parameters. The van Aerde model fits traffic data better than other single-regime models because it includes additional degrees of freedom for different conditions and facility types. The calibration of the van Aerde model is indirect and proceeds in two steps (Rakha and Crowther 2003). First, u_f , speed at capacity, capacity, and jam density are estimated in an ad hoc manner from a traffic dataset. Then, c_1 , c_2 , c_3 , and

u_e are calculated from these estimated traffic characteristics. Since ad hoc estimates of free-flow speed, speed at capacity, capacity, and jam density are very sensitive to the dataset, this two-step estimation may result in less accurate estimates of c_1 , c_2 , c_3 , and u_e . A better method would be able to estimate these four parameters simultaneously from the dataset.

Although single-regime traffic-stream models have received considerable success, it is generally acknowledged that roadways accommodate different patterns of vehicular traffic (Edie 1961; May 1990; Kockelman 2001). These distinct patterns gave rise to the development of multi-regime traffic-stream models, which attempted to describe speed-density relationships in different regimes (e.g., free-flow regime, congested regime), using different function forms, so that a better fitting of observation points can be achieved. The idea of a two-regime model was first proposed by Edie (1961), in which the Underwood model was used for the free-flow regime and the Greenberg model for the congested-flow regime:

$$u_e = \begin{cases} 54.9 \exp(-k / 163.9) & \text{for } k \leq 50 \\ 26.8 \ln(162.5 / k) & \text{for } k \geq 50 \end{cases} \quad [2-7]$$

Drake et al. (1967) proposed several multi-regime models using maximum likelihood estimation based on Quandt's work (1958, 1960). The first two-regime model proposed by

Drake et al. (1967) makes use of the Greenshields-type linear model for both the free-flow regime and the congested regime:

$$u_e = \begin{cases} 60.9 - 0.525k & \text{for } k \leq 65 \\ 40 - 0.265k & \text{for } k \geq 65 \end{cases} \quad [2-8]$$

The second two-regime model proposed by Drake et al. (1967) suggests a constant speed for the free-flow regime and a Greenberg model for the congested-flow regime:

$$u_e = \begin{cases} 48 & \text{for } k \leq 35 \\ 32 \ln(145.5/k) & \text{for } k \geq 35 \end{cases} \quad [2-9]$$

The three-regime model developed by Drake et al. (1967) takes a linear form for all three regimes as follows:

$$u_e = \begin{cases} 50 - 0.098k & \text{for } k \leq 40 \\ 81.4 - 0.913k & \text{for } 40 \leq k \leq 65. \\ 40 - 0.265k & \text{for } k \geq 65 \end{cases} \quad [2-10]$$

Multi-regime traffic-stream models provide a considerable improvement over single regime counterparts because of their flexibility in capturing different patterns of traffic flow across different regimes. However, there is a primary difficulty in developing multi-regime models, that is, the determination of breakpoints between regimes. Traditionally, the breakpoints are specified as exogenously based upon subjective judgment of the model developer (May 1990; Kockelman 2001). Such a treatment is empirical and heavily reliant on modelers' engineering experience. Kockelman (2001) developed a two-regime model in

which the membership of an observation belonging to a free-flow regime or congested regime is determined by a binary, logit-type, discrete choice model; an endogenous segmentation approach similar to the one suggested by Maddala (1983) and Bhat (1997). Sun and Zhou (2005a, 2005b) proposed using cluster analysis for specifying the breakpoints before estimating model parameters.

2.2 Shock Wave Based Traffic Flow Modeling and Prediction

One method to estimate and predict traffic is using statistics, such as parametric or non-parametric methods. With statistical methods, people try to uncover the rules underlying traffic data. This kind of method includes historical averages and local regressions. However, the most widely used one is univariate linear time series analysis, such as Auto-Regressive Integrated Moving Average (ARIMA) model. With ARIMA model, traffic data in the next time interval is connected with the data in this time period or even with data in a previous time period. Equation 2-11 gives a representation of an $ARIMA(p, d, q)$ model:

$$(1 - \phi_1 B - \dots - \phi_p B^p) \left((1 - B)^d Y_t - \mu \right) = (1 - \theta_1 B - \dots - \theta_q B^q) \varepsilon_t, \quad [2-11]$$

where, ε_t is white noise, Y_t is the time series of interest (e.g., speed, flow), μ is mean of the differenced process, B is the backward operator, θ_i and ϕ_j are parameters to be estimated from the data, and d , p , and q are respectively orders of differencing, auto-regression, and moving average. The existing methods are mostly data-driven. They

lack the ability to explain the traffic condition in the locations without traffic detector facilities and short-term traffic prediction.

This study presents a traffic estimation and prediction by simulating traffic shock waves. The traffic shock wave concept was adopted decades ago and the mathematic analysis has been performed by many researchers, such as Richard (1956). However, research on shock wave phenomena were limited to theoretical and mathematical analyses. Although the theoretical background of shock waves is thoroughly developed, there is still a need to develop a procedure for applying shock wave analysis (Abbas and Bullock 2003). Furthermore, because of the difficulty of data collection, very few studies have tracked several vehicles simultaneously over a long distance. A remarkable study is performed by Coifman (1996), who used a video recorder and manually tracked thirteen shock waves in California. In the application field, Abbas and Bullock (2003) developed an online analysis of shock waves at the downstream intersection to estimate a traffic platoon.

2.3 Travel Time Estimation

Travel time can be obtained directly and indirectly. Direct methods include instrumented vehicles, computerized and video license plate matching, point detection devices, radio navigation, automatic vehicle identification and cellular phone tracking. Indirect methods estimated from traffic data provided by point detection technologies, include inductive loop detectors and video cameras (Dailey 1993; Srinivasan and Jovanis 1996; Zhang and Kwon

1997; Oh. et al. 2003). While many of these methods are costly, deriving travel time based on point detection data is cost effective because many point loop detectors have been widely installed in existing highways due to the deployment of Intelligent Transportation Systems (ITS) over the past decade.

Both single and double loop detectors can be used for traffic data collection (e.g., occupancy, volume, speed). Only double loop detectors can provide accurate measurement of the speed of individual vehicles. Traffic speed derived from single loop detectors needs to assume an effective (but unknown) vehicle length (Hall and Persaud 1989; May 1990; Dailey 1999; Oh et al. 2003), and, therefore, it is just an approximation. In practice, reporting aggregated traffic information (e.g., speed, volume and occupancy) to form a longer interval summary (e.g., every 20 and 30 seconds, or 5 minutes) is more typically used.

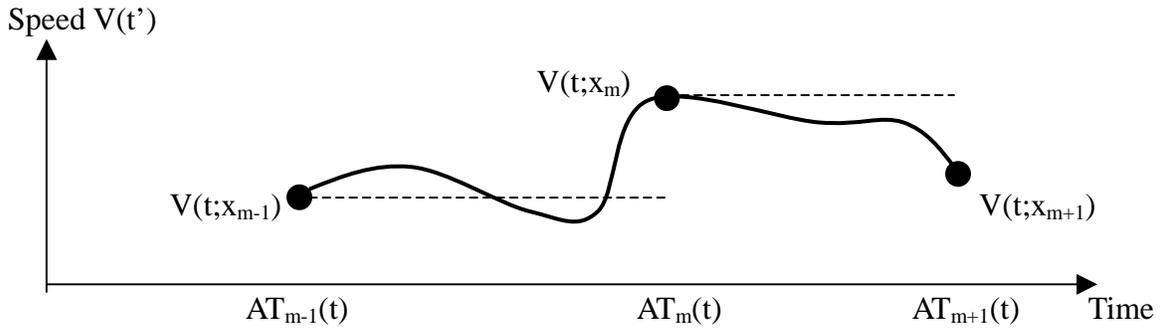
A number of methods for travel time estimation using speeds measured at detectors have been proposed in the literature (Nam and Drew 1996; Petty et al. 1998; Oh et al. 2002; van Lint and van der Zijpp 2003). Based on space mean speed, Oh et al. (2003) suggested the following definition for link travel time estimation:

$$T = \Delta x \sum_{n=1}^N \{ \min(t + \Delta t, t_n^{down}) - \max(t, t_n^{up}) \} / \sum_{n=1}^N \{ \min[x_n(t), x_n^{down}] - \max[x_n(t), x_n^{up}] \}, \quad [2-12]$$

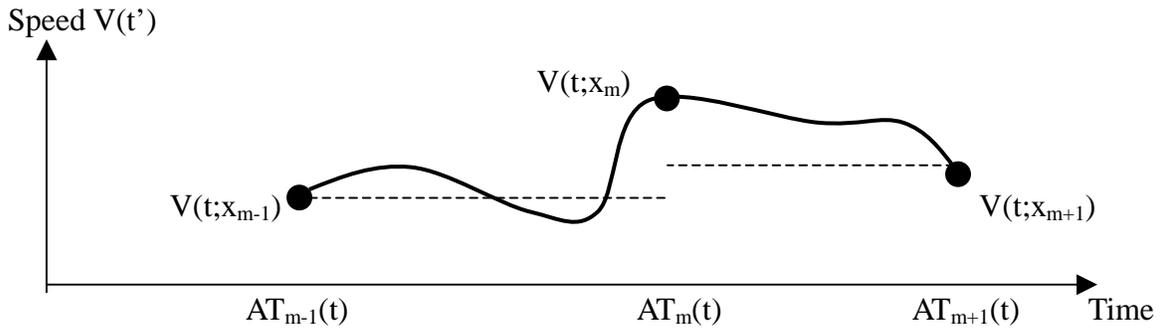
where, x_n^{down} and x_n^{up} are positions of the vehicle, n , at the downstream and upstream stations, respectively; t_n^{down} and t_n^{up} are times when the vehicle, n , passes the downstream

and upstream stations, respectively; N is the number of vehicles traversing the section during the time interval, Δt . Oh, et al. (2003) further suggests using estimated link density to improve the accuracy of travel time estimation based on equation 2-12. The link density is estimated from the subtraction of cumulative traffic counts at two adjacent detectors at a given time point, provided that no on-ramp and off-ramp exists between two adjacent detectors. The disadvantage of this method is twofold: first, since each detector may have its own cumulative count drift toward undercounting traffic, the estimated density can be inaccurate and unreliable and, second, it requires that two adjacent detectors be reset while there is no traffic between them, making the method difficult to implement and maintain in practice.

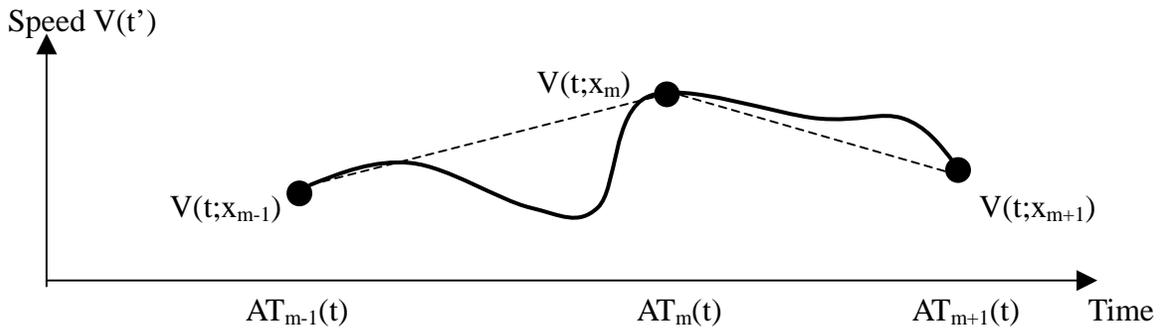
The trajectory approach estimates travel time by hypothesizing the unknown vehicle trajectory or the unknown speed trajectory between detectors. To better illustrate various trajectory approaches, Figure 2-1 shows several commonly used hypothetical speed trajectories for representing the unknown vehicle trajectory between adjacent upstream and downstream detectors.



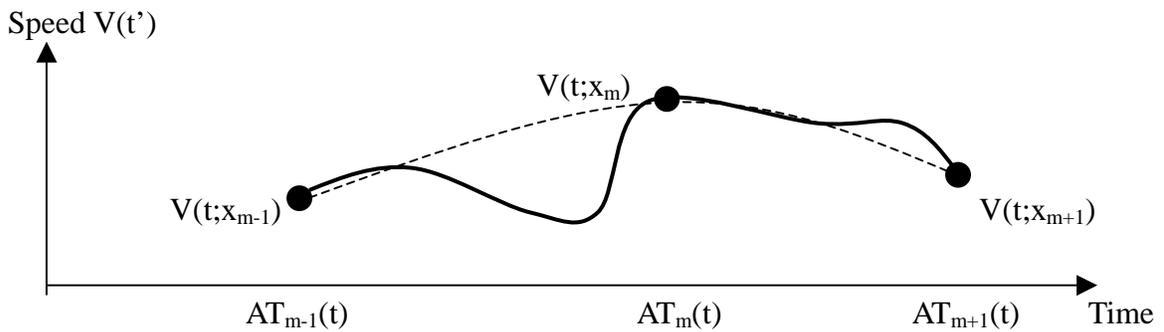
(a) Piecewise constant sample path of speeds using speed at upstream detectors



(b) Conservative piecewise constant sample path of speeds



(c) Piecewise linear sample path of speeds



(d) Piecewise quadratic sample path of speeds

Figure 2-1 Hypothetical speed trajectories between adjacent detectors

The bold line is an actual but unknown speed trajectory, while the dashed line stands for the hypothetical speed trajectories. The actual speed trajectory can only be measured and known at discrete locations where detectors (i.e., back dots in Figure 2-1) are placed. An often used, hypothetical speed trajectory is a piecewise constant interpolation of speeds measured at two adjacent detectors $[\mathbf{x}_m, \mathbf{x}_{m+1}]$, (Cortes et al. 2002, Oh et al. 2003). That is, the actual speed trajectory, $V(t')$, while traversing a link is assumed to be a constant, taking the speed measured at either an upstream detector (e.g., Figure 2-1(a)), or an adjacent downstream detector. One may also use either a conservative or an optimistic estimate of travel time. While the conservative estimate always takes a lower speed, the optimistic estimate assumes a greater speed, as measured by two adjacent detectors to estimate link travel time. In the Advanced Traveler Information Systems (ATIS) in San Antonio, Texas, a conservative estimate (e.g., Figure 2-1(b)) is indeed used for travel time estimation (Fariello 2002). Another popular hypothetical speed trajectory is piecewise, linear interpolation of speeds measured at two adjacent detectors (Cortes et al. 2001; Ishak and Al-Deek 2002; Oh et al. 2003). Figure 2-1(c) shows straight lines connecting two speeds measured at adjacent detectors.

Van Lint and van der Zijpp (2003) suggested that vehicle speed be treated as a piecewise, linear function of distance between two adjacent detectors. As a result, the vehicle speed trajectory becomes a nonlinear function of time, producing continuous speed at detectors.

When traffic is in a free-flow condition, the above-mentioned methods work well and the estimated travel time is accurate and consistent with survey results of traveler experience (Fariello 2002). However, performance of these methods decreases when traffic is in transition flow (i.e., from free flow to congestion or vice versa) and congested flow. During these flow conditions, vehicles experience stop and go oscillation. Thus, a vehicle may sojourn at a low average speed for a while. The current travel time estimation methods do not have adequate flexibility to catch up to the low average speed phenomenon, resulting in inaccurate travel time estimation.

2.4 Car-following Models

In the 1950s, researchers became interested in car-following models. The first published car-following model is a stimulus-response model. The general form of this kind of model is:

$$response_n(t) = sensitivity_n(t) \times stimulus_n(t - \tau_n), \quad [2-13]$$

where, t is the time of observation and, τ_n is the reaction time for driver, n , and, $response_n(t)$ is the acceleration applied at time, t . The reaction time, τ_n , includes the perception time (i.e. the time from the presentation of the stimulus until the foot starts to move) and the foot movement time.

Reuschel (1950) and Pipes (1953) formulated the phenomena of the motion of pairs of vehicles following each other by the expression (May and Keller 1967):

$$x_{n-1} - x_n = l_{n-1} + C_n + S(v_n), \quad [2-14]$$

where, C_n is the clearance of vehicle, n , at standstill, ($v_{n-1} = v_n = 0$). It is measured as the distance between the rear bumper of the leading vehicle and the front bumper of the following vehicle. S is a function of v_n .

The differentiation of the above equation is given as (May and Keller 1967):

$$a_n = \frac{1}{S} \Delta v_{n-1,n}, \quad [2-15]$$

where, a_n is the acceleration applied by the driver. $\Delta v_{n-1,n} = v_{n-1} - v_n$ is the measurement of the relative leader speed (that is, the speed difference between the front vehicle's speed and the subject vehicle's speed).

Chandler et al. (1958) proposed a simple linear model, which can be shown as:

$$a_n(t) = \alpha \Delta v_{n-1,n}(t - \tau_n), \quad [2-16]$$

where, $a_n(t)$ is the acceleration applied by driver, n , at time, t . $\Delta v_{n-1,n}(t - \tau_n) = v_{n-1}(t - \tau_n) - v_n(t - \tau_n)$ is the relative leader speed measured at time, $t - \tau_n$. α is a parameter.

This model was estimated using correlation analysis. Discrete measurements of acceleration, speed, spacing (i.e. the distance between the leading vehicle's front bumper and the subject vehicle's front bumper), and the relative leader speed were collected from eight drivers traveling in a two-lane freeway. Testing time was approximately 20 to 30 minutes per

driver. Measurement equipment was mounted on the subject vehicle. For each driver, the correlation between observed and predicted accelerations was computed for various combinations of α and τ_n . The combination that yielded the highest correlation was selected for the driver. The estimated, α and τ_n averaged over all samples was 0.368 second⁻¹ and 1.55 seconds respectively. The model was again tested by Herman et al. (1959). They used General Motors test track and only four drivers, and did a similar experiment as did Chandler et al. (1958).

Chandler et al. (1958) also noted that a given driver generally notices the behavior of the two vehicles ahead of him as well as that follows him (as seen through a rear view mirror or by horn signals used by the driver following him). In addition, they mentioned that the response of the following vehicle depends not only on the situation at a certain time, but also on its time history.

Komentani and Sasaki (1958) noted a spacing model, (Toledo 2003):

$$v_n(t) = f[\Delta x_{n-1,n}(t - \tau_n), v_{n-1,n}(t - \tau_n)]. \quad [2-17]$$

They used velocity as the dependent variable instead of acceleration. Both linear and quadratic (i.e. in the subject speed) formulations were proposed by the authors.

To represent the relationship between the relative leader speed and the spacing between the vehicles, Gazis et al. (1959) added $\Delta x_{n-1,n}(t - \tau_n)$ into the existing stimulus-response car-following model. Their model can be represented as follows:

$$a_n(t) = \alpha \frac{\Delta v_{n-1,n}(t - \tau_n)}{\Delta x_{n-1,n}(t - \tau_n)}, \quad [2-18]$$

where, $\Delta x_{n-1,n}(t - \tau_n)$ is the spacing between the subject vehicle and its leader measured at time, $t - \tau_n$.

This model was estimated using data collected from three locations: the Lincoln Tunnel and the Holland Tunnels in New York, and the General Motors test track. The parameters, α and τ_n , were estimated for each driver of each dataset using correlation analysis. The values of the parameters averaged over all samples were reported as the estimates. Table 2-1 below summarizes the estimation results. (Ahmed 1999; Toledo 2003)

Table 2-1 Estimation results of the car-following model proposed by Gazis et al. (1959)

Data Collection Site	Number of Drivers	α (mph)	τ (second)
Lincoln Tunnel	16	20.3	1.2
Holland Tunnel	10	18.3	1.4
GM Test Track	8	27.4	1.5

Newell (1961) developed another spacing model:

$$v_n(t) = G_n[\Delta x_{n-1,n}(t - \tau_n)], \quad [2-19]$$

where, G_n is a function that specifies the car-following behavior. No attempt to estimate this model was made.

Edie (1961) noted that previous stimulus-response models were unrealistic when the density is very low, so he modified Gazis's model to the following format:

$$a_n(t) = \alpha \frac{v_n(t - \tau_n)}{\Delta x_{n-1,n}(t - \tau_n)^2} \Delta v_{n-1,n}(t - \tau_n). \quad [2-20]$$

A more general form of the stimulus-response models was proposed by Gazis et al. (1961):

$$a_n(t) = \alpha \frac{v_n(t)^\beta}{\Delta x_{n-1,n}(t - \tau_n)^\gamma} \Delta v_{n-1,n}(t - \tau_n). \quad [2-21]$$

Unlike Edie's model, Gazis et al. used $v_n(t)$ instead of $v_n(t - \tau_n)$. They also did a similar experiment and utilized a similar model fitting method used by Chandler et al., (1958). They tested different combinations of β and γ . They noted that there was little difference in the correlation coefficient for the different sensitivity coefficients for any one driver. However, it seemed to indicate that the exponent, $v_n(t)$ is not negative, since in these cases, the correlation coefficient is generally the smallest for models where $\beta = -1$.

With the concept Chandler et al. (1958) noted, Lee (1966) developed a linear car-following model using a memory function:

$$a_n(t) = \int_0^t M(t-s) \Delta v_{n-1,n}(s) ds, \quad [2-22]$$

where, $M(\)$, is a memory (or weighting) function, which represents the way the driver acts on the information that has been received over time. Lee proposed several examples of possible memory functions, but did not apply any data to fit or test the model.

May and Keller (1967) estimated this model using two methods. The methods were applied to a set of 118, one-minute samples of time mean speeds and mean densities recorded with the pilot detection system of the Chicago Area Expressway Surveillance Project. The data were collected in the middle lane of the three-lane freeway. The first method is transferring the car-following model into the macroscopic model, traffic-stream model, and fitting the data by minimizing the squared deviations of the data points from the regression curve. They mentioned that there was little difference in the mean deviations for different sensitivity factors or parameters, and it would be difficult to give the preference to certain speed-density relations as the best fit to the data with any assurance. The second method is using traffic flow characteristics to judge the macroscopic traffic models derived from car-following models. With the second method, they provided integer and non-integer solutions:

$$a_n(t) = 0.000135 \frac{v_n(t)}{\Delta x_{n-1,n}(t - \tau_n)^3} \Delta v_{n-1,n}(t - \tau_n); \quad [2-23a]$$

$$a_n(t) = 0.000133 \frac{v_n(t)^{0.8}}{\Delta x_{n-1,n}(t - \tau_n)^{2.8}} \Delta v_{n-1,n}(t - \tau_n). \quad [2-23b]$$

Gipps (1981) proposed a new type of car-following model as given below:

$$v_n(t) = B_n \tau_n + \left\{ B_n^2 \tau_n^2 - B_n \left[2\Delta x_n(t - \tau_n) - 2l_{n-1} - v_n(t - \tau_n)\tau_n - \frac{v_{n-1}(t - \tau_n)^2}{\hat{B}_{n-1}} \right] \right\}^{1/2}. \quad [2-24]$$

This model is based on the assumption that each driver wants to achieve the maximum speed and that if he/she uses the desired braking rate, B_n , the subject vehicle can be stopped behind the leading vehicle in an emergency stop. \hat{B}_{n-1} is the braking rate of the leading vehicle in an emergency stop assumed by the driver of the subject vehicle. The model was preferred in a number of simulation projects and an in-depth numerical analysis of the model was given by Wilson (2001).

People realized that drivers may not be aware of the small changes between the lead vehicle and the subject vehicle. Therefore, a psycho-physical model was developed based on the assumption that a driver will only perform an action when a threshold expressed as a function of speed difference and distance is reached. A perceptual threshold is defined as the minimum value of the stimulus the driver will react to. Michaels and Cozan (1963), and Michaels (1963) measured the perceptual thresholds for speed differences at given distances (Leutzbach and Wiedemann 1986).

Leutzbach (1988) noted that the perceptual threshold value increases with spacing. Perceptual thresholds were found to be different for acceleration and deceleration decisions (Ahmed 1999; Toledo 2003).

The ability to perceive speed differences or estimate distances varies widely among drivers, and it is difficult to estimate and calibrate the thresholds associated with psycho-physical models.

Kikuchi and Chakroborty (1992) responded to the concern that drivers do not exercise the dichotomous decision criteria assumed in the traditional deterministic, car-following models and reported an application of "fuzzy logic principles" to the car-following models. It consisted of many straightforward natural language-based driving rules. The model divided the selected inputs into a number of fuzzy sets. Logical operators are then used to produce fuzzy output sets or rule-based car-following behaviors. However, it is difficult to estimate the membership functions, which are crucial to the operation of the model. The authors did not verify the model with field traffic data.

Addison and Low (1998) proposed the following model:

$$a_n(t) = \alpha_1 \frac{\Delta v_{n-1,n}(t - \tau_n)^l}{\Delta x_{n-1,n}(t - \tau_n)^m} v_n(t) + \alpha_2 [\Delta x_{n-1,n}(t - \tau_n) - D_n]^3, \quad [2-25]$$

where, D_n , represents the desired separation that the follower attempts to achieve. They noted that D_n is most naturally taken to be a constant multiple of the mean velocity, v_{n-1} , of the leader vehicle, $D_n = \lambda_n v_{n-1}$.

Aycin and Benekohal (1998) developed a car-following model for use in time-based simulation tools. They assumed that drivers try to attain preferred time headways with respect to their leader and to imitate the leader's speed. To ensure a continuous acceleration profile, they computed the rate of change in the acceleration for the next simulation time step based on the current spacing, speeds, and accelerations of the subject vehicle and the leader

using equations of laws of motion. The model was calibrated as follows: the preferred time headway was computed as the average of observations in which the absolute value of the relative speed was less than five feet per second. Reaction time was assumed equal to 80% of the preferred time headway. A driver was assumed to be in car-following behavior if the leader spacing was less than 250 feet. The values were rather arbitrarily selected based on values found in the literature (Toledo 2003).

Ahmed (1999) included traffic density into the stimulus-response model and made a new car-following model as shown below:

$$a_n(t) = \begin{cases} a_n^{acc}(t) & \text{if } \Delta v_{n-1,n}(t-\tau) \geq 0 \\ a_n^{dec}(t) & \text{if } \Delta v_{n-1,n}(t-\tau) < 0 \end{cases}, \quad [2-26]$$

$$a_n(t) = \alpha \frac{v_n(t - \xi\tau_n)^\beta}{\Delta x_{n-1,n}(t - \xi\tau_n)^\gamma} k_n(t - \xi\tau_n)^\delta \Delta v_{n-1}(t - \tau_n)^\rho + \varepsilon_n(t), \quad [2-27]$$

where, $k_n(t)$ is the density of traffic ahead of the subject. $\xi \in [0,1]$ is a sensitivity lag parameter and, $\varepsilon_n(t)$ is a normally distributed error term. A distance of 100 meters ahead of the subject vehicle was used to compute the density.

The model was calibrated with the video record collected on a 150 to 200 meter section of Interstate 93 at the Central Artery in downtown Boston, Massachusetts. The video was processed using the VIVA software package and local regression to get the vehicle trajectory information. In addition to the length of each vehicle, the data contains position, speed, and

acceleration of every vehicle for every second. The position measurement error for the video was estimated to be ± 1 meter. The estimation results are:

$$a_n(t) = \begin{cases} a_n^{acc}(t) = 0.0225 \frac{v_n(t)^{0.722}}{\Delta x_{n-1,n}(t)^{0.242}} k_n(t)^{0.682} |\Delta v_{n-1,n}(t - \tau_n)|^{0.600} + \varepsilon_n^{cf,acc}(t) & \text{if } \Delta v_{n-1,n}(t - \tau) \geq 0 \\ a_n^{dec}(t) = -0.0418 \frac{1}{\Delta x_{n-1,n}(t)^{0.151}} k_n(t)^{0.804} |\Delta v_{n-1,n}(t - \tau_n)|^{0.682} + \varepsilon_n^{cf,dec}(t) & \text{if } \Delta v_{n-1,n}(t - \tau) < 0 \end{cases} \quad [2-28]$$

where, $\varepsilon_n^{cf,acc}(t) \sim N(0, 0.825^2)$ is random term associated with the acceleration model of driver, n , at time t . $\varepsilon_n^{cf,dec}(t) \sim N(0, 0.802^2)$ is a random term associated with the deceleration model of driver, n , at time t .

The application of stimulus-response models and their extensions requires the parameters to be calibrated for a particular network. However, a large number of contradictory findings for the values of parameters have been reported (Brackstone and McDonald 1999).

Zhang and Kim (2001) modeled driver reaction times in the car-following models. They noted that drivers adopt a speed according to spacing and reaction times. The reaction time was modeled as a function of vehicle spacing and the amount of traffic. Although computer simulation was used in the study, no real traffic data was utilized. The model uses the following equation:

$$a_n(t) = \frac{\Delta x_{n-1,n}(t - \tau_n(t)) - L_{n-1} - b}{\tau_n(t)}, \quad [2-29]$$

where, b is the legal distance between the vehicles at standstill.

Toledo (2003) used the dataset collected in 1983 by the Federal Highway Administration (FHWA) in a section of Interstate Highway 395 (I-395) southbound in Arlington, Virginia. The four-lane highway section is 997 meters long and includes an on-ramp and two off-ramps. An hour of data at a rate of one frame per second was collected through aerial photography. His results are:

$$a_n(t) = \begin{cases} a_n^{acc}(t) = 0.0355 \frac{v_n(t)^{0.291}}{\Delta x_{n-1,n}(t)^{0.166}} k_n(t)^{0.550} \Delta v_{n-1,n}(t - \tau_n)^{0.520} + \varepsilon_n^{cf,acc}(t) & \text{if } \Delta v_{n-1,n}(t - \tau) \geq 0 \\ a_n^{dec}(t) = -0.860 \frac{1}{\Delta x_{n-1,n}(t)^{0.565}} k_n(t)^{0.143} \Delta v_{n-1,n}(t - \tau_n)^{0.834} + \varepsilon_n^{cf,dec}(t) & \text{if } \Delta v_{n-1,n}(t - \tau) < 0 \end{cases}, [2-30]$$

where, $\varepsilon_n^{cf,acc}(t) \sim N(0, 1.134^2)$ and $\varepsilon_n^{cf,dec}(t) \sim N(0, 1.169^2)$.

Hamdar and Mahmassani (2008) stated that the Gipps model is advantageous because it models driving behavior following cognitive logic that may be adopted by the driver. They modified Gipps model as follows:

$$v_n(t) = B_n \left(\frac{\tau_n}{2} \right) + \left\{ \frac{B_n^2 \tau_n^2}{4} - B_n \left[2\Delta x_n(t - \tau_n) - v_n(t - \tau_n) \tau_n - \frac{v_{n-1}(t - \tau_n)^2}{\hat{B}_{n-1}} + D_n \right] \right\}^{1/2}, [2-31]$$

where, D_n is a function of the initial risk factor representing the distance a driver is willing to travel beyond the safety threshold. When this value is positive, the driver is willing to take risks and this may increase the probability of causing an accident. If this value is negative,

the driver prefers to stay within the safety margin so he or she can come to a stop without hitting the vehicle in front of them. They used FHWA's Next Generation SIMulation Program (NGSIM) data and estimated models that fit macroscopic traffic properties.

Chapter 3 Methodology

In this chapter, the proposed methodologies for developing highway traffic models are presented. The methodologies are presented with the same sequence that people do research on traffic science, which is from macroscopic to microscopic and from estimation to prediction.

3.1 Traffic-Stream Models

It is generally acknowledged that a multi-regime traffic-stream model provides a better fit to the actual data than its single-regime counterpart. The difficulty in developing a multi-regime model lies in the determination of knot (i.e. breakpoint) positions. This is often achieved based on subjective judgment, where rigorous justification is lacking. To overcome this difficulty, this study presents a new method of adaptive semi-parametric regression for modeling the traffic speed-density relationship. It takes advantages of gradient-based optimization techniques and Genetic Algorithm (GA) to simultaneously and optimally implement parameter and knot estimation. The methodology appears to have the potential to be a substantial improvement over the existing univariate regression methodology and traffic-stream models.

The multi-regime traffic-stream model can be written as:

$$\hat{u}_e(k) = \sum_{m=1}^M B_m(k) I[k \in R_m], \quad [3-1]$$

where, M is the number of sub-regimes; I is an indicator function, which is one, if its argument is true, and zero otherwise; $\{B_m\}_1^M$ are basis functions taking one of the function forms from the basis function set B ; $\{R_m\}_1^M$ are sub-regimes and since these sub-regimes are disjoint only one basis function is nonzero for any point, k .

The model should be able to predict accurately the observed data. In other words, the predictability or accuracy of a traffic-stream model is the number one concern. For this reason, a model that can minimize an overall deviation from the actual observations should be given priority.

The notion of reasonableness depends upon the purpose for which the approximation is to be used. In almost all applications, however, accuracy is important. The lack of accuracy is often defined by the expected error, E , as:

$$E = \frac{1}{N} \sum_{i=1}^N w(k_i) d[\hat{u}_e(k_i), u_e(k_i)], \quad [3-2]$$

where, d is some measure of distance and, w is a possible weight function. If the sole purpose of the regression analysis is to obtain a rule for predicting future values of the response, u_e , for given values for k , then accuracy is the only important virtue of the model.

Often, however, one wants to use \hat{u}_e to understand the properties of the true underlying function u_e , and thereby the traffic dynamics that generated the data. Depending on the application, other desirable properties of the approximation might include rapid computability and smoothness; that is, \hat{u}_e should be a smooth function of k and at least its low-order derivatives should exist everywhere in the domain of interest.

A number of studies (May 1990; Sun and Zhou 2005a) have shown that traffic patterns can be adequately captured by free flow, transition flow, and congestion. Therefore, for the purpose of illustration of the adaptive regression method, the maximum number of regimes is set as three in this study. Models of single-, two-, and three-regimes will be screened by adaptive regression, each corresponding to 0, 1, and 2 knots, respectively, whose positions need to be estimated from the actual data.

First of all, the proposed adaptive regression function forms for each sub-regime of the domain in PPR are taken from a relatively large pool, that is, a basis function set consisting of an existing single-regime traffic-stream model and quadratic functions. Specifically, the basis function set, B is defined as:

$$B = \left\{ \begin{array}{l} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \end{array} \right\} = \left\{ \begin{array}{l} u_e = \beta_0 + \beta_1 k \\ u_e = \beta_0 \ln(\beta_1 / k) \\ u_e = \beta_0 \exp(k / \beta_1) \\ u_e = \beta_0 \exp(\beta_1 / k^2) \\ u_e = \beta_0 [1 - (k / \beta_1)^{\beta_2}]^{\beta_3} \\ u_e = \beta_0 + \beta_1 k + \beta_2 k^2 \\ k = (\beta_1 + \frac{\beta_2}{\beta_0 - u_e} + \beta_3 u_e)^{-1} \end{array} \right\}, \quad [3-3]$$

where, β_0 , β_1 , β_2 , and β_3 are coefficients and B_1 through B_7 are basis function elements, constituting the basis function set. In this way, the possible number of combinations of the piecewise multi-regime traffic-stream model is greatly enlarged. The reason that these seven basis functions in equation 3-3 are chosen is because their performances of fitting actual speed-density observations have been well investigated in the literature (Rakha and Crowther 2003; Sun and Zhou 2005a). Figure 3-1 shows the fitting of the seven basic functions.

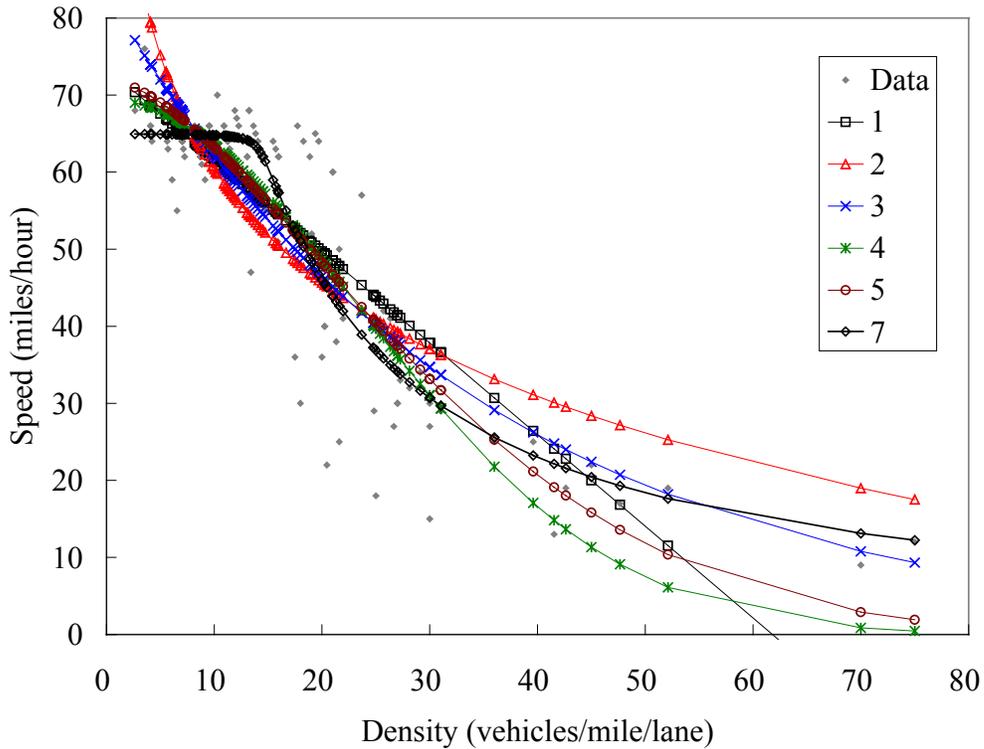


Figure 3-1 Single-regime traffic-stream models

It should, however, be noted that the basic functions described in equation 3-3 are by no means exclusive. Other types of basis functions can also be added to allow more flexible choices. Among the seven basis functions in equation 3-3, the first four basis functions only require two unknown parameters to be calibrated for each basis function. The fifth and sixth basis functions have three unknown parameters that will be estimated from actual datasets, while the seventh basis function demands four unknown parameters to be obtained from data. The larger the number of unknown parameters involved in the basis function, the more complex the developed traffic-stream model will be. In order to control the model's

complexity, (although all seven basis functions are used for developing single-regime traffic-stream models), only the first four basis functions are used in developing two-regime and three-regime traffic-stream models. The final traffic-stream model will be the one that among all single-regime, two-regime, and three-regime models has the best fitting performance.

The parameter estimation of selected basis function in equation 3-3 is achieved by minimizing the overall square error of the fitting equation 3-4 over the entire regime:

$$\{\hat{\beta}_j\}'_1 = \arg \min_{\{\beta_j\}'_1} \sum_{i=1}^N [u_i - \hat{u}_e(k_i | \{\beta_j\}'_1)]^2, \quad [3-4]$$

where, N is the total number of observations over the entire regime and, J is the total number of unknown parameters to be estimated for a particular model. It should be noted that equation 3-4 contains not only coefficients β_0 , β_1 , β_2 , and β_3 that may appear in equation 3-3, but also unknown positions of knots, $\{k_K\}'_1^{K_n}$, in which K_n is the number of unknown knots and equal $M-1$. In other words, parameters and knots, $\{\beta_j\}'_1 = \{\beta_0, \beta_1, \beta_2, \beta_3, k_1, \dots, k_K\}$ are jointly optimized using a goodness-of-fit criterion.

It is also desired that the speed-density relation should be governed by a continuous function. To achieve this requirement, the following constraint can be imposed such that function values at knots predicted from left and right functions of the multi-regime stream model are identical:

$$\hat{u}_e(k_{K-}) - \hat{u}_e(k_{K+}) = 0, \quad [3-5]$$

where, k_{K-} and k_{K+} are the traffic density values at left and right sides of knot, k_K . Hence, parameter estimation of the traffic-stream model becomes the minimization of objective function (i.e. equation 3-4) subject to constraint (i.e. equation 3-5). In principal, higher-order continuity of the speed-density relationship can also be achieved by imposing additional constraints on derivatives of the model.

The proposed adaptive regression differs from the piecewise parametric regression (PPR) in that the former has an additional model selection functionality that allows the best model to be chosen from a large set of models. Specifically, the procedure is implemented by constructing a set of globally defined basis functions that span the space of the basis function sets and fitting the coefficients of the basis function expansion to the data by ordinary least squares. Candidate models consist of all possible combinations of basis functions for M sub-regimes, each associated with a different number of knots. The resultant sum of squared error (SSE) of the model or:

$$\min \text{SSE} = \sum_{i=1}^N [u_i - \hat{u}_e(\rho_i | \{\beta_j\}'_1)]^2, \quad [3-6]$$

subject to equation 3-5, is used as the goodness-of-fit criterion for selecting the final model. A better fitting model could be achieved if the smoothness or continuity requirement on the traffic-stream model is relaxed. To this end, the following objective function, rather than equation 3-6 can be used:

$$\min \text{SSE} = \left\{ \sum_{i=1}^N [u_i - \hat{u}_e(k_i | \{\beta_j\}'_1)]^2 + w \sum_k^K [\hat{u}_e(k_{K-}) - \hat{u}_e(k_{K+})]^2 \right\}, \quad [3-7]$$

where, K_n is the total number of knots and, w is the weight function that controls the degree of continuity of the traffic-stream model. If one assigns a very large weight, w , to the second term in equation 3-7, it is equivalent to significantly penalizing the difference between function values at knots computed from left and right functions of the traffic-stream model. Therefore, this is the same thing as equation 3-6 subject to equation 3-5. If a very small weight, w is assigned to the second term, the continuity requirement can be basically ignored. Since equation 3-7 is a more generic setting, from now on only equation 3-7 will be used.

The trade-off between complexity and flexibility of the model is controlled by M . Increasing the value of M enlarges the pool of candidate models but the price is increased model complexity (that is, the number of sub-regimes and associated parameters) as well as computational burden. For instance, to one extreme when, $M = 1$, the resultant model from adaptive regression is a single-regime model. On the other extreme, when M is very large and approaches $N - 1$, the model mimics the behavior of a nonparametric, local regression model. The adaptive regression proceeds by screening all models with $1, \dots, M$ knots in a set of candidate models and selecting the one with a minimum SSE, while controlling proper model complexity as the best traffic-stream model.

3.2 Shock Wave Based Traffic Modeling and Prediction

In order to implement and test the shock wave based traffic estimation and prediction, two computer programs are developed using Visual Basic programming language (VB). The first computer program (see Appendix II) simulates a highway segment with three traffic detector stations and the second one (see Appendix III) uses two traffic detector stations.

At the beginning time, 0, the program generates two shock waves denoted as SW1 and SW2. In order to make it simple, the new shock waves are located at the midpoint between traffic detector stations, which means SW1 is of the same distance from traffic detector 1 and traffic detector 2, and SW2 is the same distance from traffic detector 2 and traffic detector 3 (see Figure 3-2 below). SW1 demarks status 1,0 and status 2,0, and SW2 demarks status 2,0 and status 3,0. Each status includes all the information measured by a certain traffic detector, such as status i,t contains speed, volume, and density data from traffic detector, i , at time, t . If the density in status 1,0 is not equal to the density in status 2,0, then the speed of shock wave 1 is calculated by equation 1-2, otherwise the speed of shock wave 1 is given as 0.

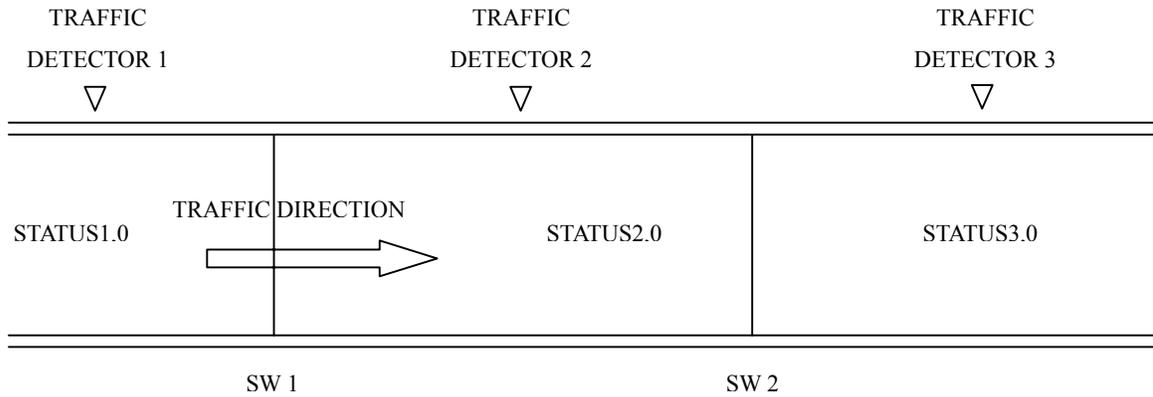


Figure 3-2 Generation of initial shock waves

After one time interval, the shock waves, SW1 and SW2, arrive at the new positions as shown in Figure 3-3 below. And new shock waves, SW3, SW4, SW5, and SW6, are generated since new traffic data is available at the time. Their initial positions are at the mid points between traffic detector stations and the closest shock waves nearby.

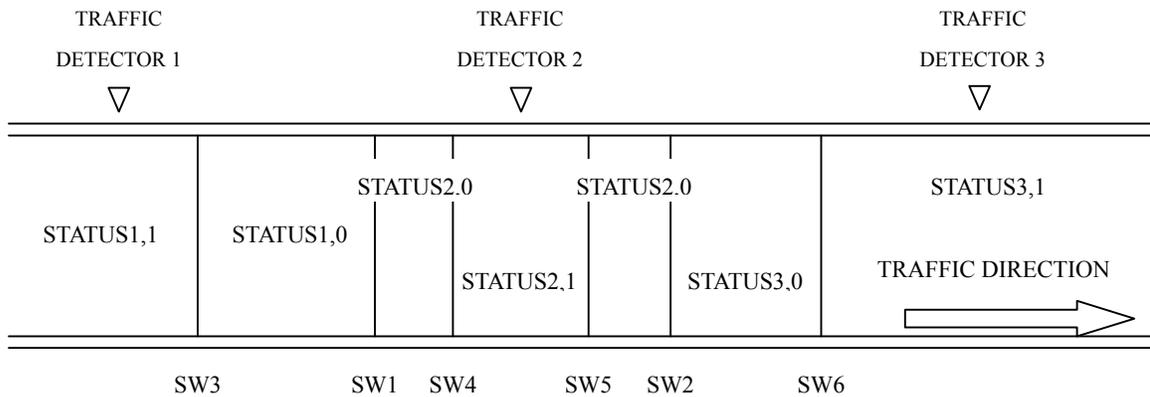


Figure 3-3 Generation of new shock waves

If two shock waves hit each other, the program will generate a new shock wave. For example, in Figure 3-4 below, shock waves SW X moves to shock wave SW Y. A new shock

wave, SW Z, will be calculated when SW X meets SW Y. The new shock wave's speed depends on Status A and C, but Status B is no longer in the system.

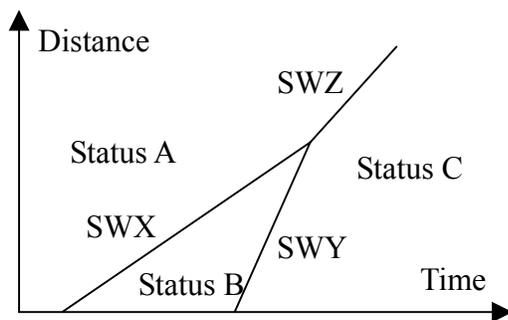


Figure 3-4 Shock Wave X meet Shock Wave Y

If a shock wave moves outside the researching road segment, it will be deleted from the simulation system.

The system is kept updated provided new data is available. When shock waves exist between traffic detector stations, the traffic status can be estimated at any point between the stations. To predict traffic information in the future, the simulation should be run without inputting any new measured data until the prediction time arrives.

The second computer program simulates a freeway segment with only two traffic detector stations. It can be used to estimate and predict the traffic information between these two traffic detector stations as well. However, without a traffic detector giving updated information, the estimation and prediction precision is less than the program using three traffic detector stations. The program uses traffic data from the first and third traffic detectors to estimate traffic information at the second traffic detector. Then the result can be compared

to the real data measured by the second traffic detector. With this program, the traffic estimation effect can be tested.

3.3 Travel Time Estimation

3.3.1 Notations

A new travel time estimation method based on speed trajectory is developed here. Suppose that there are M detectors located between and including origin, O, and destination, D. Let Δx_m be the distance between the m th and the $(m+1)$ th detectors where, $m = 1, \dots, M$. The location of detectors at the origin and the destination are labeled, \mathbf{x}_1 and \mathbf{x}_M , respectively. Let $AT_m(t)$ be the arrival time of the vehicle at the m th detector with, $AT_1(t) = t$ at the origin and, $AT_M(t) = AT(t)$ at the destination. As the vehicle traverses from the origin through each intermediate point to the destination, corridor travel time can be constructed as follows based on equation 1-3:

$$\Delta x = \sum_{m=1}^{M-1} \Delta x_m = \sum_{m=1}^{M-1} \int_{AT_m(t)}^{AT_{m+1}(t)} V(t') d\tau = \sum_{m=1}^{M-1} \bar{V}[t'_m(t)] [AT_{m+1}(t) - AT_m(t)], \quad [3-8]$$

where speed, $V(t')$ is assumed to be a continuous function over time and space, $[AT_m(t), AT_{m+1}(t)]$, real number $t'_m(t) \in [AT_m(t), AT_{m+1}(t)]$; and $\bar{V}[t'_m(t)]$ is the time mean value of speed $V(t')$ over space-time interval $[AT_m(t), AT_{m+1}(t)]$. In addition, the mean value theorem for integrals is used in the derivation of equation 3-8 (Courant, 1988).

Equation 3-8 leads to an iterative expression of arrival time at the $(m+1)$ th detector, $AT_{m+1}(t)$, in terms of arrival time at the m th detector, $AT_m(t)$:

$$AT_{m+1}(t) = AT_m(t) + \frac{\Delta x_m}{\bar{V}[t'_m(t)]}. \quad [3-9]$$

Successively applying equation 3-9 to $m = 1, \dots, M$, summing up both sides of the resulting equations, and canceling and rearranging terms leads to a concise expression of corridor travel time, $TT(t; O, D)$:

$$TT(t; O, D) = AT_M(t) - AT_1(t) = \sum_{m=1}^{M-1} TT(t; \mathbf{x}_m, \mathbf{x}_{m+1}) = \sum_{m=1}^{M-1} \frac{\Delta x_m}{\bar{V}[t'_m(t)]}, \quad [3-10]$$

where, $TT(t; \mathbf{x}_m, \mathbf{x}_{m+1})$ is link travel times between two adjacent detectors.

At the time of departure, t , the speed trajectory of a vehicle over future time interval, $[AT_1(t), AT_M(t)]$ is indeed unknown in advance. To infer such an unknown future speed trajectory, two methods can be used: “travel time estimation” and “travel time prediction”. “Travel time estimation”, means that no predicted traffic information is incorporated into the inference of travel time. The synchronously measured speeds at the departure time at all other detectors are used as substitutes of the anticipated future speeds at these detectors for inferring travel time at time, t . An implicit assumption embedded in “travel time estimation” is that speeds measured at downstream detectors at various future times can be substituted for speeds measured at these detectors at the departure time, t . It requires that, within the

space-time interval, $[AT_1(t), AT_M(t)]$, the traffic flow needs to be homogenous with respect to space and stationary with respect to time. This is a fairly strong assumption and is often violated when congestion occurs within the links. The validity of the above-mentioned assumption would be high if the space-time interval, $[AT_1(t), AT_M(t)]$ is short enough. As a result, the accuracy of estimated travel time, $TT(t; \mathbf{x}_O, \mathbf{x}_D)$, increases when travel time estimation is updated periodically as the vehicle approaches the destination. Nevertheless, “travel time estimation” is researched only in this study, since it is the basis of “travel time prediction” and can be extended to the latter.

3.3.2 Quadratic Basis Functions

Given that a number of speed measurements are known at different time-space locations, the construction of the unknown speed trajectory, $V(t')$, can be treated as a function interpolation problem, though other approaches may also apply. Polynomials can be used for function interpolation (Burden and Faires 1997; Heath 1997; Epperson 2003). A high degree polynomial for interpolating and approximating the unknown speed trajectory between multiple detectors may lead to an expensive computation burden and excessive speed oscillation. A piecewise quadratic function can be used to lessen these problems (e.g., Figure 2-1(d)).

In reality, drivers may decelerate more when getting close (either in time or in space) to a congested zone (i.e. when they slow down) or accelerate more when leaving a congested

zone to enter free flow traffic. In essence, drivers adjust their acceleration in response to foreseeing downstream traffic conditions. The benefit of having a quadratic basis function is that, under this model, vehicle acceleration becomes a linear function of time (other than a constant or zero in piecewise linear or constant speed trajectory). The acceleration rate is determined by the current position of the vehicle and traffic condition as reflected in observed speeds at three detectors. This provides flexibility for mimicking actual vehicle speed change.

A piecewise quadratic interpolation uses speed observations at three adjacent detectors to construct the unknown speed sample path over two segments, as shown in Figure 2-1(d). Suppose that the number of detectors, M is odd. If a Lagrange quadratic interpolation polynomial is used, the actual speed trajectories over two consecutive links can be approximated as follows:

$$V_{2j-1}(t') \approx V(t; \mathbf{x}_{2j-1})\ell_{2j-1}\{t'\} + V(t; \mathbf{x}_{2j})\ell_{2j}\{t'\} + V(t; \mathbf{x}_{2j+1})\ell_{2j+1}\{t'\}, \quad [3-11]$$

where, $V(t; \mathbf{x}_{2j-1})$, $V(t; \mathbf{x}_{2j})$, and $V(t; \mathbf{x}_{2j+1})$ are speeds measured at three adjacent detectors, $t' \in [AT_{2j-1}(t), AT_{2j+1}(t)]$, $j = 1, \dots, (M-1)/2$, and the Lagrange basis functions:

$$\ell_{2j-1}(t') = \frac{[\tau - AT_{2j}(t)][\tau - AT_{2j+1}(t)]}{[AT_{2j-1}(t) - AT_{2j}(t)][AT_{2j-1}(t) - AT_{2j+1}(t)]}, \quad [3-12]$$

$$\ell_{2j}(t') = \frac{[\tau - AT_{2j-1}(t)][\tau - AT_{2j+1}(t)]}{[AT_{2j}(t) - AT_{2j-1}(t)][AT_{2j}(t) - AT_{2j+1}(t)]}, \quad [3-13]$$

$$\ell_{2j+1}(t') = \frac{[\tau - AT_{2j}(t)][\tau - AT_{2j-1}(t)]}{[AT_{2j+1}(t) - AT_{2j}(t)][AT_{2j+1}(t) - AT_{2j-1}(t)]}. \quad [3-14]$$

Without loss of generality, let $AT_{2j-1}(t) \equiv t = 0$, $AT_{2j}(t) \equiv t_2$ and $AT_{2j+1}(t) \equiv t_3$. Let $V_{2j-1}(t) \equiv V_1$, $V_{2j}(t) \equiv V_2$ and $V_{2j+1}(t) \equiv V_3$. Equation 3-11 can now be simplified as:

$$V_{2j-1}(\tau) \approx V_1 \ell_{2j-1}\{\tau\} + V_2 \ell_{2j}\{\tau\} + V_3 \ell_{2j+1}\{\tau\}, \quad [3-15a]$$

where the Lagrange basis functions are now:

$$\ell_{2j-1}(t') = \frac{(t' - t_2)(t' - t_3)}{t_2 t_3}, \quad \ell_{2j}(t') = -\frac{t'(t' - t_3)}{t_2(t_3 - t_2)}, \quad \ell_{2j+1}(t') = \frac{t'(t' - t_2)}{t_3(t_3 - t_2)}. \quad [3-15b]$$

3.3.3 Constant Basis Functions

Traffic between three adjacent detectors may experience transition flow or congestion. When a vehicle at a high speed in free-flow condition enters a congestion zone, it slows down and then exhibits stop and go behavior. This behavior corresponds to a low average speed for a period of time, depending upon downstream traffic conditions. To better capture vehicle sojourn time at low-speed, piecewise truncated quadratic speed trajectory is proposed, which is equivalent to imposing a lower speed bound to the quadratic speed trajectory. When a quadratic speed trajectory reaches a lower speed bound (see Figure 3-5 below), the hypothetical quadratic speed trajectory will be replaced by the lower speed bound. Similarly, an upper speed bound will be imposed when the quadratic speed trajectory reaches an unrealistic high speed as also shown in Figure 3-5.

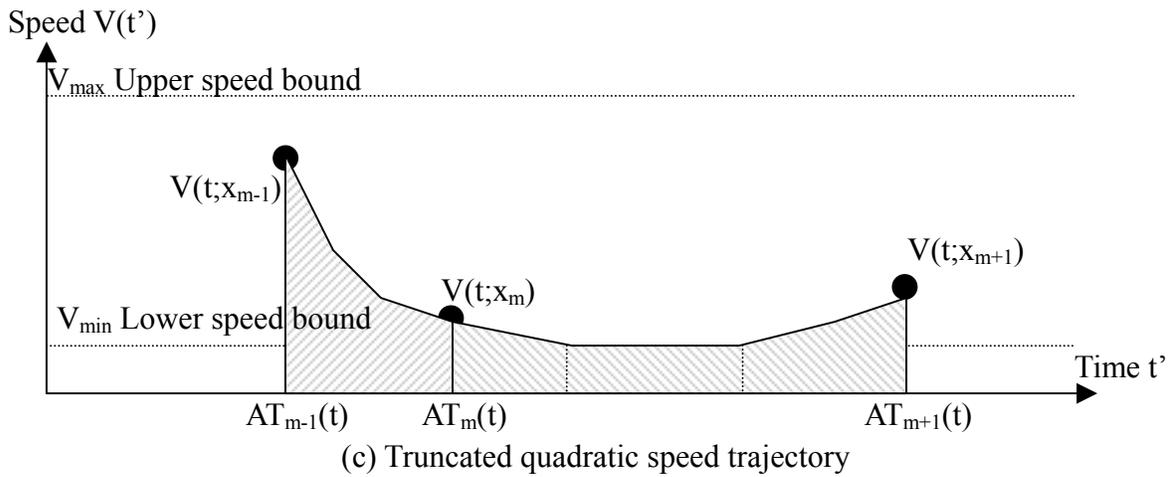
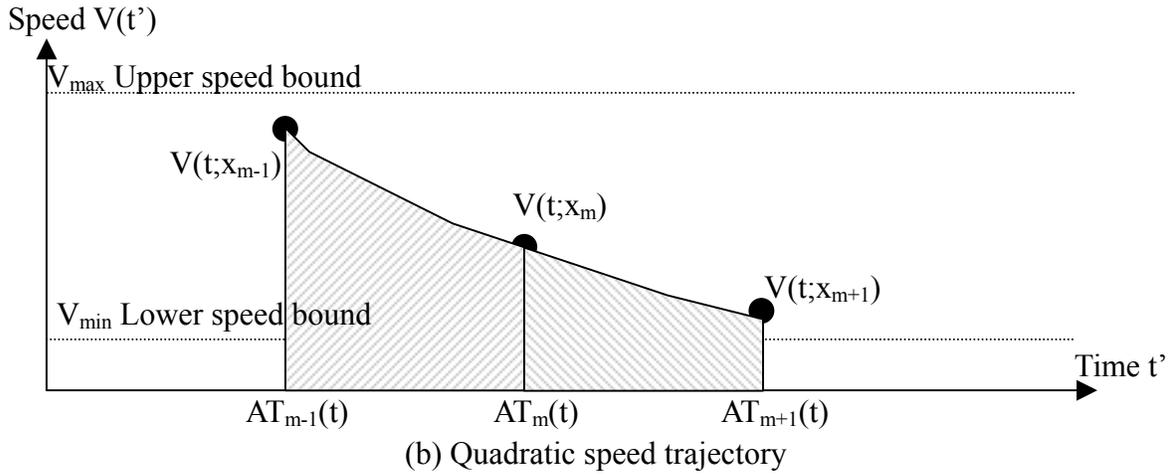
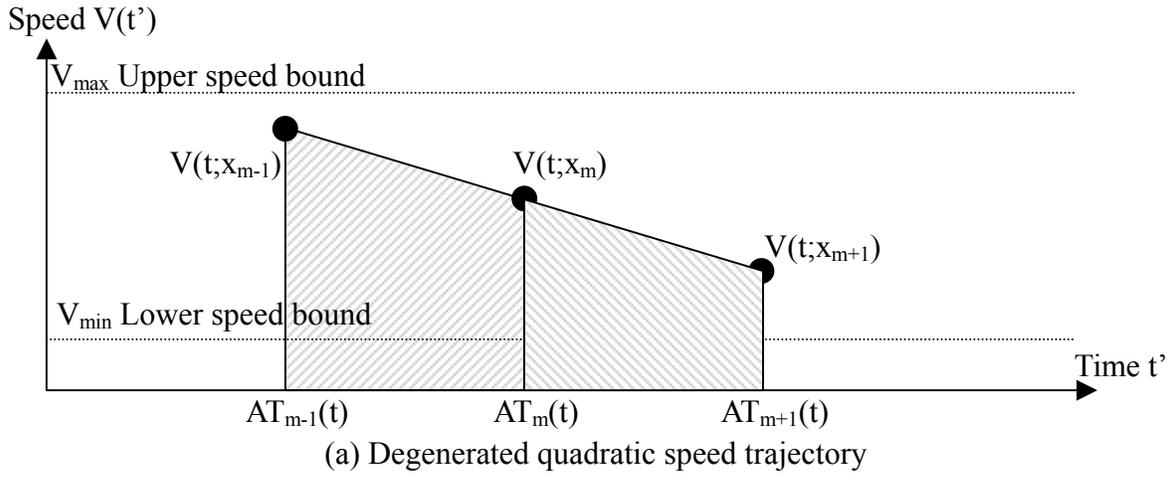


Figure 3-5 Travel distance conditions for adjacent detectors

During transition flow or congested conditions, the truncated piecewise quadratic speed trajectory over-performs a piecewise constant or a linear speed trajectory because it allows the vehicle to have a longer sojourn time at a very low speed. This indeed provides a mechanism to mimic vehicle speed change in the following conditions: entering congested conditions from free-flow traffic, staying in the congestion zone as stop-and-go traffic, and discharging from congested conditions to become free-flow traffic.

In this study, the following method is used to construct the lower and upper speed bounds. At any time, t , of a weekday or weekend, speed observations within a certain interval of t , recorded by a detector are collected. For instance, suppose that a detector reports aggregated speed every 20 seconds. At 9:00am, 36 speed observations measured during [8:58am, 9:02am] at three adjacent detectors for a specific Monday are compiled. The minimum speeds among these 36 speed observations (i.e. $3 \times 4 \times 60 / 20 = 36$) are recorded as $V_{m,i}^{\min}(t)$, with $t = 9:00$ am, in which suffix i is the date. Suppose that there are $i = 1, 2, \dots, 100$ past Mondays speed observations available. The resultant 100 low speed observations, $V_{m,i}^{\min}(t)$, can be used to construct an empirical probability density function (EPDF) of low speed. The lower speed bound can be set as the maximum likelihood estimation based on the EPDF and this is the approach used in this study. The length of the interval, [8:58am, 9:02am], may be adjusted to include more speed observations. Estimating EPDF using data from an individual weekday or weekend can mitigate the effect of weekday,

weekend, month, and season. The same approach can be applied to give an upper speed bound. It should be noted that the determination of two speed bounds is critical and can be done in many ways. For instance, instead of using the maximum likelihood estimate of EPDF, one may also use the 85th percentile of these 100 low speed observations to construct the lower speed bound for a segment of speed trajectory.

3.3.4 Combination of Quadratic and Constant Basis Functions

Let Δx_1 and Δx_2 be link distances between the $(2j-1)$ th detector and the $(2j)$ th detector, and between the $(2j)$ th detector and the $(2j+1)$ th detector, respectively, in which $j = 1, \dots, (M-1)/2$. When combining quadratic and constant basis functions, the task is to solve two unknown arrival times, t_2 and t_3 , considering a possible truncation of quadratic speed trajectory by the lower and upper speed bounds. The relationship between the quadratic speed trajectory and two speed bounds can be written as the following two equations:

$$V_1 \ell_{2j-1} \{t'\} + V_2 \ell_{2j} \{t'\} + V_3 \ell_{2j+1} \{t'\} - V_{\max} = 0, \quad [3-16a]$$

$$V_1 \ell_{2j-1} \{t'\} + V_2 \ell_{2j} \{t'\} + V_3 \ell_{2j+1} \{t'\} - V_{\min} = 0. \quad [3-16b]$$

By substituting equation 3-15b into the above two equations, one obtains:

$$f_{\max}(t') = at'^2 + bt' + c_{\max} = 0, \quad [3-17a]$$

$$f_{\min}(t') = at'^2 + bt' + c_{\min} = 0, \quad [3-17b]$$

where coefficients are $a = t_3(V_1 - V_2) - t_2(V_1 - V_3)$, $b = -t_3^2(V_1 - V_2) + t_2^2(V_1 - V_3)$,
 $c_{\max} = t_2 t_3 (t_3 - t_2)(V_1 - V_{\max})$, and $c_{\min} = t_2 t_3 (t_3 - t_2)(V_1 - V_{\min})$. Let $\Delta_{\max} = b^2 - 4ac_{\max}$ and
 $\Delta_{\min} = b^2 - 4ac_{\min}$ be discriminants of quadratic equations 3-17a and 3-17b, respectively.
 Let t_1^{\max} and t_2^{\max} , and t_1^{\min} and t_2^{\min} be two zeros of equations 3-17a and 3-17b,
 respectively. Let t'^{\max} and t'^{\min} be the maximum value of $f_{\max}(t')$ and the minimum
 value of $f_{\min}(t')$, respectively such that:

$$\begin{cases} t_1^{\max} = (-b + \sqrt{\Delta_{\max}}) / 2a \\ t_2^{\max} = (-b - \sqrt{\Delta_{\max}}) / 2a \end{cases} \text{ and } \begin{cases} t_1^{\min} = (-b + \sqrt{\Delta_{\min}}) / 2a \\ t_2^{\min} = (-b - \sqrt{\Delta_{\min}}) / 2a \end{cases}, \quad [3-18a]$$

$$t'^{\max} = t'^{\min} = -b / 2a . \quad [3-18b]$$

If zeros in equation 3-18a are real-valued, it is required that $t_1^{\max} \leq t_2^{\max}$ and $t_1^{\min} \leq t_2^{\min}$.

Also, $0 = t_1 < t_2 < t_3$ is always implicitly stated.

Due to the existence of two speed bounds, the proposed speed trajectory may fall into one of the thirteen cases or three scenarios listed in Table 3-1. Cases, 1, 2, and 3 belong to a scenario of degraded quadratic speed trajectories (i.e., linear speed trajectories). Cases 4, 5, 6, 9, 10, and 11 belong to a scenario of quadratic speed trajectories, where the quadratic speed trajectory between the $(2j - 1)$ th detector and the $(2j + 1)$ th detector intercepts with neither the upper speed bound nor the lower speed bound. Cases 7, 8, 12, and 13 belong to a scenario of truncated quadratic speed trajectories, where the quadratic speed trajectory between the $(2j - 1)$ th detector and the $(2j + 1)$ th detector intercepts with either the upper or the lower

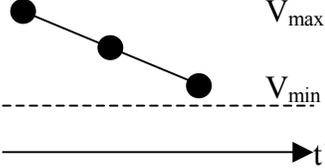
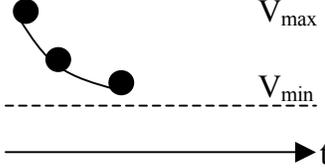
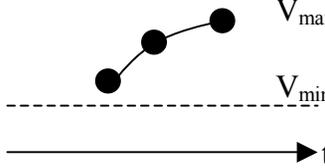
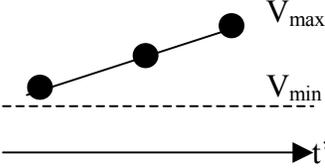
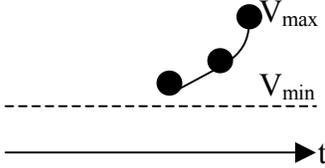
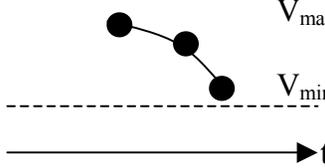
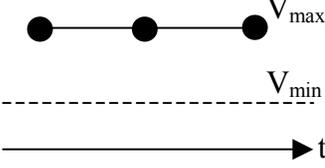
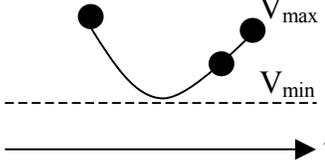
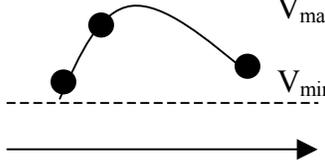
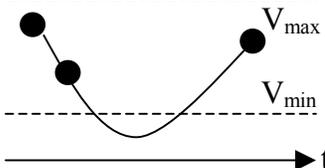
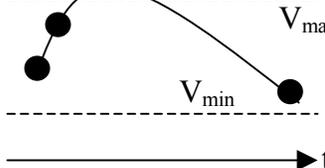
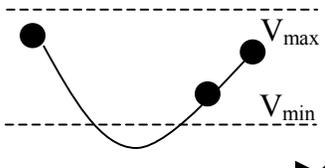
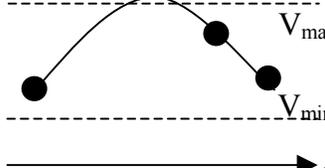
speed bound. Figure 3-5 shows these three representative scenarios. The two link distances, Δx_1 and Δx_2 , can be related to two segments of speed trajectory as follows:

$$\Delta x_1 = \int_{t_1=0}^{t_2} V_{2,j-1}(t') d\tau, \quad [3-19a]$$

$$\Delta x_2 = \int_{t_2}^{t_3} V_{2,j-1}(t') d\tau. \quad [3-19b]$$

After taking into account the thirteen cases, arrival times, t_2 and t_3 , can be uniquely solved.

Table 3-1 Scenarios of truncated quadratic speed trajectory

a=0	a>0	a<0
 <p>Case 1: $b < 0$</p>	 <p>Case 4: $t'_{\min} \geq t_3$</p>	 <p>Case 9: $t'_{\max} \geq t_3$</p>
 <p>Case 2: $b > 0$</p>	 <p>Case 5: $t'_{\min} \leq t_1 = 0$</p>	 <p>Case 10: $t'_{\max} \leq t_1 = 0$</p>
 <p>Case 3: $b = 0$</p>	 <p>Case 6: $\Delta_{\min} \leq 0,$ $0 = t_1 < t'_{\min} < t_3$</p>	 <p>Case 11: $\Delta_{\max} \leq 0,$ $0 = t_1 < t'_{\max} < t_3$</p>
<p>N/A</p>	 <p>Case 7: $\Delta_{\min} > 0,$ $0 = t_1 < t_2 < t'_{\min} < t_1'_{\min} < t_3$</p>	 <p>Case 12: $\Delta_{\max} > 0,$ $0 = t_1 < t_2 < t'_{\max} < t_1'_{\max} < t_3$</p>
<p>N/A</p>	 <p>Case 8: $\Delta_{\min} > 0,$ $0 = t_1 < t_1'_{\min} < t_1'_{\min} < t_2 < t_3$</p>	 <p>Case 13: $\Delta_{\max} > 0,$ $0 = t_1 < t_1'_{\max} < t_1'_{\max} < t_2 < t_3$</p>

3.4 Improved Gipps Car-following Model

The assumption of the Gipps category car-following model is that to avoid a rear-end collision, the subject vehicle must stop behind the rear bumper of the leading vehicle in an emergency stop, which can be displayed as:

$$x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1} \geq x_n(t) + \frac{[v_n(t) + v_n(t + \tau_n)]\tau_n}{2} - \frac{v_n(t + \tau_n)^2}{2b_n}. \quad [3-20]$$

After solving the inequality, the maximum speed that the subject vehicle can achieve is given by:

$$v_n(t + \tau_n) = \frac{1}{2}\tau_n b_n + \sqrt{\frac{1}{4}\tau_n^2 b_n^2 + v_n(t)\tau_n b_n + \frac{v_{n-1}(t)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t)}, \quad [3-21a]$$

or

$$v_n(t) = \frac{1}{2}\tau_n b_n + \sqrt{\frac{1}{4}\tau_n^2 b_n^2 + v_n(t - \tau_n)\tau_n b_n + \frac{v_{n-1}(t - \tau_n)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t - \tau_n)}. \quad [3-21b]$$

If the driver wants a buffer distance, d_n , and a relaxing time, θ_n , the inequality is rewritten as:

$$\begin{aligned} & x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1} - d_n \\ & \geq x_n(t) + \frac{[v_n(t) + v_n(t + \tau_n)]\tau_n}{2} + v_n(t + \tau_n)\theta_n - \frac{v_n(t + \tau_n)^2}{2b_n}, \end{aligned} \quad [3-22]$$

and it can be solved as:

$$v_n(t + \tau_n) \leq b_n \left(\frac{\tau_n}{2} + \theta_n \right) + \sqrt{\frac{1}{4}(\tau_n + 2\theta_n)^2 b_n^2 + v_n(t)\tau_n b_n + \frac{v_{n-1}(t)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} + 2b_n d_n - 2b_n \Delta x_{n-1,n}(t)}, \quad [3-23a]$$

or

$$v_n(t) \leq b_n \left(\frac{\tau_n}{2} + \theta_n \right) + \sqrt{\frac{1}{4}(\tau_n + 2\theta_n)^2 b_n^2 + v_n(t - \tau_n)\tau_n b_n + \frac{v_{n-1}(t - \tau_n)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} + 2b_n d_n - 2b_n \Delta x_{n-1,n}(t - \tau_n)} \quad [3-23b]$$

Gipps (1981) set $\theta_n = \frac{\tau_n}{2}$ and proposed the model of equation 2-24. d_n is not included in the model.

Hamdar and Mahmassani (2008) removed the relaxing time, (θ_n) , and defined D_n as the initial risk factor. Their model becomes model of equation 2-31.

Although the theories are sound, imaginary numbers are derived sometime when estimating with the real data. The imaginary numbers are caused by the square root in the models. To eliminate this problem, the square root can be replaced by a cube root. In order to include a cube root, the model is first simplified. Because the reaction time is short, the speed change during the reaction time can be ignored. Thus, the distance traveled by the subject vehicle during reaction time is $v_n(t)\tau_n$ instead of $\frac{[v_n(t) + v_n(t + \tau_n)]\tau_n}{2}$. The Gipps car-following model then can be derived from the following steps:

$$\begin{aligned}
x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1} &\geq x_n(t) + v_n(t)\tau_n - \frac{v_n(t + \tau_n)^2}{2b_n} \\
\Rightarrow \frac{v_n(t + \tau_n)^2}{2b_n} - v_n(t)\tau_n - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1} + x_{n-1}(t) - x_n(t) &\geq 0 \\
\Rightarrow \frac{v_n(t + \tau_n)^2}{2b_n} - v_n(t)\tau_n - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1} + \Delta x_{n-1,n}(t) &\geq 0 \\
\Rightarrow v_n(t + \tau_n)^2 - 2v_n(t)\tau_n b_n - v_{n-1}(t)^2 \frac{b_n}{b_{n-1}} - 2b_n l_{n-1} + 2b_n \Delta x_{n-1,n}(t) &\leq 0 \\
\Rightarrow v_n(t + \tau_n)^2 &\leq 2v_n(t)\tau_n b_n + v_{n-1}(t)^2 \frac{b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t) \\
\Rightarrow v_n(t + \tau_n) &\leq \sqrt{2v_n(t)\tau_n b_n + v_{n-1}(t)^2 \frac{b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t)} \\
\text{and} \\
v_n(t + \tau_n) &\geq -\sqrt{2v_n(t)\tau_n b_n + v_{n-1}(t)^2 \frac{b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t)}.
\end{aligned}$$

The second inequality is always true. So the car-following model can be written as:

$$v_n(t + \tau_n) = \sqrt{2v_n(t)\tau_n b_n + v_{n-1}(t)^2 \frac{b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t)}, \quad [3-24a]$$

or

$$v_n(t) = \sqrt{2v_n(t - \tau_n)\tau_n b_n + v_{n-1}(t - \tau_n)^2 \frac{b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n \Delta x_{n-1,n}(t - \tau_n)}. \quad [3-24b]$$

If the initial risk factor, (D_n) is included, the model is changed to:

$$\begin{aligned}
x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}} - l_{n-1} &\geq x_n(t) + v_n(t)\tau_n - \frac{v_n(t + \tau_n)^2}{2b_n} - D_n \\
\Rightarrow v_n(t + \tau_n) &\leq \sqrt{2b_n v_n(t)\tau_n + \frac{v_{n-1}(t)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n D_n - 2b_n \Delta x_{n-1,n}(t)}, \quad [3-25a]
\end{aligned}$$

or

$$v_n(t) = \sqrt{2b_n v_n(t - \tau_n) \tau_n + \frac{v_{n-1}(t - \tau_n)^2 b_n}{b_{n-1}} + 2b_n l_{n-1} - 2b_n D_n - 2b_n \Delta x_{n-1,n}(t - \tau_n)}. \quad [3-25b]$$

In all the existing models, the distance traveled by a vehicle during braking is $\frac{v_i(t)^2}{2b_i}$. It

is based on a constant braking rate. The time-speed relation is modeled as:

$$v = a - bt. \quad [3-26]$$

In reality, the braking rate of a vehicle is not constant as all the existing models assume.

With the pushing or releasing of the brake pedal, the braking rate changes. If the braking rate is modeled non-constantly, the model formation changes. A quadratic function is then tried first to model the time-speed relation as:

$$v = v(t) + b_1 t + b_2 t^2.$$

The braking/stopping distance is then calculated as:

$$\begin{aligned} T = (t|v=0) &= \frac{-b_1 \pm \sqrt{b_1^2 - 4b_2 v(t)}}{2b_2} \\ \Rightarrow \int_0^T v dt &= v(t)T + \frac{b_1}{2}T^2 + \frac{b_2}{3}T^3 \end{aligned}$$

Therefore the Gipps category car-following assumption is written as follows:

$$\begin{aligned} x_{n-1}(t - \tau_n) + \int_0^{T_{n-1}} v_{n-1} dt - l_{n-1} &\geq x_n(t - \tau_n) + v_n(t - \tau_n) \tau_n + \int_0^{T_n} v_n dt - D_n \\ \Rightarrow \int_0^{T_n} v_n dt &\leq s_n(t - \tau_n) + \int_0^{T_{n-1}} v_{n-1} dt - l_{n-1} + D_n - v_n(t - \tau_n) \tau_n \\ \Rightarrow v(t)T + \frac{b_1}{2}T^2 + \frac{b_2}{3}T^3 &\leq s_n(t - \tau_n) + \int_0^{T_{n-1}} v_{n-1} dt - l_{n-1} + D_n - v_n(t - \tau_n) \tau_n. \end{aligned}$$

That turns out to be too complex for the car-following model if solving for v_n . For the same reason, trigonometric functions also unhelpful.

Other method has to be tried. The next option is to change the braking/stopping distance.

It may be assumed that the braking/stopping is $\frac{v_i(t)^3}{B_i}$ instead of $\frac{v_i(t)^2}{2b_i}$, which is derived

from linear brake rates, where, B_i represents the braking characteristic of vehicle i. The unit

of B_i is $\frac{ft^2}{\text{sec}^3}$ and the new model can be derived as

$$\begin{aligned} \text{When } \int_0^{\tau_n} v_n dt &= \frac{v_n(t)^3}{B_n} \text{ and } \int_0^{\tau_{n-1}} v_{n-1} dt = \frac{v_{n-1}(t)^3}{B_{n-1}} \\ x_{n-1}(t - \tau_n) + \frac{v_{n-1}^3}{B_{n-1}} - l_{n-1} &\geq x_n(t - \tau_n) + v_n(t - \tau_n)\tau_n + \frac{v_n^3}{B_n} - D_n, \\ \Rightarrow v_n^3 &\leq B_n \left(\frac{v_{n-1}^3}{B_{n-1}} + \Delta x_{n-1,n}(t - \tau_n) - l_{n-1} + D_n - v_n(t - \tau_n)\tau_n \right) \end{aligned}$$

or

$$v_n(t) = \sqrt[3]{B_n \left[\frac{v_{n-1}(t - \tau_n)^3}{B_{n-1}} + \Delta x_{n-1,n}(t - \tau_n) - l_{n-1} + D_n - v_n(t - \tau_n)\tau_n \right]}. \quad [3-27]$$

So an improved Gipps car-following model with a cube root function is developed.

Comparing the existing Gipps category car-following models (i.e. see 2-24 and 2-31), the

proposed model is simpler and will not cause any imaginary number problem. The

performances of the proposed methods will be presented in the following chapters.

Chapter 4 Data Preparation

In this chapter, the data used in model estimation is introduced. Three datasets are used in this dissertation. The first two datasets include macroscopic traffic information (e.g. speed, occupancy, travel time) and are for macroscopic traffic modeling. The third one has vehicle trajectory data and is for microscopic modeling.

4.1 Traffic-stream Models and Shock Wave Based Traffic Modeling and Prediction

The wide deployment of Intelligent Transportation Systems (ITS) in the last decade has made it possible to collect a large amount of traffic information. The dataset used here is published by the Texas Department of Transportation (TxDOT). The system, TransGuide, collects traffic data throughout the highway network in San Antonio, Texas. At every location, several loop detectors collect data from each lane.

Before the data provided by the TransGuide can be used in this research, two quality control processes are applied to ensure data integrity and correctness. First, the dataset sometime shows the occupancy as zero while the speed is nonzero. These highly suspect data account for less than 1% of the entire dataset and are removed from the analysis. Second, the loop detectors used in this study is set to collect data every 40 seconds. However, many data collection intervals are not exactly 40 seconds, but have several seconds more or less.

A computer program is then developed to aggregate the original data to 40 seconds intervals using linear interpolation.

Occupancy, flow, and time mean speed are derived from loop detectors and reported to the ATMS. The lane occupancy, ρ^o is defined as (May 1990; Khisty and Lall 2003):

$$\rho^o = \frac{1}{T} \sum_{i=1}^n t_i, \quad [4-1]$$

where, T is the total observation time (40 seconds), t_i is the time that a detector is occupied by the i th vehicle, and n is the number of vehicles detected by the detector. The time mean speed, u_t is defined as the arithmetic mean of the measured speeds of all vehicles passing a fixed point during a given interval of time (40 seconds), as specified in equation 4-2:

$$u_t = \frac{1}{n} \sum_{i=1}^n v_i, \quad [4-2]$$

where, v_i is the spot speed of the i th vehicle.

To calculate density, k , the following formula is used:

$$k = \frac{q}{v}. \quad [4-3]$$

Although the space mean speed should be used in equation 4-3 for computing density, it is the time mean speed obtained from equation 4-2 that is used in this research. The reason time mean speed is utilized is that space mean speed is difficult to measure and the difference between these two speeds to be very small in this study. According to Wardrop (1952), the

time mean speed equals the space mean speed, plus the ratio of variance of the space mean speed, over the time mean speed. Since the time mean speeds are collected every 40 seconds, it is reasonable to assume that the variance of the space mean speed within 40 seconds is very small. Thus, the space mean speed can be approximated to the time mean speed.

The traffic dataset used in this study was collected from the middle lanes of Interstate Highway 35 (I-35) northbound (loop detector station L2-0035N-162.482) between 5:30pm and 7:00pm on Monday, July 9, 2001. Both free-flow patterns and congestion patterns are included in the data.

The scattered dots in Figure 4-1 shows the raw data of speed against density.

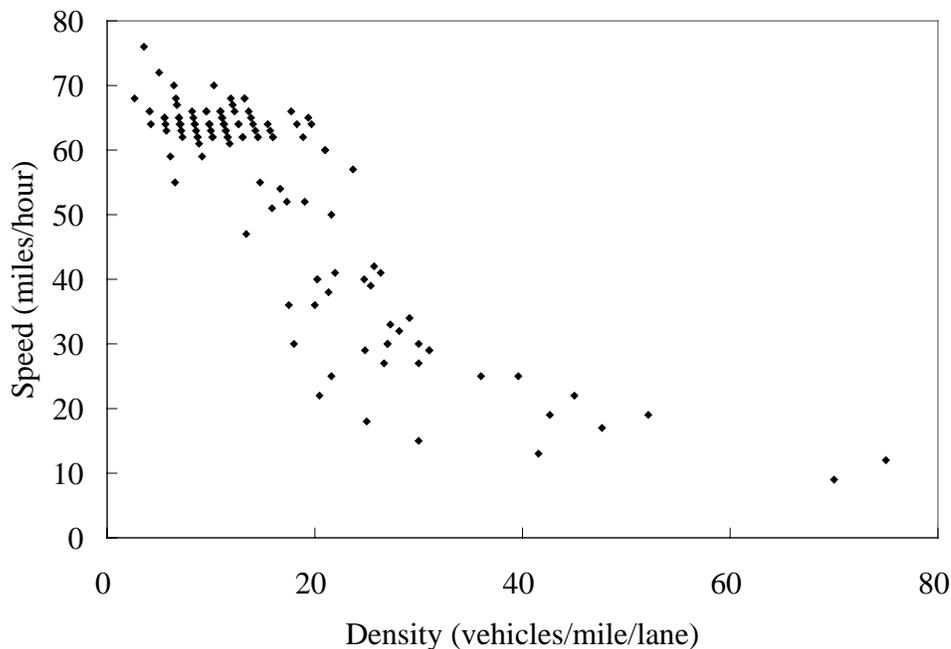


Figure 4-1 A raw data of speed against density

For the shock wave simulation, the data collected from I-35 northbound on Tuesday, July 10th, 2001 is used. All the data from the three consecutive detector stations (0035N-159.998, 0035N-160.504, and 0035N-160.892) are adopted. Each detector station contains three traffic detectors monitoring all three lanes along I-35. The data from the three lanes are averaged to represent the traffic information at the detector stations.

4.2 Travel Time Estimation

In this research, real travel time data is obtained from an experiment conducted on Thursday, May 26th, 2005. The location of the experiment is a section of Interstate Highway 66 (I-66) eastbound in Fairfax County, Virginia in the Washington, D.C. area.

Both video cameras and an instrumented vehicle are used to measure the real travel time. Three video cameras were set up on three bridges overlooking I-66. The first video camera (counting from upstream to downstream) was located at the Vienna/Fairfax-GMU Metrorail Station. The second was placed at the Dunn Loring-Merrifield Metrorail Station. And the third was at the West Falls Church Metrorail Station. The distance between the first and the second cameras was 2.38 miles, and the distance between the second and the third was 2.44 miles. The distances are measured with the on-board odometer in the instrumented vehicle. The video record is then digitized using video-capture equipment, with a precision of 1/30 seconds (i.e., 30 frames per second). An example frame that was retrieved from the video is shown in Figure 4-2 below. Each travel time is calculated from video frames by identifying

the same vehicle in all three locations. The time that a specific vehicle passes a reference point is measured from the time information displayed on the frame. The method for calculating the speed of an individual vehicle is similar to the method described in Wei et al. (2005), but the dashed pavement marking lines were used as the distance references. The average speed of three consecutive vehicles in the same lane is used as the point speed in the proposed method.



Figure 4-2 A video frame taken from travel time experiment

Speeds and travel times of the instrumented vehicle are obtained from the method mentioned above as well as from an on-board speed-meter and a watch. In total, nine real travel times are obtained using the instrumented vehicle. The relative error of travel times using the instrumented vehicle and the video cameras are all less than 5%, suggesting consistency between these two experimental methods.

4.3 Improved Gipps Car-following Model

Similar to the travel time data collection, car-following data can be obtained both directly and indirectly. The direct method involves using an instrumented vehicle. However, it is typically done with a small sample size. The other problem with this method is that driver behavior may be biased because drivers may know they are being recorded. The indirect method is to record traffic flow with video cameras, whereby vehicle trajectory data and car-following data can be retrieved from the video record. The recent availability of high-quality video cameras and advanced computers has now made it feasible to retrieve more accurate vehicle trajectory data from video records.

The dataset used in this study is published by Federal Highway Administration (FHWA) with its Next Generation SIMulation program (NGSIM). Vehicle trajectory data is collected using video cameras and object recognition software. The data is generated at every 0.1 second, with accuracy of one foot or less in position. The data is collected from several freeways and local streets. The one that is adopted in this study is from a section of Interstate Highway 80 (I-80) in Emeryville, California (near Oakland). Seven video cameras were mounted on the top of a 30-story building, Pacific Park Plaza. The building address is 6363 Christie Avenue and is adjacent to I-80. The video cameras cover 1,650 feet in length with an on-ramp at Powell Street. The off-ramp at Ashby Avenue is just downstream of the study area. There are six lanes, including one high-occupancy vehicle (HOV) lane. The dataset contains

over 15 minutes of vehicle trajectories, from 3:58:55 p.m. to 4:15:37 p.m., on Wednesday, April 13, 2005. Both low and high speed vehicles are included. The automated vehicle tracking system was developed by the University of California at Berkeley. The system automatically detects and tracks approximately 75 percent of the vehicles. In general, tracking accuracy is higher than detection accuracy. When vehicles are not automatically detected, they can be manually detected and automatically tracked (Alexiadis, V., et al, 2004).

Table 4-1 below summarizes the NGSIM data dictionary.

Table 4-1 NGSIM trajectory data dictionary (FHWA)

Name	Unit	Description
Vehicle ID	Number	Vehicle identification number (ascending by time of entry into section)
Frame ID	0.1 second	Frame identification number (ascending by start time)
Total Frames	0.1 second	Total number of frames in which the vehicle appears in this dataset
Global Time	Milliseconds	Elapsed time since January 1, 1970
Local X	Feet	Lateral (X) coordinate of the front center of the vehicle with respect to the left-most edge of the section in the direction of travel
Local Y	Feet	Longitudinal (Y) coordinate of the front center of the vehicle with respect to the entry edge of the section in the direction of travel
Global X	Feet	X coordinate of the front center of the vehicle based on California State Plane III in NAD83
Global Y	Feet	Y coordinate of the front center of the vehicle based on California State Plane III in NAD83
Vehicle Length	Feet	Length of vehicle
Vehicle Width	Feet	Width of vehicle
Vehicle Class	Text	Vehicle type: 1 – motorcycle, 2 – auto, 3 – truck

Table 4-1 (Continued) NGSIM trajectory data dictionary (FHWA)

Name	Unit	Description
Vehicle Velocity	Feet/Second	Instantaneous velocity of vehicle
Vehicle Acceleration	Feet/Second Square	Instantaneous acceleration of vehicle
Lane Identification	Number	Current lane position of vehicle. Lane 1 is the farthest left lane; lane 6 is the farthest right lane. Lane 7 is the on-ramp at Powell Street, and Lane 9 is the shoulder on the right-side.
Preceding Vehicle	Number	Vehicle ID of the lead vehicle in the same lane. A value of '0' represents no preceding vehicle - occurs at the end of the study section and off-ramp due to the fact that only complete trajectories were recorded by this data collection effort (vehicles already in the section at the start of the study period were not recorded).
Following Vehicle	Number	Vehicle Id of the vehicle following the subject vehicle in the same lane. A value of '0' represents no following vehicle - occurs at the beginning of the study section and on-ramp due to the fact that only complete trajectories were recorded by this data collection effort (vehicle that did not traverse the downstream boundaries of the section by the end of the study period were not recorded).
Spacing	Feet	Spacing provides the distance between the front-center of a vehicle to the front-center of the preceding vehicle.
Headway	Seconds	Headway provides the time to travel from the front-center of a vehicle (at the speed of the vehicle) to the front-center of the preceding vehicle. A headway value of 9999.99 means that the vehicle is traveling at zero speed (congested conditions).

To use the NGSIM dataset for the microscopic traffic modeling, a computer program is developed with Visual Basic programming language (VB) to process the original published

dataset. In the original dataset, each file contains one vehicle's information. That one file might have more than one car-following event because the car in front may be changed. The program is used first to restructure the original dataset into car-following files instead of vehicle files by identifying each car-following scenario. The program can also be used to retrieve additional information from the original dataset, so that more factors can be tested (e.g. density, second leading vehicles, and vehicles on the adjacent lanes).

In total, there are 2,052 vehicles that entered the roadway section during this time. The number of car-following scenarios is 5,284. In this study, 36 car-following scenarios are randomly chosen for the model fitting. Among the 36 files, time range of car-following behaviors is from 9.5 seconds to 60.7 seconds, with the average time at 23.7 seconds.

Chapter 5 Experimental Results

This chapter presents the results for the proposed methodologies using the data described in chapter four. The testing and verification methods are also introduced. The comparisons of the results with those from the existing methods are discussed.

5.1 Traffic-Stream Model

Since a traffic-stream model involves only two arguments (i.e. speed and density), the fitting becomes a univariate regression. Commonly used approaches include a global parametric method and a nonparametric method.

Global parametric regression (GPR) assumes one parametric function and model parameters are often obtained by least squares estimates or maximum likelihood estimates. This type of model is called single-regime model and has been reviewed in Chapter 2.

The nonparametric method is further comprised of three related paradigms: local parametric regression (LPR), roughness penalty regression (RPR), and piecewise parametric regression (PPR) (Friedman 1991; Hastie et al. 2001).

LPR differs from GPR in that the parameter values in the former are generally different at each evaluation point, k , and are obtained by locally weighted least square fitting

(Friedman 1991; Hastie et al. 2001). The weight function, $w(k, k')$ is chosen to place the dominant mass on point, k' , close to k . The properties of the approximation are determined mostly by the choice of w and to a lesser extent by the particular local parametric function used. Although it is shown that, asymptotically, higher-order polynomials (e.g., local linear fitting, local quadratic fitting) can have superior convergence rates when used with simple weight functions (Stone 1977; Cleveland 1979; Cleveland and Devlin 1988), the most commonly studied local parametric function is the simple constant (Parzen 1962; Friedman 1991). The most common choice of weight function takes the form of (Hastie et al. 2001):

$$w(k, k') = K_K(|k - k'|/b(k)), \quad [5-1]$$

where, $|k - k'|$ is a possibly weighed distance between k' and k , $b(k)$ is a scale factor (bandwidth), and K_K is a kernel function of a single argument. The kernel is usually chosen so that its absolute value decreases with the increasing value of its argument. Commonly used scale functions are a constant, $b(k) = b_0$ (kernel smoothing) or $b(k) = b_0 / \hat{p}(k)$ (near neighbor smoothing), where $\hat{p}(k)$ is some estimate of the local density of the design point.

RPR are defined as

$$\hat{u}_e(k, \lambda) = \arg \min_{u_e} \left\{ \sum_{i=1}^N [u_{e,i} - u_e(k_i)]^2 + \lambda R(u_e) \right\}, \quad [5-2]$$

where, $R(u_e)$ is a penalty function that increases with increasing roughness of the function, $u_e(k)$. The minimization is performed over all u_e for which $R(u_e)$ is defined. The

parameter, λ , regulates the tradeoff between the roughness of, u_e and its fidelity to the data. The properties of RPR are similar to those of LPR. The popular cubic smoothing spline for one-dimensional inputs use the integrated squared Laplacian as the roughness function (Hastie et al. 2001):

$$R(u_e) = \int [u_e'(k)]^2 dk. \quad [5-3]$$

The penalty, R , corresponds to a log-prior, $\hat{u}_e(k, \lambda)$, to the log-posterior distribution, and minimizing, $\hat{u}_e(k, \lambda)$ amounts to finding the posterior mode (Hastie et al. 2001).

For model simplicity, the number of regimes of a multi-regime model should be limited. As a result, LPR and RPR are not good choices because model parameters developed using these methods vary at every point of density, k . Finally, PPR is chosen for developing traffic-stream models. PPR is used to approximate the function, u_e , by several simple parametric functions (usually low-order polynomials). PPR contains GPR as a special case in which a single parametric model covers the entire region of interest. The most popular piecewise polynomial fitting procedures are based on splines, where the parametric functions are taken to be polynomials of degree, q and derivatives of order $q-1$ are required to be continuous. For instance, Sun and Zhou (2005a) used quadratic functions ($q=3$), the most popular choice in the practice of spline regression to develop multi-regime traffic-stream models.

Unknown knots plus model coefficients in equation 3-3 are parameters to be estimated. In many cases the calculation of the estimated parameters is possible with a mathematically derived formula, such as in linear regression. However, in many interesting instances this is not possible. Parameter estimation of a single-regime model using the least square error can be implemented by solving a nonlinear programming problem. Classical gradient-based nonlinear optimization methods such as the steepest-ascent algorithm is difficult to apply directly, because parameter estimation of two-regime or three-regime models (using the least square error as the criterion) involves the estimation of knot position, which is discontinuous for taking the derivative. Furthermore, a new derivation of the solution is necessary every time the model is varied to change the knot positions. Genetic Algorithm (GA) can be a good approach for these complex applications.

For the sake of computational efficiency and ease of implementation, model coefficients and knots are estimated respectively in two phases. In the first phase, the knots are estimated using GA; and, in the second phase, model coefficients are estimated using the steepest-ascent algorithm. That is, the adaptive regression iteratively searches for the locations of knots while minimizing the global optimization objective function through the following steps: (1) applying cluster analysis to obtain an initial estimate of knot positions for a specified number of sub-regimes; (2) for a combination of basis functions for different sub-regimes, the GA and steepest ascent method are used respectively for estimating optimal

knots and model coefficients; and (3) all candidate models are compared using the goodness-of-fit criterion, and the best one is automatically selected for representing the traffic-stream model. The advantage of this hybrid approach is that GA can conveniently deal with discontinuous functions like equation 3-5 and does not require gradient information to be known in order to search the best estimate of knots, while the classical gradient-based optimization algorithm is very efficient in terms of finding the optimal solution when gradients of the objective function can be analytically formulated and easily evaluated.

The steepest ascent method for model coefficient estimation can be found in many standard references of nonlinear optimization (Winston 1993; Hillier and Lieberman 2005), and therefore is not outlined in detail here. This method requires the gradient of the objective function to be evaluated. Using basis functions in equation 3-3, the gradients of the objective function in equation 3-7 for each regime can be very complicated because they are composed of gradients of sum-of-squared error and sum-of-squared differences at discrete knots. Since basis functions used for two adjacent regimes might differ from each other, the total number of combination of gradients for the entire regime in adaptive regression can be large. Hence, the detailed gradient information is not presented in this dissertation. However, the basic components consisting of these gradients are dependent upon the specific type of basis functions used in the multi-regime traffic-stream model (see Appendix I).

Starting values of knots are necessary for initiating the GA program. Since good starting values can save unnecessary GA iterations, a wise strategy has to be used to initialize the knot positions. In multivariate adaptive regression spline (Friedman 1991), the regression process is implemented by optimally and recursively partitioning the regime using the goodness-of-fit criterion, and generating an increasing number of knots. The resulting sub-regimes are then recombined in a reverse manner until an optimal set is achieved, based on a criterion that penalize both the lack-of-fit and the increased number of regimes (Breiman, et al. 1984; Friedman 1991).

In this study, a different strategy is used. Specifically, cluster analysis, an unsupervised learning technique, is applied in order to split data into M regimes (classes). The breakpoints between two consecutive regimes are adopted as starting knots. The grouping of the patterns is accomplished through clustering by defining and quantifying dissimilarities between the individual data points or patterns. The patterns that are similar to the highest extent are assigned to the same cluster. The K-means algorithm, one of the most popular cluster analysis methods, is used here to segment datasets. A detailed description of the application of cluster analysis can be found in Sun and Zhou (2005a).

A gene in GA is a string of bits representing a possible solution. In this study, a population of 50 genes is initially created in a random manner. Decision variables are first multiplied by, 10^n , where n is the number of comma-places (or the desired precision). Then,

the amplified decision variables are converted to binary form. The desired bit strings are binary representations of these amplified decision variables. The length of the bit string depends on the problem to be solved. In this study, 36 bits are used for each gene. For indicating the sign of a decision variable, one can either add the lowest allowed value to the variable and transform the variables to positive ones, or reserve one bit per variable for the sign.

GA consists of three processes: selection, mating and crossover, and mutation (Goldberg 1989). Selection is used to extract a subset of genes from a predecessor population of genes, according to a defined fitness. The fitness function used in this study is sum of squared error (SSE) value as defined in equation 3-7. Genes producing smallest SSE values have a higher likelihood of being chosen for the next generation. The next steps in creating a new population of genes are mating and crossover. The following mating and crossover strategy is used here: first, 30 percent of a new population of genes (after selection) will be randomly mated in pairs; secondly, a crossover point (Figure 3-1) will be randomly chosen for each pair; and finally, the information after the crossover-point will be exchanged between the two genes of each pair. Mutation means that each bit in the resulting gene has a defined probability (10% in this study) of getting inverted.

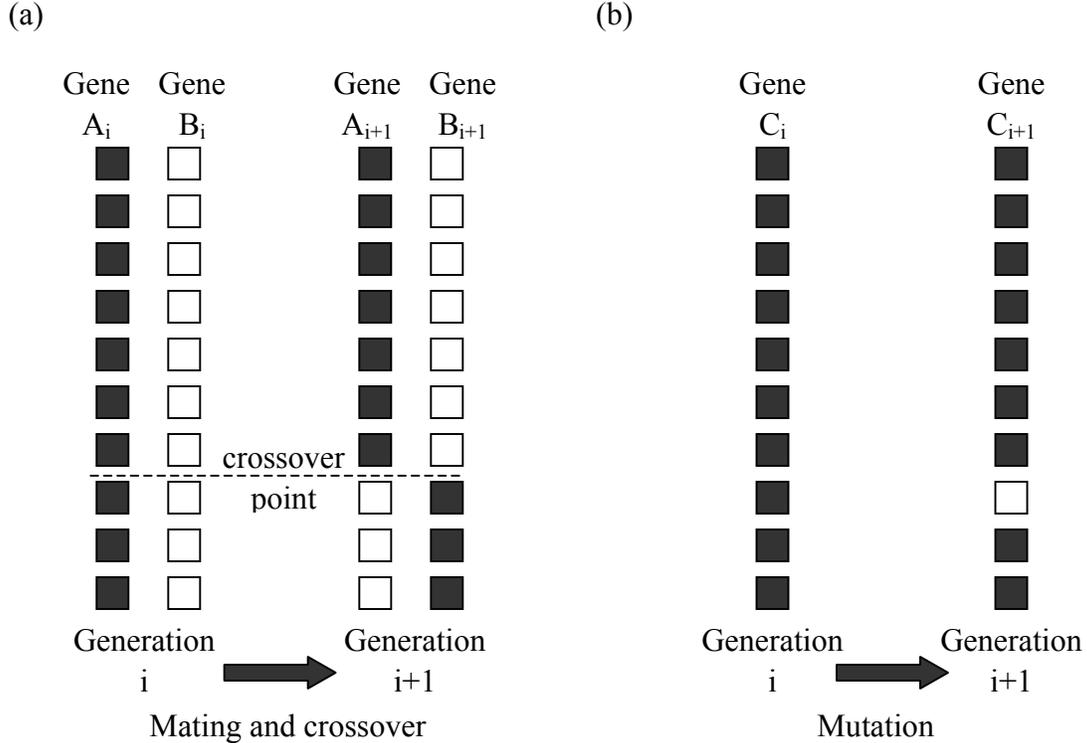


Figure 5-1 Mating and crossover as well as mutation in GA: (a) mating and crossover; (b) mutation.

After the mutation step, convert the bit string of each gene back to a decimal representation of the decision variable and the test SSE values. If SSE is small enough, then terminate the iteration. Otherwise, restart the algorithm and go through all three processes until a minimum SSE is achieved.

GA can be slow, particularly, for problems involving a large number of decision variables. In this study, because there are only very few decision variables (e.g. positions of knots) that need to be estimated using GA and other model parameters are estimated using the steepest ascent method, the proposed two-phase algorithm is very efficient. By trial and

error, it is observed that variations after 30 generations is very small and can be ignored and, therefore, 30 is the number of generations that will be used in this research. The above procedure for parameter estimation and knot selection has been programmed into a computer program using Visual Basic programming language (VB). The program can provide an estimate of all parameters almost instantaneously.

While developing single-regime models, the steepest-ascent algorithm is used for estimating parameters in basis functions, B_1, \dots, B_6 , while a genetic algorithm is used to calibrate parameters in basis function, B_7 . Figure 4-1 shows single-regime traffic-stream models using seven basis functions, respectively. The symbols 1, ..., 7 in this figure represents basis functions, B_1, \dots, B_7 . Table 5-1 gives estimated parameters and model error in terms of mean residual squares, which is the sum of differences between predicted and observed speed, divided by the number of total observations. From this table it can be seen that basis function, B_7 , has the least mean residual square of 46.13. The last model in Table 5-1 represents the best-fitting, single-regime traffic-stream model, with the least mean residual square of 46.13.

Table 5-1 Single-regime traffic-stream models

Basis Function	One-regime Stream Methods	Mean Residual Square
1	$u_e = 73.53 - 1.19\rho$	73.58
2	$u_e = 21.29 \ln(170.73 / \rho)$	88.90
3	$u_e = 83.33 \exp(\rho / 34.25)$	64.73
4	$u_e = 69.44 \exp(23.64 / \rho^2)$	54.54
5	$u_e = 72.08 [1 - (\rho / 229.51)^{1.60}]^{19.92}$	55.68
6	$u_e = 81.01 - 1.97\rho + 0.01\rho^2$	60.62
7	$\rho = (0.09 + \frac{9.45}{64.96 - u_e} + 1.74u_e)^{-1}$	46.13

While developing two-regime and three-regime traffic-stream models, the dataset was first sorted by density in ascending order. The dataset is then partitioned into two or three sub-datasets using GA. For each regime, the four basis functions are applied to fit the speed-density relation using the steepest-ascend algorithm, while the genetic algorithm is simultaneously used for searching the optimal knots (i.e. breakpoints).

Continuity conditions can be imposed at knots for two-regime or three-regime traffic-stream models. For illustration purposes, models with and without continuity are considered in this case study. Figures 5-2 and 5-3 present two-regime traffic-stream models with and without continuity. The symbols, for example, “1+4” used in these figures mean that basis function, B_1 and B_4 are used for fitting the first and the second regimes, respectively.

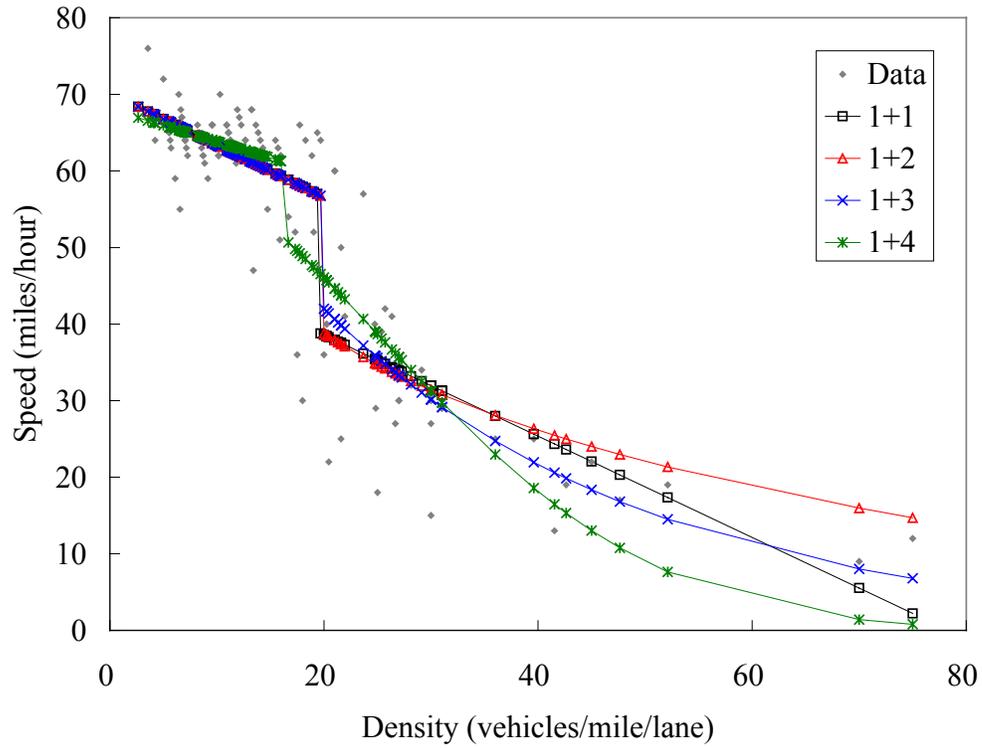


Figure 5-2a Two-regime traffic-stream models without continuity (using B1 for the first regime)

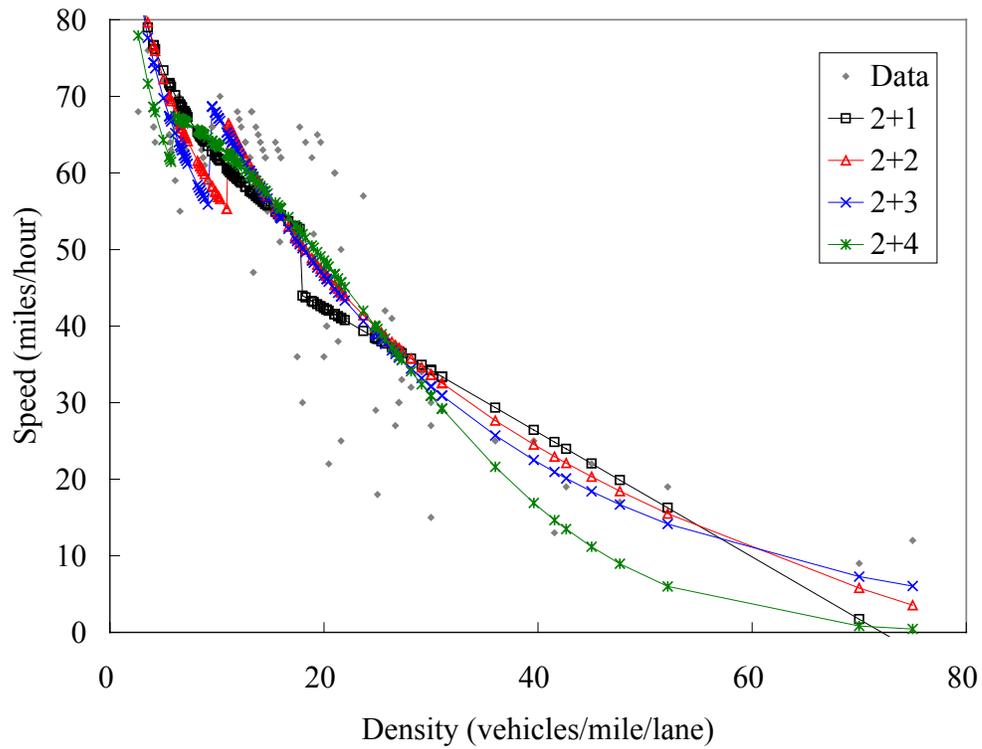


Figure 5-2b Two-regime traffic-stream models without continuity (using B2 for the first regime)

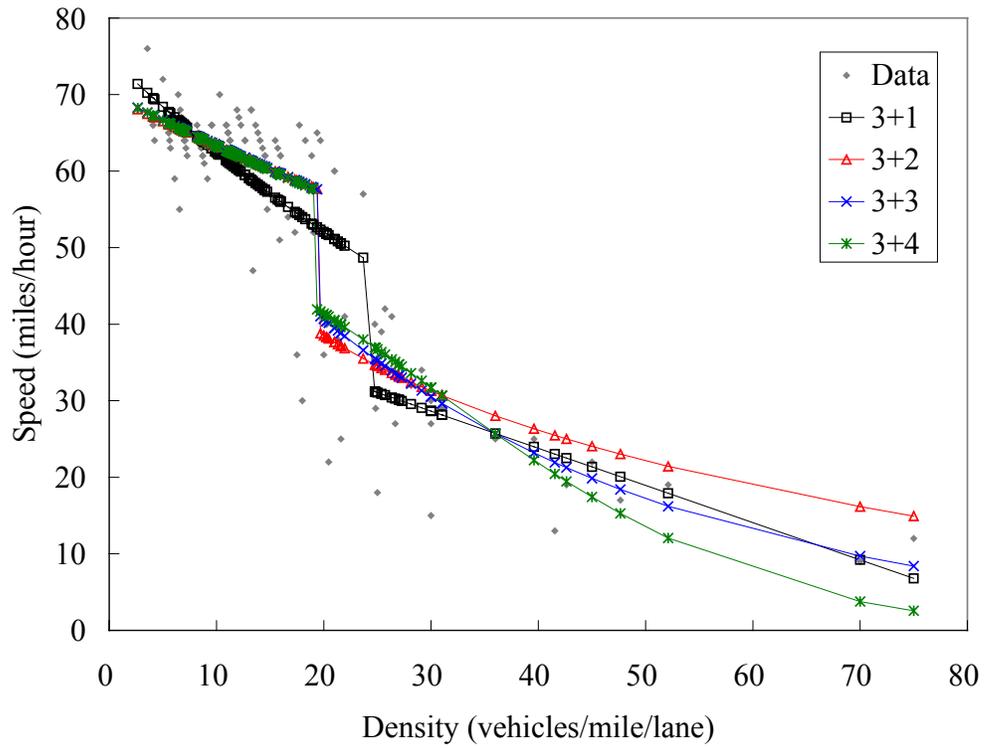


Figure 5-2c Two-regime traffic-stream models without continuity (using B3 for the first regime)

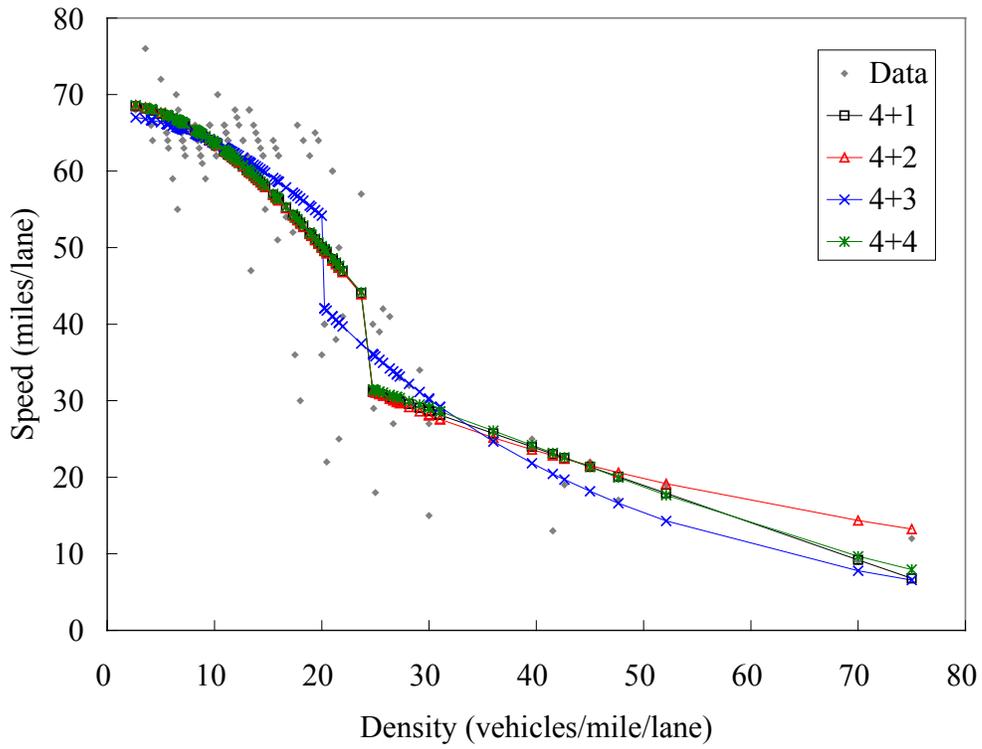


Figure 5-2d Two-regime traffic-stream models without continuity (using B4 for the first regime)

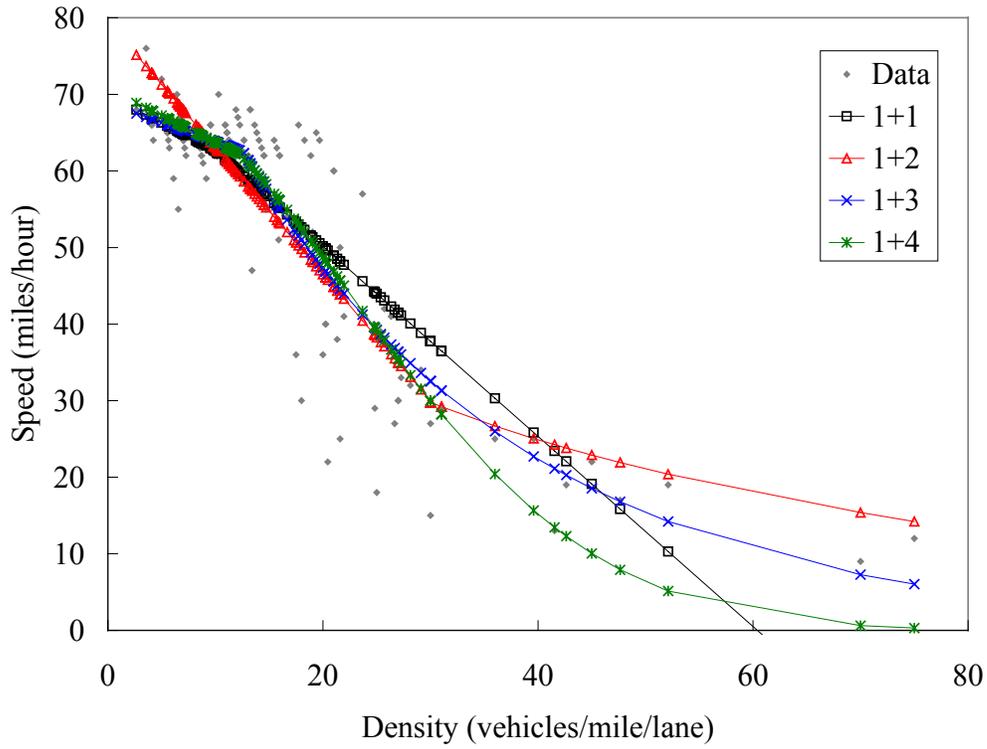


Figure 5-3a Two-regime traffic-stream models with continuity (using B1 for the first regime)

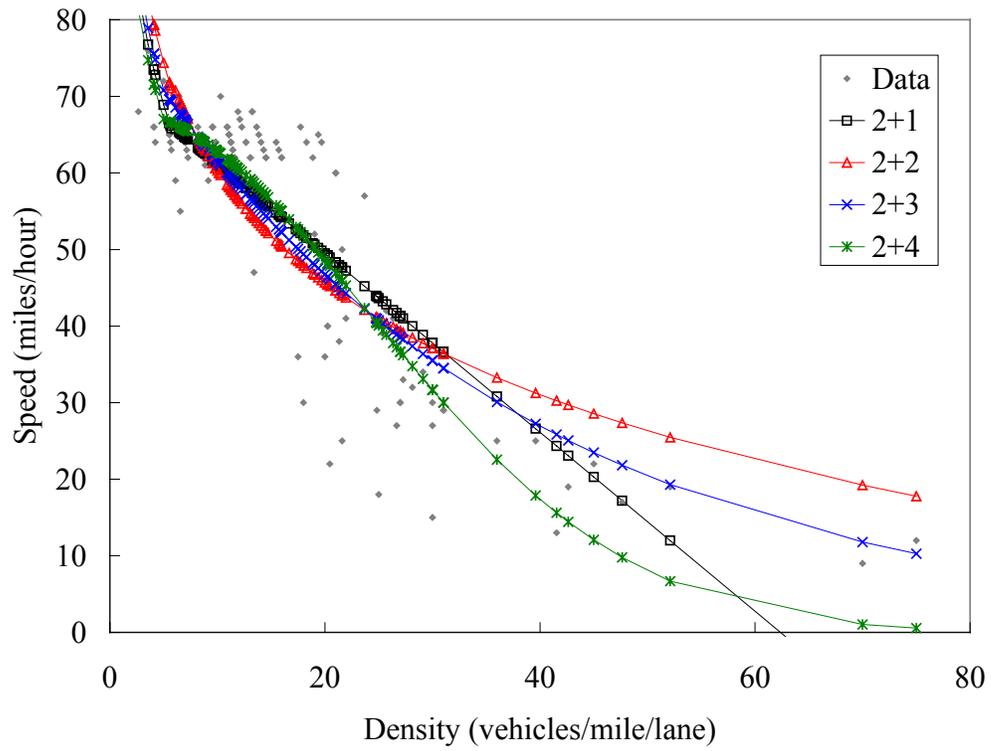


Figure 5-3b Two-regime traffic-stream models with continuity (using B2 for the first regime)

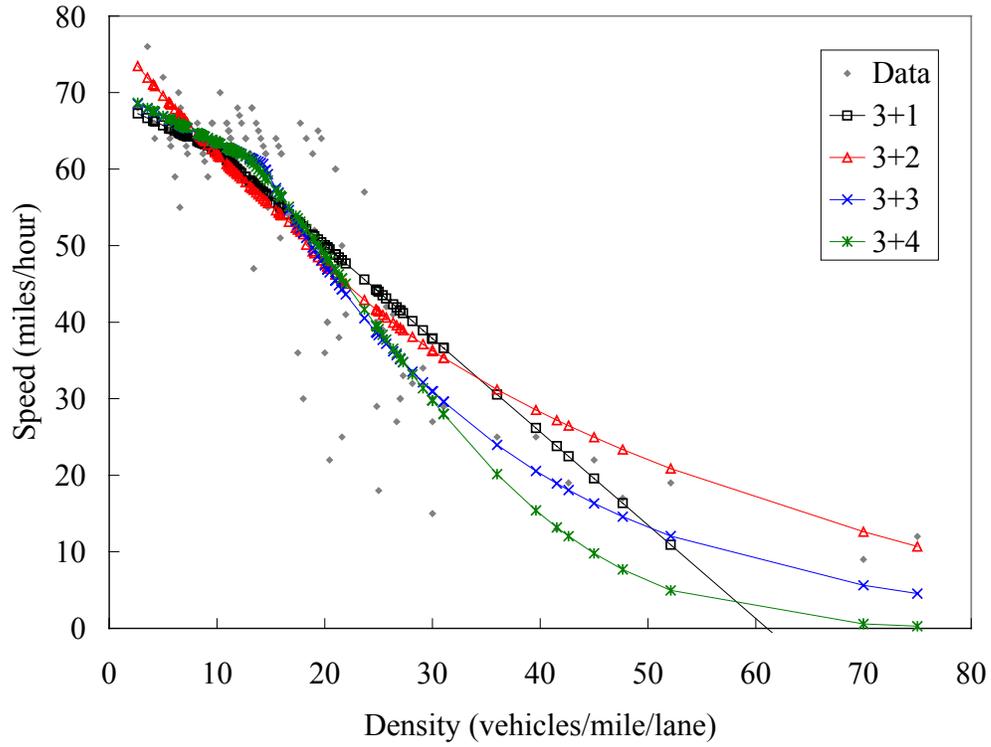


Figure 5-3c Two-regime traffic-stream models with continuity (using B3 for the first regime)

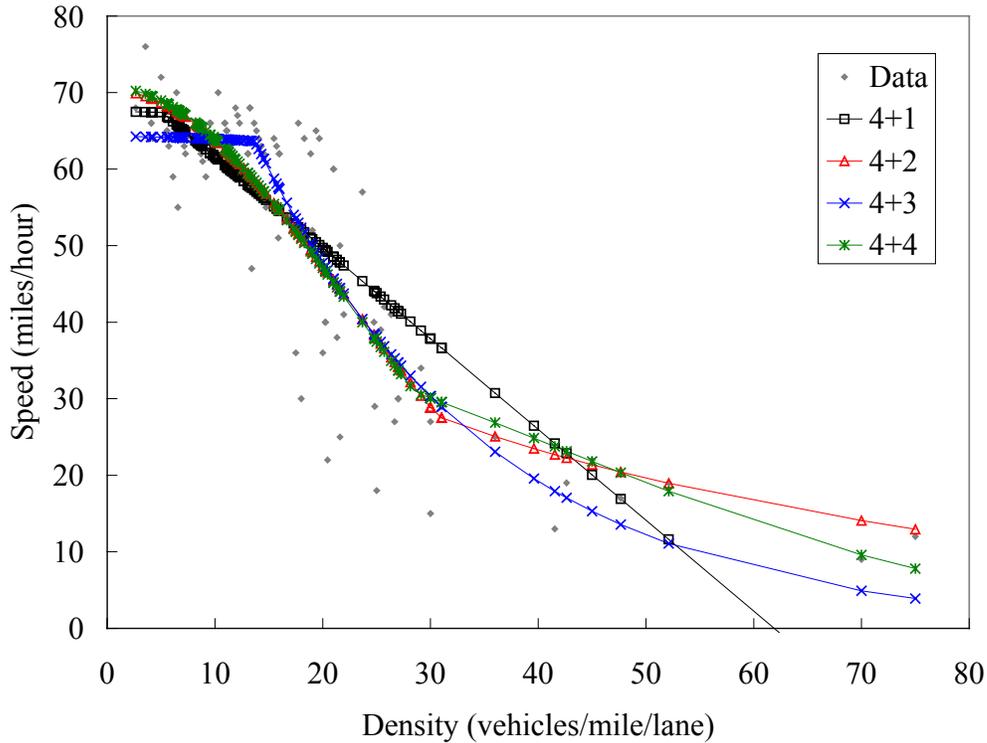


Figure 5-3d Two-regime traffic-stream models with continuity (using B_4 for the first regime)

Table 5-2 shows the mean residual squares of adaptive regression of two-regime models, with and without continuity. The minimum least mean residual square of two-regime models without continuity is 43.22. This model is specified in equation 5-4, which consists of basis function, B_1 for the first regime, and B_3 for the second regime. The minimum least mean residual square of two-regime models with continuity is 47.18. This model is specified in equation 5-5, and consists of basis function B_4 for the first regime and B_3 for the second regime.

$$u = \begin{cases} 70.19 - 0.68k & \text{for } k \leq 19.81 \\ 81.49 \exp(k/30.19) & \text{for } 19.81 \leq k \end{cases} \quad [5-4]$$

$$u = \begin{cases} 64.27 \exp(97.95/k^2) & \text{for } k \leq 13.80 \\ 118.80 \exp(k/21.96) & \text{for } 13.80 \leq k \end{cases} \quad [5-5]$$

Table 5-2 Mean residual squares of two-regime traffic-stream models

Mean Residual Square (models without continuity)		Basis Function of the Second Regime			
		1	2	3	4
Basis Function of the First Regime	1	46.00	46.02	43.22	49.75
	2	68.98	63.86	55.80	55.15
	3	51.60	46.29	43.57	48.39
	4	47.44	47.47	45.31	47.89
Mean Residual Square (models with continuity)		Basis Function of the Second Regime			
		1	2	3	4
Basis Function of the First Regime	1	73.32	56.96	48.88	52.78
	2	76.35	89.22	68.62	56.51
	3	73.25	62.93	47.54	52.67
	4	73.62	50.04	47.18	50.67

For three-regime model, the combination of basis functions can be large ($4 \times 4 \times 4 = 64$). Therefore, the detailed plot and tabular information is not presented, and data is limited to a list of the optimal three-regime traffic-stream models. The minimum least mean residue square of three-regime models without continuity is 40.82 and given by equation 5-6, which consists of basis function, B_1 , B_4 , and B_3 for the first, second, and third regimes, respectively. The minimum least mean residue square of three-regime models with continuity is 48.71 and is given by equation 5-7, which consists of basis function, B_1 , B_4 , and B_3 for the first, second, and third regimes, respectively.

$$u = \begin{cases} 70.19 - 0.68k & \text{for } k \leq 19.76 \\ 43.16 \exp(101.21/k^2) & \text{for } 19.76 \leq k \leq 24.76 . \\ 56.69 \exp(k/43.40) & \text{for } 24.76 \leq k \end{cases} \quad [5-6]$$

$$u = \begin{cases} 75.09 - 1.67k & \text{for } k \leq 6.48 \\ 64.44 \exp(100.40/k^2) & \text{for } 6.48 \leq k \leq 11.63 . \\ 99.03 \exp(k/26.96) & \text{for } 11.63 \leq k \end{cases} \quad [5-7]$$

5.2 Shock Wave Based Traffic Flow Modeling and Prediction

To estimate the traffic status between two traffic detector stations, the shock wave simulation program with two traffic detector stations is performed. In this study, the estimation point is set to the location of traffic detector 2, so that the estimation results can be compared to the data measured by traffic detector 2. The results are shown in Figure 5-4 to Figure 5-6. To make it clearer, only data between 16:55 to 18:07 are shown in the figures because both congestion and non-congestion situations are included in this period of time. For the dataset used in this study, the estimated speed is 1.7 miles per hour less than the average of measured speed; the estimated volume is 36 vehicles per hour less than the average of measured volume; the estimated density is 0.4 vehicles per mile less than the average of density from measured occupancy calculated. The results suggest that the shock wave simulation method is capable of estimating traffic information with acceptable error.

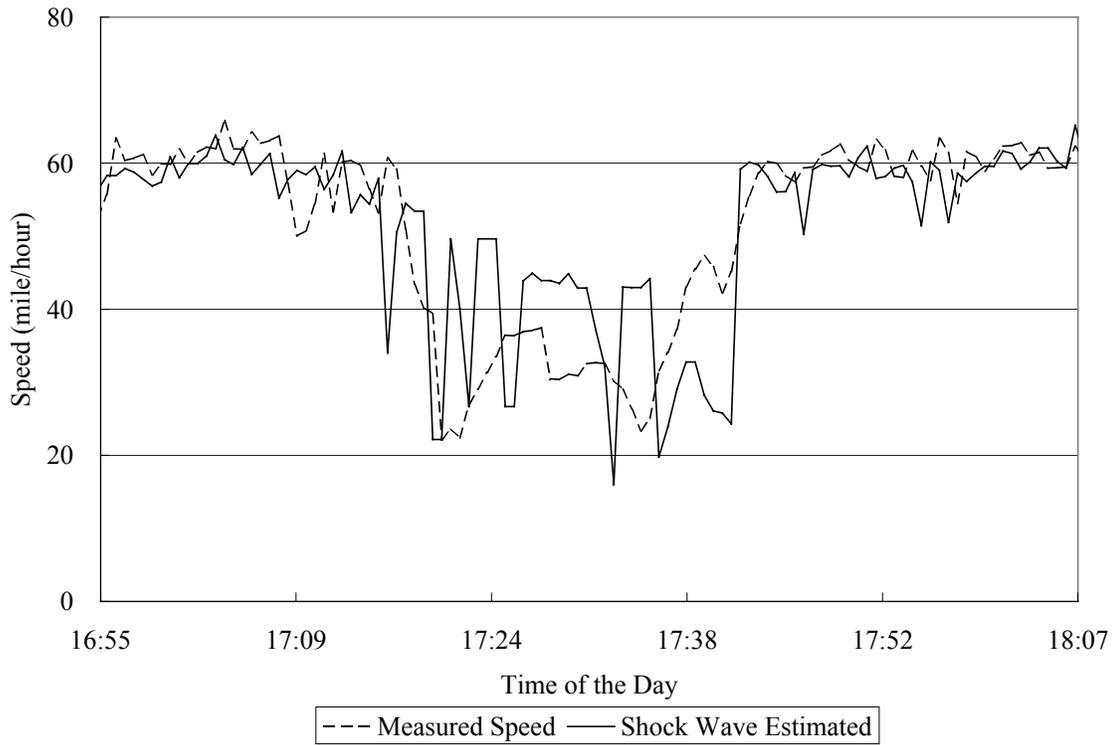


Figure 5-4 Speed estimation results

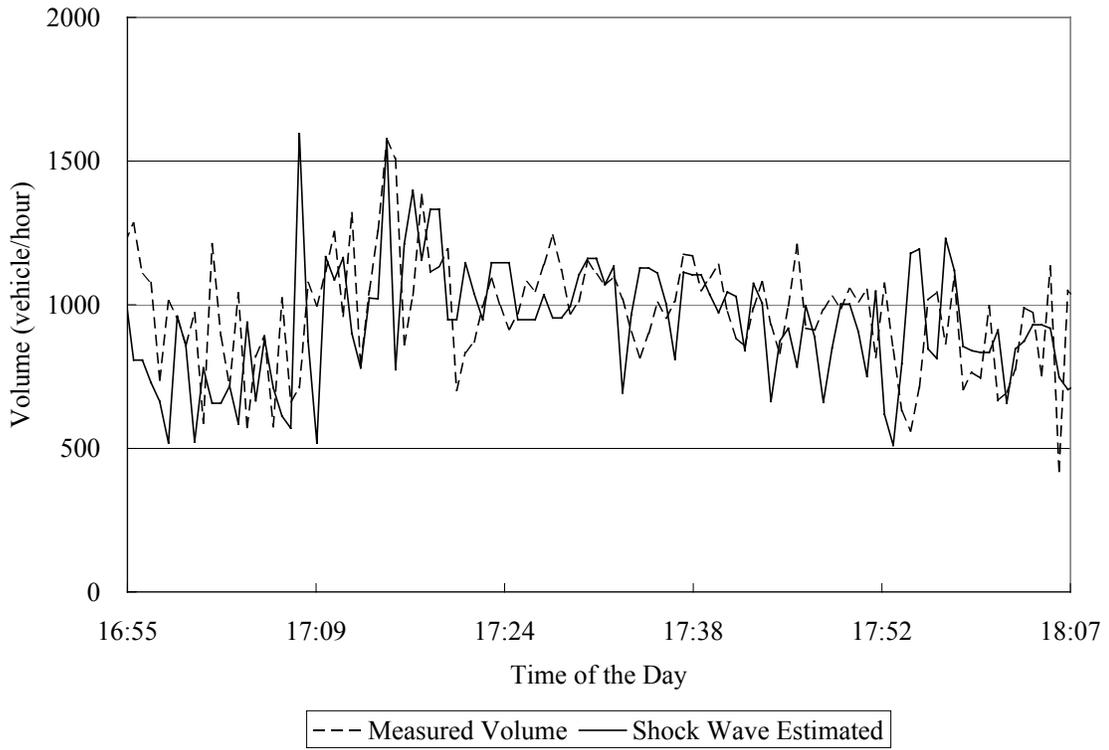


Figure 5-5 Traffic volume estimation results

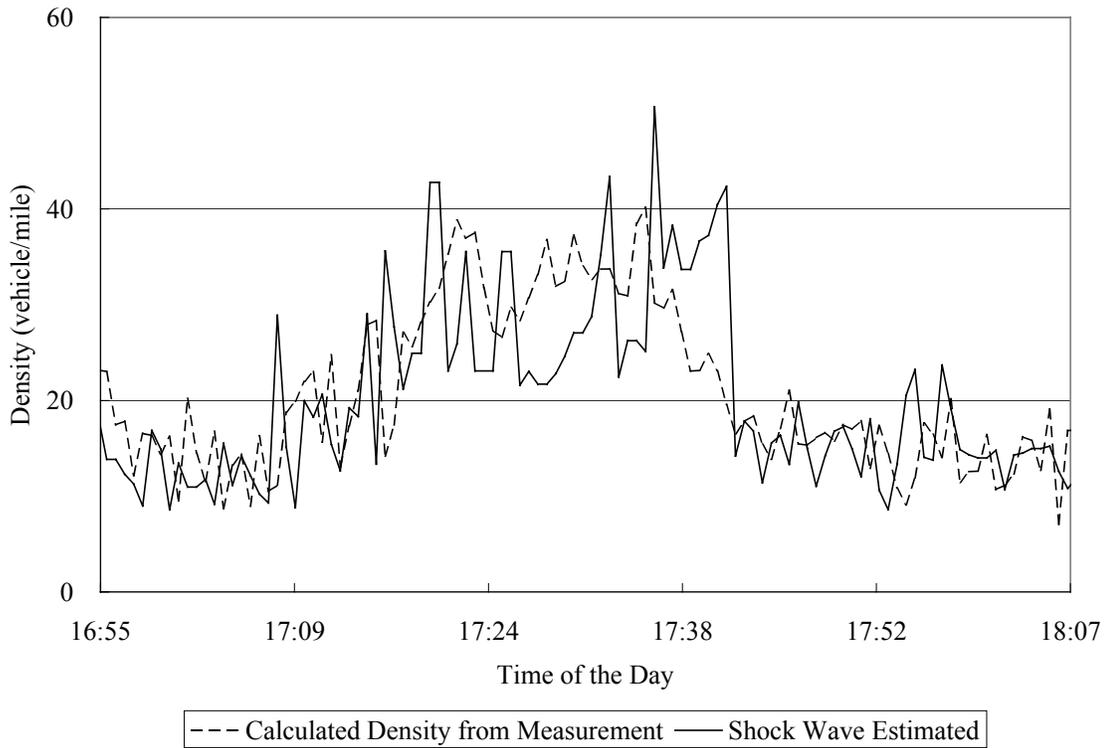


Figure 5-6 Density estimation results

Figure 5-7 shows the speed estimation between traffic detector 1 and 3 at 10:23 p.m. It is clear that using shock wave simulation, more information can be derived from the locations where traffic detector stations are absent.

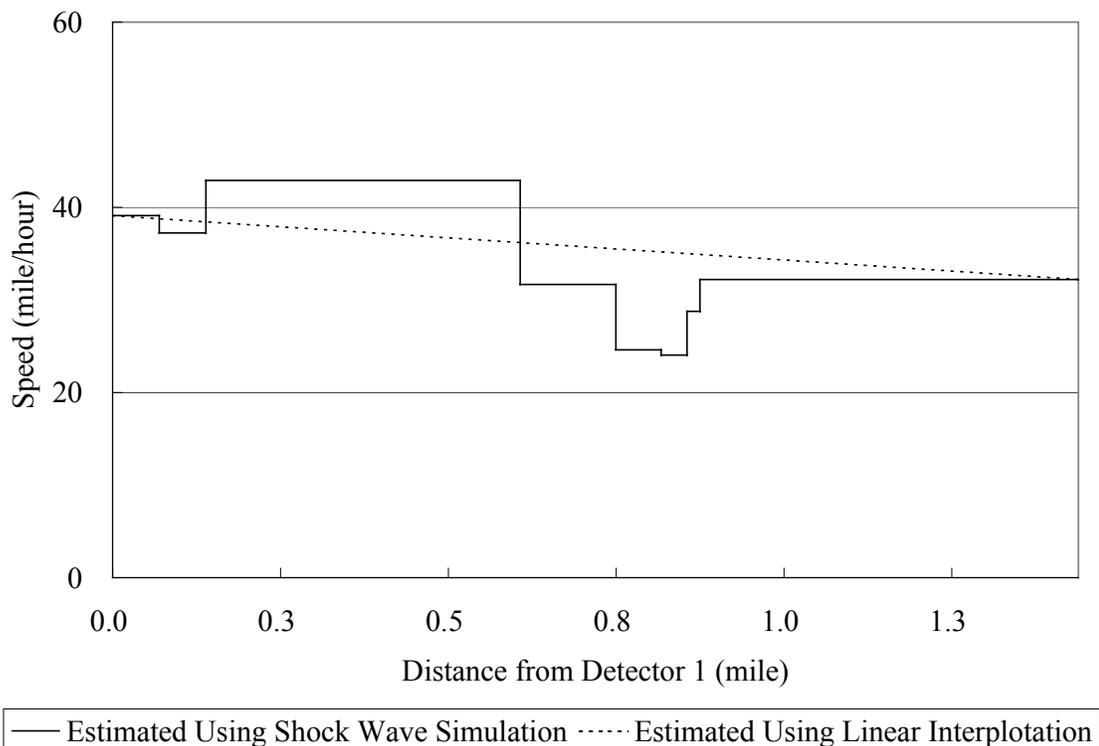


Figure 5-7 Comparison of shock wave simulation and linear interpolation

To test the prediction function with shock wave simulation, 28 points of time are used as reference times for which the prediction is made. Two-, ten-, and sixty-minute prediction errors of shock wave simulation and Auto-Regressive Integrated Moving Average (ARIMA) methods are calculated. The shock wave simulation is done by the proposed computer program with three traffic detector stations (see Appendix II) and the ARIMA method is performed using the SAS program. These 28 points are 50 intervals between consecutive points, or 33 minutes and 20 seconds apart because of 40-second intervals each.

The results are shown in Figure 5-8 to 5-10. It can be seen that the ARIMA model method and shock wave simulation method have similar prediction errors. No statistical difference is found between the prediction errors by the two methods.

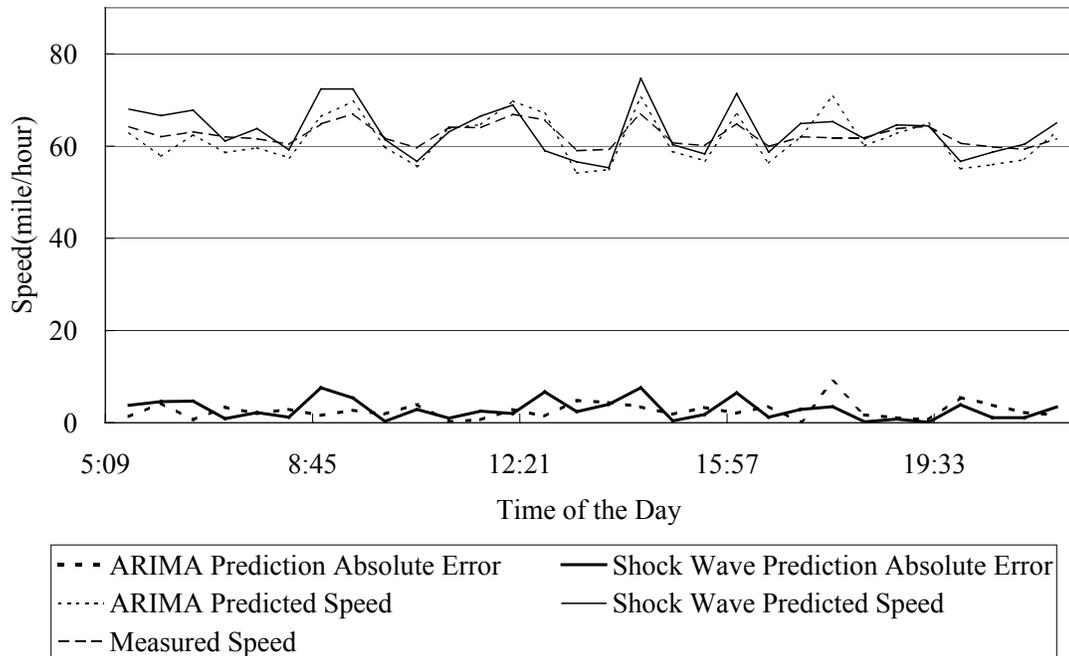


Figure 5-8 Two minutes speed prediction results

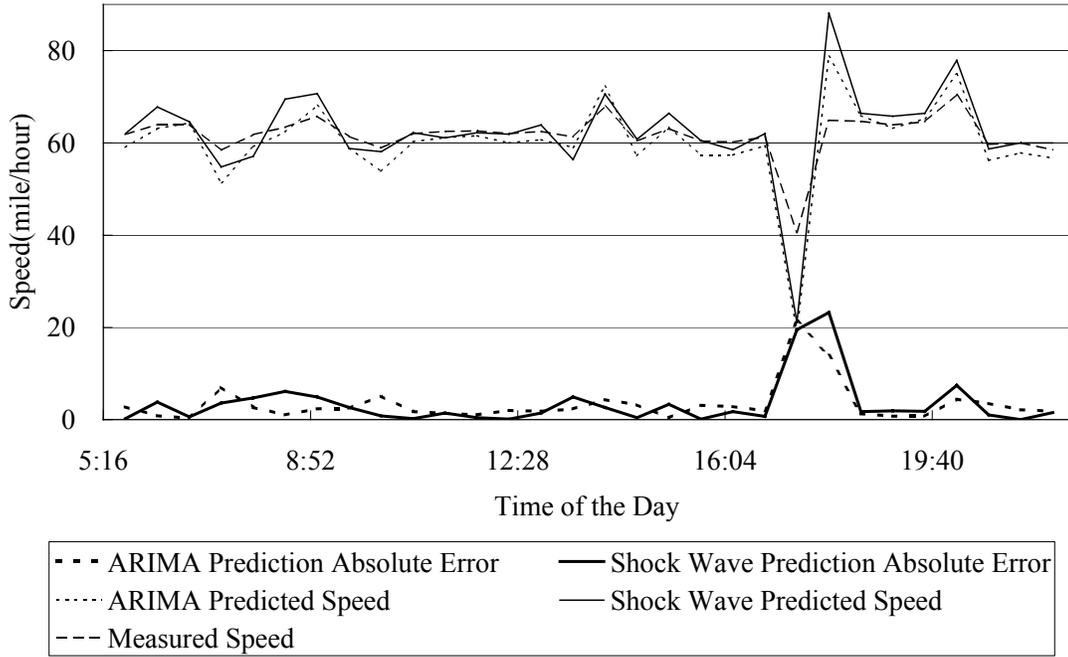


Figure 5-9 Ten minutes speed prediction results

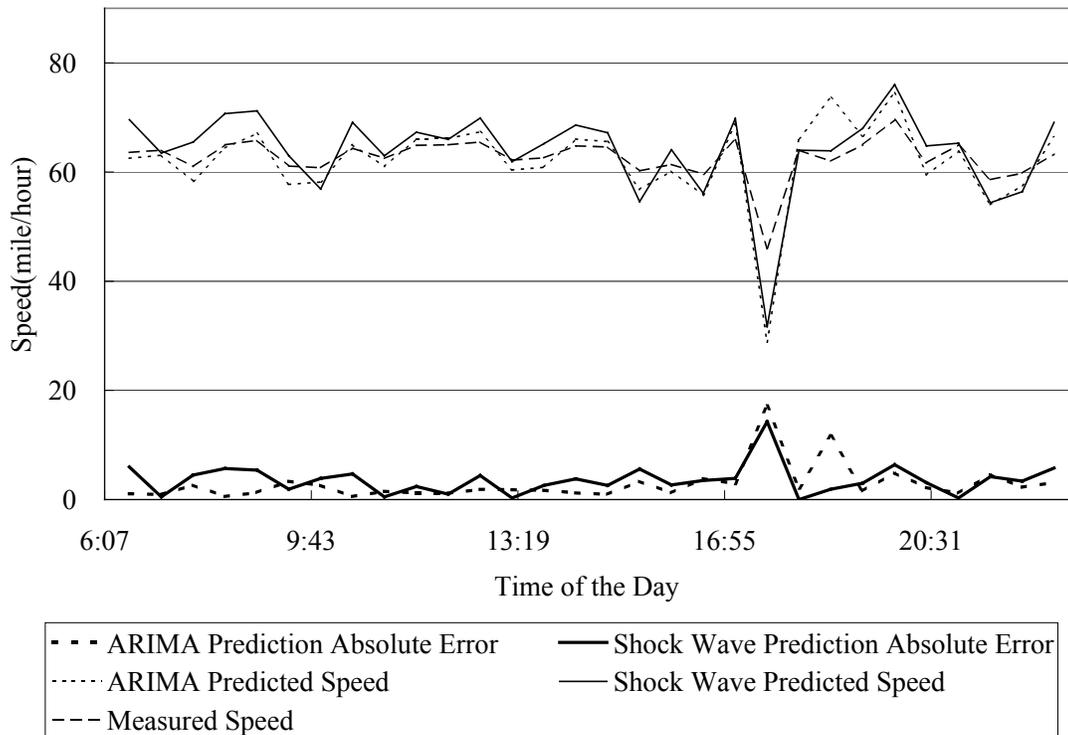


Figure 5-10 Sixty minutes speed prediction results

The ability to predict traffic congestion is very important for traffic management. However, there is no effective method available, especially for the congestion caused by unpredicted incidents such as traffic accidents. Thus, the time that a system spends in reflecting the congestion is very important. Figure 5-11 shows the 16-minute prediction of congestion around 5:30 p.m. Congestion can be predicted faster with shock wave simulation than with the ARIMA model.

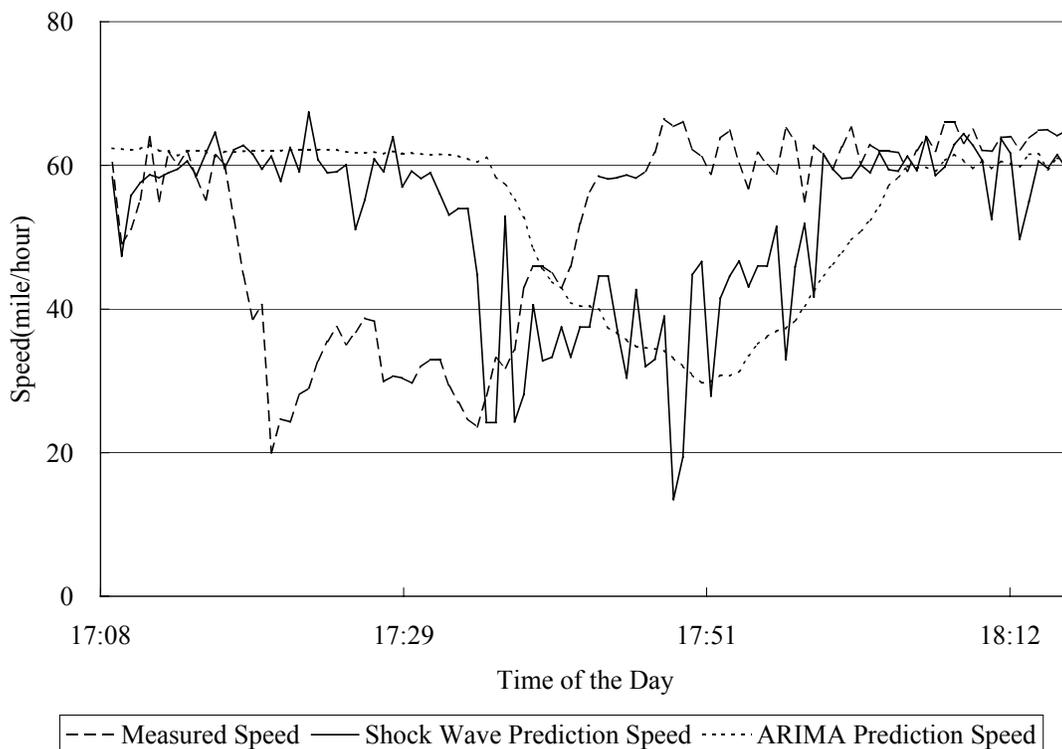


Figure 5-11 Congestion prediction

5.3 Travel Time Estimation Based on Piecewise Truncated Quadratic Speed Trajectory

In this section, travel time is estimated using piecewise truncated quadratic sample paths of speeds compared with collected travel time data and travel time estimated from three other trajectory-based approaches, namely: piecewise conservative constant (equation 3-17a); piecewise aggregative constant (equation 3-17b); and piecewise linear interpolations. An efficient computer program suitable for online, real-time travel time estimation is developed using VB (see Appendix IV).

The method for setting speed bound explained in section 3.3.3 cannot be used here because there are no historical speed observations available for the test site. Instead, the upper speed bound is directly set at 80 mph, a value between the speed limit and the design speed of the roadway. The lower speed bound is set at ten mph because 95% of observed speeds at three locations during the experiment are above this value. Figure 5-12 shows the actual and estimated travel time. Figure 5-13 shows the relative error calculated using the following:

$$\text{Relative Error \%} = \frac{\text{Estimated Travel Time} - \text{Actual Travel Time}}{\text{Actual Travel Time}} \times 100\% . \quad [5-8]$$

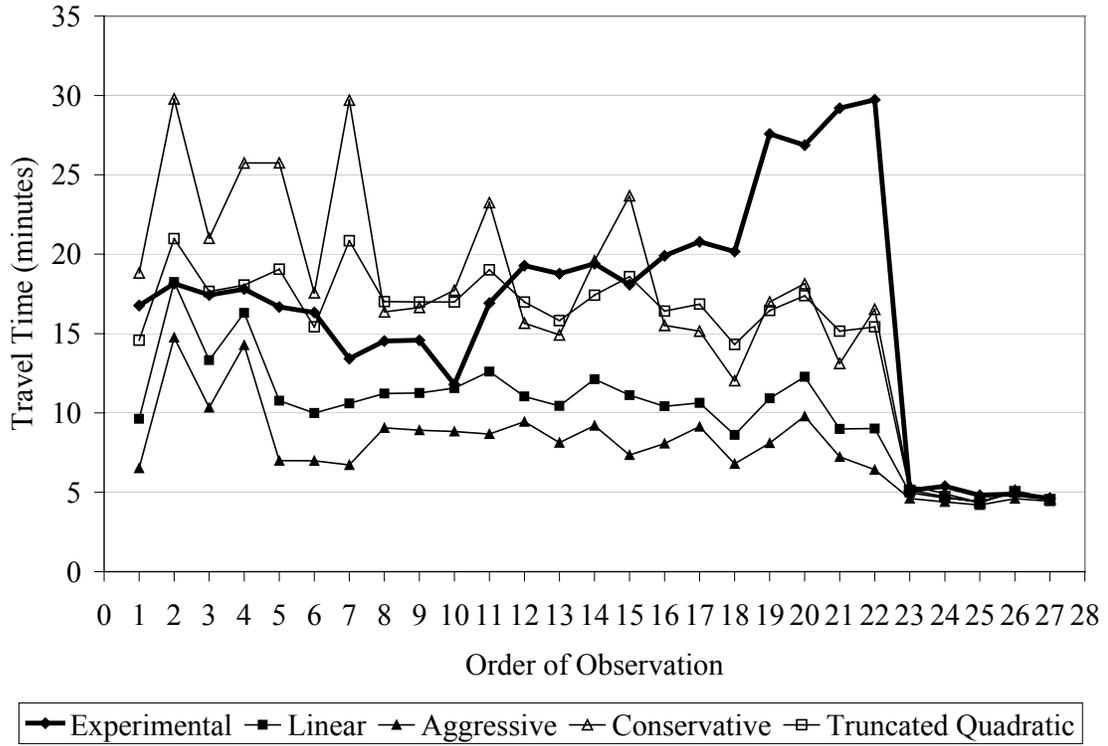


Figure 5-12 Comparison of real travel time and estimated travel time using four methods

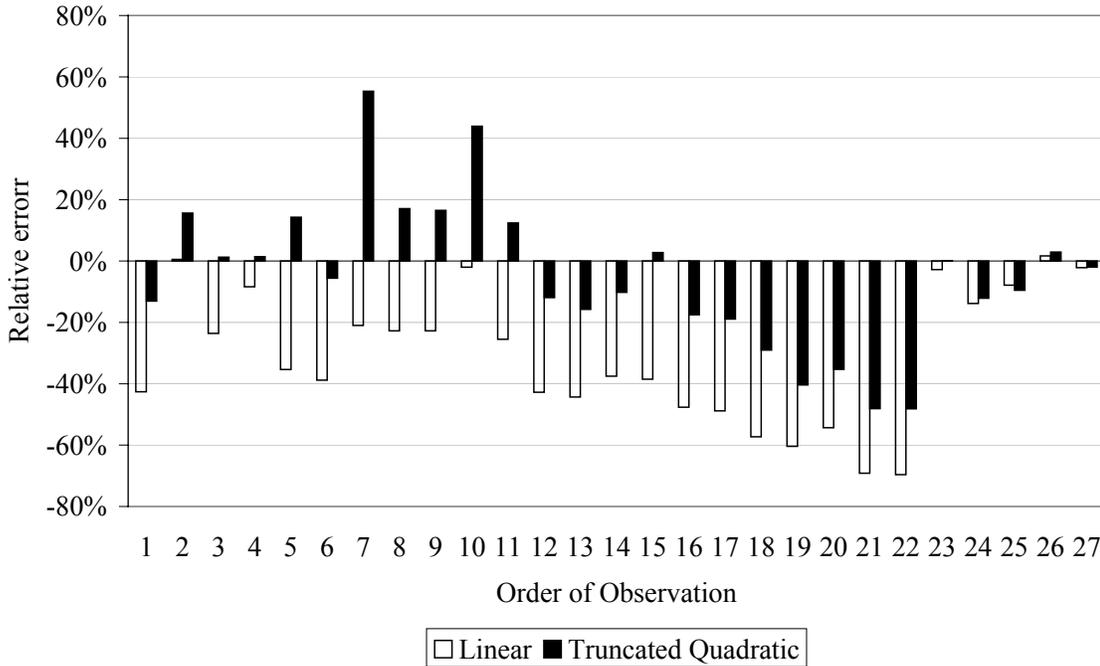


Figure 5-13 Relative errors of travel time estimated using linear and truncated quadratic speed trajectories

Table 5-3 lists 27 experimental observations and estimated travel times using different methods. Table 5-4 provides paired t-test statistics of the relative error derived from using, linear, aggressive and conservative speed trajectory methods, and that of the proposed piecewise, truncated, quadratic speed trajectory. It can be observed that linear, aggressive and truncated quadratic speed trajectories have similar variance of relative error, while the conservative speed trajectory generates a large variance of relative error (15.4%). In addition, the truncated speed trajectory produces the smallest mean relative error. The p-values (1.81×10^{-8} , 1.91×10^{-9} , and 3.11×10^{-3}) of the paired t-tests indicate that all other

three-speed trajectories yield a significantly larger mean relative error than the truncated speed trajectory.

Table 5-3 Comparison between estimated travel times and real travel times

ID	Record Time	Speed (mph)			Real Travel Time (minutes)	Estimated Travel Time (minute)			
		Detector 1	Detector 2	Detector 3		Linear	Aggressive	Conservative	Truncated Quadratic
1	8:54:17	10.00	32.27	69.77	16.77	9.63	6.52	18.82	14.58
2	8:57:54	8.41	11.44	64.43	18.15	18.25	14.76	29.78	20.98
3	8:59:25	11.36	17.35	69.43	17.43	13.32	10.34	21.01	17.65
4	9:00:00	10.76	11.74	69.32	17.80	16.31	14.27	25.74	18.05
5	9:02:51	6.82	30.45	63.64	16.67	10.77	6.99	25.75	19.05
6	9:03:56	11.36	29.32	69.32	16.33	9.99	6.98	17.56	15.42
7	9:09:16	5.68	32.05	64.55	13.42	10.60	6.72	29.70	20.85
8	9:14:02	15.38	20.68	67.95	14.53	11.22	9.06	16.36	17.02
9	9:14:06	14.70	21.14	67.95	14.58	11.26	8.91	16.64	16.98
10	9:19:23	13.11	21.44	67.39	11.80	11.56	8.83	17.72	16.98
11	9:18:45	8.64	21.82	68.86	16.92	12.61	8.67	23.24	19.02
12	9:20:13	17.50	19.55	68.41	19.28	11.04	9.45	15.65	16.98
13	9:25:24	15.68	25.23	59.32	18.77	10.44	8.13	14.91	15.82
14	9:27:37	22.73	14.77	50.23	19.40	12.12	9.20	19.58	17.42
15	9:34:49	7.73	28.18	64.32	18.08	11.12	7.34	23.67	18.58
16	9:47:22	15.23	23.86	70.23	19.90	10.42	8.07	15.51	16.42
17	9:50:01	20.68	19.09	65.68	20.78	10.63	9.13	15.15	16.85
18	9:53:05	19.77	30.45	69.55	20.17	8.61	6.79	12.03	14.32
19	9:48:36	12.95	24.55	64.09	27.58	10.92	8.10	16.99	16.45
20	9:50:32	13.64	19.09	63.41	26.87	12.28	9.79	18.14	17.38
21	9:48:51	28.64	22.05	65.23	29.20	8.99	7.23	13.12	15.15
22	9:50:35	33.18	17.50	69.09	29.73	9.02	6.42	16.53	15.42
23	15:04:51	58.30	53.41	67.95	5.11	4.97	4.60	5.41	5.12
24	15:13:13	67.59	58.86	64.32	5.38	4.64	4.39	4.91	4.73
25	15:28:39	59.20	68.86	64.20	4.81	4.43	4.20	4.35	4.35
26	15:44:06	50.23	56.93	70.00	4.89	4.97	4.60	5.41	5.03
27	16:05:47	62.25	63.52	66.70	4.62	4.52	4.44	4.60	4.53

Table 5-4. t-test: paired two sample means

Statistics	Linear	Aggressive	Conservative	Quadratic
Mean	-31.0%	-44.3%	6.3%	-5.0%
Variance	4.8%	5.0%	15.4%	5.9%
Observations	27	27	27	27
Pearson Correlation	0.72	0.49	0.91	N/A
Hypothesized Mean Difference	0	0	0	N/A
df	26	26	26	N/A
t Statistic	-7.70	-8.67	2.98	N/A
P(T<=t) one-tail	1.81×10^{-8}	1.91×10^{-9}	3.11×10^{-3}	N/A
t Critical one-tail	1.71	1.71	1.71	N/A

Several observations can be made based on Table 5-3, Figure 5-12 and 5-13. First, when traffic is operating at free flow condition, as indicated by high speed flows shown in observations ID 23~27 in Table 5-3, the travel time estimates obtained from using four different methods are very similar and are consistent with actual travel time. The relative error of estimated travel time is on the order of 10%.

Second, for transition flow or congestion conditions as suggested by lower speeds detected, the conservative speed trajectory tends to overestimate travel time for most of the observations, while aggressive speed trajectory and linear speed trajectory underestimate travel time. The estimated travel time using a linear speed trajectory lies between those of conservative and aggressive speed trajectories, and is more close to the actual travel time than the other two.

Third, the truncated quadratic speed trajectory seems to overestimate travel time for observations when speeds measured at three adjacent detectors follow the patterns in cases two, five, and nine (Table 3-1), while it seems to underestimate travel time for observations when speeds measured at three adjacent detectors follow the patterns in cases six, seven and eight (Table 3-1). From Figure 5-13 it can be seen that, in terms of the accuracy of travel time estimation, the performance of truncated, quadratic speed trajectory is better than that of the linear speed trajectory, particularly when speeds at three adjacent detectors fall into cases six, seven and eight (Table 3-1).

In conventional speed trajectory methods, the hypothetical speed trajectory can never exceed the speed range set by observed speeds at adjacent detectors, which can be unrealistic during transition flow and congestion conditions. The proposed piecewise, truncated quadratic speed trajectory method allows vehicle speed to go above or below the observed speeds at detectors (but still within some speed range based on physical constraints). Moreover, it allows a vehicle to stay at a relatively low speed for a certain amount of time , thus providing a more plausible representation of the unknown speed trajectory that a vehicle might experience while traversing the link, and resulting in an improved accuracy of travel time estimation over other trajectory approaches.

5.4 Improved Gipps Car-following Model

To estimate the Gipps category car-following models, suppose there are n observations, (x_i, y_i) $i = 1, 2, \dots, n$ with a known functional relationship such that:

$$y_i = f(x_i, \theta^*) + \varepsilon_i \quad (i = 1, 2, \dots, n), \quad [5-9]$$

where, $E(\varepsilon_i) = 0$, x_i is a $k \times 1$ vector, and θ^* is a vector of parameters. The nonlinear least squares estimate of θ^* minimizes the error sum of squares,

$$SSE(\theta) = \sum_{i=1}^n [y_i - f(x_i, \theta^*)]^2. \quad [5-10]$$

Note that the nonlinear least squares may have several relative minima in addition to the absolute minimum. If the joint distribution of the ε_i 's in the model is assumed known, the maximum-likelihood estimate is obtained by maximizing the likelihood function:

$$L(\theta) = \prod_{i=1}^n f(x_i, \theta). \quad [5-11]$$

According to Jennrich (1969), the least squares estimate is not only the maximum-likelihood estimate but also, under appropriate regularity conditions, is asymptotically efficient, i.e. $a'\theta$ is an asymptotically minimum-variance for every a .

It is assumed that errors follow a normal distribution, i.e. the ε_i 's are independent and identically distributed with $N(0, \sigma^2)$. Seber and Wild (1989) point out that under this assumption both MLE and nonlinear least squares estimators are equivalent.

The coefficient of determination, R^2 , is defined as the ratio of the sum of squares explained by a regression model and the total sum of squares around the mean, i.e.,

$$R^2 = 1 - SSE / SST \text{ where } SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \text{ and } SST = \sum_{i=1}^n (y_i - \bar{y})^2 .$$

The adjusted R^2 , labeled as R_{adj}^2 , is a rescaling of R^2 by degrees of freedom so that it involves a ratio of mean squares rather than the sums of squares, $R_{adj}^2 = 1 - MSE / MST$.

While R^2 will never increase when a predictor is dropped from a regression equation, the R_{adj}^2 may be larger. Specifically, if the t-ratio for a predictor is less than one, dropping that predictor from the model will increase the R_{adj}^2 . Sometimes researchers keep everything with a t larger than 1 in the model. The motivation for doing this is to obtain as large an R_{adj}^2 as possible. R_{adj}^2 removes the impact of the degrees of freedom and gives a quantity that is more comparable than R^2 over models involving different numbers of parameters. Unlike R^2 , R_{adj}^2 does not always increase as variables are added to the model. The value of R_{adj}^2 will tend to stabilize around some upper limit as variables are added. The simplest model with R_{adj}^2 near this upper limit is chosen as the “best” model.

The following table presents the average estimation results for the existing Gipps category car-following models and the proposed model. Totally, there are five out of 36 car-following scenarios have imaginary number problem for Hamdar and Mahmassani model. These cases are not included in the Table 5-5. The table shows that the proposed model

provides a higher R_{adj}^2 and lower MSE, which suggests a better fit. The numbers in the table are the average of results.

Table 5-5 Average estimation results of Gipps category car-following models

Model	R_{adj}^2	MSE	τ_n (sec)	b_n (ft/sec ²)	b_{n-1} (ft/sec ²)	B_n (ft ² /sec ³)	B_{n-1} (ft ² /sec ³)	D_n (ft.)
Gipps	0.57	2.52	1.2	-4.1	-6.6	N/A	N/A	N/A
Hamdar and Mahmassani	0.62	2.50	1.3	-4.6	-9.1	N/A	N/A	24.9
Proposed	0.64	2.55	1.5	N/A	N/A	195.0	554.1	23.7

To verify the proposed car-following model, simulation is used to compare the performance of the proposed model with the existing models. It represents the model validity in the traffic simulation applications. The velocity information of the subject vehicle is simulated based on the car-following models. The simulation steps are represented in equations 5-12 to 5-18. The simulation starts at time, $\tau_n + 1$. The observed leading vehicle data is used in the model, while only the observed subject vehicle data is adopted within time, $\tau_n + 1$ to time, $2\tau_n$. The estimated velocity is then compared with the observed data.

$$\hat{v}_n(t) = \sqrt[3]{\hat{B}_n \left(\frac{v_{n-1}(t - \tau_n)^3}{\hat{B}_{n-1}} + \Delta x_{n-1,n}(t - \tau_n) - l_{n-1} - \hat{D}_n - v_n(t - \hat{\tau}_n) \hat{\tau}_n \right)}, \tau_n + 1 \leq t \leq 2\tau_n; \quad [5-12]$$

$$\hat{v}_n(t) = \sqrt[3]{\hat{B}_n \left(\frac{v_{n-1}(t - \tau_n)^3}{\hat{B}_{n-1}} + \Delta \hat{x}_{n-1,n}(t - \tau_n) - l_{n-1} - \hat{D}_n - \hat{v}_n(t - \hat{\tau}_n) \hat{\tau}_n \right)}, t > 2\tau_n; \quad [5-13]$$

$$\hat{a}_n(t) = \hat{v}_n(t) - v_n(t-1), \quad t = \tau_n + 1; \quad [5-14]$$

$$\hat{a}_n(t) = \hat{v}_n(t) - \hat{v}_n(t-1), \quad t > \tau_n + 1; \quad [5-15]$$

$$\hat{x}_n(t+1) = x_n(t) + v_n(t)\Delta t + \frac{1}{2}\hat{a}_n(t)\Delta t, \quad t = \tau_n + 1; \quad [5-16]$$

$$\hat{x}_n(t+1) = \hat{x}_n(t) + \hat{v}_n(t)\Delta t + \frac{1}{2}\hat{a}_n(t)\Delta t, \quad t > \tau_n + 1; \quad [5-17]$$

$$\Delta\hat{x}_{n-1,n}(t) = \hat{x}_{n-1}(t) - \hat{x}_n(t), \quad t > 2\tau_n. \quad [5-18]$$

Figure 5-16 below shows eight randomly chosen examples of the estimated and observed velocities from the 36 car-following files. There are no simulated data from Hamdar and Mahmassani model in Figure 5-16e and 5-16f because of imaginary number problem.

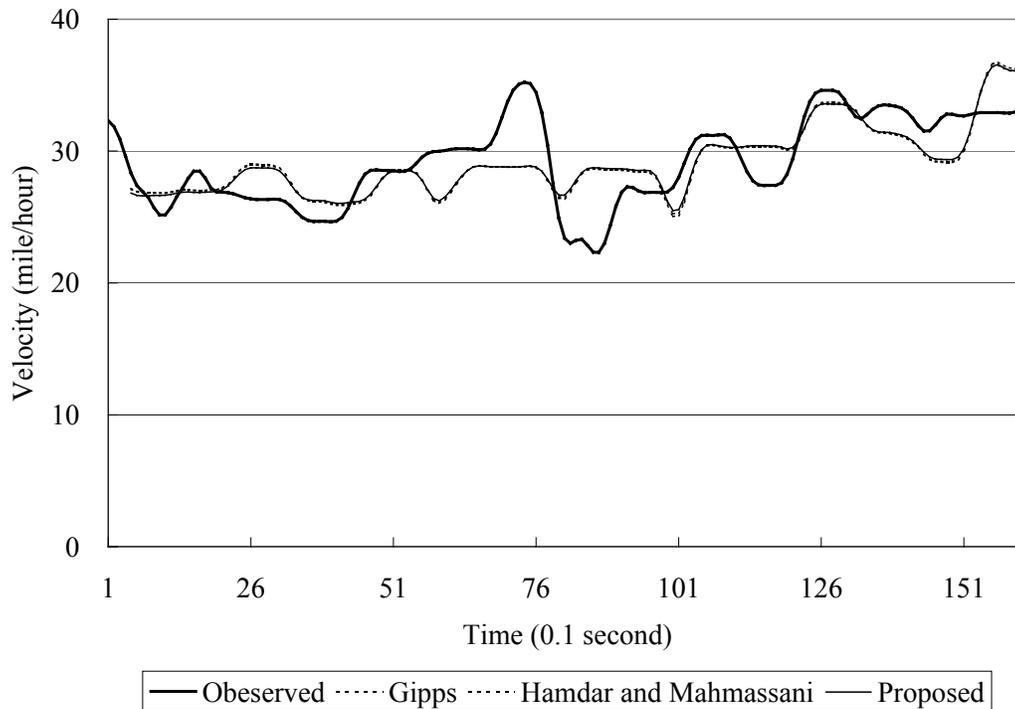


Figure 5-14a Performance example 1 of the Gipps category car-following models

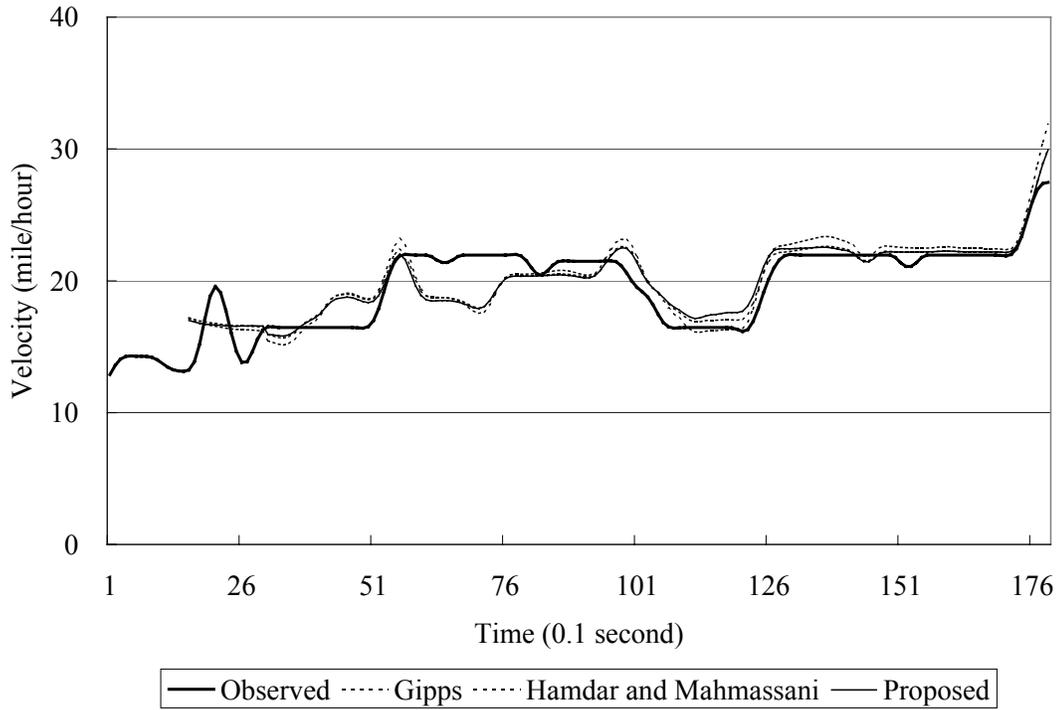


Figure 5-14b Performance example 2 of the Gipps category car-following models

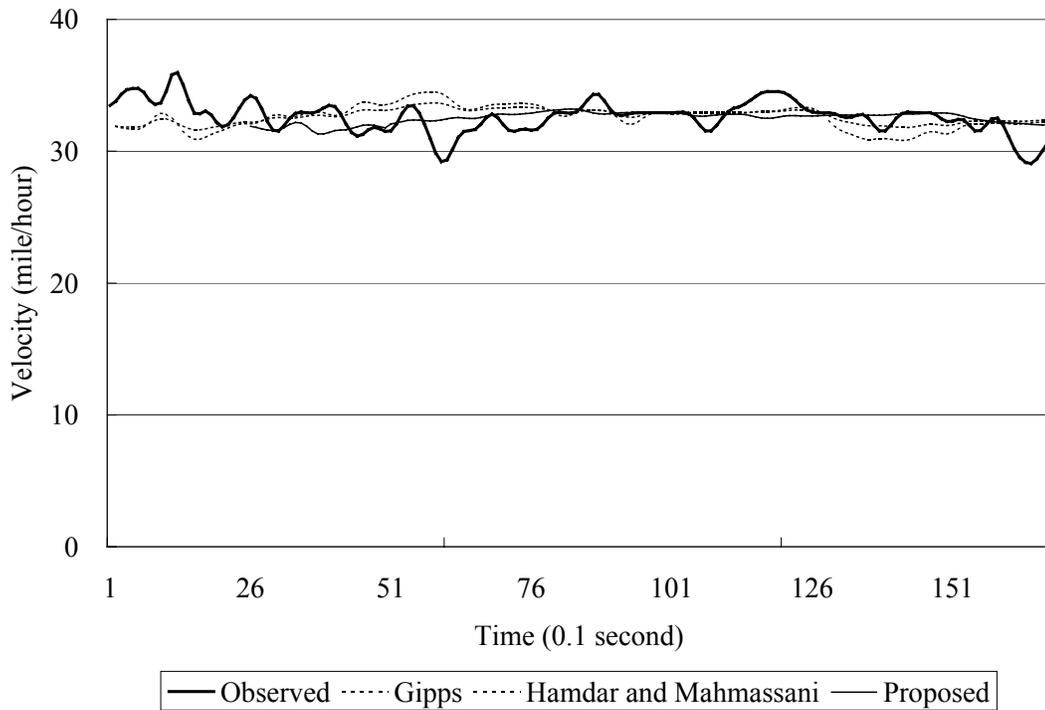


Figure 5-14c Performance example 3 of the Gipps category car-following models

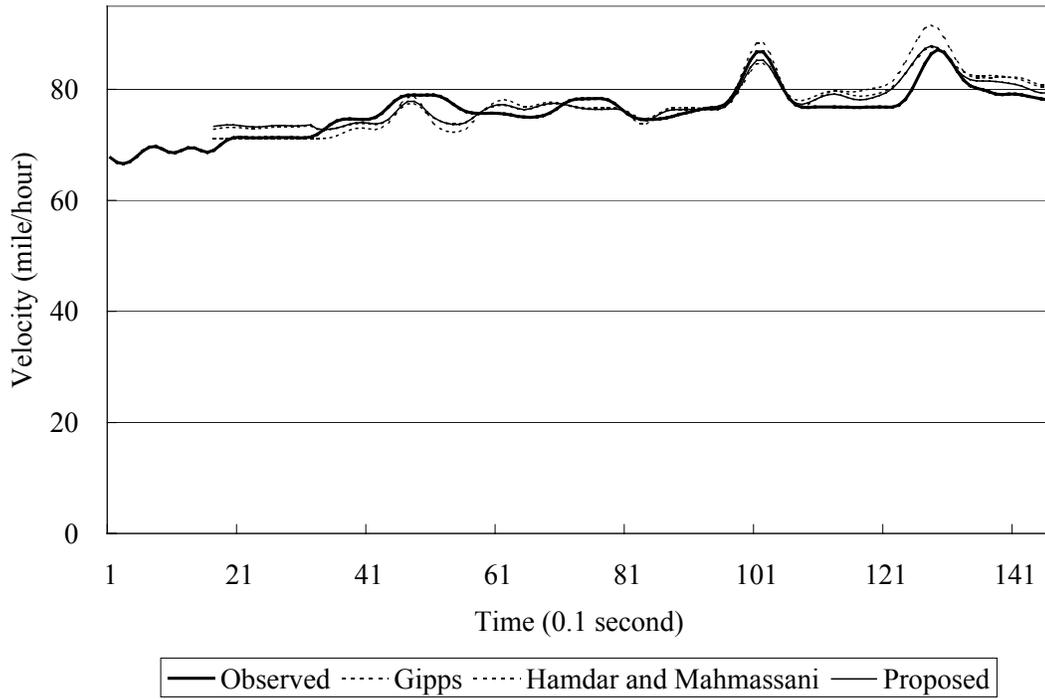


Figure 5-14d Performance example 4 of the Gipps category car-following models

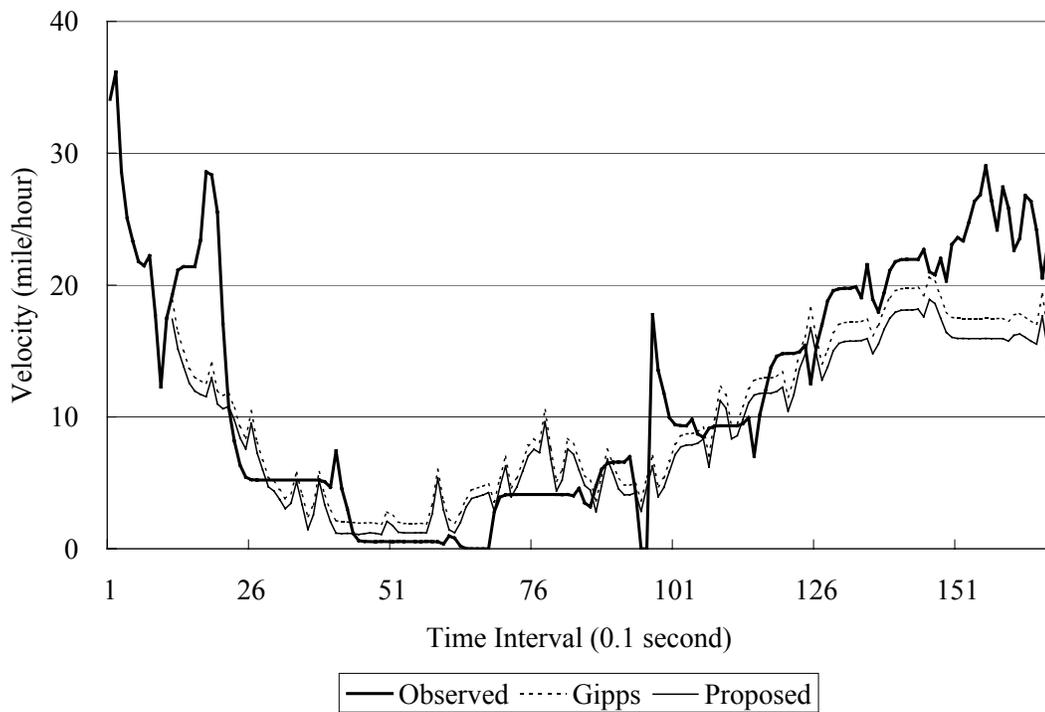


Figure 5-14e Performance example 5 of the Gipps category car-following models

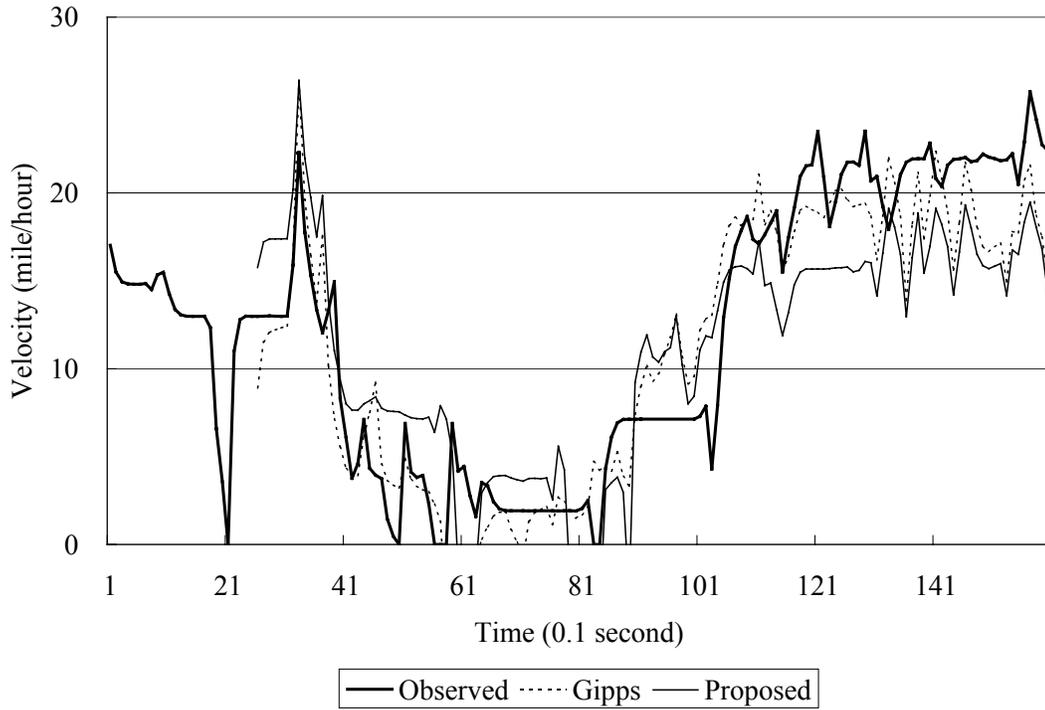


Figure 5-14f Performance example 6 of the Gipps category car-following models

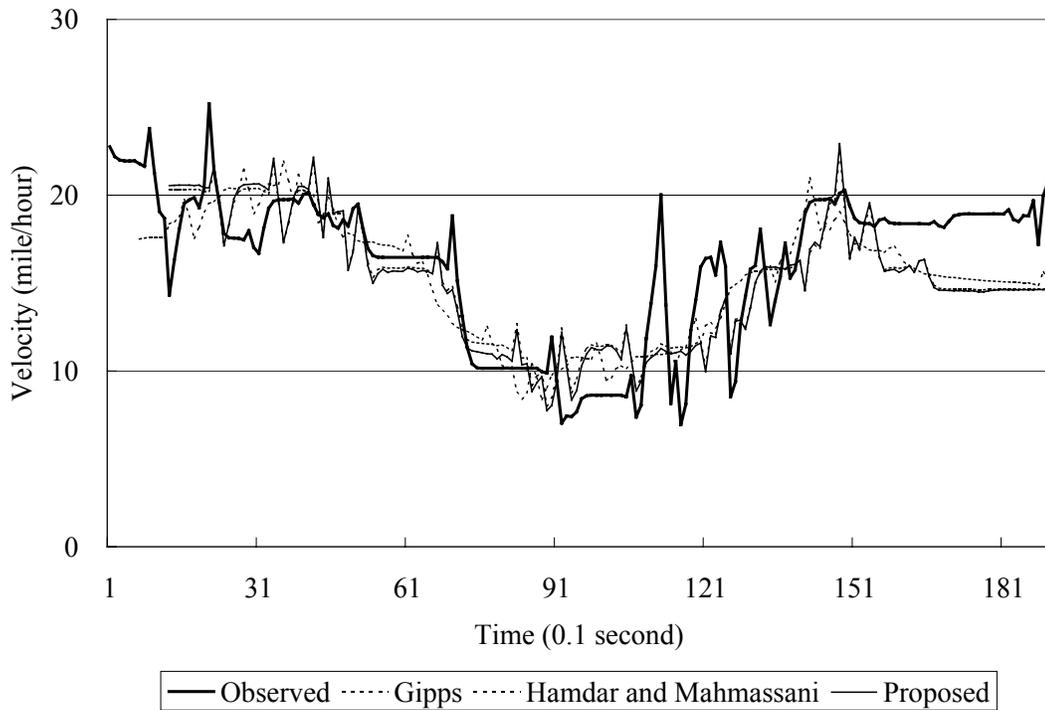


Figure 5-14g Performance example 7 of the Gipps category car-following models

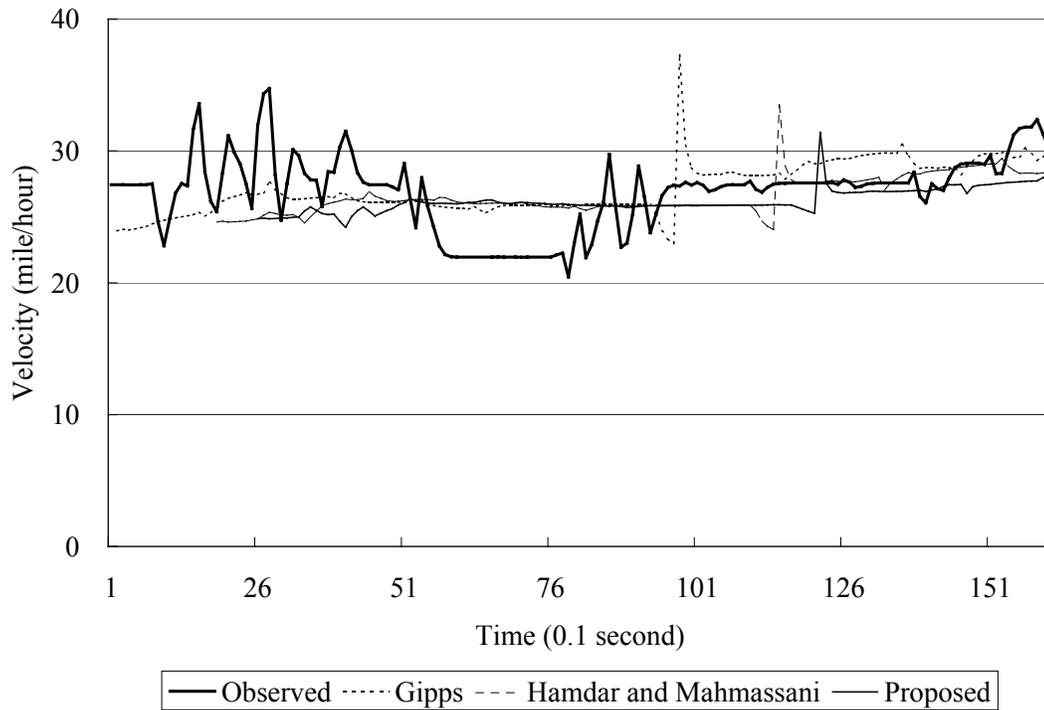


Figure 5-14h Performance example 8 of the Gipps category car-following models

The mean square errors (MSE) are reported in Table 5-6. It shows that the proposed model has similar performance in simulation with the existing Gipps category car-following models.

Table 5-6 Simulation results of the Gipps category car-following models

MSE	Gipps Model	Hamdar and Mahmassani Model	Proposed Model
Minimum	2.19	1.66	1.02
Mean	8.11	7.23	8.24
Median	6.55	6.55	6.47
Maximum	17.80	17.33	21.29

Chapter 6 Conclusions and Future Research

This dissertation presents the development of new methods to model highway traffic. The first part is a data mining-based adaptive regression method for developing a traffic-stream model. The second part proposes a shock wave simulation method for traffic estimation and prediction. Travel time estimation is then studied with a new method based on piecewise, truncated quadratic speed trajectories. Finally, an improved Gipps car-following model is developed.

The originality and contribution of the data mining based adaptive regression method is that the best traffic-stream model can be automatically and efficiently screened and selected from candidate models. A computer program is developed with Visual Basic programming language (VB). It uses raw speed and density data as well as a user-specified maximum number of regimes (M) and continuity requirement as input, and the output is an optimal traffic-stream model with parameters and function forms for each regime. This method holds several practical advantages. First, the need for calibrating site-specific, equilibrium speed-density relations is precluded, because the adaptive regression automatically identifies an optimal traffic-stream model from candidate models that fits the best to the actual data. Second, the method is flexible because users may choose the degree of continuity and add new basis functions. Third, the determination of knots is done optimally and the estimation of

parameters is done simultaneously, avoiding subjective judgment of breakpoints between different regimes (for instance, free-flow or congestion), and eliminating the ad hoc nature of two-step parameter estimation for multi-regime traffic-stream modeling. Finally, the program has a high computational efficiency and can be used to perform on-line parameter estimates and updates of traffic-stream models, given real-time observations.

In order to estimate and predict traffic information between traffic detectors, a new method using shock wave simulation is proposed. The method is easy to apply and can be used to estimate and predict the traffic information in practice and nonlinear way. Tested with the real traffic data from San Antonio, Texas, the general prediction precision of the proposed method is similar to the prediction precision using time series models such as the Auto-Regressive Integrated Moving Average (ARIMA) model. However, the prediction precision for traffic congestion using the proposed method is even better than the ARIMA model.

Based on the observed or estimated traffic information, travel time can be estimated. A piecewise truncated quadratic speed trajectory is proposed to mimic the unknown speed trajectory between detectors. The basis functions of the proposed trajectory consist of quadratic and constant functions of time. The constant functions, corresponding to upper and lower speed bounds, are determined using the maximum likelihood estimates (or 85th percentile) of the highest and the lowest speeds that have been historically observed within a

time interval. Using the actual travel time obtained from the field experiment, the new method yields more accurate travel time estimation than other trajectory-based methods. Computational implementation of the new trajectory method is tractable and can be done efficiently; making it suitable for online, real time travel time estimation.

This dissertation also presents an improved Gipps car-following model. All the existing Gipps category car-following models have square root functions derived from the linear braking rate assumption. This study uses a more realistic, nonlinear braking rate, and develops a new model with a cube root function. Therefore, it eliminates the imaginary number problem when fitting the car-following models with the real traffic data.

The research framework proposed in this dissertation can be extended in the following major directions. First, new equipment and methods are needed to acquire new information and more accurate data. The data available today is much more accurate than before, however new information like reaction time will contribute to improve the models. Second, different types of leading vehicles, the second leading vehicle, and other traffic conditions need to be tested to verify their influences. Third, this study assumes drivers do not change their driving behaviors. However, driver behaviors always change and it should be considered in the future work. Fourth, roadway layout and condition, weather, and other factors may have meaningful effects on the traffic modeling and this needs to be further

studied. Finally, the proposed methods should be extended to the modeling of traffic at intersections, because traffic patterns and driving behaviors are different at intersections.

Appendix I Gradients for the Basic Functions in Traffic-Stream Models

Let z be sum of squared error (SSE). Using basis function B_1 , the SSE can be expressed

as $z = \sum (u_i - u_e)^2 = \sum [u_i - (\beta_0 + \beta_1 \rho_i)]^2$, whose gradient is given by

$$\nabla \mathbf{Z} = \begin{bmatrix} \frac{\partial z}{\partial \beta_0} \\ \frac{\partial z}{\partial \beta_1} \end{bmatrix} = \begin{bmatrix} \sum 1 \\ \sum \rho_i \end{bmatrix}. \quad [\text{A-1}]$$

With B_2 , the SSE can be expressed as $z = \sum (u_i - u_e)^2 = \sum [u_i - \beta_0 \log(\beta_1 / \rho_i)]^2$, whose gradient is given by

$$\nabla \mathbf{Z} = \begin{bmatrix} \frac{\partial z}{\partial \beta_0} \\ \frac{\partial z}{\partial \beta_1} \end{bmatrix} = \begin{bmatrix} \sum \{-2 \log(\beta_1 / \rho_i) [u_i - \beta_0 \log(\beta_1 / \rho_i)]\} \\ \sum \{-2 \beta_0 [u_i - \beta_0 \log(\beta_1 / \rho_i)] / \beta_1\} \end{bmatrix}. \quad [\text{A-2}]$$

With B_3 , the SSE can be expressed as $z = \sum (u_i - u_e)^2 = \sum [u_i - \beta_0 \exp(\rho_i / \beta_1)]^2$, whose gradient is given by

$$\nabla \mathbf{Z} = \begin{bmatrix} \frac{\partial z}{\partial \beta_0} \\ \frac{\partial z}{\partial \beta_1} \end{bmatrix} = \begin{bmatrix} \sum \{-2 \exp(\rho_i / \beta_1) [u_i - \beta_0 \exp(\rho_i / \beta_1)]\} \\ \sum \{2 \beta_0 \rho_i \exp(\rho_i / \beta_1) [u_i - \beta_0 \exp(\rho_i / \beta_1)] / \beta_1^2\} \end{bmatrix}. \quad [\text{A-3}]$$

With B_4 , the SSE can be expressed as $z = \sum (u_i - u_e)^2 = \sum [u_i - \beta_0 \exp(\beta_1 / \rho_i^2)]^2$, whose gradient is given by

$$\nabla \mathbf{Z} = \begin{bmatrix} \frac{\partial z}{\partial \beta_0} \\ \frac{\partial z}{\partial \beta_1} \end{bmatrix} = \begin{bmatrix} \sum \left\{ -2 \exp(\beta_1 / \rho_i^2) [u_i - \beta_0 \exp(\beta_1 / \rho_i^2)] \right\} \\ \sum \left\{ -2 \beta_0 \exp(\beta_1 / \rho_i^2) [u_i - \beta_0 \exp(\beta_1 / \rho_i^2)] / \rho_i^2 \right\} \end{bmatrix}. \quad [\text{A-4}]$$

With B_5 , the SSE can be expressed as $z = \sum (u_i - u_e)^2 = \sum \left\{ u_i - \beta_0 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \right\}^2$,

whose gradient is given by

$$\nabla \mathbf{Z} = \begin{bmatrix} \frac{\partial z}{\partial \beta_0} \\ \frac{\partial z}{\partial \beta_1} \\ \frac{\partial z}{\partial \beta_2} \\ \frac{\partial z}{\partial \beta_3} \end{bmatrix} = \begin{bmatrix} \sum \left\langle -2 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \left\{ u_i - \beta_0 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \right\} \right\rangle \\ \sum \left\langle -2 \beta_0 \beta_2 \beta_3 \rho_i (\rho_i / \beta_1)^{\beta_2 - 1} [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3 - 1} \left\{ u_i - \beta_0 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \right\} / \beta_1^2 \right\rangle \\ \sum \left\langle 2 \beta_0 \beta_3 \ln(\rho_i / \beta_1) (\rho_i / \beta_1)^{\beta_2} [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3 - 1} \left\{ u_i - \beta_0 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \right\} \right\rangle \\ \sum \left\langle -2 \beta_0 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \left\{ u_i - \beta_0 [1 - (\rho_i / \beta_1)^{\beta_2}]^{\beta_3} \right\} \ln [1 - (\rho_i / \beta_1)^{\beta_2}] \right\rangle \end{bmatrix}. \quad [\text{A-5}]$$

With B_6 , the SSE can be expressed as $z = \sum (u_i - u_e)^2 = \sum [u_i - (\beta_0 + \beta_1 \rho_i + \beta_1 \rho_i^2)]^2$,

whose gradient is given by

$$\nabla \mathbf{Z} = \begin{bmatrix} \frac{\partial z}{\partial \beta_0} \\ \frac{\partial z}{\partial \beta_1} \\ \frac{\partial z}{\partial \beta_2} \end{bmatrix} = \begin{bmatrix} \sum 1 \\ \sum \rho_i \\ \sum \rho_i^2 \end{bmatrix}. \quad [\text{A-6}]$$

Appendix II Three-detector Shock Wave Simulation VB Program

Option Explicit

Private Const DISTANCE1TO2 As Single = 0.506 * 1.609344

Private Const DISTANCE2TO3 As Single = 0.388 * 1.609344

Public RECORD\$, LENGTHOFTHERECORD%, POINTER%, SPACE%,

NUMBEROFRECORD%

Public ID%, HOUR#, MINUTE%, SECOND%

Public TIME#, BASETIME#, STOPTIME#

Private Const PREDICTTIME As Single = 0.66667

Private Const TIMEINTERAL As Single = 0.66667

Private TIMEOFRECORD(3000) As Single

Private SPEED(3, 3000) As Single

Private VOL(3, 3000) As Single

Private OCC(3, 3000) As Single

Private DENSITY(3, 3000) As Single

Public NUMBEROFSHOCKWAVE%, NUMBEROFSHOCKWAVEATBASETIME%

Public PREDICT_SPEED!, PREDICT_VOL!, PREDICT_DENSITY!

Private W(2000) As W

Private Type W

A_SPEED As Single

A_VOL As Single

A_DENSITY As Single

B_SPEED As Single

B_VOL As Single

B_DENSITY As Single

X_AXIS As Single

SPEED As Single

End Type

Private WATBASETIME(2000) As WATBASETIME

Private Type WATBASETIME

A_SPEED As Single

A_VOL As Single

A_DENSITY As Single

B_SPEED As Single

B_VOL As Single

B_DENSITY As Single

X_AXIS As Single

SPEED As Single

End Type

Public I%, J%, MINIMUMTIME!

Private Sub Command2_Click()

Open "SHOCK WAVE INPUT.txt" For Input As #1

Open "SHOCK WAVE OUTPUT1.txt" For Output As #3

Open "SHOCK WAVE OUTPUT2.txt" For Output As #4

ID = -1

Do While Not EOF(1)

Line Input #1, RECORD: ID = ID + 1

If ID >= 1 Then

LENGTHOFTHERECORD = Len(RECORD)

POINTER = 1: SPACE = InStr(POINTER + 1, RECORD, ":")

HOUR = CSng(Left(RECORD, SPACE - 1))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, ":")

MINUTE = CSng(Mid(RECORD, POINTER, 2))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

SECOND = CSng(Mid(RECORD, POINTER, 2))

TIMEOFRECORD(ID) = 60 * HOUR + MINUTE + SECOND / 60

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

SPEED(1, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER)) * 1.609344 /

60

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

VOL(1, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER)) * 60 / 40

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

OCC(1, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

If SPEED(1, ID) <> 0 Then

DENSITY(1, ID) = VOL(1, ID) / SPEED(1, ID)

Else: DENSITY(1, ID) = 0

End If

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

SPEED(2, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER)) * 1.609344 /

60

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

VOL(2, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER)) * 60 / 40

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

OCC(2, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

If SPEED(2, ID) <> 0 Then

DENSITY(2, ID) = VOL(2, ID) / SPEED(2, ID)

Else: DENSITY(2, ID) = 0

End If

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

SPEED(3, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER)) * 1.609344 /

60

```

    POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
    VOL(3, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER)) * 60 / 40
    POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
    OCC(3, ID) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
    If SPEED(3, ID) <> 0 Then
        DENSITY(3, ID) = VOL(3, ID) / SPEED(3, ID)
    Else: DENSITY(3, ID) = 0
    End If
End If

Loop
NUMBEROFRECORD = ID
Print "DATA INPUT COMPLETED."

ID = 1: TIME = TIMEOFRECORD(ID): BASETIME = TIME:
NUMBEROFSHOCKWAVE = 0
Print #4, "ID + PREDICTTIME / TIMEINTERAL", "PREDICT_SPEED",
"PREDICT_VOL", "PREDICT_DENSITY", "NUMBEROFSHOCKWAVE"
Do While ID <= NUMBEROFRECORD

    For I = 1 To NUMBEROFSHOCKWAVE
        If W(I).X_AXIS <= 0 Or W(I).X_AXIS = DISTANCE1TO2 Or W(I).X_AXIS >=
DISTANCE1TO2 + DISTANCE2TO3 Then

```

```

If I < NUMBEROFSHOCKWAVE Then
  For J = I To NUMBEROFSHOCKWAVE - 1
    W(J).A_SPEED = W(J + 1).A_SPEED
    W(J).A_VOL = W(J + 1).A_VOL
    W(J).A_DENSITY = W(J + 1).A_DENSITY
    W(J).B_SPEED = W(J + 1).B_SPEED
    W(J).B_VOL = W(J + 1).B_VOL
    W(J).B_DENSITY = W(J + 1).B_DENSITY
    W(J).SPEED = W(J + 1).SPEED
    W(J).X_AXIS = W(J + 1).X_AXIS
  Next J
End If
W(NUMBEROFSHOCKWAVE).A_SPEED = 0
W(NUMBEROFSHOCKWAVE).A_VOL = 0
W(NUMBEROFSHOCKWAVE).A_DENSITY = 0
W(NUMBEROFSHOCKWAVE).B_SPEED = 0
W(NUMBEROFSHOCKWAVE).B_VOL = 0
W(NUMBEROFSHOCKWAVE).B_DENSITY = 0
W(NUMBEROFSHOCKWAVE).SPEED = 0
W(NUMBEROFSHOCKWAVE).X_AXIS = 0
NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE - 1
End If

```

Next I

If NUMBEROFSHOCKWAVE = 0 Then

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

W(1).A_SPEED = SPEED(2, ID)

W(1).A_VOL = VOL(2, ID)

W(1).A_DENSITY = DENSITY(2, ID)

W(1).B_SPEED = SPEED(1, ID)

W(1).B_VOL = VOL(1, ID)

W(1).B_DENSITY = DENSITY(1, ID)

W(1).X_AXIS = 0.5 * DISTANCE1TO2

If DENSITY(2, ID) <> DENSITY(1, ID) Then

W(1).SPEED = (VOL(2, ID) - VOL(1, ID)) / (DENSITY(2, ID) - DENSITY(1, ID))

Else: W(1).SPEED = 0

End If

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

W(2).A_SPEED = SPEED(3, ID)

W(2).A_VOL = VOL(3, ID)

W(2).A_DENSITY = DENSITY(3, ID)

W(2).B_SPEED = SPEED(2, ID)

W(2).B_VOL = VOL(2, ID)

W(2).B_DENSITY = DENSITY(2, ID)

```

W(2).X_AXIS = DISTANCE1TO2 + 0.5 * DISTANCE2TO3

If DENSITY(3, ID) <> DENSITY(2, ID) Then

    W(2).SPEED = (VOL(3, ID) - VOL(2, ID)) / (DENSITY(3, ID) - DENSITY(2, ID))

Else: W(2).SPEED = 0

End If

Else:

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

For I = NUMBEROFSHOCKWAVE To 2 Step -1

    W(I).A_SPEED = W(I - 1).A_SPEED

    W(I).A_VOL = W(I - 1).A_VOL

    W(I).A_DENSITY = W(I - 1).A_DENSITY

    W(I).B_SPEED = W(I - 1).B_SPEED

    W(I).B_VOL = W(I - 1).B_VOL

    W(I).B_DENSITY = W(I - 1).B_DENSITY

    W(I).SPEED = W(I - 1).SPEED

    W(I).X_AXIS = W(I - 1).X_AXIS

Next I

If W(2).X_AXIS < DISTANCE1TO2 Then

    W(1).A_SPEED = W(2).B_SPEED

    W(1).A_VOL = W(2).B_VOL

    W(1).A_DENSITY = W(2).B_DENSITY

    W(1).B_SPEED = SPEED(1, ID)

```

```

W(1).B_VOL = VOL(1, ID)

W(1).B_DENSITY = DENSITY(1, ID)

W(1).X_AXIS = W(2).X_AXIS / 2

If W(1).A_DENSITY <> W(1).B_DENSITY Then

    W(1).SPEED = (W(1).A_VOL - W(1).B_VOL) / (W(1).A_DENSITY -
W(1).B_DENSITY)

Else: W(1).SPEED = 0

End If

For I = 3 To NUMBEROFSHOCKWAVE

    If W(I).X_AXIS > DISTANCE1TO2 Then

        NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

        For J = NUMBEROFSHOCKWAVE To I + 1

            W(J).A_SPEED = W(J - 1).A_SPEED

            W(J).A_VOL = W(J - 1).A_VOL

            W(J).A_DENSITY = W(J - 1).A_DENSITY

            W(J).B_SPEED = W(J - 1).B_SPEED

            W(J).B_VOL = W(J - 1).B_VOL

            W(J).B_DENSITY = W(J - 1).B_DENSITY

            W(J).SPEED = W(J - 1).SPEED

            W(J).X_AXIS = W(J - 1).X_AXIS

        Next J

        W(I).A_SPEED = W(I + 1).B_SPEED

```

W(I).A_VOL = W(I + 1).B_VOL

W(I).A_DENSITY = W(I + 1).B_DENSITY

W(I).B_SPEED = SPEED(2, ID)

W(I).B_VOL = VOL(2, ID)

W(I).B_DENSITY = DENSITY(2, ID)

W(I).X_AXIS = DISTANCE1TO2 + (W(I + 1).X_AXIS - DISTANCE1TO2) /

2

If W(I).A_DENSITY <> W(I).B_DENSITY Then

 W(I).SPEED = (W(I).A_VOL - W(I).B_VOL) / (W(I).A_DENSITY -

W(I).B_DENSITY)

Else: W(I).SPEED = 0

End If

W(I - 1).A_SPEED = W(I).B_SPEED

W(I - 1).A_VOL = W(I).B_VOL

W(I - 1).A_DENSITY = W(I).B_DENSITY

W(I - 1).X_AXIS = DISTANCE1TO2 - (DISTANCE1TO2 - W(I - 1).X_AXIS)

/ 2

If W(I - 1).A_DENSITY <> W(I - 1).B_DENSITY Then

 W(I - 1).SPEED = (W(I - 1).A_VOL - W(I - 1).B_VOL) / (W(I -

1).A_DENSITY - W(I - 1).B_DENSITY)

Else: W(I - 1).SPEED = 0

End If

```

Exit For

End If

Next I

Else:

W(1).A_SPEED = SPEED(2, ID)

W(1).A_VOL = VOL(2, ID)

W(1).A_DENSITY = DENSITY(2, ID)

W(1).B_SPEED = SPEED(1, ID)

W(1).B_VOL = VOL(1, ID)

W(1).B_DENSITY = DENSITY(1, ID)

W(1).X_AXIS = DISTANCE1TO2 / 2

If W(1).A_DENSITY <> W(1).B_DENSITY Then

    W(1).SPEED = (W(1).A_VOL - W(1).B_VOL) / (W(1).A_DENSITY -
W(1).B_DENSITY)

Else: W(1).SPEED = 0

End If

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

For I = NUMBEROFSHOCKWAVE To 3 Step -1

    W(I).A_SPEED = W(I - 1).A_SPEED

    W(I).A_VOL = W(I - 1).A_VOL

    W(I).A_DENSITY = W(I - 1).A_DENSITY

    W(I).B_SPEED = W(I - 1).B_SPEED

```

W(I).B_VOL = W(I - 1).B_VOL

W(I).B_DENSITY = W(I - 1).B_DENSITY

W(I).SPEED = W(I - 1).SPEED

W(I).X_AXIS = W(I - 1).X_AXIS

Next I

W(2).A_SPEED = W(3).B_SPEED

W(2).A_VOL = W(3).B_VOL

W(2).A_DENSITY = W(3).B_DENSITY

W(2).B_SPEED = SPEED(2, ID)

W(2).B_VOL = VOL(2, ID)

W(2).B_DENSITY = DENSITY(2, ID)

W(2).X_AXIS = DISTANCE1TO2 + (W(3).X_AXIS - DISTANCE1TO2) / 2

If W(2).A_DENSITY <> W(2).B_DENSITY Then

W(2).SPEED = (W(2).A_VOL - W(2).B_VOL) / (W(2).A_DENSITY -

W(2).B_DENSITY)

Else: W(2).SPEED = 0

End If

End If

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

W(NUMBEROFSHOCKWAVE).A_SPEED = SPEED(3, ID)

W(NUMBEROFSHOCKWAVE).A_VOL = VOL(3, ID)

W(NUMBEROFSHOCKWAVE).A_DENSITY = DENSITY(3, ID)

W(NUMBEROFSHOCKWAVE).B_SPEED = W(NUMBEROFSHOCKWAVE -
1).A_SPEED

W(NUMBEROFSHOCKWAVE).B_VOL = W(NUMBEROFSHOCKWAVE -
1).A_VOL

W(NUMBEROFSHOCKWAVE).B_DENSITY = W(NUMBEROFSHOCKWAVE -
1).A_DENSITY

W(NUMBEROFSHOCKWAVE).X_AXIS = DISTANCE1TO2 + DISTANCE2TO3 -
(DISTANCE1TO2 + DISTANCE2TO3 - W(NUMBEROFSHOCKWAVE - 1).X_AXIS) / 2

If W(NUMBEROFSHOCKWAVE).A_DENSITY <>

W(NUMBEROFSHOCKWAVE).B_DENSITY Then

W(NUMBEROFSHOCKWAVE).SPEED =
(W(NUMBEROFSHOCKWAVE).A_VOL - W(NUMBEROFSHOCKWAVE).B_VOL) /
(W(NUMBEROFSHOCKWAVE).A_DENSITY -
W(NUMBEROFSHOCKWAVE).B_DENSITY)

Else: W(NUMBEROFSHOCKWAVE).SPEED = 0

End If

End If

For I = 1 To NUMBEROFSHOCKWAVE - 1

If W(I).B_VOL = 0 Then

For J = I To NUMBEROFSHOCKWAVE - 1

W(J).A_SPEED = W(J + 1).A_SPEED

W(J).A_VOL = W(J + 1).A_VOL

W(J).A_DENSITY = W(J + 1).A_DENSITY

W(J).B_SPEED = W(J + 1).B_SPEED

W(J).B_VOL = W(J + 1).B_VOL

W(J).B_DENSITY = W(J + 1).B_DENSITY

W(J).SPEED = W(J + 1).SPEED

W(J).X_AXIS = W(J + 1).X_AXIS

Next J

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE - 1

End If

Next I

If W(NUMBEROFSHOCKWAVE).B_VOL = 0 Then

W(NUMBEROFSHOCKWAVE - 1).A_SPEED =

W(NUMBEROFSHOCKWAVE).A_SPEED

W(NUMBEROFSHOCKWAVE - 1).A_VOL =

W(NUMBEROFSHOCKWAVE).A_VOL

W(NUMBEROFSHOCKWAVE - 1).A_DENSITY =

W(NUMBEROFSHOCKWAVE).A_DENSITY

End If

For I = 1 To NUMBEROFSHOCKWAVE

WATBASETIME(I).A_SPEED = W(I).A_SPEED

WATBASETIME(I).A_VOL = W(I).A_VOL

WATBASETIME(I).A_DENSITY = W(I).A_DENSITY

WATBASETIME(I).B_SPEED = W(I).B_SPEED

WATBASETIME(I).B_VOL = W(I).B_VOL

WATBASETIME(I).B_DENSITY = W(I).B_DENSITY

WATBASETIME(I).SPEED = W(I).SPEED

WATBASETIME(I).X_AXIS = W(I).X_AXIS

Next I

NUMBEROFSHOCKWAVEATBASETIME = NUMBEROFSHOCKWAVE

BASETIME = TIME: STOPTIME = BASETIME + PREDICTTIME

Call Calculation

If W(NUMBEROFSHOCKWAVE).X_AXIS < DISTANCE1TO2 Then

PREDICT_SPEED = W(NUMBEROFSHOCKWAVE).A_SPEED * 60 / 1.609344

PREDICT_VOL = W(NUMBEROFSHOCKWAVE).A_VOL

PREDICT_DENSITY = W(NUMBEROFSHOCKWAVE).A_DENSITY

ElseIf W(1).X_AXIS > DISTANCE1TO2 Then

PREDICT_SPEED = W(1).B_SPEED * 60 / 1.609344

PREDICT_VOL = W(1).B_VOL

PREDICT_DENSITY = W(1).B_DENSITY

Else:

For I = 1 To NUMBEROFSHOCKWAVE - 1

```

If W(I).X_AXIS < DISTANCE1TO2 And W(I + 1).X_AXIS > DISTANCE1TO2
Then
    PREDICT_SPEED = W(I).A_SPEED * 60 / 1.609344
    PREDICT_VOL = W(I).A_VOL
    PREDICT_DENSITY = W(I).A_DENSITY
    Exit For
ElseIf W(I).X_AXIS = DISTANCE1TO2 Then
    PREDICT_SPEED = 0.5 * (W(I).A_SPEED + W(I).B_SPEED) * 60 / 1.609344
    PREDICT_VOL = 0.5 * (W(I).A_VOL + W(I).B_VOL)
    PREDICT_DENSITY = 0.5 * (W(I).A_DENSITY + W(I).B_DENSITY)
    Exit For
End If
Next I
End If
Print #4, ID + PREDICTTIME / TIMEINTERAL, PREDICT_SPEED, PREDICT_VOL,
PREDICT_DENSITY, NUMBEROFSHOCKWAVE

TIME = BASETIME: STOPTIME = TIMEOFRECORD(ID + 1):
NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVEATBASETIME

For I = 1 To NUMBEROFSHOCKWAVE
    W(I).A_SPEED = WATBASETIME(I).A_SPEED
    W(I).A_VOL = WATBASETIME(I).A_VOL

```

W(I).A_DENSITY = WATBASETIME(I).A_DENSITY

W(I).B_SPEED = WATBASETIME(I).B_SPEED

W(I).B_VOL = WATBASETIME(I).B_VOL

W(I).B_DENSITY = WATBASETIME(I).B_DENSITY

W(I).SPEED = WATBASETIME(I).SPEED

W(I).X_AXIS = WATBASETIME(I).X_AXIS

Next I

Call Calculation

ID = ID + 1

Loop

Print "MISSION COMPLETED."

End Sub

Private Sub Calculation()

Do

MINIMUMTIME = -1

For I = 1 To NUMBEROFSHOCKWAVE - 1

If W(I).SPEED <> W(I + 1).SPEED Then

```

    If (W(I + 1).X_AXIS - W(I).X_AXIS) / (W(I).SPEED - W(I + 1).SPEED) >= 0 Then
        If MINIMUMTIME = -1 Or (W(I + 1).X_AXIS - W(I).X_AXIS) / (W(I).SPEED -
W(I + 1).SPEED) < MINIMUMTIME Then MINIMUMTIME = (W(I + 1).X_AXIS -
W(I).X_AXIS) / (W(I).SPEED - W(I + 1).SPEED)

        End If

    End If

Next I

If MINIMUMTIME <> -1 And TIME + MINIMUMTIME <= STOPTIME Then

    TIME = TIME + MINIMUMTIME

    For I = 1 To NUMBEROFSHOCKWAVE

        W(I).X_AXIS = W(I).X_AXIS + W(I).SPEED * MINIMUMTIME

    Next I

    For I = 1 To NUMBEROFSHOCKWAVE - 1

        If Abs(W(I).X_AXIS - W(I + 1).X_AXIS) < 0.0001 Then

            W(I).A_SPEED = W(I + 1).A_SPEED

            W(I).A_VOL = W(I + 1).A_VOL

            W(I).A_DENSITY = W(I + 1).A_DENSITY

            If W(I).B_DENSITY <> W(I).A_DENSITY Then

                W(I).SPEED = (W(I).B_VOL - W(I).A_VOL) / (W(I).B_DENSITY -
W(I).A_DENSITY)

                Else: W(I).SPEED = 0

```

End If

$W(I).X_AXIS = 0.5 * (W(I).X_AXIS + W(I + 1).X_AXIS)$

For J = I + 1 To NUMBEROFSHOCKWAVE - 1

$W(J).A_SPEED = W(J + 1).A_SPEED$

$W(J).A_VOL = W(J + 1).A_VOL$

$W(J).A_DENSITY = W(J + 1).A_DENSITY$

$W(J).B_SPEED = W(J + 1).B_SPEED$

$W(J).B_VOL = W(J + 1).B_VOL$

$W(J).B_DENSITY = W(J + 1).B_DENSITY$

$W(I).SPEED = W(J + 1).SPEED$

$W(J).X_AXIS = W(J + 1).X_AXIS$

Next J

$W(NUMBEROFSHOCKWAVE).A_SPEED = 0$

$W(NUMBEROFSHOCKWAVE).A_VOL = 0$

$W(NUMBEROFSHOCKWAVE).A_DENSITY = 0$

$W(NUMBEROFSHOCKWAVE).B_SPEED = 0$

$W(NUMBEROFSHOCKWAVE).B_VOL = 0$

$W(NUMBEROFSHOCKWAVE).B_DENSITY = 0$

$W(NUMBEROFSHOCKWAVE).SPEED = 0$

$W(NUMBEROFSHOCKWAVE).X_AXIS = 0$

$NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE - 1$

End If

```
Next I
Else:
  For I = 1 To NUMBEROFSHOCKWAVE
    W(I).X_AXIS = W(I).X_AXIS + W(I).SPEED * (TIMEOFRECORD(ID + 1) -
TIME)
  Next I
  TIME = STOPTIME
  Exit Do
End If
Loop

End Sub
```

Appendix III Two-detector Shock Wave Simulation VB Program

Option Explicit

Private Const DISTANCE1TO2 As Single = 0.506

Private Const DISTANCE2TO3 As Single = 0.388

Public RECORD\$, LENGTHOFTHERECORD%, POINTER%, SPACE%,

NUMBEROFRECORD%

Public T%, HOUR#, MINUTE%, SECOND%

Public TIME#, BASETIME#, STOPTIME#

Private Const PREDICTTIME As Single = 0

Private Const TIMEINTERAL As Single = 40 / 3600

Private TIMEOFRECORD(3000) As Single

Private SPEED(3, 3000) As Single

Private VOL(3, 3000) As Single

Private OCC(3, 3000) As Single

Private DENSITY(3, 3000) As Single

Public NUMBEROFSHOCKWAVE%, NUMBEROFSHOCKWAVEATBASETIME%

Public PREDICT_SPEED!, PREDICT_VOL!, PREDICT_DENSITY!

Private W(2000) As W

Private Type W

A_SPEED As Single

A_VOL As Single

A_DENSITY As Single

B_SPEED As Single

B_VOL As Single

B_DENSITY As Single

X_AXIS As Single

SPEED As Single

End Type

Private WATBASETIME(2000) As WATBASETIME

Private Type WATBASETIME

A_SPEED As Single

A_VOL As Single

A_DENSITY As Single

B_SPEED As Single

B_VOL As Single

B_DENSITY As Single

X_AXIS As Single

SPEED As Single

End Type

Public I%, J%, MINIMUMTIME!

Private Sub Command2_Click()

Open "SHOCK WAVE INPUT.txt" For Input As #1

Open "SHOCK WAVE OUTPUT1.txt" For Output As #3

Open "SHOCK WAVE OUTPUT2.txt" For Output As #4

T = 0

Line Input #1, RECORD

Do While Not EOF(1)

T = T + 1

Line Input #1, RECORD

LENGTHOFTHERECORD = Len(RECORD)

POINTER = 1: SPACE = InStr(POINTER + 1, RECORD, ":")

HOUR = CSng(Left(RECORD, SPACE - 1))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, ":")

MINUTE = CSng(Mid(RECORD, POINTER, 2))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

SECOND = CSng(Mid(RECORD, POINTER, 2))

TIMEOFRECORD(T) = HOUR + MINUTE / 60 + SECOND / 3600

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

SPEED(1, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))

POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")

```

VOL(1, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
OCC(1, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
DENSITY(1, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
SPEED(2, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
VOL(2, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
OCC(2, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
DENSITY(2, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
SPEED(3, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
VOL(3, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
POINTER = SPACE + 1: SPACE = InStr(POINTER + 1, RECORD, " ")
OCC(3, T) = CSng(Mid(RECORD, POINTER, SPACE - POINTER))
DENSITY(3, T) = CSng(Right(RECORD, LENGTHOFTHERECORD - SPACE))

```

Loop

NUMBEROFRECORD = T

Print "DATA INPUT COMPLETED."

T = 1: TIME = TIMEOFRECORD(T): BASETIME = TIME: NUMBEROFSHOCKWAVE
= 0

Print #4, "ID+PREDICTTIME/TIMEINTERAL", "PREDICT_SPEED(MILE/HOUR)",
"PREDICT_VOL(VEHICLE/HOUR)", "PREDICT_DENSITY(VEHICLE/MILE)",
"NUMBEROFSHOCKWAVE"

Do While T <= NUMBEROFRECORD

For I = 1 To NUMBEROFSHOCKWAVE

If W(I).X_AXIS <= 0 Or W(I).X_AXIS >= DISTANCE1TO2 + DISTANCE2TO3

Then

If I < NUMBEROFSHOCKWAVE Then

For J = I To NUMBEROFSHOCKWAVE - 1

W(J).A_SPEED = W(J + 1).A_SPEED

W(J).A_VOL = W(J + 1).A_VOL

W(J).A_DENSITY = W(J + 1).A_DENSITY

W(J).B_SPEED = W(J + 1).B_SPEED

W(J).B_VOL = W(J + 1).B_VOL

W(J).B_DENSITY = W(J + 1).B_DENSITY

W(J).SPEED = W(J + 1).SPEED

W(J).X_AXIS = W(J + 1).X_AXIS

Next J

End If

W(NUMBEROFSHOCKWAVE).A_SPEED = 0

W(NUMBEROFSHOCKWAVE).A_VOL = 0

W(NUMBEROFSHOCKWAVE).A_DENSITY = 0

W(NUMBEROFSHOCKWAVE).B_SPEED = 0

W(NUMBEROFSHOCKWAVE).B_VOL = 0

W(NUMBEROFSHOCKWAVE).B_DENSITY = 0

W(NUMBEROFSHOCKWAVE).SPEED = 0

W(NUMBEROFSHOCKWAVE).X_AXIS = 0

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE - 1

End If

Next I

If NUMBEROFSHOCKWAVE = 0 Then

NUMBEROFSHOCKWAVE = 1

W(1).A_SPEED = SPEED(3, T)

W(1).A_VOL = VOL(3, T)

W(1).A_DENSITY = DENSITY(3, T)

W(1).B_SPEED = SPEED(1, T)

W(1).B_VOL = VOL(1, T)

W(1).B_DENSITY = DENSITY(1, T)

```

W(1).X_AXIS = 0.5 * (DISTANCE1TO2 + DISTANCE2TO3)

If DENSITY(3, T) <> DENSITY(1, T) Then

    W(1).SPEED = (VOL(3, T) - VOL(1, T)) / (DENSITY(3, T) - DENSITY(1, T))

Else: W(1).SPEED = 0

End If

Else:

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

For I = NUMBEROFSHOCKWAVE To 2 Step -1

    W(I).A_SPEED = W(I - 1).A_SPEED

    W(I).A_VOL = W(I - 1).A_VOL

    W(I).A_DENSITY = W(I - 1).A_DENSITY

    W(I).B_SPEED = W(I - 1).B_SPEED

    W(I).B_VOL = W(I - 1).B_VOL

    W(I).B_DENSITY = W(I - 1).B_DENSITY

    W(I).SPEED = W(I - 1).SPEED

    W(I).X_AXIS = W(I - 1).X_AXIS

Next I

W(1).A_SPEED = W(2).B_SPEED

W(1).A_VOL = W(2).B_VOL

W(1).A_DENSITY = W(2).B_DENSITY

W(1).B_SPEED = SPEED(1, T)

W(1).B_VOL = VOL(1, T)

```

```

W(1).B_DENSITY = DENSITY(1, T)

W(1).X_AXIS = W(2).X_AXIS / 2

If W(1).A_DENSITY <> W(1).B_DENSITY Then
    W(1).SPEED = (W(1).A_VOL - W(1).B_VOL) / (W(1).A_DENSITY -
W(1).B_DENSITY)

Else: W(1).SPEED = 0

End If

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE + 1

W(NUMBEROFSHOCKWAVE).A_SPEED = SPEED(3, T)

W(NUMBEROFSHOCKWAVE).A_VOL = VOL(3, T)

W(NUMBEROFSHOCKWAVE).A_DENSITY = DENSITY(3, T)

W(NUMBEROFSHOCKWAVE).B_SPEED = W(NUMBEROFSHOCKWAVE -
1).A_SPEED

W(NUMBEROFSHOCKWAVE).B_VOL = W(NUMBEROFSHOCKWAVE -
1).A_VOL

W(NUMBEROFSHOCKWAVE).B_DENSITY = W(NUMBEROFSHOCKWAVE -
1).A_DENSITY

W(NUMBEROFSHOCKWAVE).X_AXIS = DISTANCE1TO2 + DISTANCE2TO3 -
(DISTANCE1TO2 + DISTANCE2TO3 - W(NUMBEROFSHOCKWAVE - 1).X_AXIS) / 2

If W(NUMBEROFSHOCKWAVE).A_DENSITY <>
W(NUMBEROFSHOCKWAVE).B_DENSITY Then

```

```
W(NUMBEROFSHOCKWAVE).SPEED =  
(W(NUMBEROFSHOCKWAVE).A_VOL - W(NUMBEROFSHOCKWAVE).B_VOL) /  
(W(NUMBEROFSHOCKWAVE).A_DENSITY -  
W(NUMBEROFSHOCKWAVE).B_DENSITY)
```

```
Else: W(NUMBEROFSHOCKWAVE).SPEED = 0
```

```
End If
```

```
End If
```

```
For I = 1 To NUMBEROFSHOCKWAVE - 1
```

```
  If W(I).B_VOL = 0 Then
```

```
    For J = I To NUMBEROFSHOCKWAVE - 1
```

```
      W(J).A_SPEED = W(J + 1).A_SPEED
```

```
      W(J).A_VOL = W(J + 1).A_VOL
```

```
      W(J).A_DENSITY = W(J + 1).A_DENSITY
```

```
      W(J).B_SPEED = W(J + 1).B_SPEED
```

```
      W(J).B_VOL = W(J + 1).B_VOL
```

```
      W(J).B_DENSITY = W(J + 1).B_DENSITY
```

```
      W(J).SPEED = W(J + 1).SPEED
```

```
      W(J).X_AXIS = W(J + 1).X_AXIS
```

```
    Next J
```

```
  NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE - 1
```

```
End If
```

```

Next I

If W(NUMBEROFSHOCKWAVE).B_VOL = 0 Then

    W(NUMBEROFSHOCKWAVE - 1).A_SPEED =
W(NUMBEROFSHOCKWAVE).A_SPEED

    W(NUMBEROFSHOCKWAVE - 1).A_VOL =
W(NUMBEROFSHOCKWAVE).A_VOL

    W(NUMBEROFSHOCKWAVE - 1).A_DENSITY =
W(NUMBEROFSHOCKWAVE).A_DENSITY

End If

```

```

For I = 1 To NUMBEROFSHOCKWAVE

    WATBASETIME(I).A_SPEED = W(I).A_SPEED

    WATBASETIME(I).A_VOL = W(I).A_VOL

    WATBASETIME(I).A_DENSITY = W(I).A_DENSITY

    WATBASETIME(I).B_SPEED = W(I).B_SPEED

    WATBASETIME(I).B_VOL = W(I).B_VOL

    WATBASETIME(I).B_DENSITY = W(I).B_DENSITY

    WATBASETIME(I).SPEED = W(I).SPEED

    WATBASETIME(I).X_AXIS = W(I).X_AXIS

```

```

Next I

NUMBEROFSHOCKWAVEATBASETIME = NUMBEROFSHOCKWAVE

```

BASETIME = TIME: STOPTIME = BASETIME + PREDICTTIME

Call Calculation

If W(NUMBEROFSHOCKWAVE).X_AXIS < DISTANCE1TO2 Then

PREDICT_SPEED = W(NUMBEROFSHOCKWAVE).A_SPEED

PREDICT_VOL = W(NUMBEROFSHOCKWAVE).A_VOL

PREDICT_DENSITY = W(NUMBEROFSHOCKWAVE).A_DENSITY

ElseIf W(1).X_AXIS > DISTANCE1TO2 Then

PREDICT_SPEED = W(1).B_SPEED

PREDICT_VOL = W(1).B_VOL

PREDICT_DENSITY = W(1).B_DENSITY

Else:

For I = 1 To NUMBEROFSHOCKWAVE - 1

If W(I).X_AXIS < DISTANCE1TO2 And W(I + 1).X_AXIS > DISTANCE1TO2

Then

PREDICT_SPEED = W(I).A_SPEED

PREDICT_VOL = W(I).A_VOL

PREDICT_DENSITY = W(I).A_DENSITY

Exit For

ElseIf W(I).X_AXIS = DISTANCE1TO2 Then

PREDICT_SPEED = 0.5 * (W(I).A_SPEED + W(I).B_SPEED)

PREDICT_VOL = 0.5 * (W(I).A_VOL + W(I).B_VOL)

PREDICT_DENSITY = 0.5 * (W(I).A_DENSITY + W(I).B_DENSITY)

```

Exit For

End If

Next I

End If

Print #4, T + PREDICTTIME / TIMEINTERAL, PREDICT_SPEED, PREDICT_VOL,
PREDICT_DENSITY, NUMBEROFSHOCKWAVE

TIME = BASETIME: STOPTIME = TIMEOFRECORD(T + 1):
NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVEATBASETIME

For I = 1 To NUMBEROFSHOCKWAVE

W(I).A_SPEED = WATBASETIME(I).A_SPEED

W(I).A_VOL = WATBASETIME(I).A_VOL

W(I).A_DENSITY = WATBASETIME(I).A_DENSITY

W(I).B_SPEED = WATBASETIME(I).B_SPEED

W(I).B_VOL = WATBASETIME(I).B_VOL

W(I).B_DENSITY = WATBASETIME(I).B_DENSITY

W(I).SPEED = WATBASETIME(I).SPEED

W(I).X_AXIS = WATBASETIME(I).X_AXIS

Next I

If NUMBEROFSHOCKWAVE > 1 Then Call Calculation

```

T = T + 1

Loop

Print "MISSION COMPLETED."

End Sub

Private Sub Calculation()

If T = 1578 Then

Print

End If

Do

MINIMUMTIME = -1

For I = 1 To NUMBEROFSHOCKWAVE - 1

If W(I).SPEED <> W(I + 1).SPEED Then

If (W(I + 1).X_AXIS - W(I).X_AXIS) / (W(I).SPEED - W(I + 1).SPEED) >= 0 Then

If MINIMUMTIME = -1 Or (W(I + 1).X_AXIS - W(I).X_AXIS) / (W(I).SPEED -

W(I + 1).SPEED) < MINIMUMTIME Then

MINIMUMTIME = (W(I + 1).X_AXIS - W(I).X_AXIS) / (W(I).SPEED - W(I +

1).SPEED)

End If

```

    End If

End If

Next I

If MINIMUMTIME <> -1 And TIME + MINIMUMTIME <= STOPTIME Then

    TIME = TIME + MINIMUMTIME

    For I = 1 To NUMBEROFSHOCKWAVE

        W(I).X_AXIS = W(I).X_AXIS + W(I).SPEED * MINIMUMTIME

    Next I

    For I = 1 To NUMBEROFSHOCKWAVE - 1

        If Abs(W(I).X_AXIS - W(I + 1).X_AXIS) < 0.00001 Then

            W(I).A_SPEED = W(I + 1).A_SPEED

            W(I).A_VOL = W(I + 1).A_VOL

            W(I).A_DENSITY = W(I + 1).A_DENSITY

            If W(I).B_DENSITY <> W(I).A_DENSITY Then

                W(I).SPEED = (W(I).B_VOL - W(I).A_VOL) / (W(I).B_DENSITY -

W(I).A_DENSITY)

            Else: W(I).SPEED = 0

            End If

            W(I).X_AXIS = 0.5 * (W(I).X_AXIS + W(I + 1).X_AXIS)

            For J = I + 1 To NUMBEROFSHOCKWAVE - 1

                W(J).A_SPEED = W(J + 1).A_SPEED

```

W(J).A_VOL = W(J + 1).A_VOL

W(J).A_DENSITY = W(J + 1).A_DENSITY

W(J).B_SPEED = W(J + 1).B_SPEED

W(J).B_VOL = W(J + 1).B_VOL

W(J).B_DENSITY = W(J + 1).B_DENSITY

W(I).SPEED = W(J + 1).SPEED

W(J).X_AXIS = W(J + 1).X_AXIS

Next J

W(NUMBEROFSHOCKWAVE).A_SPEED = 0

W(NUMBEROFSHOCKWAVE).A_VOL = 0

W(NUMBEROFSHOCKWAVE).A_DENSITY = 0

W(NUMBEROFSHOCKWAVE).B_SPEED = 0

W(NUMBEROFSHOCKWAVE).B_VOL = 0

W(NUMBEROFSHOCKWAVE).B_DENSITY = 0

W(NUMBEROFSHOCKWAVE).SPEED = 0

W(NUMBEROFSHOCKWAVE).X_AXIS = 0

NUMBEROFSHOCKWAVE = NUMBEROFSHOCKWAVE - 1

End If

Next I

Else:

For I = 1 To NUMBEROFSHOCKWAVE

W(I).X_AXIS = W(I).X_AXIS + W(I).SPEED * (TIMEOFRECORD(T + 1) - TIME)

Next I

TIME = STOPTIME

Exit Do

End If

Loop

End Sub

Appendix IV Travel Time Estimation VB Program

Option Explicit

Dim TIME\$, RECORDOFTHETIME\$, E\$

Dim POINTER%, SPEEDSTART%, SPEEDEND%, LENGTHOFTHERECORD%

Private SPEED(5) As Single

Private DISTANCE(4) As Long

Private TRAVELTIME(4, 20) As Single

Private CALCULATEDDISTANCE(4, 20) As Single

Private ATS2(20) As Double

Private ATS3(20) As Double

Dim TT1!, TT2!

Dim AT2#, AT3#

Dim SENSOR%, DATENUMBER%, MINIMUMSPEEDOFTHETIME!

Dim TIMEUPPERLIMIT1!, TIMELOWERLIMIT1!, TIMEUPPERLIMIT2!,

TIMELOWERLIMIT2!

Dim CALCULATEDDISTANCE1!, CALCULATEDDISTANCE2!, MC1!, MC2!

Dim MINIMUMVARIANCE!

Dim S1%, S2%, S3%, D1%, D2%, T1%, T2%

Dim A#, B#, C#, A1#, B1#, C1#, A2#, B2#, C2#

Dim DELTA#, DELTA1#, DELTA2#

Dim P%, I%, K%, N%, J#, FRONT#, REAR#

Dim BASICSP#, SP#

Dim ROOT1#, ROOT2#, R0!, R1!, R2!, R3!, R4!

Dim M%, NUMBEROFANSWERS1%, NUMBEROFANSWERS2%

Dim TROUBLE As Boolean

Dim SP2!

Dim FAT2J#

Dim ATS2J!, ATS3J!

Private Sub Start_Click()

DISTANCE(1) = 636: DISTANCE(2) = 417: DISTANCE(3) = 522: DISTANCE(4) = 475

Open "Speed Data.txt" For Input As #1

Open "Travel Time.txt" For Output As #3

Do While Not EOF(1)

Line Input #1, RECORDOFTHETIME

LENGTHOFTHERECORD = Len(RECORDOFTHETIME)

TIME = Left(RECORDOFTHETIME, 10)

POINTER = 11

For DATENUMBER = 1 To 19

Status.Cls: Status.Print "The computer is calculating the data of time "; TIME; " and
date August "; DATENUMBER; "."

For SENSOR = 1 To 5

If DATENUMBER = 19 And SENSOR = 5 Then

SPEED(SENSOR) = CSng(Right(RECORDOFTHETIME,
LENGTHOFTHERECORD - POINTER)) * 1609 / 3600

Else:

SPEEDSTART = POINTER

SPEEDEND = InStr(POINTER + 1, RECORDOFTHETIME, " ")

SPEED(SENSOR) = CSng(Mid(RECORDOFTHETIME, SPEEDSTART + 1,
SPEEDEND - SPEEDSTART - 1)) * 1609 / 3600

POINTER = SPEEDEND

End If

Next SENSOR

Print #3, TIME; " Date"; DATENUMBER, Int((SPEED(1) * 3600 / 1609) * 10 + 0.5) /
10; " "; Int((SPEED(2) * 3600 / 1609) * 10 + 0.5) / 10; " ";

Print #3, Int((SPEED(3) * 3600 / 1609) * 10 + 0.5) / 10; " "; Int((SPEED(4) * 3600 /
1609) * 10 + 0.5) / 10; " "; Int((SPEED(5) * 3600 / 1609) * 10 + 0.5) / 10,

For P = 1 To 2

If P = 1 Then

S1 = 1: S2 = 2: S3 = 3: D1 = 1: D2 = 2: T1 = 1: T2 = 2: NUMBEROFANSWERS1
= 0

```

Else:
    S1 = 3: S2 = 4: S3 = 5: D1 = 3: D2 = 4: T1 = 3: T2 = 4: NUMBEROFANSWERS2
= 0
End If
M = 0

Call CALCULATEAT2AT3

For I = 1 To M
    A = (SPEED(S3) * ATS2(I) - SPEED(S2) * ATS3(I) + SPEED(S1) * ATS3(I) -
SPEED(S1) * ATS2(I)) / (ATS2(I) * ATS3(I) * (ATS3(I) - ATS2(I)))
    B = (SPEED(S2) - SPEED(S1) - A * ATS2(I) * ATS2(I)) / ATS2(I)
    C = SPEED(S1)
    If A = 0 Then
        CALCULATEDDISTANCE1 = (SPEED(S1) + SPEED(S2)) * ATS2(I) / 2
        CALCULATEDDISTANCE2 = (SPEED(S2) + SPEED(S3)) * (ATS3(I) -
ATS2(I)) / 2
        If Abs(DISTANCE(D1) - CALCULATEDDISTANCE1) + Abs(DISTANCE(D2)
- CALCULATEDDISTANCE2) > 100 Then Call TROUBLECASE
        Call SAVEANSWERS
    ElseIf A > 0 And B * B / (4 * A) - B * B / (2 * A) + C < 1 * (1609 / 3600) Then

```

```

If ((-1) * B + Sqr(B * B - 4 * A * (C - 1 * (1609 / 3600)))) / (2 * A) <= 0 Or ((-1)
* B - Sqr(B * B - 4 * A * (C - 1 * (1609 / 3600)))) / (2 * A) >= ATS3(I) Then

    TT1 = ATS2(I): TT2 = ATS3(I) - ATS2(I)

    CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B *
TT1 * TT1 + C * TT1

    CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) *
(TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -
CALCULATEDDISTANCE1

    If Abs(DISTANCE(D1) - CALCULATEDDISTANCE1) +
Abs(DISTANCE(D2) - CALCULATEDDISTANCE2) > 100 Then Call TROUBLECASE

        Call SAVEANSWERS

    Else:

        Call TROUBLECASE

        Call SAVEANSWERS

    End If

ElseIf A < 0 And B * B / (4 * A) - B * B / (2 * A) + C > 135 * (1609 / 3600) Then

    If ((-1) * B + Sqr(B * B - 4 * A * (C - 135 * (1609 / 3600)))) / (2 * A) <= 0 Or
((-1) * B - Sqr(B * B - 4 * A * (C - 135 * (1609 / 3600)))) / (2 * A) >= ATS3(I) Then

        TT1 = ATS2(I): TT2 = ATS3(I) - ATS2(I)

        CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B *
TT1 * TT1 + C * TT1

```

CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) *
(TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -

CALCULATEDDISTANCE1

If Abs(DISTANCE(D1) - CALCULATEDDISTANCE1) +
Abs(DISTANCE(D2) - CALCULATEDDISTANCE2) > 100 Then Call TROUBLECASE

Call SAVEANSWERS

Else:

Call TROUBLECASE

Call SAVEANSWERS

End If

Else:

TT1 = ATS2(I); TT2 = ATS3(I) - ATS2(I)

CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B *
TT1 * TT1 + C * TT1

CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) *
(TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -

CALCULATEDDISTANCE1

If Abs(DISTANCE(D1) - CALCULATEDDISTANCE1) + Abs(DISTANCE(D2)
- CALCULATEDDISTANCE2) > 100 Then Call TROUBLECASE

Call SAVEANSWERS

End If

Next I

Next P

Call PRINTOUTPUT

Print #3,

Next DATENUMBER

Print #3,

Loop

Status.Cls: Status.Print "Calculating completed."

Close #1: Close #3

End Sub

Private Sub CALCULATEAT2AT3()

MINIMUMVARIANCE = -1

If SPEED(S1) = SPEED(S2) And SPEED(S2) = SPEED(S3) Then

M = M + 1

ATS2(M) = DISTANCE(D1) / SPEED(S1): ATS3(M) = DISTANCE(D2) / SPEED(S2) +

ATS2(M)

Else:

A2 = SPEED(S1) * SPEED(S1) + SPEED(S1) * SPEED(S2) + SPEED(S2) * SPEED(S2)

+ 3 * SPEED(S1) * SPEED(S3) + 3 * SPEED(S2) * SPEED(S3)

$B2 = (-6) * DISTANCE(D1) * (SPEED(S1) + SPEED(S2) + SPEED(S3))$

$C2 = 9 * DISTANCE(D1) * DISTANCE(D1)$

$SP = 4$

Do

For N = -1 To 1 Step 2

J = 10

Do While J <= DISTANCE(D1) / (1 * (1609 / 3600))

Call SEARCHROOTS

J = J + SP

Loop

Next N

$SP = SP / 4$

Loop Until M > 0 Or SP < 1 / 128

End If

If M < 1 And ATS2J > 0 And ATS3J > 0 Then

M = 1: ATS2(M) = ATS2J: ATS3(M) = ATS3J

End If

End Sub

Private Sub SEARCHROOTS()

TROUBLE = False: FAT2J = FAT2(J, S1, S2, S3, D1, D2)

If $SP < 1 / 8$ And $SP \geq 1 / 32$ And $AT3 \geq J + 5$ And $AT3 - J \leq \text{DISTANCE}(D2) / (1 * (1609 / 3600))$ Then

TT1 = J: TT2 = AT3 - TT1

Call JIFEN

If MINIMUMVARIANCE = -1 And TROUBLE = False Then

MINIMUMVARIANCE = $\text{Abs}(\text{DISTANCE}(D1) - \text{CALCULATEDDISTANCE1}) + \text{Abs}(\text{DISTANCE}(D2) - \text{CALCULATEDDISTANCE2})$

ATS2J = TT1: ATS3J = TT1 + TT2

ElseIf $\text{Abs}(\text{DISTANCE}(D1) - \text{CALCULATEDDISTANCE1}) + \text{Abs}(\text{DISTANCE}(D2) - \text{CALCULATEDDISTANCE2}) < \text{MINIMUMVARIANCE}$ And TROUBLE = False Then

MINIMUMVARIANCE = $\text{Abs}(\text{DISTANCE}(D1) - \text{CALCULATEDDISTANCE1}) + \text{Abs}(\text{DISTANCE}(D2) - \text{CALCULATEDDISTANCE2})$

ATS2J = TT1: ATS3J = TT1 + TT2

End If

End If

If $\text{Abs}(\text{FAT2J}) \leq 1$ And TROUBLE = False Then

M = M + 1: ATS2(M) = J

J = J + 2

$\text{ATS3}(M) = ((-1) * B1 + N * \text{Sqr}(B1 * B1 - 4 * A1 * C1)) / (2 * A1)$

```

If  $ATS3(M) \leq ATS2(M) + 5$  Or  $ATS3(M) - ATS2(M) > DISTANCE(D2) / (1 * (1609 / 3600))$  Then

     $M = M - 1$ :  $J = J - 1$ 

End If

ElseIf  $Abs(FAT2J) > 1$  And  $FAT2J * FAT2(J + SP, S1, S2, S3, D1, D2) < 0$  And  $TROUBLE = False$  Then

     $FRONT = J$ :  $REAR = J + SP$ 

    Do

        If  $FAT2(FRONT, S1, S2, S3, D1, D2) * FAT2((FRONT + REAR) / 2, S1, S2, S3, D1, D2) < 0$  And  $TROUBLE = False$  Then

             $REAR = (FRONT + REAR) / 2$ 

        ElseIf  $TROUBLE = False$  Then

             $FRONT = (FRONT + REAR) / 2$ 

        End If

    Loop Until  $REAR - FRONT < 0.001$  Or  $TROUBLE = True$ 

    If  $TROUBLE = False$  Then

         $M = M + 1$ :  $ATS2(M) = FRONT$ 

         $A1 = 6 * DISTANCE(D1) - 3 * SPEED(S1) * ATS2(M) - 3 * SPEED(S2) * ATS2(M)$ 

         $B1 = 4 * SPEED(S1) * ATS2(M) * ATS2(M) - 6 * DISTANCE(D1) * ATS2(M) + 2 * SPEED(S2) * ATS2(M) * ATS2(M)$ 

         $C1 = (SPEED(S3) - SPEED(S1)) * ATS2(M) * ATS2(M) * ATS2(M)$ 

         $J = J + 1$ 

```

$$ATS3(M) = ((-1) * B1 + N * Sqr(B1 * B1 - 4 * A1 * C1)) / (2 * A1)$$

If $ATS3(M) \leq ATS2(M) + 5$ Or $ATS3(M) - ATS2(M) > DISTANCE(D2) / (1 * (1609 / 3600))$ Then

$$M = M - 1; J = J - 1$$

End If

End If

End If

End Sub

Private Function FAT2(AT2, S1, S2, S3, D1, D2) As Double

$$A1 = 6 * DISTANCE(D1) - 3 * SPEED(S1) * AT2 - 3 * SPEED(S2) * AT2$$

$$B1 = 4 * SPEED(S1) * AT2 * AT2 - 6 * DISTANCE(D1) * AT2 + 2 * SPEED(S2) * AT2 * AT2$$

$$C1 = (SPEED(S3) - SPEED(S1)) * AT2 * AT2 * AT2$$

$$DELTA1 = B1 * B1 - 4 * A1 * C1$$

If $DELTA1 \geq 0$ Then

$$AT3 = ((-1) * B1 + N * Sqr(DELTA1)) / (2 * A1)$$

$$FAT2 = (AT3 - AT2) * ((-1) * SPEED(S1) * (AT3 - AT2) * (AT3 - AT2) + SPEED(S2) * AT3 * AT3 + 2 * SPEED(S2) * AT2 * AT3 + 2 * SPEED(S3) * AT2 * AT3 + AT2 * AT2 * SPEED(S3)) - 6 * AT2 * AT3 * DISTANCE(D2)$$

Else:

TROUBLE = True

End If

End Function

Private Sub TROUBLECASE()

MINIMUMVARIANCE = -1

SP2 = 8

$\text{TIMELOWERLIMIT1} = \text{DISTANCE(D1)} / (135 * (1609 / 3600))$: $\text{TIMEUPPERLIMIT1} =$
 $\text{DISTANCE(D1)} / (1 * (1609 / 3600))$

$\text{TIMELOWERLIMIT2} = \text{DISTANCE(D2)} / (135 * (1609 / 3600))$: $\text{TIMEUPPERLIMIT2} =$
 $\text{DISTANCE(D2)} / (1 * (1609 / 3600))$

If $\text{Abs}(\text{SPEED(S1)} - \text{SPEED(S2)}) < 10 * (1609 / 3600)$ And $\text{Abs}(\text{SPEED(S2)} - \text{SPEED(S3)})$
< $10 * (1609 / 3600)$ And $\text{Abs}(\text{SPEED(S1)} - \text{SPEED(S3)}) < 10 * (1609 / 3600)$ Then

$\text{TIMELOWERLIMIT1} = \text{DISTANCE(D1)} / (((\text{SPEED(S1)} + \text{SPEED(S2)} + \text{SPEED(S3)}) /$
 $3 + 30) * (1609 / 3600))$

$\text{TIMEUPPERLIMIT1} = \text{DISTANCE(D1)} / (((\text{SPEED(S1)} + \text{SPEED(S2)} + \text{SPEED(S3)}) /$
 $3 - 30) * (1609 / 3600))$

$\text{TIMELOWERLIMIT2} = \text{DISTANCE(D2)} / (((\text{SPEED(S1)} + \text{SPEED(S2)} + \text{SPEED(S3)}) /$
 $3 + 30) * (1609 / 3600))$

$$\text{TIMEUPPERLIMIT2} = \text{DISTANCE(D2)} / (((\text{SPEED(S1)} + \text{SPEED(S2)} + \text{SPEED(S3)}) / 3 - 30) * (1609 / 3600))$$

If $\text{TIMELOWERLIMIT1} < \text{DISTANCE(D1)} / (135 * (1609 / 3600))$ Then
$$\text{TIMELOWERLIMIT1} = \text{DISTANCE(D1)} / (135 * (1609 / 3600))$$

If $\text{TIMELOWERLIMIT2} < \text{DISTANCE(D2)} / (135 * (1609 / 3600))$ Then
$$\text{TIMELOWERLIMIT2} = \text{DISTANCE(D2)} / (135 * (1609 / 3600))$$

If $\text{TIMEUPPERLIMIT1} < \text{DISTANCE(D1)} / (1 * (1609 / 3600))$ Then
$$\text{TIMEUPPERLIMIT1} = \text{DISTANCE(D1)} / (1 * (1609 / 3600))$$

If $\text{TIMEUPPERLIMIT2} < \text{DISTANCE(D2)} / (1 * (1609 / 3600))$ Then
$$\text{TIMEUPPERLIMIT2} = \text{DISTANCE(D2)} / (1 * (1609 / 3600))$$

End If

Do

For $\text{TT1} = \text{TIMELOWERLIMIT1}$ To TIMEUPPERLIMIT1 Step SP2

For $\text{TT2} = \text{TIMELOWERLIMIT2}$ To TIMEUPPERLIMIT2 Step SP2

Call JIFEN

If $\text{MINIMUMVARIANCE} = -1$ Then

$$\text{MINIMUMVARIANCE} = \text{Abs}(\text{DISTANCE(D1)} - \text{CALCULATEDDISTANCE1})$$

$$+ \text{Abs}(\text{DISTANCE(D2)} - \text{CALCULATEDDISTANCE2})$$

$$\text{ATS2(I)} = \text{TT1}; \text{ATS3(I)} = \text{TT2} + \text{TT1}; \text{MC1} = \text{CALCULATEDDISTANCE1};$$

$$\text{MC2} = \text{CALCULATEDDISTANCE2}$$

ElseIf Abs(DISTANCE(D1) - CALCULATEDDISTANCE1) + Abs(DISTANCE(D2) - CALCULATEDDISTANCE2) < MINIMUMVARIANCE Then

MINIMUMVARIANCE = Abs(DISTANCE(D1) - CALCULATEDDISTANCE1) + Abs(DISTANCE(D2) - CALCULATEDDISTANCE2)

ATS2(I) = TT1: ATS3(I) = TT2 + TT1: MC1 = CALCULATEDDISTANCE1:
MC2 = CALCULATEDDISTANCE2

End If

Next TT2

If MINIMUMVARIANCE < 100 Then TT1 = TT1 + 2

Next TT1

CALCULATEDDISTANCE1 = MC1: CALCULATEDDISTANCE2 = MC2

SP2 = SP2 / 4

Loop Until MINIMUMVARIANCE < 100 Or SP2 <= 0.5

End Sub

Private Sub JIFEN()

A = (SPEED(S3) * TT1 - SPEED(S2) * (TT1 + TT2) + SPEED(S1) * TT2) / (TT1 * (TT1 + TT2) * TT2)

B = (SPEED(S2) - SPEED(S1) - A * TT1 * TT1) / TT1

C = SPEED(S1)

If $A > 0$ And $B * B - 4 * A * (C - 1 * (1609 / 3600)) \leq 0$ Then

$CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1$
 $+ C * TT1$

$CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2)$
 $+ (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -$

$CALCULATEDDISTANCE1$

ElseIf $A > 0$ And $B * B - 4 * A * (C - 1 * (1609 / 3600)) > 0$ Then

$ROOT1 = ((-1) * B - Sqr(B * B - 4 * A * (C - 1 * (1609 / 3600)))) / (2 * A)$

$ROOT2 = ((-1) * B + Sqr(B * B - 4 * A * (C - 1 * (1609 / 3600)))) / (2 * A)$

If $ROOT2 \leq 0$ Or $ROOT1 \geq TT1 + TT2$ Then

$CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1 + C * TT1$

$CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -$

$CALCULATEDDISTANCE1$

ElseIf $TT1 \leq ROOT1$ Then

$R0 = 0$

$R1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1 + C * TT1$

$R2 = (1 / 3) * A * ROOT1 * ROOT1 * ROOT1 + (1 / 2) * B * ROOT1 * ROOT1 + C * ROOT1$

$R3 = (1 / 3) * A * ROOT2 * ROOT2 * ROOT2 + (1 / 2) * B * ROOT2 * ROOT2 + C * ROOT2$

$$R4 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2)$$

$$CALCULATEDDISTANCE1 = R1 - R0$$

$$CALCULATEDDISTANCE2 = R2 - R1 + 1 * (1609 / 3600) * (ROOT2 - ROOT1) + R4 - R3$$

ElseIf TT1 >= ROOT2 Then

$$R0 = 0$$

$$R1 = (1 / 3) * A * ROOT1 * ROOT1 * ROOT1 + (1 / 2) * B * ROOT1 * ROOT1 + C * ROOT1$$

$$R2 = (1 / 3) * A * ROOT2 * ROOT2 * ROOT2 + (1 / 2) * B * ROOT2 * ROOT2 + C * ROOT2$$

$$R3 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1 + C * TT1$$

$$R4 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2)$$

$$CALCULATEDDISTANCE1 = R1 - R0 + 1 * (1609 / 3600) * (ROOT2 - ROOT1) + R3 - R2$$

$$CALCULATEDDISTANCE2 = R4 - R3$$

End If

ElseIf A = 0 Then

$$CALCULATEDDISTANCE1 = (SPEED(S1) + SPEED(S2)) * TT1 / 2$$

$$CALCULATEDDISTANCE2 = (SPEED(S2) + SPEED(S3)) * TT2 / 2$$

ElseIf A < 0 And B * B - 4 * A * (C - 135 * (1609 / 3600)) <= 0 Then

CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1
+ C * TT1

CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2)
+ (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -

CALCULATEDDISTANCE1

ElseIf A < 0 And B * B - 4 * A * (C - 135 * (1609 / 3600)) > 0 Then

ROOT1 = ((-1) * B - Sqr(B * B - 4 * A * (C - 135 * (1609 / 3600)))) / (2 * A)

ROOT2 = ((-1) * B + Sqr(B * B - 4 * A * (C - 135 * (1609 / 3600)))) / (2 * A)

If ROOT2 <= 0 Or ROOT1 >= TT1 + TT2 Then

CALCULATEDDISTANCE1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 *
TT1 + C * TT1

CALCULATEDDISTANCE2 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 +
TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2) -

CALCULATEDDISTANCE1

ElseIf TT1 <= ROOT1 Then

R0 = 0

R1 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1 + C * TT1

R2 = (1 / 3) * A * ROOT1 * ROOT1 * ROOT1 + (1 / 2) * B * ROOT1 * ROOT1 + C *
ROOT1

R3 = (1 / 3) * A * ROOT2 * ROOT2 * ROOT2 + (1 / 2) * B * ROOT2 * ROOT2 + C *
ROOT2

$$R4 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2)$$

$$CALCULATEDDISTANCE1 = R1 - R0$$

$$CALCULATEDDISTANCE2 = R2 - R1 + 135 * (1609 / 3600) * (ROOT2 - ROOT1) + R4 - R3$$

ElseIf TT1 >= ROOT2 Then

$$R0 = 0$$

$$R1 = (1 / 3) * A * ROOT1 * ROOT1 * ROOT1 + (1 / 2) * B * ROOT1 * ROOT1 + C * ROOT1$$

$$R2 = (1 / 3) * A * ROOT2 * ROOT2 * ROOT2 + (1 / 2) * B * ROOT2 * ROOT2 + C * ROOT2$$

$$R3 = (1 / 3) * A * TT1 * TT1 * TT1 + (1 / 2) * B * TT1 * TT1 + C * TT1$$

$$R4 = (1 / 3) * A * (TT1 + TT2) * (TT1 + TT2) * (TT1 + TT2) + (1 / 2) * B * (TT1 + TT2) * (TT1 + TT2) + C * (TT1 + TT2)$$

$$CALCULATEDDISTANCE1 = R1 - R0 + 135 * (1609 / 3600) * (ROOT2 - ROOT1) + R3 - R2$$

$$CALCULATEDDISTANCE2 = R4 - R3$$

End If

End If

End Sub

Private Sub SAVEANSWERS()

If P = 1 Then

 NUMBEROFANSWERS1 = NUMBEROFANSWERS1 + 1

 TRAVELTIME(1, NUMBEROFANSWERS1) = AT2(I): TRAVELTIME(2,
NUMBEROFANSWERS1) = AT3(I) - AT2(I)

 CALCULATEDDISTANCE(1, NUMBEROFANSWERS1) =
CALCULATEDDISTANCE1

 CALCULATEDDISTANCE(2, NUMBEROFANSWERS1) =
CALCULATEDDISTANCE2

ElseIf P = 2 Then

 NUMBEROFANSWERS2 = NUMBEROFANSWERS2 + 1

 TRAVELTIME(3, NUMBEROFANSWERS2) = AT2(I): TRAVELTIME(4,
NUMBEROFANSWERS2) = AT3(I) - AT2(I)

 CALCULATEDDISTANCE(3, NUMBEROFANSWERS2) =
CALCULATEDDISTANCE1

 CALCULATEDDISTANCE(4, NUMBEROFANSWERS2) =
CALCULATEDDISTANCE2

End If

End Sub

```
Private Sub PRINTOUTPUT()
```

```
For I = 1 To NUMBEROFANSWERS1
```

```
For K = 1 To NUMBEROFANSWERS2
```

```
Print #3, Int(TRAVELTIME(1, I) * 10 + 0.5) / 10, Int(TRAVELTIME(2, I) * 10 + 0.5) /  
10, Int(TRAVELTIME(3, K) * 10 + 0.5) / 10, Int(TRAVELTIME(4, K) * 10 + 0.5) / 10,
```

```
Print #3, Int(TRAVELTIME(1, I) * 10 + 0.5) / 10 + Int(TRAVELTIME(2, I) * 10 + 0.5) /  
10 + Int(TRAVELTIME(3, K) * 10 + 0.5) / 10 + Int(TRAVELTIME(4, K) * 10 + 0.5) / 10,
```

```
Print #3, Int((CALCULATEDDISTANCE(1, I) - DISTANCE(1)) * 1 + 0.5) / 1,  
Int((CALCULATEDDISTANCE(2, I) - DISTANCE(2)) * 1 + 0.5) / 1,
```

```
Print #3, Int((CALCULATEDDISTANCE(3, K) - DISTANCE(3)) * 1 + 0.5) / 1,  
Int((CALCULATEDDISTANCE(4, K) - DISTANCE(4)) * 1 + 0.5) / 1,
```

```
Next K
```

```
Next I
```

```
End Sub
```

Bibliography

- Abbas, M. M., and Bullock, D. (2003). "On-line measure of shockwaves for ITS applications." *Journal of Transportation Engineering*, 1(1), 1-6.
- Addison, P. S., and Low, D. J. (1998). "A novel nonlinear car-following model." *Chaos*, 8(4), 791-799.
- Ahmed, K. I. (1999). *Modeling Drivers' Acceleration and Lane Changing Behaviors*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Alexiadis, V., Colyar, J., Halkias, J., Hranac, R., and McHale, G. (2004). "The Next Generation Simulation Program." *ITE Journal*, August, 22-26.
- Aycin, M. F., and Benekohal, R. (1998). "A linear acceleration car following model development and validation." *Proceedings of the 77th annual meeting*, Transportation Research Board, Washington, DC.
- Aycin, M. F., and Benekohal, R. F. (2001). "Stability and performance of car following models in congested traffic." *Journal of Transportation Engineering*, 127(1), 2-12.
- Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. N., and Rabi, M. (2001). "Network state estimation and prediction for real-time transportation management applications." *Networks and Spatial Economics*, 1(3-4), 293-318.

- Bexelius, S. (1968). "An extended model for car following." *Transportation Research*, 2, 13-21.
- Bhat, C. (1997). "Endogenous segmentation mode choice model with an application to intercity travel." *Transportation Science*, 31(1), 34-48.
- Brackstone, M., and McDonald, M. (1999). "Car-following: a historical review." *Transportation Research, Part F*, 2, 181-196.
- Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and Regression Trees*, Wadsworth, Belmont, CA.
- Chandler, R., Herman, R., and Montroll, E. W. (1958). "Traffic dynamics: studies in car following." *Operations Research*, 6(2), 165-184.
- Cleveland, W. S. (1979). "Robust locally weighted regression and smoothing scatter plots." *Journal of American Statistical Association*, 74, 829-836.
- Cleveland, W. S., and Devlin, S. J. (1988). "Locally weighted regression: an approach to regression analysis by local fitting." *Journal of American Statistical Association*, 83, 596-610.
- Coifman, B. (1996). "Time space diagrams for thirteen shock waves." *Report of California Partners for Advanced Highways and Transit (PATH) Program*, University of California, Berkeley, CA.

- Cortes, C. E., Lavanya, R., Oh, J. S., and Jayakrishnan, R. (2002). "A general purpose methodology for link travel time estimation using multiple point detection of traffic". *Transportation Research Record*, 1802, 181-189
- Courant, R. (1988). *Differential and integral calculus*, John Wiley & Sons, London, UK.
- Dailey, D. J. (1993). "Travel time estimation using cross-correlation techniques." *Transportation Research, Part B*, 27, 97-107.
- Dailey, D. J. (1999). "A statistical algorithm for estimating speed from single loop volume and occupancy measurement." *Transportation Research, Part B*, 33, 313-322.
- Drake, J. S., Schofer, J. L., and May, A. D. (1967). "A statistical analysis of speed density hypothesis." *Highway Research Record*, 154, 53-87.
- Drew, D. R. (1968). *Traffic Flow Theory and Control*, McGraw-Hill, New York, NY.
- Eddie, L. C. (1961). "Car-following and steady-state theory for noncongested traffic." *Operations Research*, 9(1), 66-76.
- Epperson, J. (2003). *An introduction to numerical methods and analysis*, John Wiley & Sons, New York, NY.
- Fancher, P. S., and Bareket, Z. (1998). "Evolving model for studying driver-vehicle system performance in longitudinal control of headway." *Transportation Research Record*, 1631, 13-19.

- Fariello, B. (2002). "ITS America RFI Travel Time Projects in North America." *Report of San Antonio TransGuide Program*, Texas Department of Transportation, Austin, TX.
- Friedman, J. H. (1991). "Multivariate adaptive regression splines." *The Annals of Statistics*, 19(1), 1-67.
- Gazis, D. C., Herman, R., and Potts, R. B. (1959). "Car following theory of steady-state traffic flow." *Operations Research*, 7(4), 499-505.
- Gazis, D. C., Herman, R., and Rothery, R. W. (1961). "Nonlinear following-the-leader models of traffic flow." *Operations Research*, 9, 545-567.
- Gipps, P. G. (1981). "A behavioral car-following model for computer simulation." *Transportation Research, Part B*, 15, 105-111.
- Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*, Addison Wesley Publishing, Reading, MA.
- Greenberg, H. (1959). "An analysis of traffic flow." *Operations Research*, 7(1), 255-275.
- Greenshields, B. D. (1935). "A study of traffic capacity." *Proceedings of the Highway Research Board*, 14(1), 448-477.
- Hall, F., and Persaud, B. (1989). "Evaluation of speed estimates made with single detector data from freeway traffic management systems." *Transportation Research Record*, 1232, 9-16.

- Hamdar, S. H., and Mahmassani, H. S. (2008). "From existing accident-free car-following models to colliding vehicles: exploration and assessment." *Proceedings of the 87th annual meeting*, Transportation Research Board, Washington, DC.
- Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The elements of statistical learning: data mining, Inference, and Prediction*, Springer, New York, NY.
- Heath, M. (1997). *Scientific Computing: An introductory survey*, McGraw-Hill, New York, NY.
- Helbing, D. (2001). "Traffic and related self-driven many-particle systems." *Reviews of Modern Physics*, 73(4), 1067-1141.
- Helbing, D., and Treiber, M. (1998). "Jams, waves, and clusters." *Science*, 282, 2001-2003.
- Herman, R., Montroll, E. W., Potts, R. B., and Rothery, R. (1959). "Traffic dynamics: analysis of stability in car following." *Operations Research*, 7(1), 86-106.
- Hillier, F. S., and Lieberman, G. J. (2005). *Introduction to operations research*, McGraw Hill, New York, NY.
- Helly, W. (1961). "Simulation in bottlenecks in single-lane traffic flow." *Proceedings of the Symposium on the Theory of Traffic Flow*, Amsterdam, Netherlands, 207-238.
- Khisty, C. J., and Lall, B. K. (2003). *Transportation engineering: an introduction*, Prentice Hall, Upper Saddle River, NJ.

- Kikuchi, C., and Chakroborty, P. (1992). "Car following model based on a fuzzy inference system." *Transportation Research Record*, 1365, 82-91.
- Kockelman, K. M. (2001). "Modeling traffic's flow-density relation: accommodation of multiple flow regimes and traveler types." *Transportation*, 28(4), 363-374.
- Ishak, S., and Al-Deek, H. (2002). "Performance evaluation of short-term time series traffic prediction model." *Journal of Transportation Engineering*, 128(6), 490-498.
- Lee, G. (1966). "A generalization of linear car following theory." *Operations Research*, 14(4), 595-606.
- Leutzbach, W., and Wiedemann, R. (1986). "Development and applications of traffic simulation models at the Karlsruhe Institut für Verkehrswesen." *Traffic Engineering and Control*, 27(5), 270-278.
- Leutzbach, W. (1988). *Introduction to the theory of traffic flow*, Springer-Verlag, Berlin, Germany.
- Lighthill, M. J., and Whitham, G. B. (1955). "On kinematic waves II. A theory of traffic flow on long crowded roads." *Proceedings of the Royal Society (London), Series A*, 229(1178), 317-345.
- Maddala, G. S. (1983). *Limited-dependent and qualitative variables in econometrics*, *econometric society monographs in quantitative economics*, Cambridge University, Cambridge, MA.

- Mahmassani, H. S. (2001). "Dynamic network traffic assignment and simulation methodology for advanced system management application." *Networks and Spatial Economics*, 1, 267-292.
- May, A. D. (1990). *Traffic flow fundamentals*, Prentice Hall, Upper Saddle River, NJ.
- May, A. D., and Keller, H. E. M. (1967). "Non-integer car following models." *Highway Research Record*, 199, 19-32.
- Michaels, R. M. (1963). "Perceptual factors in car following." *Proceedings of the Second International Symposium on the Theory of Road Traffic Flow*, 44-59.
- Michaels, R. M., and Cozan, L. W. (1963). "Perceptual and field factors causing lateral displacement." *Highway Research Record*, 25, 1-13.
- Nam, D. H., and Drew, D. R. (1996). "Traffic dynamics: methods for estimating freeway travel times in real time from flow measurements." *Journal of Transportation Engineering*, 122, 185-191.
- Newell, G. F. (1961). "Nonlinear effects in the dynamics of car following." *Operations Research*, 9(2), 209-229.
- Oh, S., Ritchie, S. G., and Oh, C. (2002). "Real time traffic measurement from single loop inductive signatures." *Proceedings of the 81st annual meeting*, Transportation Research Board, Washington, DC.

- Oh, J. S., Jayakrishnan, R., and Recker, W. (2003). "Section travel time estimation from point detection data." *Proceedings of the 82nd annual meeting*, Transportation Research Board, Washington, DC.
- Parzen, E. (1962). "On estimation of a probability density function and mode." *Annals of Mathematical Statistics*, 33, 1065-1076.
- Petty, K. F., Bickel, P. Ostland, M., Rice, J. Schoenberg, F., Jiang, J., and Rotov, Y. (1998). "Accurate estimation of travel time from single loop detectors." *Transportation Research, Part A*, 32, 1-17.
- Pipes, L. A. (1953). "An operational analysis of traffic dynamics." *Journal of Applied Physics*, 24, 274-281.
- Pipes, L. A. (1967). "Car-following models and the fundamental diagram of road traffic." *Transportation Research*, 1(1), 21-29.
- Quandt, R. E. (1958). "The estimation of the parameters of a linear regression system obeying two separate regimes." *Journal of the American Statistical Association*, 53, 873-880.
- Quandt, R. E. (1960). "Tests of hypothesis that a linear regression system obeys two separate regimes." *Journal of the American Statistical Association*, 53, 324-330.

- Rakha, H., and Crowther, B. (2003). "Comparison and calibration of FRESIM and INTEGRATION steady-state car-following behavior." *Transportation Research, Part A*, 37, 1-27.
- Reuschel, A. (1950). "Fahrzeugbewegungen in der Kolonne." *Oesterreichisches Ingenieurarchiv*, 4, 193-215.
- Richard, P. I. (1956). "Shock waves on the highway." *Operations Research*, 4(1), 42-51.
- Jennrich, R. I. (1969). "Asymptotic properties of non-linear least squares estimators." *Annals of Mathematical Statistics*, 40(2), 633-643.
- Seber, G. A. F., and Wild, C. J. (1989). *Nonlinear regression*, John Wiley & Sons, New York, NY.
- Srinivasan, K., and Jovanis, P. (1996). "Determination of number of probe vehicles required for reliable travel time measurement in urban network." *Transportation Research Record*, 1537, 15-22.
- Stone, C. J. (1977). "Nonparametric regression and its applications (with discussion)." *Annals of Mathematical Statistics*, 5, 595-645.
- Sun, L., and Zhou, J. (2005a). "Developing multi-regime speed-density relationships using cluster analysis." *Transportation Research Record*, 1934, 64-71.

- Sun, L., and Zhou, J. (2005b). "Development of multi-regime speed-density relation by cluster analysis." *Proceedings of the 16th International Transportation and Traffic Symposium*, University of Maryland, College Park, MD, 20-24.
- Toledo, T. (2003). *Integrated Driving Behavior Modeling*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Van Aerde, M. (1995). "Single regime speed-flow-density for congested and uncongested highways." *Proceedings of the 74th annual meeting*, Transportation Research Board, Washington, DC.
- Van Aerde, M., and Rakha, H. (1995). "Multivariate calibration of single-regime speed-flow-density relationships." *Proceedings of Vehicle Navigation and Information Conference*, Institute of Electrical and Electronics Engineers, Seattle, WA.
- Van Lint, J. W. C., and Van der Zijpp, N. J. (2003). "An improved travel-time estimation algorithm using dual loop detectors." *Proceedings of the 82nd annual meeting*, Transportation Research Board, Washington, DC.
- Wardrop, J. G. (1952). "Some theoretical aspects of road traffic research." *Proceedings of the Institution of Civil Engineers, Part II*, I, 325-362.
- Wei, H. Feng, C. Meyer, E., and Lee, J. (2005). "Video-capture-based approach to extract multiple vehicular trajectory data for traffic modeling." *Journal of Transportation Engineering*, 131(7), 496-505.

- Wilson, R. E. (2001). "An analysis of Gipps' car-following model of highway traffic." *Journal on Applied Mathematics*, 66(5), 509-537.
- Winston, W. L. (1993). *Operations research applications and algorithms*, Duxbury, Belmont, CA.
- Zhang, H. M., and Kim, T. (2001). "A car-following theory for multiphase vehicular traffic flow." *Proceedings of the 80th annual meeting*, Transportation Research Board, Washington, DC.
- Zhang, H. M., and Kwon, E. (1997). "Travel time estimation on urban arterials using loop detector data." *Research Report*, University of Iowa, Iowa City, IA.