DYNAMIC ESTIMATION OF TRAVEL TIME ON ARTERIAL ROADS BY USING AUTOMATIC VEHICLE LOCATION (AVL) BUS AS A VEHICLE PROBE

by

Sanghoon Bae

Dissertation submitted to the Faculty of the
Virginia Polytechnique Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
IN
CIVIL ENGINEERING

APPROVED:

A.G. Hobeika, Chairman

H. VanLandingham

A.A. Trani

I.L. Al-Qadi

R. Sivanandan

May, 1995

Blacksburg, Virginia

Dynamic Estimation of Travel Time on Arterial Roads by Using Automatic Vehicle Location (AVL) Bus as a Vehicle Probe

by

Sanghoon Bae
Antoine G. Hobeika, Chairman

Civil Engineering

(ABSTRACT)

A method of reducing congestion is to reduce the demand by providing the road network users with accurate and reliable travel time information for their pretrip planning and enroute guidance, and/or attracting more people to the public transit. For this purpose, this research concenturates on using Automatic Vehicle Location (AVL) system-equipped bus as a probe vehicle for estimating bus arrival times and auto travel times. Since many transit organizations throughout the North America are currently operating these AVL buses on their bus routes, in a sense, existing AVL bus would be the most cost-effective traffic probe which can be utilized for data collection in the proactive mode.

Therefore, the goals of this research are to enhance the current use of AVL systems by introducing a new module to estimate bus arrival time information for transit travelers, and use AVL systems-equipped bus as a probe vehicle to estimate the nontransit travel time for auto travelers.

The initial objective for the first goal was to model and simulate the dynamic bus behaviors at a single and multiple bus stops using time varying passenger arrival rate and

passenger boarding rate. Then, a prototype arrival time estimation model was simulated by adopting the Parameter Adaptation Algorithm (real-time identification). Later in the research, a dynamic link travel time function was developed by dividing the conventional arterial link into two regions. An integrated travel time estimation model, which combines the prototype arrival time estimation model and dynamic link travel time function, was developed and validated. Three stops-ahead link travel time estimation, dynamic link flow estimation and on-line parameter estimation were the main tasks conducted in the development of integrated travel time estimation model.

To fulfill the second goal, regression analysis was conducted to identify the correlation between the bus travel time and the auto travel time. Another modeling technique, Artificial Neural Network (ANN), was applied in order to interpret auto travel time directly from the bus travel time. A multilayer perceptron, adopting a supervised backpropagation learning algorithm, was constructed to map the bus travel time to auto travel time. Dynamic and static data which affect the travel time of bus and auto were collected by field observations from Blacksburg and Norfolk, Virginia and used to validate the auto travel time prediction model. ANN results outperformed the regression analysis, and it has high potential to be incorporated in a Traffic Management Center (TMC) to generate the estimated travel time information for both transit and nontransit travelers. Also, recommendations for further research were discussed.

ACKNOWLEDGMENT

No individual can accomplish his or her goals without any assistance from other people. I certainly have many people who helped me toward a better understanding of transportation engineering. First, I would like to thank my advisor, Dr. A.G. Hobeika, for giving me the opportunity to do this study and for his support. Acknowledgment is also given to the other members of my graduate committee for their help and advice.

I would like to express my special thanks to Dr. P. Kachroo, who introduced me to an entirely new world of research area in the application of engineering, and his warm advice, efforts and talent should not be forgotten.

Many people contributed in numerous ways toward the successful completion of this study and to whom I extend my sincere thanks, especially, those people who participated in time-consuming data collection.

Last but not least, I wish to dedicate this research to my parents, who supported me throughout my study at Virginia Tech.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER I INTRODUCTION

1.1 BACKGROUND

The term *intelligence* is not only a popular word in the area of transportation but in all areas of engineering and science. This trend in transportation is believed to be caused by utilizing the current sources of infrastructures and technologies to battle with, and then eventually overcome the chronic problems that we have in late 20th century. The goals of the Intelligent Transportation Systems (ITS) are to relieve congestion, improve safety, save energy, and solve environmental problems in surface transportation systems as well as all other modes of transportation systems - marine and air. Among these problem domains, *congestion* has the first priority that must be coped with an urban transportation system. Congestion, specifically nonrecurring congestion (i.e., accidents, disabled vehicles, signal malfunction, temporary maintenance, spill loads), causes up to 60 percent of freeway delay in the United States (Lindley, 1987). Due to the complicated nature of the surface street, nonrecurrent congestion causes more total delay time on arterial roads than on the freeway.

A method of reducing congestion is reducing the demand through providing the road network users with accurate and reliable travel time information for their pretrip planning and dynamic route guidance, and/or attracting more people to public transit. At this point, ITS engineers and planner started paying more attention to the quality of information. Previous surveys revealed that if alternate route information for detouring a congested link was provided for motorists, the majority of them would take the suggested alternate route (Sherazi et al., 1986). From the survey, we also learned the need for better and more accurate information, so that motorists would trust and accept the information that is provided for altering their travel behaviors. A more recent study - *User Route Choice Behavior under ATIS* - conducted by Yang et al. (1993), also emphasized the importance of accurate information. If drivers are given poor (inaccurate) information, they are unlikely to follow the system advice in immediately subsequent trips. With these problems in mind, this research was initiated and focused on attracting more people to public transit via forecasting reliable travel time (arrival time) to the riders at downstream bus stops.

In addition to this fact, the applications of travel time variable in transportation are versatile. Accurately predicted short-term travel time is a good Measure of Effectiveness (MOE) for the proactive mode, for anticipating problems and immediately implementing corrective control strategies in ATMS, for route guidance in ATIS, and for high quality of user services in APTS (Kirby et al., 1994). The various application areas of travel time are shown on Figure 1.1.1. The essential factors which will be considered specifically in the travel time prediction model are accuracy, timeliness of processing speed, and adaptability. Thus, the development of a methodology for estimating the accurate delay time and, therefore, the travel time is a critical factor for increasing the level of service for the urban transportation network users.

In collecting real-time travel time data, the choices of detection technologies are broad (Klein et al., 1993). Microwave radar, passive and active infrared, ultrasound, acoustic, video image processing, inductive loop, and probe vehicles have been tested in various locations throughout the U.S. for their applicabilities. Currently, the inductive loop detector, predominantly used in vehicle detection, provides the ability to measure flow, speed, density, travel time, and vehicle turning movements simultaneously. Yet, to meet the requirements of ITS, the most feasible and most effective detection technology should have capability to provide the less minimal force-alarm rate and timeliness to detect. Due to this reason and the specific needs of real-time data for traffic management and information systems, probe vehicle proved to be appropriate to provide useful data, especially without flow interruption during the process of installation and maintenance.

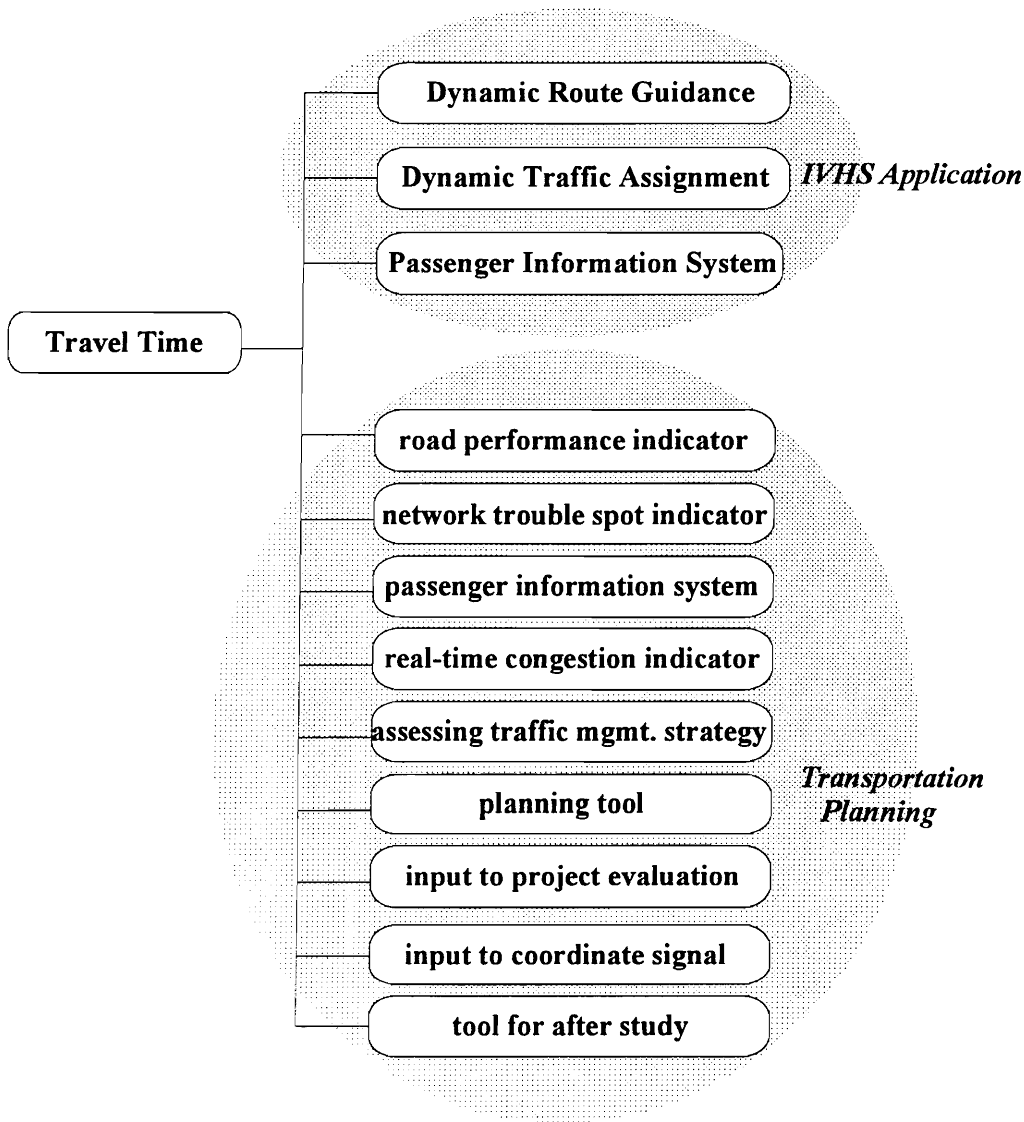


Figure 1.1.1 Travel Time Application Areas

A detailed study of advantages of using the probe vehicles to collect the traffic data was conducted by Westerman et al. (1993). Since the disturbances in the traffic flow due to incidents can only be detected when they have propagated to the location of fixed detector, it causes a time lag in detection and this requires more expenses in order to be a proactive surveillance system. To meet real-time information for ATMS and ATIS, an advanced information processing and network data collection technique have to be developed. In this process, probe vehicles are capable of collecting both real-time location data and road traffic data (speed, travel time, incident) simultaneously. Figure 1.1.2 illustrates proposed and potential application areas of probe vehicle in the context of ITS. Also, the supplementary data collection sources for IVHS along with the probe vehicle would be police car, local authority, weather station, air surveillance, and advanced sensors. The probe vehicles utilized to estimate bus arrival time in this study are the Automatic Vehicle Location (AVL) system equipped buses. Since many transit organizations throughout the United States and Canada are currently operating the AVL buses on their various routes, in a sense, existing AVL buses would be the most cost-effective traffic probe to be utilized for data collection in the proactive mode. Additionally, bus routes cover almost all the urban networks on a regular time basis and throughout the day. Some of the AVL systems are already operational in full scale, others are in the stage of upgrade or reinstallation (Schweiger et al., 1994). Among these, AVL system of MTA in Baltimore, Maryland, and RDT in Denver, Colorado are two potential candidates for AVL data source.

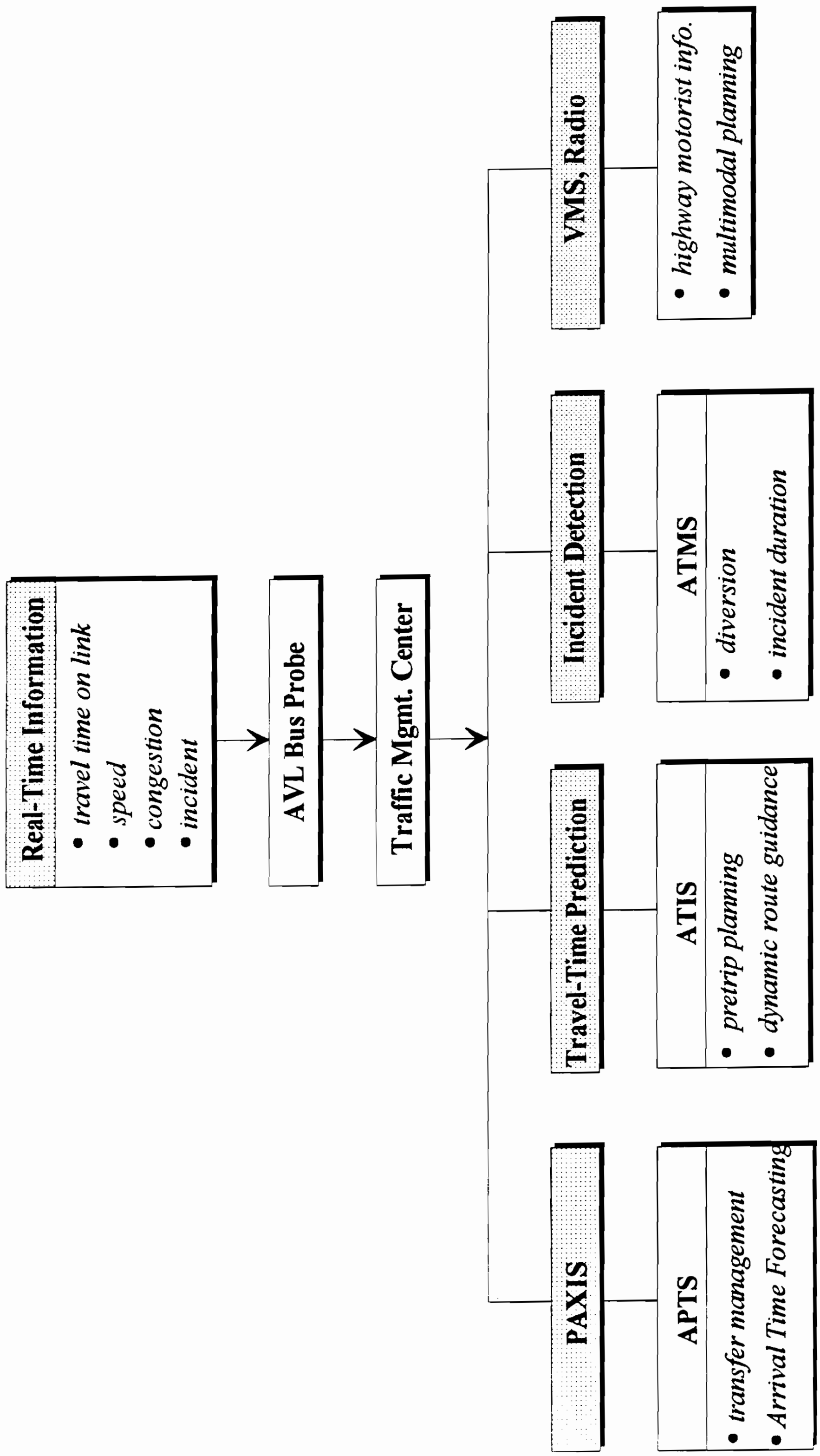


Figure 1.1.2 Proposed and Potential Application areas of Probe Vehicle

In making pre-trip planning and during the en-route decision-making process, accurate real-time travel time data that describes the status of the network is a key factor. To this extent, the prediction model should be formulated as accurately as possible. Also, the rational reasoning behind the prediction model in Intelligent Transportation Systems (ITS) application is well-identified by Smith and Demetsky (1994). They suggested that the final output for the users in ITS would be the smart highway because dynamic control of traffic is a key issue in ITS. In order to achieve the dynamic control of traffic, making and updating predictions of traffic flow and link travel time for the short term horizon will be the critical issues. Therefore, the success of ITS relies heavily on the accurate traffic forecasting model development. Another subject to be studied in this research will be Artificial Intelligence (AI).

The area of AI has been credited with a large portion of the technological leap in transportation operation, control, and planning, especially in the field of ITS, which adopts advanced AI techniques, such as Neural Networks, Fuzzy Set Theory, and Expert Systems. Artificial Neural Networks (ANN) was introduced in transportation engineering in the mid 1980's, and its application areas cover a variety of problem domains - prediction, pattern recognition, control, classification, etc. Conventional mathematical models to deal with very complicated problems, such as arterial travel time, travel time in weaving section, etc. have a great deal of difficulties in empirical model formulation, calibration, and validation due to the fact that some of factors are unknown. ANN is characterized to handle all of these problems easily.

1.2 PROBLEM STATEMENT

The utilization of mass transportation in the United States has been decreasing year by year, especially in nonwork-related trips. Social behaviors against public transit, level of service, and on-time performance are the main issues causing this chronic downward trend in the public transit. In order to cope with this problem, the concept of Advanced Public Transportation System (APTS) has been introduced as a component of Intelligent Transportation Systems (ITS). In this research, the latter two issues were dealt by introducing the mathematical and Artificial Neural Networks models for forecasting accurate and reliable arrival time, and interpreting auto travel time from the bus travel time.

Maintaining on-time schedule is important for customer satisfaction and for efficient management and control in a mass transit operation. In this manner, providing the riders with reliable real-time schedule status (arrival time) is critical for the credibility of the transit organization. Current conventional arrival time forecasting modules or schedule status checking modules in most of the Automatic Vehicle Location systems are based solely on the static methodology. There are numbers of reference locations which are associated with the static time schedule (average historical travel time) throughout the route. Since the real-time location of a bus is a known factor through the use of an AVL probe bus, scheduled time and bus location are compared to identify the performance status of each bus through a simple calculation of average speed and distance to travel on that link without consideration of any mathematical algorithmic approaches.

However, the condition of links on arterial roads changes rapidly and unpredictably due to nonrecurrent congestion, such as a temporary lane blockage caused by a delivery truck, road maintenance, or an accident. Time headway between buses changes unpredictably so that on-schedule performance can be disturbed. Therefore, without knowing the dynamic behaviors of arterial roads on a real-time basis, forecasting, especially short-term forecasting of travel time for riders at downstream stops, dynamic route guidance, and pre-trip planning would be unrealistic. Along with the AVL probe bus data, real-time based volume data on critical links of arterial road, if available, can be another good indicator for describing the situation of the link.

As stated, the behavior of arterial road is so complicated that obtaining the real-time based data from the arterial roads is essential to interpret the real situation into a physical sense. Currently, on-line data collection efforts on arterial roads are emerging so that it will be feasible in the very near future. However, as far as short-term (a minute or 5 minute) real-time link volume data is concerned, it is not easy to obtain the data as of today. Thus, in this study, instead of utilizing on-line volume data from the arterial links, historical data (15 minute volume) will be utilized to test the travel time forecasting model.

Two approaches will be investigated for this study. First, mathematical models for travel time estimation on arterial roads will be formulated by adopting parameter estimation algorithms. In this model, several steps of modeling efforts will be investigated. In order to test and validate the mathematical model, simulation will be conducted by using real world data. On the other hand, for the conversion of car travel time from the bus travel time, feed forward multilayered neural networks utilizing a backpropagation learning algorithm will be applied. Detailed descriptions on the evolution of the development of the mathematical travel time forecasting model and a neural networks model will be made in the following chapters.

1.3 GOALS, OBJECTIVES, AND SCOPE

The first goal of this research is to enhance the current usage of AVL systems. This goal will be accomplished by constructing an advanced mechanism for estimating the travel time of the bus by utilizing the data from AVL-equipped probe vehicles. Hence, the outcome of the first goal will provide both accurate and reliable forecasting of bus arrival time information to the regular and potential bus transit users. With the successful accomplishment of the first goal, obtaining normal traffic (nontransit) travel time from the probe bus travel time on the arterial road is the final goal of this research. By considering the conversion factors of lane usage of buses and dynamic characteristics of bus behaviors, and by excluding passenger loading times, a platform can be established to interpret the normal traffic travel time from the probe bus travel time information. In order to meet these goals, several objectives were identified.

Several objectives to be accomplished in this research are as follows:

- Studying the dynamics of bus behavior at a single stop
- Extending the dynamics of bus behavior to multiple stops
- Developing a prototype bus arrival time prediction model
- Developing a dynamic link travel time model considering signalized intersection
- Developing an integrated bus travel time estimation model
- Developing a model for converting normal traffic travel time from bus travel time

The initial study of dynamic behavior of bus performances at a single stop was simulated according to the constant and time varying loading time and passenger arrival rate. A number of passengers arriving at a certain stop was the main variable focused in this simulation. Then, dynamics of bus behaviors at multiple stops were simulated based on the ratio between passenger arrival rate and passenger boarding rate. The main variable focused in this simulation was the departure time headway (loading time variation). For the third objective, prototype bus arrival time prediction model based on parameter estimation algorithm was developed. In this model, least square estimator with forgetting factors was adopted, and tested. Two simulation scenarios, constant and time varying actual parameters at each stops, were tested in order to identify the parameter update capability. The estimation model was then analyzed according to parameter errors and estimation errors. Dynamic link travel time model was developed for the forth objective. Conventional link was divided into two (congested and uncongested) regions, and validated in the integrated travel time estimation model by adopting real data. In the integrated travel time estimation model, all the previously developed models were combined, and tested in the simulation. Parameter convergences and estimation errors were the main outputs in the evaluation of the integrated model. For the last objective, Artificial Neural Networks was adopted for mapping the bus travel time to auto travel time. Multilayer perceptrons utilizing backpropagation learning algorithm was used in this phase. Sum of Square Errors were used for testing mapping accuracy. Figure 1.3.1 illustrates the evolution of model development.

The data and information flow of developing model is illustrated in Figure 1.3.2. Real-time data from probe vehicle will be the input to the forecasting models. Then, the processed data will be conveyed to the riders at bus stops, and non-transit travelers.

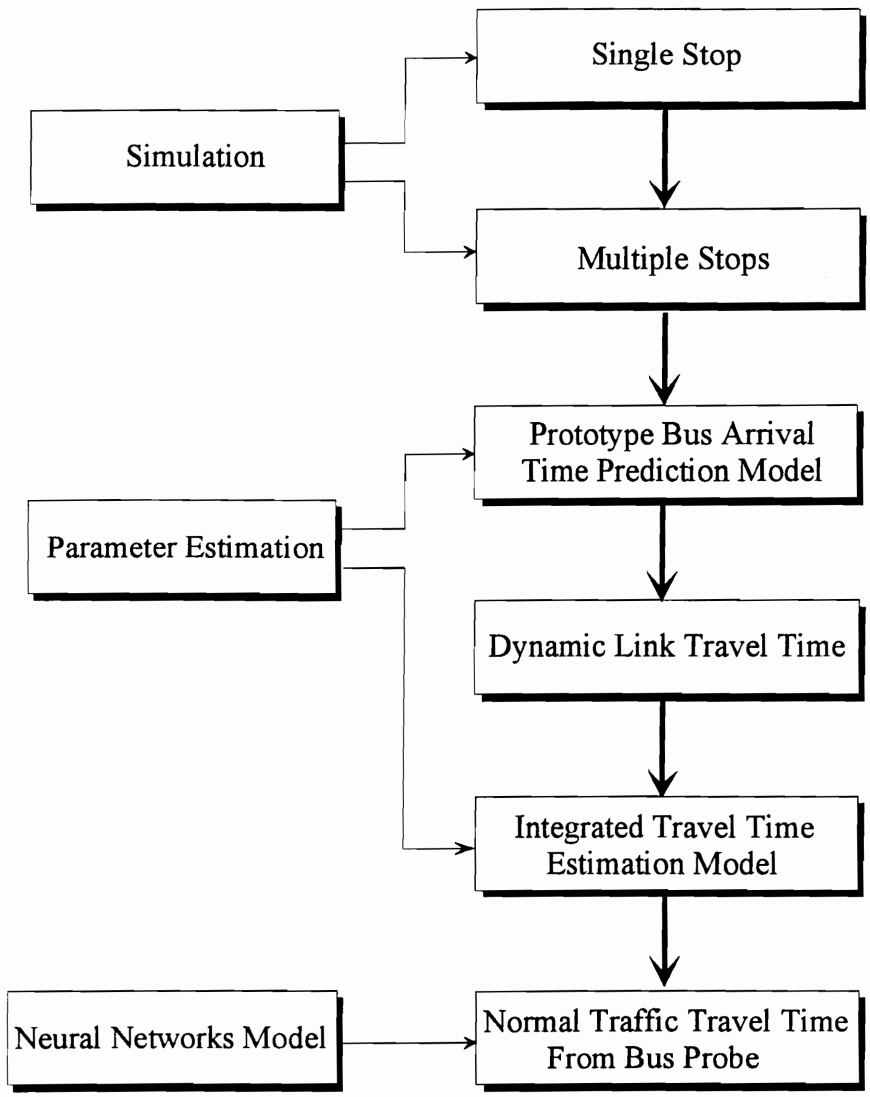


Figure 1.3.1 Evolution of Model Development

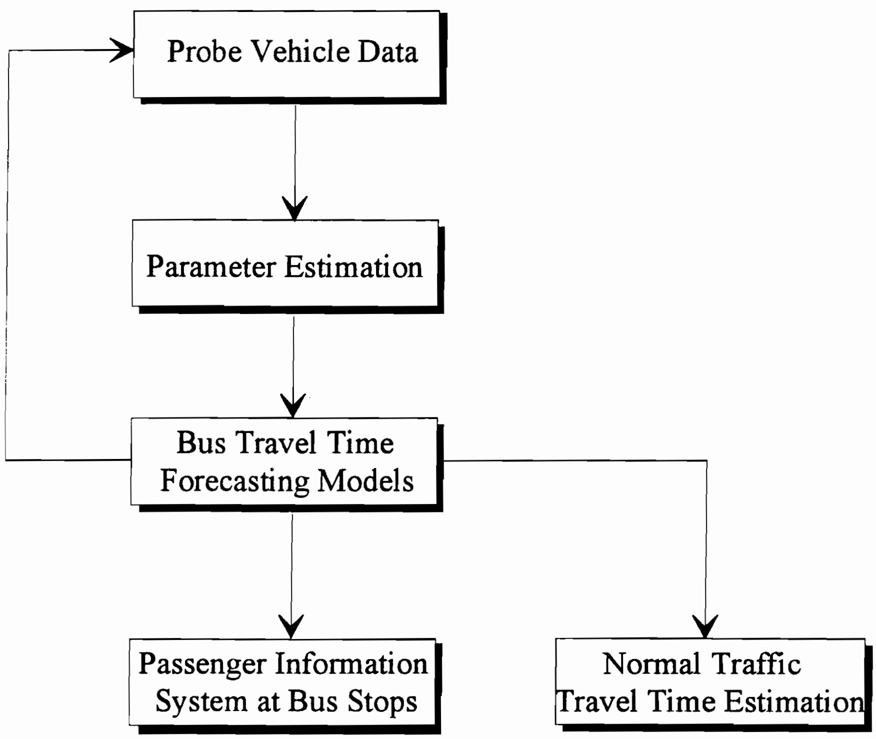


Figure 1.3.2 Data and Information Flow of Model Development

Forecasted arrival time will be updated continuously based on the most recent arrival time of the bus at the same stop.

1.4 OUTLINE OF REMAINING CHAPTERS

Introduction, including background, problem statement, objectives and scope of the research were discussed in Chapter 1.

In Chapter 2, extensive literature review of link travel time function on interrupted flow conditions, fundamentals of Artificial Neural Networks (ANN), and applications of ANN in the field of transportation and ITS are introduced. World-wide research efforts on travel time are categorized into nine approaches, and ten of the current ANN applications in transportation engineering are reviewed in this chapter.

Various analytical modeling aspects of the bus travel time are simulated and estimated in Chapter 3. Dynamics of bus behaviors at single bus stop, and multiple bus stops are discussed in the early part of this chapter. At the end of this chapter, prototype of bus arrival time prediction model is formulated and tested in the simulation.

Dynamic link travel estimation model was developed in Chapter 4. This model is formulated based on the division of conventional link into two regions, i.e., congested and uncongested regions. Travel time of these two regions are formulated separately, and then combined together for the link travel time.

Integrated travel time estimation model, combining prototype arrival time model and dynamic link travel time model, is proposed and evaluated in the simulation. Linear parameterization of dynamic link travel time model, integrated travel time estimation model, three stop-ahead bus arrival time estimation, dynamic flow estimation, and parameter estimation issues are the main topics in this chapter, Chapter 5.

In Chapter 6, a statistical approach (regression) is applied to identify the correlation between transit and non-transit travel time. Along with the regression, Artificial Neural Networks to convert the non-transit travel time from the bus travel time is also discussed. Input identification, network training, and testing are the main issues in the application of Artificial Neural Networks.

In the last chapter of the dissertation, Chapter 7, conclusions of the research, and recommendations for the future research are discussed.

CHAPTER II LITERATURE REVIEW

2.1 INTRODUCTION

Travel time information is one of the most important measure of effectiveness (MOE) to interpret the traffic condition and volume of traffic on physical network. It provides an important input to analyze flow on transportation networks for short- and long-term planning model.

Evolution of travel time functions on *uninterrupted flow* (freeway) were researched by Rose et al. (1989). They identified the definition of the uninterrupted flow as "vehicle traversing a roadway are not impeded by any causes external to the traffic stream, such as signs and signals, however, vehicles may be stopped by causes internal to the traffic stream." The initial travel time function was suggested by Beurou of Public Road (BPR, 1964) and named as BPR function.

$$T = t_0\left[1+0.15\left(\frac{v}{c}\right)^4\right]$$

where, t_0 = free flow travel time

Currently, most of the travel time estimation is dwell on static BPR travel time function. However, BPR function generates too approximate results to be realistic in a real-time sense and its function is focused on uninterrupted flow (freeway) so that it cannot predict delay time on arterial (interrupted flow) properly. Davidson (1966) characterized the functional form of travel time function which satisfy the requirements of traffic assignment model. He proposed by using three parameters which are free flow travel time (t_0), the link capacity (C), and a parameter (J) which determines the degree of curvature of the function. The equation would be

$$T = t_0(1+\frac{Jv}{c-v})$$

where, T = link travel time

v = volume.

In order to overcome the asymptotic nature of the Davidson's function, Akcelik (1978) proposed a modified function which taking a linear extension to the function at a volume close to capacity. In this way the function is defined for all volumes and therefore causes no computational problems. Recently, Taylor (1988) provided a rigorous derivation and application of a non-linear least squares procedure for estimating the parameters of the Davidson's function. Rigorous estimation of the travel time function provides an opportunity for testing the specifications or formulation of the function. Whereas detail study on *interrupted flow* (arterial) will be reviewed at following sections.

Meanwhile, Rose et al. also proposed a family of travel time functions which is developed on the basis of a network classification scheme (area, corridor, route, link) that incorporates increasing detail as the network unit decreases in size. Coupled with the network dimension is a functional classification for route and links that includes more details at the finer levels of network disaggregation. Rose et al. also stated that the costs and benefits increases as the travel time functions are estimated at finer levels of network detail. Brief description on the network classification scheme (area, corridor, route, link) is as follows:

● area level travel time ... At this level the data would focus on the general propensity for movement within the spatial unit. Average unit travel time and traffic intensity are the variables which enter the travel time function. The application areas of this level is suitable for broad-brush policy type studies that require a basic differentiation of the traffic carrying capacity of arterial roads on a broad zonal basis. Study area could be broken up into a number of sub-areas based on an appraisal of differences in basic land use and development pattern, and network connectivity between parts of the metropolitan area. Separate travel time functions would be developed for each sub-area.

● corridor level travel time ... Direction of flow is identified to account for more of the variation in flow characteristics that may be identified between roads. The network at this level is not homogenous. Differences are likely to emerge at this level (flow in east-west and north-south) in terms of different signalized intersection densities and signal priorities.

● route level travel time ... This level consists of sequence of connected links and a primary component of a corridor. A three dimensional classification system would appear useful with the zone and direction-of-flow dimensions supplemented by a function type classification. It would be possible to classify the roads according to posted speed limits, i.e., freeway, 100 km/hr arterial, etc.

● link level travel time ... The three dimensional classification scheme may be continued at the level of the network link, with further disaggregation of the facility type dimensions. The richer classification of road types are the main difference between this and route travel time function. Even finer level of analysis could focus on individual link elements. The link element approach assumes that the overall travel time on the link is the sum of the travel time on the link elements that form the link. However, this level of network representation has not been used in practice.

Travel time information would be applied for tremendous areas both in IVHS and transportation planning. Within the context of IVHS, dynamic travel time information (travel cost) can be a key factor for Dynamic Route Guidance (DRG), which would be a key project area in ATIS, and it must be calibrated in an efficient way for all the links of the network. The requirement of travel time prediction model would be the capacity to represent the time dependent behavior of the network, including disruption on flow due to recurrent and non-recurrent congestion. Travel time function would be also a key factor in time dependent shortest path estimation for Dynamic Traffic Assignment (DTA). For the purpose of APTS reliable and accurate information on bus arrival time for regular and potential passengers are essential for pre-trip planning and multimodal system. Another application area for travel time in bus operation will be the minimization of the schedule deviation (headway adjustment on high frequency routes, redistribution, reassignment, catch up pauses), rerouting, real-time arrival time information for passengers, etc.

Some of the examples of applications areas for transportation planning are as follows:
1) link travel time is an essential parameter required to pin-point trouble spots in the network (Longfoot, 1991); 2) it can be utilized as an indicator of the overall road system performance; 3) it is a good indicator for a real-time measure of congestion; 4) it is an appropriate means for assessing traffic management strategy; 5) it is good for a planning tool; 6) it is an input to project evaluation; 7) it is an input to coordinate signal control system; 8) it is a good measurement for assigning the effectiveness of signal change; 9) it is a good measure of effectiveness (MOE) for readily quantifiable improvement for after

study; 10) it is a good means for assessing traffic management strategies as a it is a planning tool; and 11) it can be an input to project evaluation.

Longfoot (1991) suggested the system criteria for travel time measurement. These are: 1) sample size should be statistically significant in order not to be bias; 2) vehicle measured should be representative of the population; 3) low cost for purchase, installation, and operation of the system.

Travel Time Data Collection Methodologies which would be applied for real-time detection and control for IVHS are as follows:

1) *floating car*: It measures the travel time while volume data may be collected from the traffic control system during the specified travel time. This method is primitive and requires the man-power;

2) *license plate matching*: It can be used to measure travel time in conjunction with the use of standard traffic volume counter. The number plate data have to be collected on a sample basis and an estimation of the total population size is required. This method is costly to conduct and analyzed. This method could be structured to provide individual vehicle data which could be used to construct individual vehicle time profile;

3) *Automatic Vehicle Identification (AVI)*: AVI is the subset of Automatic Vehicle Monitoring system. Inductive loop AVI, radio communication systems, microwave AVI,

infrared AVI, and optical AVI are reviewed by Bell and Cowell (1988). For the suitability of bus monitoring system inductive loop and microwave system were concluded to be the most appropriate, however, infrared/roadside communication were appropriate for the DRG purpose. The feature, advantages, and drawbacks of some of the detection methodologies are summarized in Table 2.1.1;

4) *loop detector output*: It can provide volume, occupancy, and arrival patterns of vehicle on surface. Travel time can be inferred from these data. It can provide the historical data profiles and supplement probe vehicle reports with on-line information on traffic condition. Also, it provides a sound alternative data source to be used in the absence of probe vehicle data;

5) *cellular phone*: This method utilizes the preestablished reference locations which can be utilized for motoring the travel time that takes for travelers to journey in between the reference points. Then this information is transferred to the control center via cellular phone. Travel time can be directly obtained from the probe vehicle which is equipped with the cellular phone. The important issues to be resolved in this methodology are the number and spacing of reporting locations, and the number of probe vehicles for certain network. Also, probe vehicle should be uniformly distributed throughout the time and network. Otherwise it would provide insufficient data for the network;

6) *input-output data survey*: The number of vehicle arrivals in successive time interval is recorded at up stream and down stream stations in the test stations and these data are used to obtain an estimate of the mean travel time.

Table 2.1.1 Features, Advantages, and Disadvantages of the AVI

	FEATURES	*ADVANTAGES*	*DISADVANTAGES*
Inductive loop AVI	• *use radio frequency (RF)* • *less than 1MHz freq.* • *applied in SCOOT*	• *cheap and robust* • *vandal-proof*	• *low information transfer rate* • *inconvenient to install*
radio communication system	• *use radio frequency (RF)* • *higher freq. up to 150 Mhz* • *Germany, BUSCO in London, BON system in Hanover*	• *cable free* • *short range or line of sight transmission* • *cheap and easy to set up*	• *hard to get broadcast license* • *data corruption*
microwave AVI	• *use radio frequency (RF)* • *freq. greater than 1 GHz*	• *larger spectrum of freq. is available* • *higher transmission rate than inductive loop AVI* • *low energy consumption rate*	• *expensive than radio transmitters*
infrared AVI	• *an optical system* • *shorter radiation wavelength than microwave AVI*	• *similar to microwave AVI* • *high data transmission rate*	• *requirement for greater accuracy in aligning transmitter and receiver* • *decreasing signal strength caused by dirt or snow*
optical AVI	• *requires close contact between vehicle and detector*	• *variety of commercial usage*	• *impractical in the harsh and uncontrolled environment* • *sensitive to the fault on bar code of vehicle* • *same drawbacks as infrared systems* • *high error rate (5%)*

2.2 TRAVEL TIME FUNCTIONS ON INTERRUPTED FLOW

Previous studies on the main topic of the dissertation are reviewed in this section. The most recent and various algorithms from several countries are categorized in this section. One of the common deficiencies imbedded in these approaches is that current algorithms are limited to link specific so that it can not be applied directly either to section-specific or route-specific transportation network environment. In addition, majority of the current algorithms still require more concrete validations.

Some statistical approaches (probabilistic model, ARIMA, cross-correlation technique with inductance loop detector, recursive identification algorithm), detector output, straight line algorithm, manual (floating car, license plate matching), AVI, probe vehicle, other theoretical approaches (sandglass, delay time method) are the major methods of formulating the algorithms. Following section describes the world-wide efforts on the various algorithm developments for travel time function. Some of the algorithms to be reviewed in this section are as follows: link specific algorithm, strait line algorithm, cross-classification technique with inductance loop data, probabilistic model in the *Unites States*, *Canadian* approach, sandglass algorithm in *Japan*, autocorrelation algorithm in *England*, flow-density relationship from detector output in *Netherlands*, Automatic Network Travel Time System in *Australia*.

2.2.1 United States

2.2.1.1 Link Specific Approach (Ran et al. 1992)

The main theme of the approach is focused on time dependent travel time function for signalized network links. They define the link travel time as the travel time of a vehicle from existing the upstream intersection to existing downstream intersection. Hence, link travel time would be the summation of the cruise time (uncongested travel time), delay time (congested travel time), and the travel time through the intersection. The formulation of the model starts from the state equation.

$$\frac{dx_a(t)}{dt} = u_a(t) - v_a(t) \qquad \forall a.$$

where, x_a(t) = the number of vehicles traveling on link a at time t

u_a(t) = the inflow rate into link a at time t (veh/hr)

v_a(t) = the exit flow rate from link a at time t (veh/hr)

The number of vehicles on link *a* at an initial time t=0 is assumed to be equal to zero, i.e., $x_a(t)=0$ for $\forall a$.

$$x_a(t) = \int_0^t [u_a(w) - v_a(w)]dw \quad \forall a.$$

where, $x_a(t) \geq 0$, $u_a(t) \geq 0$, and $v_a(t) \geq 0$.

Instantaneous travel time $C_a(t)$ would be defined as the travel time that would be experienced by each vehicle traversing link a if prevailing traffic conditions do not change. Thus, $C_a(t)$ would be defined as

$$C_a(t) = D_{a1}[x_a(t), u_a(t)] + D_{a2}[x_a(t), v_a(t)]$$

where, $D_{a1}[x_a(t), u_a(t)]$ = instantaneous cruise time

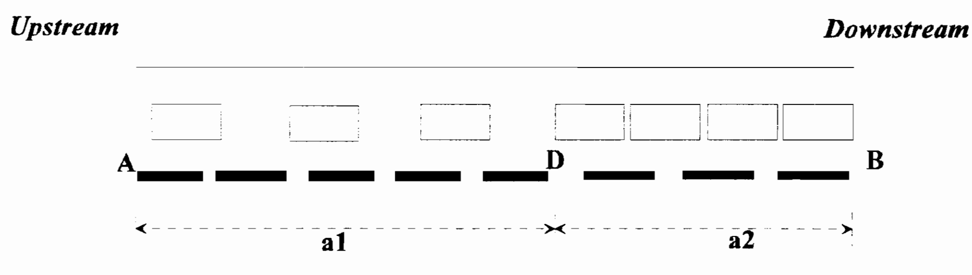
$D_{a2}[x_a(t), v_a(t)]$ = instantaneous queuing delay.

Under the assumption that stable traffic condition would remain until vehicle reaches its destination, route travel time can be estimated by getting the summation of individual links. However, network traffic conditions are changing over time, the actual route travel time may be significantly different from the instantaneous route travel time. When the route or travel time is long and severely oversaturated, the difference will be large. Because of this reason Ran's model was strictly based on instantaneous link travel time. Instantaneous link travel time is also much easier to obtain and estimate than the actual route travel time since the prevailing traffic flow data can be obtained from a probe vehicle or a roadside detector.

They classified link travel time prediction models in two categories, long term and short term, respectively. For long term (5 to 30 minutes) forecasting flow and capacity information on link is required. In this category model estimates average queue length, and average delay per vehicle over a complete traffic peak. Stochastic delay constitutes a significant part of the total delay in long term forecasting. They used six link flow variables to build the link travel time function. They suggested that the travel time function in this category is suitable to be used for off-line evaluation of ATMS and ATIS. On the other hand, for short time scale forecasting stochastic delay would be negligible and deterministic oversaturation delay must be considered. This type of forecasting is appropriate for real-time, on-line ATMS and ATIS.

2.2.1.1.1 Link Travel Time Function for Longer Time Horizon

They divide the physical length of the link a=(A,B) into two dummy links: link a1=(A,D) and a2=(D, B). The location of D is flexible. Further, they assumed that link a1 contained uncongested vehicle stream and link a2 is filled with a vehicle queue.



The state equation for the two dummy links are:

$$\frac{dx_{a1}(t)}{dt} = u_{a1}(t) - v_{a1}(t) \qquad \forall a_1 \in a; a;$$

$$\frac{dx_{a2}(t)}{dt} = u_{a2}(t) - v_{a2}(t) \qquad \forall a_2 \in a; a.$$

$v_{a1}(t) = u_{a2}(t)$ $\forall a_1, a2 \in a; a$ for flow conservation.

Denote $\Delta k = [k, k+1]$ as the length of the time interval in hours. Hence the discrete-time form of equation would be

$$x_{a1}(k+1) = x_{a1}(k) + u_{a1}(k)\Delta k - v_{a1}(k)\Delta k \qquad \forall a_1 \in a; a; k;$$

$$x_{a2}(k+1) = x_{a2}(k) + u_{a2}(k)\Delta k - v_{a2}(k)\Delta k \qquad \forall a_2 \in a; a; k.$$

$$v_{a1}(k) = u_{a2}(k) \qquad \forall a_1, a_2 \in a; a; k.$$

Hence, the average link travel time $\tau_a(k)$ (in second) per vehicle during time interval k can be expressed as the sum of two components.

$$\tau_a(k) = D_{a1}(k) + D_{a2}(k) \qquad \forall \in a.$$

where, $Da1(k)$ = a flow-dependent cruising time over the first part of link a

$Da2(k)$ = a queuing delay.

$Da1(k)$ would be defined as

$$D_{a1}(k) = 3600 \frac{l_a - x_{a2}(k) / e_{am}}{w_{a1}(k)}$$

Here, *la* as the length of link *a* (miles) and the derivation of the formula is based on Greenshield's (macroscopic) model. e_{am} is defined as jam density, thus, the notation $l_a - x_{a2}(k) / e_{am}$ denotes the actual length of cruise from inflow until reaching the queue. From the flow-density-speed relationship, cruise speed w_{a1} can be expressed as:

$$w_{a1}(k) = \frac{w_{a0}}{2e_{am}} \left\{ e_{am} + \sqrt{[e_{am}]^2 + 4u_{a1}(k)\frac{e_{am}}{w_{a0}}} \right\}$$

$$\text{where, } u_{a1}(k) = e_{a1}(k)w_{a1}(k) = e_{am}w_{a1}(k) - \frac{e_{am}}{w_{a0}}[w_{a1}(k)]^2$$

There are three link elements: *entry, midblock road section, exit* of the link. Since, both capacities at entry and midblock are usually higher than at the exit point, they refer only to the exit capacity when considering the flow capacity of a link. Capacity calculation methods also depend on the type of junction such as *signalized intersection, major/minor priority,* and *roundabouts.* However, Ran et al. deal with signals only.

The mathematical models used to estimate intersection delay are *queuing models*. Steady state queuing theory has been used, but it predicts infinite queues and delays when the demand reaches the capacity available to it. Deterministic queuing has been applied when demand and capacity vary in time. However, deterministic queuing ignores the random nature of traffic arrivals and departures within a rather long time interval, and leads to serious underestimates in the delay unless the capacity is exceeded by a considerable margin. Also, when the demand just reaches capacity, zero delays are predicted by the deterministic model. Therefore, they deal solely with queues and the corresponding approach delays instead of covering the entire range of demand and capacity.

The average delay per vehicle, D_{a2} *(k),* for vehicles arriving at the downstream intersection of link *a* during time interval *k* are expressed as the sum of two delay terms:

$$D_{a2}(k) = d_{a1}(k) + d_{a2}(k) \qquad \forall \in a$$

where, d_{a1} = non-random delay, which caused by signal cycle effects calculated assuming non-random arrivals at the average inflow rate, and d_{a2} (k) is the overflow delay including effects of random arrivals as well as any oversaturation delays experienced by vehicles arriving during the specified flow period. The uniform delay (non-random delay) adopts the Webster's formula.

$$d_{a1}(k) = \frac{0.5c[1 - g(k)/c]^2}{1 - \rho_a(k)g(k)/c}$$

where, the degree of saturation at the exit from link *a* during time interval *k* would be expressed as $\rho_a(k) = \frac{u_{a2}(k)}{\mu_a(k).}$ Here, $\mu_a(k)$ is the capacity at the exit of a link in function of time. Also, *c* denotes the signal cycle time in seconds, and *g(k)* is the effective green time in seconds during time interval *k,* respectively.

Messer introduced converting the above uniform delay formula into a non-uniform arrival term to account for the progression effects. Then, Akcelik and Rouphail presented more general delay for this uniform delay term.

$$d_{a1}(k) = 0.5[c - g(k)] \qquad \text{for } \rho_a(k) \rangle 1.0$$

$$= \frac{0.5c[1 - g(k)/c]^2}{1 - \rho_a(k)g(k)/c} \qquad \text{for } \rho_a(k) \le 1.0$$

However, since a smooth function is necessary for the solution of dynamic traffic assignment problems, above formula would be smoothed for the framework of their research, and this topic is remained as further study.

Akcelik observed that the Highway Capacity Manual (HCM) equation predicted higher delays for oversaturated condition than did the other Australian and Canadian formula. He recommended a general formula for the overflow delay as

$$da2(k) = 900\Delta k[\rho a(k)]^n \left\{ [\rho a(k) - 1] + \sqrt{[\rho a(k) - 1]^2 + \frac{m[\rho a(k) - \rho a0(k)]}{\mu a(k)\Delta k}} \right\}$$

where, $\rho_{a0}(k)$ would be the degree of saturation below which the overflow delay $d_{a2}(k)$ is negligible. This can be expressed as $\rho_{a0}(k) = a + bs(k)g(k)$.

Here, *s(k)* would be the saturation flow rate in vehicles per second during time interval *k*, and *s(k)g(k)* represents the capacity per cycle during time interval *k*. The rest of the notation are the calibration parameters (a, b, m, n).

Ran et al., however, adopt the Burrow's generalized version of Akcelik's model for overflow delay. This model is:

$$da2(k) = 900\Delta k[\rho a(k)]^n \left\{ [\rho a(k) - 1] + \alpha + \sqrt{[\rho a(k) - 1]^2 + \frac{m[\rho a(k) - \beta]}{\mu a(k)\Delta k}} \right\}$$

where, α = additional term used to encompass the more general form above

β = a term related to $\rho_a(k)$ term in Akcelik's model.

The queuing delay caused by initial queue x_{a2} can be expressed as:

$$3600 \frac{x_{a2}(k)}{\mu_a(k)}.$$

Therefore, they suggest the following overflow delay equation for dynamic problem:

$$da2(k) = 3600 \frac{xa2(k)}{\mu a(k)} + 900\Delta k[\rho a(k)]^n \left\{ [\rho a(k) - 1] + \sqrt{[\rho a(k) - 1]^2 + \frac{m\rho a(k)}{\mu a(k)\Delta k}} \right\}$$

2.2.1.1.2 Link Travel Time Function for Short Time Horizon

For short time horizon they assumed that the impact of randomness of traffic arrivals at the signalized intersection is negligible during such a short time period. Hence, they further assumed that there are only deterministic uniform and oversaturation delays in this case, and if the time interval is less than one minute, offset of signal and uniform delay should be considered. The structure of the link travel time formula is similar to the former case:

$$\tau_a(k) = D_{a1}(k) + d_{a1}(k) + d_{a2}(k) \qquad \forall \in a$$

Here, D_{a1} and d_{a1} is the same as the terms in the longer time horizon. However, the average deterministic queuing discharge time or queuing delay $D_{a2}(k)$ per vehicle arriving during time interval [*k*, *k+1*] can be expressed as:

$$d_{a2}(k) = 3600\frac{[u_{a2}(k) - v_{a2}(k)]\Delta k / 2 + x_{a2}(k)}{\mu_a(k)} = 1800\frac{[u_{a2}(k) - v_{a2}(k)]\Delta k + 2x_{a2}(k)}{\mu_a(k)}$$

where, $x_{a2}(k)$ is the initial deterministic queue encountered be the first vehicle arriving at the beginning of time interval [*k*, *k+1*]. The deterministic queue encountered by the last vehicle arriving at the end of time interval [*k*, *k+1*] is $[ua2(k) - va2(k)]\Delta k + xa2(k)$. The average queue for vehicles arriving during time interval [*k*, *k+1*] can be expressed as $[u_{a2}(k) - v_{a2}(k)]\Delta k / 2 + x_{a2}(k)$.

The selection of suitable link travel time functions for discrete-time dynamic traffic assignment largely depends on the length of the analysis time interval. The models that developed by Ran et al. need to be further analysis and validation for on-line usage for IVHS application.

2.2.1.2 Straight Line Algorithm (Sussman et al., 1974)

Sussman et al. (1974) suggested the series of Straight Line models for travel time estimation. They identified that travel time between arbitrary points i and j on highway network varies from observation to observation. Hence, it has been hypothesized that this variation stems from three components: 1) regular condition-dependent variations (e.g., recurrent congestion, day of the week, seasonal variation); 2) irregular condition-dependent variation (e.g., non-recurrent congestion, incident); 3) random variation (e.g., driver behavior, pedestrian, traffic light at different points in its cycle). They differentiate the category 2 and 3 by admitting any travel time variation that are caused by stochastic incidents to such an extent that the delay in travel time is noticeable for an extended period of time are defined as category 2. On the other hand, if the occurrence only affects a single trip and has no effect on immediately following trips through the same location then it would be category 3 (random variation). Six different series of algorithms (straight line, zoned straight line, zoned straight line with linear relation, rotating zones, optimal coordinates, and grid) are discussed.

However, their approach has not considered the dynamic aspect of travel time estimation, in which parameter would be updated in real-time to reflect changing traffic network condition.

2.2.1.2.1 Straight Line Algorithm

This method assumes that the travel time is proportional to the straight line distance between origin and destination. They formulate the equation as

$$t_{pij} = kd_{ij} \qquad \text{(a)}$$

where, t_{pij} = estimate of condition-dependent mean travel time between i and j k = constant of proportionality relating distant and speed

d_{ij} = straight line distance between i and j.

Here, d_{ij} would be estimated by using Cartesian coordination of origin and destination. This algorithm has some of its characteristics: 1) it is computationally efficient; 2) economical computer storage; 3) it handles condition dependent variations merely by varying *k* as circumstances change; 4) it is often inaccurate, especially in densely settled areas where anything approaching straight line trips are out of the question; and 5) large estimation errors may occur in regions where driving speeds vary widely.

2.2.1.2.2 Zoned-Straight Line Algorithm

This method is identical to the former (straight line) algorithm except that the area is divided into n zones. Also, instead of single *k* value, there are n^2 constants of proportionality k_{lm}.

$$t_{pij} = k_{lm}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

where, point i is in zone 1 and point j is in zone m. Better accuracy is achieved since the technique is more capable to take account of local aberrations in conditions.

2.2.1.2.3 Zoned Straight Line with Linear Relation

This method is identical to the former (zoned straight line) algorithm except that the additional constant specific to the zone pair term l and m.

$$t_{pij} = k_{lm}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + k'_{lm}$$

2.2.1.2.4 Rotating Zones

Experiments with the linear relation method of the previous section revealed that the travel time between various points of a pair of zones was very weakly correlated with their geographic distance. Imagine that the zones rotated about their center points, the correlation between the geographic distance and driving distance is made stronger. Hence former values (k_{lm} and k'_{lm}) are suitably adjusted, more accurate estimates could be obtained. The coordinates (x_r, y_r) of a point in the rotated zone can be computed from the relations:

$$x_r = (x - x_0)\cos\phi - (y - y_0)\sin\phi + x_0$$

$$y_r = (x - x_0)\sin\phi - (y - y_0)\cos\phi + y_0$$

where, (x, y) = the original coordinates

(x_0, y_0) = the coordinates of the center of rotation of the zone

ϕ = the angle of rotation.

2.2.1.2.5 Optimal Coordinates

Accuracy is increased by using pseudo-coordinates which provide a better fit to the actual travel times than geographic coordinates. Given a set of n points and the matrix of travel times between them, find a set of coordinates (x_l, y_l) for each point such that the sum of the error squared in a straight line approximation to the travel times is minimized. That is, minimize the objective function Z.

$$Z = \sum_i \sum_j [M(t_{aij}) - d_{ij}]^2$$

where, $M(t_{aij})$ = condition-dependent mean actual travel time.

This is an optimization problem in *2n* variables. Iterative approximation computation is considerable for a moderate number of points.

2.2.1.2.6 Grid Algorithms

A Cartesian grid is superimposed on the geographic area of interest. Then, travel time is predicted as follows:

$$t_{pij} = k_l|x_i - x_j| + k_2|y_i - y_j|.$$

When velocity is direction independent, k_l and k_2 are equal. There are computational advantages of this method over the straight line methods in that simple subtractions suffice. However, predicted travel times are a function of the positioning of the grid. While this positioning is straightforward, less regular areas present certain problems.

They also discussed about the errors involved in travel time estimation algorithms. These errors are as follows: 1) inability to adjust parameters (k, k_{lm}, k'_{lm}) to changing in condition; 2) coarseness in the estimating algorithms; and 3) random errors.

2.2.1.3 Cross-classification technique with incident loop data (Dailey, 1993)

Inductance loop detector placed in parts at short distance measures volume and occupancy, then convert them into speed and travel time. Although the closely spaced loops are capable of measuring the transit time of single vehicles, there are a number of problems associated with methods for estimating speed and travel time from single loop data. Hence, cross-correlation technique is proposed to overcome this problem.

Traffic management system have approximated speed by using the ratio of volume (flow) to occupancy with a correlation factor to convert occupancy to density. However, researchers have determined that the correction factor is at least a function of location and occupancy and perhaps weather. The continuous version of the model is developed first. However, because traffic is not continuous in reality, the methodology for making discrete estimates of the continuous functions from real traffic measurement is presented in his paper. Some of the important aspects of the correlation technique would be: 1) it provides a means to estimate time delay between widely separated single inductance loop sets 2) this estimate is independent of the actual mean value of the volume or occupancy 3) for system that rely on volume and occupancy ratio, this combination can provide a powerful tool for converting volume and occupancy data to speed estimates.

2.2.1.4 Probabilistic model (Fisk, 1991)

Current travel time function in traffic assignment treats travel time as a function of the link flow and it is derived from an empirical base using observed travel time/flow data to determine the appropriate functional form and to estimate the parameter values. One exception is Davidson's model and it visualizes a road segment to consist of an infinite sequence of queues. However, the analogy is only approximate and the model parameter 'J' has no direct physical interpretation.

A probabilistic model describing flow characteristic on a single lane of traffic adopted to produce link travel time and flow relationship appropriate for use on some links in traffic assignment by Fisk. In his approach the model parameters are the mean and variance in free flow travel time, and the minimum time headway between successive vehicle traveling in a platoon.

2.2.2 Canada

Metropolitan Toronto Transportation Information Production System (MTIPS) has been developed in the Ministry of Transportation of Ontario (MTO), Canada (Berinzon 1993). The goal of MTIPS is to collect, organize, process, and disseminate the real-time transit and traffic data. MTIPS would generate accurate and reliable road travel time measurements to be used for a traveler information system called Travelguide. In order to minimize the travel time knowing the time to reach one's destination by different modes for certain time of day are essential to travelers. Their goal is very similar to what this dissertation is proposing. However, in terms of methodology to estimate the travel time for the bus arrival time, more sophisticated approach is proposed in the dissertation. But, it is still worth to review the similar project that is on the way. Therefore, more advanced system would be established.

Link travel time for specific routes, expected bus arrival times at specified bus stops for the next two buses, schedule deviation from normal travel time on specified routes, bus stop close to specified locations can be supported by the proposed system. MTIPS database allows more than one program to access the same data at essentially the same time. The data which can be collected from the probe vehicle would be odometer data, door open status, engine sensor data, and schedule adherence data. The configuration of the overall system is depicted in Figure 2.2.1.

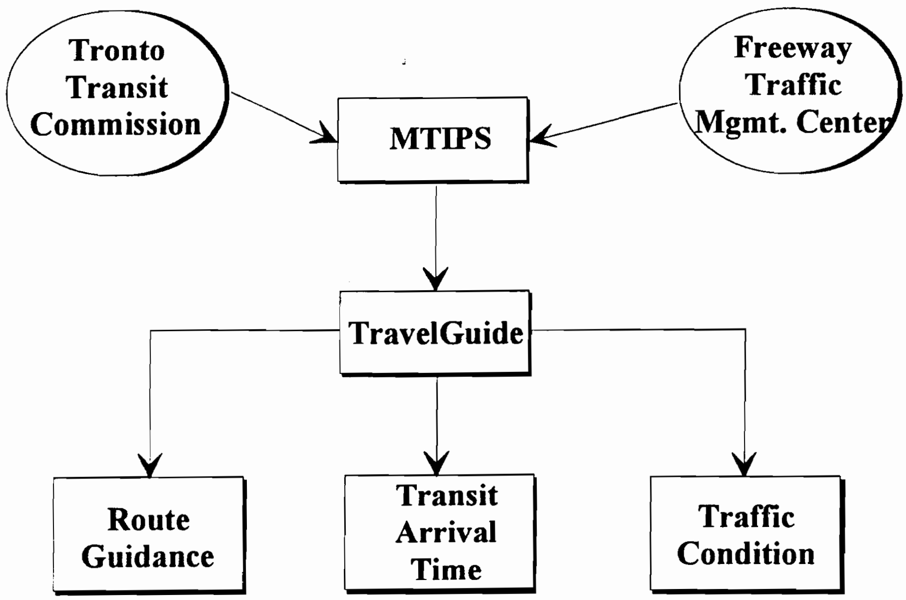


Figure 2.2.1 Configuration of MTIPS

Since the transit data inherently reflect the background traffic condition, any change relative to scheduled travel time might be the result of a change in traffic condition. Hence, with some logical approach normal traffic (auto) information would be calibrated from the probe vehicle.

The algorithms that are employed by the MTO are *cluster analysis, linear filtering*, and *neural network.* Cluster analysis is a statistical method for grouping a set of discrete input data into distinct classes of patterns. Linear filtering would be used for the increasing the accurate result by eliminating the noise. Neural network is to use for pattern recognition problem and useful for nonlinear relationship. Calculation of link travel time would require a modification of Split, Cycle time and Offset Optimization Technique (SCOOT) software. Other traffic information such as incident location and general information are coded using the International Traveler Information Interchange Standard (ITIS).

2.2.3 Japan

2.2.3.1 Sandglass Model

Takaba et al. (1991) suggested the model and vehicle detector information including queue length (vehicle speed) and flow were utilized as the data source for the model formulation. The number of vehicles in the queue (sandglass) and output flow rate at the intersection (down flow at the bottom of glass) are the two main factor in the model. The procedure for the model development was: 1) estimate the queue length; 2) measure the outflow; 3) measure the number of vehicles in queue length; and estimate travel time. The estimation of travel time would follow

$$T_{si} = \frac{Ni}{Qi} + \frac{(Loi - Li)}{Va} \quad \text{(a)}$$

where, N_i = number of vehicles in queue length

Q_i = flow volume at each subsection

L_{oi} = length of subsection i

L_i = queue length through each section

V_a = travel speed in uncongested section.

Here the density k would be

$$k = \frac{N_i}{L_i} = k_m - kQ_i \quad \text{(b)}$$

Therefore, equation (a) can be

$$T_{si} = k_m \frac{Li}{Qi} - kL_i + \frac{(L_{oi} - L_i)}{V_a} \quad \text{(c)}$$

Equation (c) is the namely the Sandglass model and practically utilizing in real applications.

Takaba et al. also proposed the **delay time** model. The travel time in congested section of the link was given by the summation of delay time Di.

$$D_i = d_i \times m_i \tag{d}$$

where, di = delay time per cycle

m_i = number of cycles while vehicles run through subsection i.

Delay time per cycle would be estimated as

$$d_i = C - G_i$$

$$= C - C \times \frac{Q_i}{s} \tag{e}$$

where, C is the cycle length, s is the saturation flow rate, and Gi is the effective green time. The number of cycles while vehicles run through subsection i would be estimated as

$$m_i = k_m \times \frac{L_i}{Q_i \times C} \tag{f}$$

The running time through each subsection i using running speed v would be

$$F_i = \frac{L_i}{v} \tag{g}$$

Finally, estimated travel time T_{si} in delay model would be

$$T_{si} = D_i + \frac{L_i}{v} + \frac{(L_{oi} - L_i)}{v_a}$$

$$= (C - C \times \frac{Q_i}{s}) \times k_m \times \frac{L_i}{(Q_i \times C)} + \frac{L_i}{v} + \frac{(L_{oi} - L_i)}{v_a}$$

$$= k_m \times \frac{L_i}{Q_i} - L_i(\frac{k_m}{s} - \frac{1}{v}) + \frac{(L_{oi} - L_i)}{v_a} \quad \text{(h)}$$

Where the first and second items in equation (h) mean travel time in congested section, and the third item means travel time in uncongested section. If regression coefficient k would be assumed to equal $k_m/s - 1/v$, Tsi using the delay time model equals that using the Sandglass model. In practice, since jam density and saturation flow rate are utilized in the Sandglass model, this model is impractical to be applied in real situation than the delay model due to the difficulty in obtaining the data.

2.2.4 England

2.2.4.1 Autoregressive Model For Travel Time Estimation: Bus Journey Time Estimation (Bell and Cowell, 1988)

A recursive identification procedure for an autoregressive model is formulated and field tested in England. A memory discounting factor is introduced to enable increased emphasis to be placed on the most recently observed journey time. Discounting is appropriate if the degree of autocorrelation varies with time to weight the more recently realized travel time more heavily, as prior consideration of the dynamics of bus operation would lead one to expect. For evaluation of the model alternative recursive procedure for bus journey time prediction using data collected for and by an experimental passenger information system were applied. For validation of the developed model they compared the autoregressive model with three different statistical properties. These properties are: 1) alternative predictions based on a recursively estimated mean; 2) on exponential smoothing; and 3) simply on the latest observed value.

2.2.4.1.1 Model Formulation: A hypothetical route having a stop

It is important to understand the dynamics of bus service operation to forecast bus travel time. Deterministic model of bus operation is first formulated by Newell and Potts in 1964. Since then, their model has been used for the guidance for forecasting the bus travel time because of its simplicity and illumination. The model proposed and explored in this section also adopted Newell and Potts model. The model constructed by Bell and Cowell are as follows:

$$x_i = p(h - \alpha x_{i-1} + \alpha x_i) \qquad \text{(a)}$$

where, x_i = the number of passengers boarding the ith bus

h = headway (1/h = bus arrival rate)

p = passenger arrival rate

α = boarding time per passenger ($1/\alpha$ = boarding rate).

Above formula would be further expressed by the arrangement:

$$x_i = \frac{-\alpha p}{1 - \alpha p} x_{i-1} + \frac{ph}{1 - \alpha p} \qquad \text{(b)}$$

When x is an equilibrium value, then

$$x = \frac{-\alpha p}{1 - \alpha p} x + \frac{ph}{1 - \alpha p} \qquad \text{(c)}$$

which implies that x = ph and $\alpha p \neq 1$. If equation (c) is subtracted from (b), we can examine the consequences of a perturbation to the system in equilibrium.

$$x_i - x = \frac{-\alpha p}{1 - \alpha p}(x_{i-1} - x) \qquad \text{(d)}$$

For perturbations to be damped over successive buses

$$\left| \frac{-\alpha p}{1 - \alpha p} \right| \langle 1 \qquad \text{(e)}$$

which in turn implies that

$$p < \frac{1}{2\alpha} \quad . \tag{f}$$

The equation (f) implies that the arrival rate of passengers must be less than half the boarding rate for the system stability. In addition to this, if the system is stable we note that

$$\frac{-\alpha p}{1-\alpha p} < 0 \tag{g}$$

indicating oscillatory behavior in the numbers boarding successive buses, and therefore in the stopped time. Let z_i be the departure headway from the stop. Then

$$z_i = h - \alpha p z_{i-1} + \alpha p z_i \tag{h}$$

Under equilibrium

$$z = h - \alpha p z + \alpha p z = h . \tag{i}$$

Suppose now that the arrival headway (h_i) between buses i and i-1 differs from h. If the system had previously been in equilibrium, then the corresponding departure headway is given by

$$z_i = h_i - \alpha p z + \alpha p z_i \tag{j}$$

If the equation (i) is subtracted from (j), then

$$z_i - z = h_i - h + \alpha p(z_{i-1} - z) \tag{k}$$

$$= \frac{h_i - h}{1 - \alpha p} \tag{l}$$

If the system is stable, we have that $0 < \alpha p < 1/2$, which in turn implies that

$$2 \rangle \frac{1}{1-\alpha p} \rangle 1 \qquad \text{(m)}$$

The ratio of the passenger arrival rate to the passenger boarding rate magnifies headway perturbation at a stop. Thus, the passenger arrival rate increases, service regularity is likely to deteriorate, and irregularities in headways tend to get magnified along the route.

Suppose that successive buses have identical speed profiles, so that the time from passing A until coming to rest at the stop, and then from leaving the stop to passing B, denoted by C, is constant.

$$t_i = c + \alpha x_i \qquad \text{(n)}$$

where, t_i is the time the ith bus takes to pass from A to B. Since it is assumed that the time stopped depends on the number of passengers boarding, using equation (n) to eliminate x_i from (a).

$$t_i = c + \alpha p(h - t_{i-1} + t_i) \qquad \text{(o)}$$

Equation (o) represents a first order, autoregressive model for the journey time. For non-pathological values of α and p, the system will oscillate. The degree of autocorrelation for a succession of stops depending positively on the passenger arrival rates at the intervening stops, or negatively on the passenger boarding rate. However, the system is no longer a first-order autoregressive process.

2.2.4.1.2 Multiple Stops

In the absence of information about how the process is changing over time, the value of past travel times in the estimation process is reduced. The memory of past journey times should therefore be successively discounted. Consequently, a recursive algorithm to minimize the discounted sum of squared errors is derived. Such an algorithm has been suggested by Ljung and Soderstrom (1983).

The autoregressive process would imply that

$$t_i = a_0 + a_1 t_{i-1} + a_2 t_{i-2+ \ldots +} a_n t_{i-n} \quad \text{(p)}$$

where, t_i is the time the ith bus takes to pass between two points in a multiple stops in a network. Equation (p) will be stable if the polinomial equation in w has all its roots strictly within the unit circle.

$$w^n - a_1 w^{n-1} - \ldots - a_n = 0 \quad \text{(q)}$$

A first-order autoregressive process, implying n=1, will be stable if $|a_1| < 1$. If we define

e_i = an error term, sometimes referred to as equation error

$y_i = t_i$

$\underline{t}_i = [1, t_{i-1}, t_{i-2}, \ldots, t_{i-n}]^T$

$\underline{b} = [a_0, a_1, a_2, a_3, \ldots, a_n]^T$

Then equation (p) may be reexpressed as

$$y_i = \underline{x}_i^T \underline{b} + e_i \quad \text{(r)}$$

The discounted sum of squared error is

$$E = \sum_{i=0}^{k-1} (e_{k-1})^2 \lambda^i = \sum_{i=0}^{k-1} (y_{k-1} - \underline{b}^T \underline{x}_{k-i})^2 \lambda^i \qquad \text{(s)}$$

where $0 < \lambda < 1$ is a discount factor. Let j=k-1, then

$$E = \sum_{j=1}^{k} (y_j - \underline{b}^T \underline{x}_j)^2 \lambda^{k-j}. \qquad \text{(t)}$$

Equations for the parameters, which in this case are necessary and sufficient for a minimum discounted sum of squared errors, are obtained by partially differentiating E with respect to the elements of $\underline{b}$ and setting the result equal to zero

$$\frac{\partial E}{\partial \underline{b}} = -2 \sum_{j=1}^{k} \lambda^{k-j} (y_j - \hat{\underline{b}}_k^T \underline{x}_j) \underline{x}_j^T = \underline{o}^T \qquad \text{(u)}$$

The parameter values uniquely defined by these equations are denoted by $\hat{\underline{b}}_k$. Rearrangement of equation (u) yields

$$\sum_{j=1}^{k} \lambda^{k-j} y_j \underline{x}_j = \sum_{j=1}^{k} \lambda^{k-j} \hat{\underline{b}}_k \underline{x}_j \underline{x}_j^T \qquad \text{(v)}$$

Let

$$\underline{Z}_k = \sum_{j=1}^{k} \lambda^{k-j} y_j \underline{x}_j \text{ and } \underline{X}_k = \sum_{j=1}^{k} \lambda^{k-j} \underline{x}_j \underline{x}_j^T. \text{ Hence, } \underline{Z}_k = \underline{X}_k \hat{\underline{b}}_k.$$

Note that

$$\underline{X}_k = \lambda \underline{X}_{k-1} + \underline{x}_k \underline{x}_k^T \qquad \text{(w)}$$

$$\underline{Z}_k = \lambda \underline{Z}_{k-1} + \underline{x}_k \underline{y}_k \qquad \text{(x)}$$

Define

$$\underline{P}_k = \underline{x}_k^{-1}.$$

Then from equation (w), we have

$$\underline{P}_k^{-1} = \lambda \underline{P}_{k-1}^{-1} + \underline{x}_j \underline{x}_j^T \qquad \text{(y)}$$

By the use of the matrix inversion lemma we obtain

$$\underline{P}_k = \frac{1}{\lambda}[\underline{P}_{k-1} + \underline{P}_{k-1}\underline{x}_k(\lambda + \underline{x}_k^T \underline{P}_{k-1}\underline{x}_k)^{-1}\underline{x}_k^T \underline{P}^T{}_{k-1}] \qquad \text{(z)}$$

Finally, the best fit parameters after k observations are given by

$$\hat{\underline{b}}_k = \underline{P}_k \underline{Z}_k . \qquad \text{(aa)}$$

A recursive relationship for $\hat{\underline{b}}_k$ may be developed by substituting equation (z) and (x) into (aa) to yield after much simplification

$$\hat{\underline{b}}_k = \hat{\underline{b}}_{k-1} + \underline{g}_k(y_k - \underline{x}^T{}_k \hat{\underline{b}}_{k-1}) \qquad \text{(ab)}$$

where,

$$\underline{g}_k = \underline{P}_{k-1}\underline{x}_k(\lambda + \underline{x}_k^T \underline{P}_{k-1}\underline{x}_k)^{-1} \qquad \text{(ac)}$$

Equation z can now be written as

$$\underline{P}_k = \frac{1}{\lambda}[\underline{P}_{k-1} - \underline{g}_k \underline{x}_k^T \underline{P}_{k-1}] \qquad \text{(ad)}$$

Note that because of the definition of $\underline{x}_k$ the procedure is only defined for k > n. Consider an algorithm suitable for the case where n=1 and λ=1. Two parameters, a_0 and a_1, are to be calculated recursively. $\underline{P}_1$ and $\underline{b}_1$ are initialized to

$$\begin{bmatrix} 9999 & 0 \\ 0 & 9999 \end{bmatrix} \text{ and } \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

If the process is non-stationary, discounting parameter values would be reduced from 1 to, say, 0.1.

2.2.4.1.3 Evaluation and Validation of Model

In order to evaluate and test the model manually collected one day sample and partial samples from the microwave AVI system were compared. These results suggested that random variations in travel time between stops or in passenger arrivals at stops prevent any detectable oscillations from establishing themselves.

When the predictive accuracy is compared in terms of total absolute error, the recursive fitting of an autoregressive model, first- or second-order, with or without a discounted memory, appears to yield worse predictions than all three alternative procedures for all three periods surveyed. The introduction of the discounted factor appears to reduce predictive accuracy in all cases.

The least squares estimates of coefficients and their standard errors for first-order autoregressive model and three periods in the manually collected data are compared in detail. The overall results showed that there is little evidence in practice of first- or second-order autoregressive behavior.

In order to test the alternative prediction procedure further, a simulation model was developed. The simulation is achieved by permitting the components of the Newell and Potts representation to vary randomly in accordance to appropriate statistical distribution. But travel time between stops was treated as a normally distributed variate with constant mean. The passenger arrival rate is treated as a Poisson process with a constant mean rate. The time taken per passenger to board is assumed to be a normally distributed variate. The predictions of the travel time between beacon 1 to another were obtained from past realized travel times by an autoregressive model as well as by the three alternative procedures. Despite a statistically significant degree of autocorrelation, total absolute prediction error for the autoregressive model remains greater than for two of the three alternative procedures. The introduction of a discounting factor to allow predictions to adjust better to non-stationary processes did not appear to improve predictive accuracy.

Finally, it appears that a good estimate of mean travel time, obtained perhaps by recursive estimation, provides as good and generally better predictive accuracy than more sophisticated autoregressive models.

2.2.5 The Netherlands

2.2.5.1 Flow-Density Relationship from Detector Output

This approach (Sisiopicu et al. 1993, Westerman 1993) is one of the most conventional approach to estimate the travel time on either freeway or signalized arterial roads. The algorithm is based on obtaining the flow and occupancy data from the loop detector. Under low traffic demand traffic time is independent from both flow and occupancy. As the demand increases, the correlation between travel time and occupancy becomes more significant. Then, regression analysis is utilized to predict the travel time. Due to the correlation between the independent variables (flow and occupancy), these two variables can never be used simultaneously. Hence, other supplementary independent variables such as signal setting (cycle length, red time), queue length, speed (running and desired) has been used for regression. The reported algorithm is, however, limited to correlate the through movement travel time function and detector flow and occupancy observed for arterial roads environment.

For validation license plate matching (both manual and AVI), floating car, wheelbase data matching techniques has been used. These techniques would be used for not only regression model validation purpose, but also appropriate for other algorithms. Also, all the current approaches for estimating travel time functions neglect the turning movement and adaptability (route specific). Thus, further study is required for adaptive algorithm and model. Another area of further research would be the development of theoretical mathematics model for converting detector data into travel time.

2.2.5.1.1 Travel Time from Flow Measurement

Unique methodologies which determines travel time directly from traffic flow measurement instead of relating basic quantities (speed, density, and capacity) has been proposed by Westerman in Netherlands. Other recent literatures were reviewed, and travel time estimation algorithms were classified into six categories. Brief descriptions are as follows:

1) individual vehicle identification via footprint

This technique is based on comparing the footprints (certain degree of induction power) of successive measuring sites. This technique is, however, reliable only on short distances. Another disadvantage of this technique would be distortion of signals when vehicles pass the loop slowly or halfway.

2) platoon of vehicle identification via footprints

This technique compares the group of footprints passing the measuring sites. One major disadvantage of this technique would be caused by lane change, on ramps, and weaving section. It is reliable only in 5 kilometer distance. Also, in terms of cost effectiveness in application, this technique is not appropriate.

3) in- and outgoing speeds and intensities

In the case of free flow traffic the mean travel time on a road segment with length L_k can roughly be calculated using the incoming and outgoing speed ($v_k(t_i)$ and $v_{k+1}(t_i)$) in that section, assuming that the speed of the traffic flow has the incoming speed until half way the section and that it has the outgoing speed in the second half of the section. In this way, the travel time at the measuring point in time t_i ($\iota_k(t_i)$) between sites k and k+1, in the case of free flow traffic, is

$$\tau(t_i) = \frac{1}{2}\frac{L_k}{v_k(t_i)} + \frac{1}{2}\frac{L_k}{v_k + 1(t_i)}$$

The results of this model is sufficiently accurate, $(t_{i+1} - t_i) < 2$ minutes, and $L_k \leq 1$ kilometer. A better method would be to use the ratio of the intensity I_k and incoming speed v_k for obtaining a valuation of the local traffic density. Then, the total number of vehicles ($N_k(t_i)$) in the section would be

$$N_k(t_i) = \frac{1}{2} L_k \times \left\{ \frac{I_k(t_i)}{v_k(t_i)} + \frac{I_{k+1}(t_i)}{v_{k+1}(t_i)} \right\}$$

$N_k(t_i)$ being known, is updated by adding the difference between the incoming and outgoing intensity

$$N_k(t_{i+1}) = N_k(t_i) + \left\{ I_k(t_{i+1}) - I_{k+1}(t_{i+1}) \right\} \times \delta t$$

where, $\delta t = t_{i+1} - t_i$. The travel time on the given section would be

$$\tau(t_i) = \frac{N_k(t_i)}{I_{k+1}(t_i)}.$$

The proposed methodology provide accurate results when the successive measuring sites does not exceed 500 meters. However, when the distance is greater than 1 kilometer, the accuracy decreases significantly.

4) intensity-flow correlation

This technique is based on the comparison of intensity between two sites. Hence, obtaining the stochastic signals from the detector is the input for the algorithm. However, it is not suitable for the congested network.

5) cumulative distribution

This technique is based on the individually registered vehicle (probe vehicle) passing. If vehicle categories drive at a different speed this may considerably improve the accuracy of the method. Disadvantage of this method is that when a certain percentage of vehicles is not or wrongly detected this could lead to an error in travel time estimation.

6) two-way communication

This technique is based on the beacons and Global Positioning System. Reliable, simple, and better data collection on short period of time for travel time estimation would be possible via advanced telecommunication systems.

His paper deals with simple counting, by matching cumulative distributions of successive measuring sites, in combination with correlation of fluctuations in the traffic flow, which would be used to guarantee a continuing reliability and accuracy of the travel time estimation. This method has been applied for real freeway operation in Netherlands, and producing reliable travel time data for both free flow and congested flow.

2.2.6 Australia

2.2.6.1 Automatic Network Travel Time System (ANTTS)

ANTTS has been developed for position specific incident detection and broadcasting congestion information for traffic network users in Australia (Longfoot 1991). In his paper, the requirements for a comprehensive network travel time measurement system are discussed and the characteristics of the developed systems are described. ANTTS is based on AVI to measure travel time. Data communication in AVI is conducted through the wide area traffic control system, SCATS (Sydney Coordinated Adaptive Traffic System). ANTTS does not collect travel time as raw data. It is inferred from a collection of vehicle identification events. The time stamped identity messages are stored in a vehicle time point data base which is a dynamic binary tree structure with a node allocated to each active vehicle. The tree is periodically scanned to determine if there is more than one location/time record for a given vehicle and if the time separation between events satisfies certain criteria a vehicle travel time record is generated.

A feasibility study of using AVI with SCATS to implement a travel time measurement system for metropolitan Sydney area was conducted in 1989. The major findings were: 1) 3700 taxis passing surveyed nodes satisfied the sample size criterion; 2) 200 nodes are required to cover Sydney metropolitan area; 3) 50 samples are required to give reasonable level of accuracy for travel time estimates; and 4) travel time distributions are considerably skewed and the most appropriate measures of central tendency are the median and the interquartile range. For full system to cover entire network in Sydney 4000 tagged vehicles and 250 interrogators are necessary.

2.3 ARTIFICIAL NEURAL NETWORKS

2.3.1 Introduction

Artificial Intelligence has been introduced in problem solving in the field of engineering for decades. Recently, school of interests focused on Artificial Neural Network (ANN) brought special interests in the field of recognition, classification, signal processing, and parallel processing. With the advent of Intelligent Transportation Systems, newly developed ANN has been investigated for its adaptability in various areas. Since ANN as problem solving technique in transportation is fairly new concept, it is worth while reviewing in this section. Some definitions, algorithms, and fundamental concepts were reviewed at the following sections. At the end of this section, current ANN applications in ITS were reviewed.

2.3.2 Fundamentals of ANN

2.3.2.1 Human Brain vs. ANN

The human brain consists of 10 billion to 500 billion neurons (Faghri and Hua, 1992). A cell body, dendrites, and an axon make up a biological neuron. Synapses which is connected to 100 to 10,000 other neurons. The brain is highly sophisticated massive computer. It interprets complicated information from the senses at very high speeds and creates a useful skill from the learning process. Human brain learns from a neuron which collects signals from dendrites. Then, the neuron sends spikes of electrical activity through an axon. Further, a synapse converts the electrical activity into an electrical effects to excite activity in the connected neurons. Then, it sends spike of electrical activity down its axon. The learning occurs by increasing the effectiveness of the synapses where the influence to other neurons changes.

The definition of ANN is "an information processing system that is nonalgorithmic, nondigital, and intensely parallel." In simple terms, ANN can be defined as a system that mimics the biological neural networks. Artificial neurons (processing units or processing elements) take after the function of biological neuron, i.e., process the inputs presented to them and compute the total value as output(s) with a transfer function. The principle that can approximately compute any reasonable function in an artificial neural networks is called architecture.

ANN connect mathematically modeled neurons through synapses which are represented as modifiable weights. Hence, the ANNs are adaptive, trainable, and massively parallel.

In general, more hidden units and hidden layer yield more efficient learning and better performance, and training bias to each neuron are trained as regular weights to hasten the convergence of the training process. The unpredictable nature inherent in the system may be accurately represented by the bias neuron.

ANN do not impose a relationship directly between the input and output variables, but seek to deduce the strength to be attached to different relationships by *learning* from examples (Kirby et al., 1994). However, if a direct mathematical relationship is derived from the data, then the relationship is explicit.

In order not to have scale bias, all the input vector should be scaled in between 0 and 1. Factors to be considered in model building are: 1) activation function output range, 2) allowable error, 3) step size, and 4) number of hidden layers and neurodes.

If the relation between the inputs and outputs are complex, a linear activation function (transfer function) is usually inadequate for function fitting process. For performance evaluation, sum of square error (SSE) can be adopted.

$$SQE = \sum_{i=1}^{N} (Z_i - T_i)^2$$

The squared error (SQE) represents the derivation of predicted outputs from the expected targets. Another measure of error is the percentage error (PE) for each test data point.

$$PE = \frac{(Z_i - T_i)}{T_i} \times 100\%$$

2.3.2.2 Definitions

- processing element

It is the basic building block in ANN. It is also called processing unit or artificial neuron. Figure 2.3.1 illustrates structure of neural networks.

- weights

It is designed to connect processing elements. The strength of connection among the neurons are represented by the weight.

- transfer function

It is an operator that add all the input values from the processing units to produce the outputs. The main reason of using transfer function is to obtain nonlinear output (i.e., normalizing the output) nor from linear input. Usually, *sigmiodal squashing function* or *hyperbolic tangent* are adopted. It is also called as activation function.

Sigmoidal squashing function can be expressed as

$$y = \frac{1}{(1+e^{-I})} \qquad 2.3.1$$

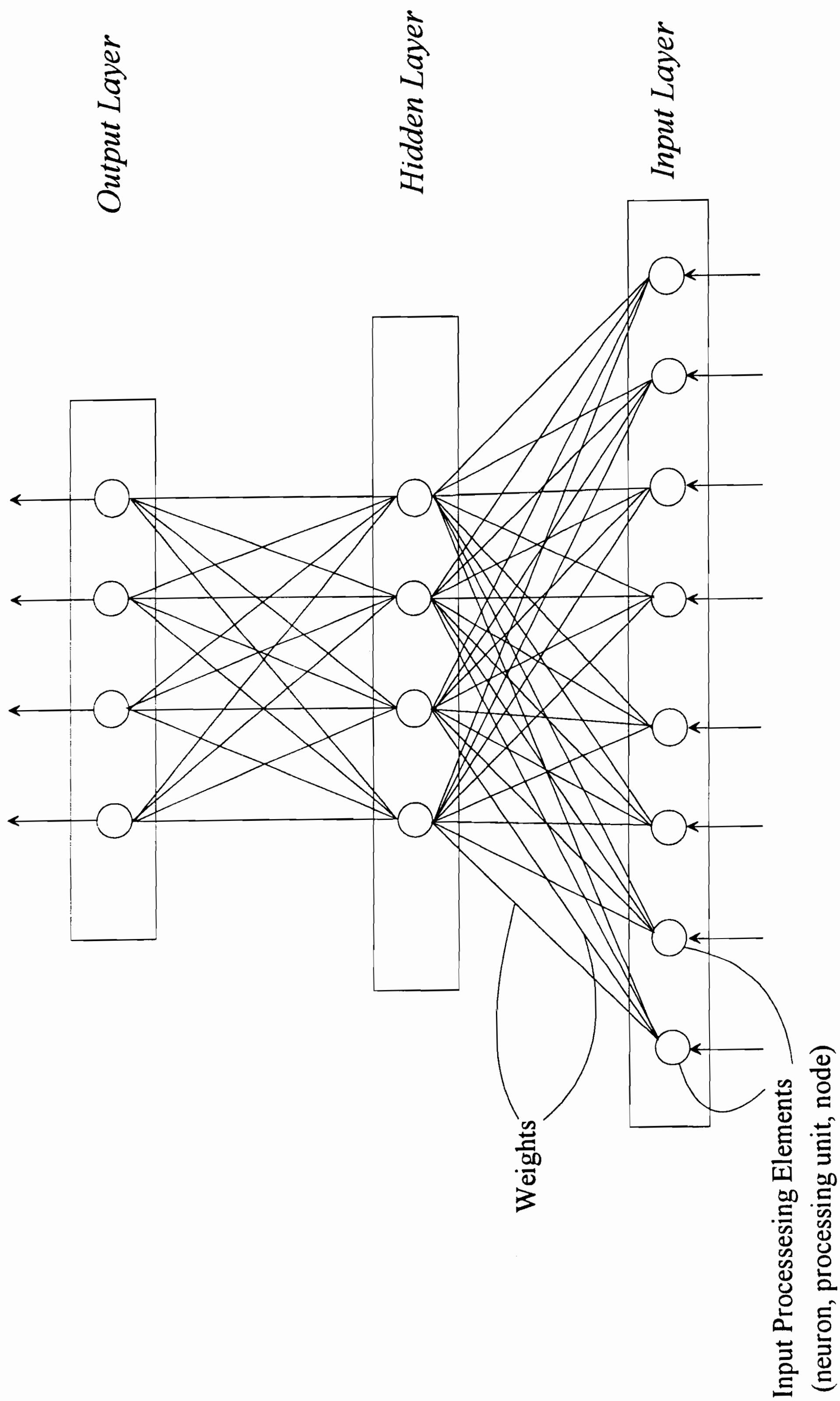


Figure 2.3.1 Structure of Neural Networks

where, y = output

$$I = \sum_i weight(i) \times output\ from\ the\ previous\ layer's\ neuron(i)$$

It squashes the output into 0 to 1 range. Derivation of sigmoidal squashing function can be obtained by getting derivative of y based on input value (I).

$$\frac{dy}{dI} = -1 * (1+e^{-I})^{-2} * (-e^{-I}) = \frac{e^{-I}}{(1+e^{-I})^2} \qquad 2.3.2$$

However,

$$y(1-y) = \left(\frac{1}{1+e^{-I}}\right)\left(\frac{e^{-I}}{1+e^{-I}}\right) = \frac{e^{-I}}{(1+e^{-I})^2} \qquad 2.3.3$$

Therefore, we use $y(1\text{-}y)$ as derivative of sigmoidal squashing function in real application instead of using derivative of sigmoidal squashing function.

- input layer

It is the initial layer in the ANN structure and composed of sets of input processing elements. It has function to transfer input signal to the hidden layer. No transfer function is conducted in this layer.

- hidden layer

It is the middle layer in the structure of ANN. Outputs from the input layer can be the inputs of the hidden layer. Transfer function is usually activated in this layer.

- output layer

It is the top layer in the structure of ANN. Outputs from the hidden layer can be the inputs of the output layer. Transfer function does not always activated in this layer.

- training

It is the process of changing weight.

- learning

It is the effect of training.

- learning rate

It influences the amount of adjustment to the connection weights between successive iterations. It ranges between 0 and 1 (0 means no change in weight).

- training tolerance

It indicates the range for which outputs are considered correct. If it set to too low, it takes longer time to train the network. If is set to too high, it tends to trap in local minima.

• momentum

It is used for improving the training time and stability of process. It involves adding the term that is proportional to the amount of previous weight change in the weight adjustment process.

• bias input

A constant - usually binary value - input to each processing elements. It is usually connected to the input processing units and output processing units directly.

2.3.3 Training Algorithms

There are multiple numbers of training algorithms and researches are focused on decreasing the training time in many aspects.

2.3.3.1 Supervised Learning

It requires the presence of external guidance and known desired output set. The network output will be compared with desired output response during the training process. The difference between desired output and actual output is used as indicator to correct the network.

2.3.3.2. Unsupervised Learning

It does not require an apriori knowledge of desired output. The unknown training data is fed into the network and then it forms internal clusters. Eventually, presented data set is classified into inherent classes after multiple iterations.

2.3.3.3 Self-Supervised Learning

It investigates internal performances. If error exists, error correction feed-back occurs. After the multiple iteration correct response will be produced.

2.3.3.4 Backpropagation Learning

Its algorithm consists of two phases - forward and backward pass. In order to understand the backpropagation learning algorithm easily, let's consider the simple hypothetical networks. It is consists of two input processing elements, one hidden layer with two processing elements, and two processing elements in output layer. Figure 2.3.2 illustrates the configuration of hypothetical neural networks.

2.3.3.4.1 Forward Pass

The input processing elements just pass the input values (bipolar values) and no squashing functions are associated with these. Hence,

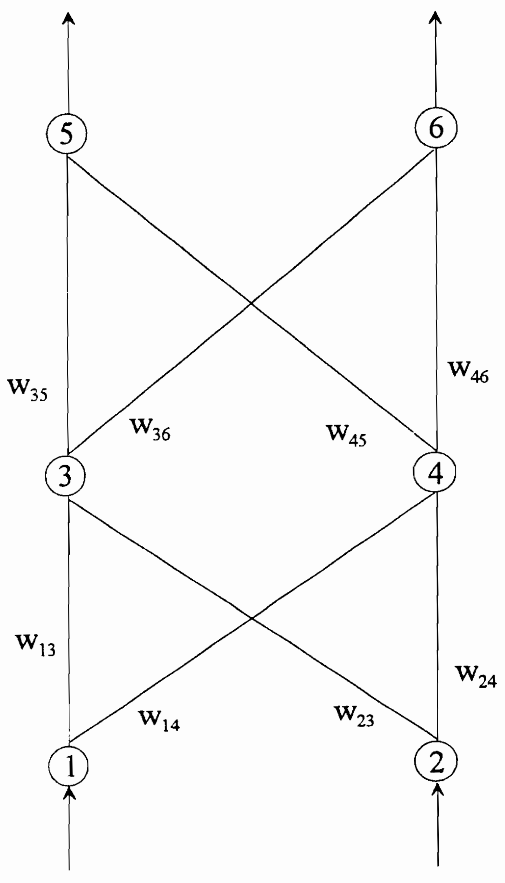


Figure 2.3.2 Hypothetical Neural Networks

$$y_i = x_i \text{ for all } i \qquad 2.3.4$$

In the hidden layer,

$$I_j = \sum_i w_{ij} \times y_i \qquad 2.3.5$$

where, i = neurons at input layer

j = neurons at hidden layer

To get the outputs from the hidden layer, sigmoidal transfer function is adopted.

$$yi = f(Ii) = \frac{1}{1 + e^{-Ii}} \qquad 2.3.6$$

Since, outputs from hidden layer neurons can be inputs to the output layer neurons, the same process from equation 2.3.5 to 2.3.6 will be repeated.

2.3.3.4.2 Backward Pass

Backward error correction process starts from the processing elements at the output layer. Since the processing element(s) in the output layer is associated with the sigmoid squashing function, error (difference between desired output and actual output) should be scaled. This process is as follows:

$$e_j' = e_j \times y_j(1 - y_j) \quad \text{for all j}$$

where, e_j' = scaled error at processing element j

e_j = error at processing element j

y_j = output from the processing element j

Then, calculate the changes in hidden to output layer weights.

$$\Delta w_{ij} = c \times e_j' \times y_i \qquad \text{for all i and j}$$

where, Δw_{ij} = weight correction term from neuron i in hidden layer to neuron j in output layer

c = learning rate

y_i = output from the hidden layer neuron j (i.e., input to the output layer neuron j)

Next step will be the weight change.

$$w_{ij}' = w_{ij} + \Delta w_{ij} \quad \text{for all i and j}$$

where, w_{ij}' = adjusted weights from hidden layer to output layer

w_{ij} = previous weights from hidden layer to output layer

Now next step will be the adjustments of weights from neurons in the input layer to neurons in the output layer. Since we have desired output(s) for output neuron(s), it is easy to calculate the Δw. However, at the hidden layer no desired output(s) is associated with it. Thus, weight changing law in hidden layer is different than that of output layer. The weight changing mechanism in hidden layer is as follows:

$$e_i' = [y_i(1 - y_i)]\sum e_j' w_{ij} \qquad \text{for all } i \text{ and } j$$

where, i = neurons in hidden layer

j = neurons in output layer

Hence,

$$\Delta w_{ij} = c_1 \times e_j' \times y_i, \quad \text{and}$$

$$w_{ij}' = w_{ij} + \Delta w_{ij}$$

where, w_{ij}' = new weights from input layer to hidden layer

w_{ij} = previous weights from input layer to hidden layer

2.3.4 Pro's and Con's

Through the extensive literature review, some of artificial neural networks' advantages and disadvantages were reviewed in this section.

2.3.4.1 Advantage of Neural Networks

- It is suitable for real-time application where significantly short development time is required.
- It responses to unknown situation.
- It provides the practical solutions for difficult or almost impossible problems.
- It has adaptive, memorizing, and parallel characteristics.
- It requires lack of need for technical statistical expertise.
- It encompasses the computation of non-linear complex statistical model in practical application.
- It outperforming a more simple linear model.
- It has capability of speedy computation (suitable for real-time application).
- It has ability to cope with missing or incomplete data.
- It has ability to be easily transformed into C code.
- It is nonparametric so that there is no need to make assumption about the functional form of the distribution of data.

- It is particularly useful when there is no effective way to explain reasoning.
- It is more responsive to dynamic conditions (feasible for ITS application)
- ANN model is also able to deal with analogical input database
- ANN can handle more complex problem, since multiple input and output vectors can be processed individually
- ANN learn from data and can be numerically quite versatile and robust
- it has on-line learning capability
- low cost development and high speed performance
- Causal relations between any input and output may be adequately captured because of intensive connection between adjacent layer

2.3.4.2 Disadvantage of Neural Networks

- It is perceived as a black box process.
- Heuristic approach is the only solution to make the determination on learning parameter, momentum factor, etc.
- It requires lengthy data preparation, otherwise biased results would be derived.
- It is impossible to determine in advance how much data is needed.
- It has so many paradigms.
- Interpretation of output is difficulty.

- High quality and success of the results obtained can be variable.
- It has tendency to fall into local minima.
- There is no guarantee to end the training.
- If too much extraneous information is presented to the network, the network can be overwhelmed. In reverse case network would be under-informed.

2.3.5 ANN Application Areas in Intelligent Transportation Systems

Some of the current ANN applications in ITS were reviewed at following sections. These were: short-term traffic flow prediction,

2.3.5.1 Short-Term Flow Prediction (Smith and Demetsky, 1994)

The previous approach of prediction model would be categorized into three types. These are: 1) data-based and historical algorithm; 2) time-series model; and 3) simulation. Historical and data-based algorithm has been famous for its simple and easily implementable capability. However, it has lack of ability in coping with dynamic behavior of network. AUTOGUIDE in London UTCS is based on this algorithm. Meanwhile, time-series model (ARIMA model) has been broadly used for prediction model in transportation. In general, time-series model generates unsatisfactory goodness of fit and high errors. In case of simulation model, no model is advisable due to complexity of traffic behavior.

Recent development of neural network model was compared with historical, data-base model, and time series model. Volume (t), volume (t-15 minutes), historical volume (t), historical volume (t+15 minutes), average speed (t), and wet pavement (t) were fed into the model as input variables. Backpropagation learning algorithm was applied to train the network.

Although the results of three different cases of short-term volume prediction models generated adequately, the neural network model was clearly superior. The neural network model responded more actively to dynamic conditions than the historical, data-base model. Also, neural network model did not experience the lag and overprediction characteristics of the time series model. Smith and Demetsky concluded that due to neural networks model's ability to run in parallel computing environment, it can be a good candidate for the real-time application in the Intelligent Transportation Systems.

2.3.5.2 Dynamic Traffic Pattern Classification (Area-Wide Signal Timing Control, Hua and Faghri, 1993)

Currently, time-interval-dependent control is the most common area signal-timing control strategy in dynamic traffic. In this approach time of day is splited into several time intervals, e.g., peak-hour interval, normal day time, night time. However, it assumes that the traffic volume on each roadway of the network is constant over each time interval. Then, the model determines the control parameters for each set for each time interval. However, it is known that the dynamic behavior of traffic volume can be expressed by traffic patterns. Hence, traffic pattern should be classified analogically, since numerical comparisons of volume difference in more than one links will lead to wrong classification. Hua and Faghri discussed the requirements of the traffic pattern classifications. These are as follows: 1) analogic pattern recognition and classification, and 2) acceptable tolerance between traffic pattern.

The adaptive Resonance Theory 1 (ART1), which deals with binary integer, was selected as a paradigm for their model. The underlying reasons for adopting ART1 are that their "networks automatically stabilize pattern categories and automatically activate new processing units when they are needed to create new categories." Also, ART1 can classify analogic patterns into appropriate categories, and it can be trained on-line.

A hypothetical transportation network (consists of 6 nodes and 7 links) is applied under the assumption that no turning movement, one way, and capacity as 1800 veh/hr. Data preprocessing was conducted for the input vectors. All the link volume was transformed into binary vectors. For the binary conversion, volume to link capacity ratio was calculated (under the assumption that volume is less than or equal to capacity). Then, the ratio was transformed into a 10 elements of binary vector.

It was concluded that ANN approach is more natural and reasonable than the conventional method, dynamic programming, due to the reason that as the network complexity increases, computational burden increases, i.e., impractical in on-line application. Also, ANN model is more effective and efficient in determining appropriate number of time intervals, and it provides parallel process so that it can deal with large dynamic data base.

2.3.5.3 Vehicle Detection (Bullock et al., 1992)

Current image processing technology has drawbacks on varying lighting conditions, camera perspective and shadow. This project is initiated under the umbrella of Autonomous Land Vehicle in a Neural Networks (ALVINN) project in Carnegie Mellon University. Basic idea is to train a network to continuously monitor a general region or location of a video image and to recognize when the image contains a vehicle. Hence, the network should be trained to recognize when a vehicle has entered the detection zone, is within the zone, and has left. The way the network indicates the relative position of the vehicle within the image is an important decision regarding this approach. In their research, several analog output units, where each output unit corresponds to a particular zone, were selected to represented a Gaussian type distribution indicating the most possible location of the vehicle. When a vehicle reaches at the zone, the output begins to rise toward 1.00. Then the vehicle reaches the center of the zone and begins exiting the zone, the output begins declining towards 0.00.

The architecture of image processing using neural networks are: 1) digitize a video image by a matrix (512 pixel by 480 pixel), and 2) pixel aggregation into 25 by 25. In their paper, 140 inputs were used (7 by 70) and these were empirically selected by visual observation. For the neural network model development, 2 hours of live traffic was recorded and VCR was used to generate video signal. Then, it is digitized, manually classified, and digital image was aggregated. 233 training images were collected. 150 images were utilized for training and 83 were reserved for testing.

According to the analysis of the model, 12 percent (9 out of 83) outputs were erroneous for the test case. However, since the vehicle just entered and moving out the detection zones were considered to be error by the erroneous condition, the result was not so bad. In the conclusion, they suggested that the initial result of ANN approach for vehicle detection provide better alternative than more traditional image processing architectures.

2.3.5.4 Freeway Incident Detection (Cheu et al., 1991)

Conventional California-type algorithm has been broadly used for incident detection. However, its high false alarm rate, low detection rate, and long mean time to detect incident are not suitable for real-time ATMS application in ITS and integrated freeway and arterial traffic management. Hence, new approach utilizing ANN were adopted to construct the effective incident detection model, since the problem is viewed as pattern recognition. ANN are information processing structures that are based on simplified theoretical models of the functioning of the human brain. Information can be processed in a parallel and distributed fashion in ANN due to its architecture.

They divide the freeway sections into shorter segments, based on the availability of the data and the level of detail. Volume and occupancy data were obtained from the loop detector for the input data. As it were, detector data from the upstream and downstream end of a freeway section at 30 second interval are the input data for the neural networks and desired outputs are the four states (incident free, beginning of queue, queue section, end of queue section). Network training is done by using synthetic data set.

The areas of occupancy above 0.26 and volume under 1000 vphpl are considered as the region for congestion in volume-occupancy plot. For volume and occupancy for 30 second freeway sections, Monte Carlo simulation methods for the upstream and downstream are utilized. Then, the desired state (e.g., if upstream is congested and downstream is uncongested, then category defined as state 2 - beginning of congestion) of traffic conditions at the freeway section were assigned. After all of these assumptions and data preprocessing, 1000 training vectors were generated and used in the training process.

Instead of using volume itself, the ratio of volume against maximum flow is applied in order not to have scaled bias. The structure of neural networks model was composed of four input features (occupancy and volume/max. flow in upstream and downstream), 10 hidden units, and 4 output neurons. After 5000 iteration, the total sum of square error (SSE) reached at 0.085. At the test, simulated set of 1000 pattern yielded an overall recall of 99.1 percent for all four states.

A 6 mile section of the eastbound SR-91 freeway in Orange county, CA was selected as off-line case study. Integrated Traffic Simulation (INTRAS) model was used to generate a microscopic simulation of freeway operation under the incident conditions and to output detector occupancy and volume as needed for incident detection purpose. The result of multilayer perceptron was compared with California #8 algorithm. The results revealed that neural networks model detects incident in 360 seconds and the latter in 480 seconds. Also, the latter model was not capable of showing the queue condition during the period of interest. At the conclusion Cheu et al. believed the performance of the multilayered perceptron model in this test was encouraging one.

2.3.5.5 Freeway Incident Detection (Chang, 1992)

This study explores the applications of advanced technology to enhance conventional freeway incident detection techniques and improve operational efficiency for computerized surveillance application. detection rate, false alarm rate, time to detection are the three main performance measurements in incident detection. The conventional comparative occupancy-based incident detection logic is known to have difficulties when there are natural changes in occupancy due to grade and geometry. INCDET 8 - one the most popular revised version of FHWA's Algorithm 8 - requires continuity in input data, otherwise the result will be severely affected.

A neural network (ANN) can be trained by the historical data to simulate the functions of INCDET 8. Another ANN was trained to make up the major disadvantage of INCDET 8 (time series problem). A model for a combination of the two ANN and INCDET 8 can be a potential solution to incident detection. ANN can be trained by the data to recognize volume patterns in freeways and to detect incidents. The development process of an ANN includes data preprocessing, a training procedure, and testing. For data preprocessing, occupancy rate, interval, state variable were selected. Here, incident states were the output of the ANN. In training, selection of proper parameters are important ,e.g., learning rate, training tolerance, blurring effect, system noise, network size. He also claimed that a fuzzy system can be also applied to INCDET 8.

Six freeway stations were used in the test database (five intervals were used). Each station was tested for consecutive 32 time periods. In model 1, the best result is 3 errors out of 18 unknown results, and the difference amount to 3. The correctness was 83 %. Chang suggested that by experience, if the range of accuracy is up to 90 %, result would be acceptable. Hence, he concluded from the first model that since the amount of the trained data was not enough, the factor blurring effect causes a *blurred* duplicate of the input data. Because of this reason, the amount of the trained data become larger. However, as the amount of input data was small, the result was not good enough. For INCDET 8 and INCDET_NN2 model, the error rate was 28 percent. In the combination of NN1 and NN2, the error rate was 33 percent. Here, even though the accuracy of the results of this ANN was far inferior to NN1, the advantage of NN2 was that it requires far less data. The combination of INCDET_NN2 and INCDET 8 predicts missing inputs, fast, moderately accurate, and loosely dependent on previous inputs. Although loosing some accuracy, this model becomes the best choice for incident detection. As it was mentioned, the ANN that he developed required more data to train them properly.

2.3.5.6 Route Choice Behavior Under Advanced Traveler Information System (Yang et al., 1993)

Prediction accuracy of the model, the acceptance rate of advice, and the quality of advice are closely connected

The data used for analysis was collected from learning experiments using a computer simulation. It was found that most subjects made route choices based mainly on their recent experiments. Hence, if drivers are given poor (inaccurate) information, they are unlikely to follow the system advice in immediately subsequent trips. Route choice behavior was also related to the characteristics of the respective routes and varied significantly from driver to driver.

Their neural networks model was identified as 9 processing elements in the input layer, 5 neurons in hidden layer, single processing element in output layer to indicate a choice between freeway and side road. For the input vectors, current advice, agreement satisfaction, disagreement satisfaction, speed, delay from the side road are fed into the network. For the freeway side of the input, agreement satisfaction, disagreement satisfaction, speed, delay are fed into the network for the training purpose. The input variables are as follows:

x1: binary input variables, (1 if the freeway is recommended by the information system, otherwise 0, x2: previous day's weighted sum of the driver's level of satisfaction when they followed the advice to take the side road,

x3: previous day's weighted sum of the driver's level of satisfaction when they did not follow the advice to take the side road,

x4: a weighted sum of the driver's estimate of their perceived speeds on the side road chosen in previous days,

x5: a weighted sum of the delays on the side road in previous days,

x6: weighted agreement satisfaction on the freeway,

x7: weighted disagreement satisfaction on the freeway,

x8: weighted speed on the freeway,

x9: weighted delay on the .freeway,

x10: driving frequency,

x11: gender (1 for male).

They suggested that if the route choice behaviors of a group of drivers are to be analyzed, drivers individual attributes (driver's gender and frequency) should be taken into account. The scale which has been used for evaluation of the categories are as follows:

	-2	*-1*	*0*	*1*	*2*
Evaluation of Choice	Incorrect Choice	Probably Incorrect	Don't Know	Probably Correct	Correct Choice
Evaluation of Speed	Incredibly Slow	Fairly Slow	Moderate	Reasonabl y Fast	Fastest Possible
Evaluation of Driving Frequency	Do not drive	Drive Infrequentl y	Never commuted but drive frequently	Don't know but formerly commuted	Currently Commute

The desired output is set to be 1 if the freeway is chosen and, and 0 otherwise. Boundary value is set to 0.5 during the prediction so that the freeway is resolved to be chosen if the output value is greater or equal to 0.5. They identified that two pieces of information are major factors in the route choice behavior model development. These were: 1) the route advice provided by the information system, and 2) the driver's perceptions of travel conditions on the freeway and side road.

Driver's knowledge or perception of travel conditions comes from the travel experience. Hence, these must be updated in choice sequences by combining historical information and experience. Perception updating is formulated as follows:

$$x(w) = (1-\lambda)x(w-1) + \lambda u(w-1),\ 0\leq\lambda\leq 1$$

where, u(w-1): a vector of driver's evaluations on his travel on last day,

x(w-1): a vector of driver's previous historical perception of road condition,

λ: experience factor which reflects the relative impacts of the last day experience and the accumulated historical knowledge on the individual's perception updating.

It is reported from the model testing and validation that the neural networks had an overall prediction rate of about 90 percent after convergence. Hence, the result suggested that the constructed neural networks model can be used reliably predict driver's route choice behaviors. Authors also suggested that in practice, it may be difficult to use a unified simple approach to describe driver's route choice behavior, because of diversity and variety of dynamic route choice behaviors across different subjects and across different types of road.

2.3.5.7 Trip Generation (Faghri and Hua, 1993)

In Urban Transportation Planning Process (UTPP), the main goal of trip generation step is to develop a mathematical model that can be utilized for forecasting the person trip production of the forecasting zone according to a number of socioeconomic indexes - population, number of vehicles in household, income, etc. ADALINE (Adaptive Linear Element) was used for the neural networks (ANN) approach for trip generation. The crucial difference between regression analysis and ANN is that coefficients in regression will tend to produce the minimum error on the surveyed data, whereas the training of ADALINE pursues the best value of the weights that will allow the model to obtain good results on the testing data set - not on the training data set. Backpropagation (BP) model was also examined to compare the performance with ADALINE. Here, instead of using popular nonlinear sigmoid function for transfer function, linear transfer function was adopted for ease of comparison with ADALINE.

Four input vectors (no. of single family houses, permanent single family residents, no. of multi-family dwelling units, no. of families with single auto ownership), two processing elements in hidden layer, and a desired output (trip rate) were identified for the ANN architecture. Data set was divided into three categories for training, testing and validation purposes. In training, conventional Delta Learning Rule was adopted. After 10,000 iteration, ADALINE model reached at its allowable mean square error (MSE). Three models (regression, ADALINE, BP) were tested by using validation data sets. The result revealed that both of the ANN model excelled the regression model. Comparison between the ADALINE and BP model, the former model performed slightly better than the latter one. However, in terms of training efficiency, BP performed better (2,500 iteration) than the ADALINE model. In conclusion authors envisioned that the artificial neural networks are to become powerful tools for transportation problems and ITS application as well.

2.3.5.8 Travel Time on Uninterrupted Flow (Hua and Faghri, 1993)

A demonstrated version of neural network travel time estimation model for a two-lane roadway segment which is assumed to be arterial with 30 mph free flow speed was investigated. The underlying assumptions were normal condition, exclude signal, blockage in one lane. For their study short length (1,000 feet) of arterial link was selected in order to avoid complexity of signal effect. The available lane reduction and weaving movement phenomena would be caused by lane blockage. The length and position of blockages are also important factors influencing the travel time estimation.

Backpropagation (BP) learning algorithm was adopted for a flexible neural network paradigm for travel time estimation modeling. The raw data for training the network was obtained from simulation run on the TRAF-NETSIM. The simulation period was five minute and simulation was divided into three groups (normal traffic, blockage on the right lane, blockage on left lane). The architecture of BP consists of three input vectors (5 minute traffic volume, blockage index for the right lane, blockage index for the left lane), five hidden processing elements, and one output neuron (average through travel time in second). Blockage index in input vectors are using binary value such as 0.0 for no blockage, and 1.0 for right lane blockage, respectively. The total of 96 data sets were divided into two groups, one for training and the other for testing, with a random order.

After 4,900 iterations training was completed with acceptable minimal error range. The BP estimated the travel time on a studied segment for average error of 5.5 seconds compared with the results obtained by NETSIM. The results of neural networks model was considered conceptually reasonable, since no other models were available to compare with.

2.3.5.9 Traffic Volume and Classification Monitoring (Mead et al., 1994)

A newly developed mapping neural networks - Connectionist Hyperprism Classification (CHC) was utilized to learn from data and to form a mapping of arbitrary topology. The CHC recognize multi dimensional patterns presented by the various vehicle signatures produced by the traffic volume and classification monitoring (TVCM) sensors and to produce corresponding classification outputs. The goal of the research was to provide TVCM system featuring high occupancy, adaptability to wide sensor and environmental variations, and continuous fault detection by using CHC networks.

The FHWA's 13 vehicle classification scheme divides vehicle largely according to application and/or axle configuration. The TVCM typically combines piezoelectric strip sensor(s) with magnetic loop sensor(s) to detect vehicles. Reliability and accuracy issues are always embedded in vehicle detection technology and TVCM utilizing above-mentioned sensing technologies are not exceptional. They defined the adaptive network and ANN as "an adaptive network is a collection of many identical processing elements, connected into a network, together with training algorithm that allows the network to *learn* or *adapt* from a database of examples. ANN is a kind of adaptive network that is patterned loosely after biological neural system, such as the brain or the visual cortex."

Commercial TVCM was tested to compare the result with neural networks model. In volume and occupancy estimation, commercial system provided 99 percent of accuracy. However, in classification of vehicle of 2 and 3, the results were poor (yet 95% in NN case). Hence, it does indicate the difficulty of obtaining accurate and reliable results from current system, and it puts the accuracy obtained here for the NN-based system in a favorable light.

2.3.5.10 Multiperiod Travel Time Estimation in Transportation Networks (Wei and Schonfeld, 1993)

The selection and timing of improvement project is a challenging problem when the benefits or costs of projects are interdependent. Hence, multiperiod network design model was proposed to select the best combination of improvement project and schedules. They define the multiperiod network design problem (MPNDP) as "given a transportation network and a finite set of improvement projects, find the optimal combination and schedule of projects such as the total cost is minimize throughout a finite number of planning periods." The MPNDP may be solved by exact procedures or heuristic methods. However, it is time consuming and has not cost-effectively achieved. Thus, the NN approach for estimating total travel time corresponding to various project selection and scheduling decision was made. The advantage of using ANN for MPNDP is that after the network is trained, it may be repeatedly used for complex problems with little calculation time.

The assumptions for the NN applications for MPNDP were that traffic demand was expressed in terms of peak hour O-D trip table for each year, and was independent of network improvements, and for five-year planning horizon. In order to identify the critical location of network V/C ratio of each link, six critical segments were extracted from the network. Adding one additional lane to each critical link in both directions was considered for all segments. All other characteristics (signal control type, free flow speed, etc) were assumed to be unchanged. Traffic assignment for five times had to be performed to obtain the total travel time for each improvement combination. ANN was used to estimate the resulting total travel time over a multiperiod planning horizon for various combinations of project schedules.

In conclusion, they identified that the relatively simple ANN model was very powerful tool for predicting accurate network outcomes. Some of the findings of ANN training related issues were 1) training time decreases when the output range size decreases, 2) avoid over training because it will distort the purely random nature of the original problem, 3) when the accuracy in the training process concerns, the trade between computational time and prediction accuracy should be made, and 4) identical step size (training coefficient) between input and hidden, and between hidden and output shouldn't be used (i.e., larger training coefficient should be used for hidden and input rather than hidden to output layers.

CHAPTER III BUS ARRIVAL TIME ESTIMATION: A PROTOTYPE

3.1 INTRODUCTION

Deterministic model of bus operation was first introduced by Newell and Potts (1964), and then Bell and Cowell (1988) suggested the more descriptive dynamic model which covers bus journey times between the single stop and multiple stops. They expanded Newell and Potts' model by introducing recursive autoregressive model. However, one of the unrealistic assumptions that both of the former researchers had made was that the passenger arrival rate at bus stop and boarding rate are time independent values. In reality this assumption is not valid and, therefore, it is assumed in this research that passenger arrival rate and boarding rate follow a distribution, e.g., Poisson, Negative Exponential, etc, or time dependent characteristics. Thus, time varying passenger arrival rate and passenger loading rate were adopted and simulated in the PC MATLAB.

Model developments in this chapter consist of three phases. The first phase was focused on the study of dynamics of bus behavior at a single stop. The number of boarding passengers were simulated based on the constant and time dependent passenger arrival rate and boarding rate. The second phase was based on the study of bus behaviors at multiple bus stops. Simulation on loading time variation was the main focus of this phase. In the last phase, a prototype bus arrival time model was formulated and evaluated by adopting constant and varying passenger arrival rate and boarding rate. Parameter Adaptation Algorithm (PAA) was applied in this phase, and parameter errors and estimation errors were evaluated.

An assumption adopted in the formulation of prototype arrival time estimation model was that the bus location data, and its arrival and departure time at each bus stop was acquired by the AVL system. Another assumption was that arriving passenger at each bus stop will be all accommodated by the bus arrived.

3.2 SINGLE BUS AT THE SINGLE STOP

3.2.1 Model Formulation

Dynamics of bus operation should be thoroughly understood in order to forecast the bus travel time in between the arbitrary locations in transportation network. The initial approach of development of travel time estimation model in this research relies on the extended autoregressive model of Bell and Cowell. In this research, however, the averaged distribution of passenger arrival rate was considered in order to be more realistic and accurate model for interpreting the real-world behavior of transportation system as well as to be readily applied to the upcoming ITS. The significant advantage of adopting recursive model is that the output of the model can be used for adjusting the model itself. Hence, the most recent output affects the current status of model the most through the adaptive process.

The first modeling approach was concentrated on the simplest architecture of the environment as illustrated in Figure 3.2.1. Dynamics of bus operations in between the stops were investigated. Although the model is simple, it retains the essential dynamics of the bus operations. The formulation of model is as follows:

$$z_{i+1} = h_{i+1} - \alpha_{avg,i} x_i + \alpha_{avg,i+1} x_{i+1} \qquad 3.2.1$$

where, z_{i+1} = departure time headway between bus i and bus i+1 (minute/vehicle)

h_{i+1} = arrival time headway between bus i and bus i+1 (minute/vehicle)

$\alpha_{avg,i}$ = average boarding time at stop i (minute/passenger)

x_i = no. of passenger boarding on the bus at stop i (no. of passenger/vehicle)

From Figure 3.2.1, $p_{avg,i+1}$ and $\alpha_{avg,i+1}$ can be calculated as

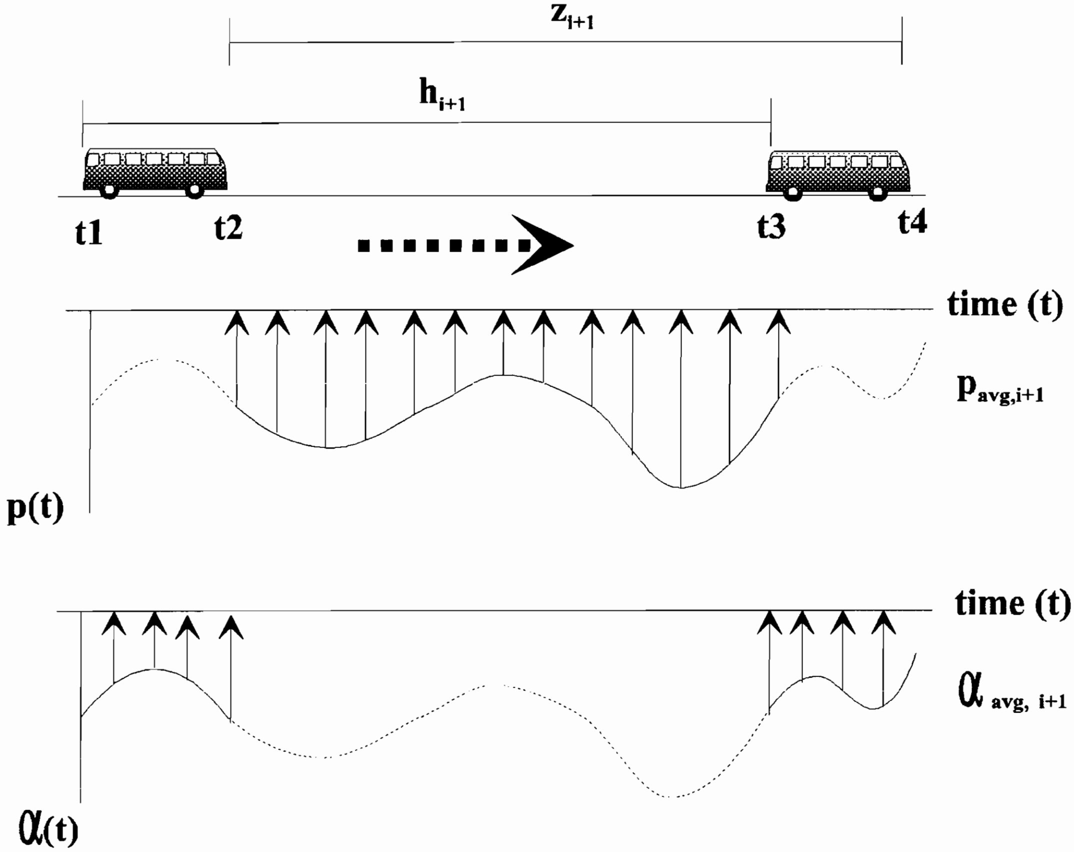


Figure 3.2.1 Travel Time Estimation on Simple Bus Stop

$$p_{avg,i+1} = \frac{\int_{t2}^{t4} p(t)dt}{z_{i+1}} \quad 3.2.2$$

where, p(t) = passenger arrival rate at time t(passenger/minute)

$$\alpha_{avg,i+1} = \frac{\sum_{k=0}^{x_{i+1}} \alpha_k}{x_{i+1}} \quad 3.2.3$$

where, α_k = passenger boarding time (minute/passenger)

The number of passengers boarding the $i+1^{th}$ bus will be the integration of passenger arrival rate between t_2 and t_4. As noticed here, passenger arrival rate (p) is continuous function, but passenger boarding time (α)is discontinuous function.

$$x_{i+1} = \int_{t2}^{t4} p(t)dt \quad 3.2.4$$

Hence, from equation 3.2.1 and 3.2.2, equation 3.2.4 can be rewritten as

$$x_{i+1} = P_{avg,i+1} z_{i+1} = p_{avg,i+1}(h_{i+1} - \alpha_{avg,i} x_i + \alpha_{avg,i+1} x_{i+1}). \quad 3.2.5$$

$$(1 - \alpha_{avg,i+1} p_{avg,i+1}) x_{i+1} = -\alpha_{avg,i} p_{avg,i+1} x_i + p_{avg,i+1} h_{i+1} \quad 3.2.6$$

Therefore,

$$x_{i+1} = \frac{-\alpha_{avg,i} p_{avg,i+1}}{(1 - \alpha_{avg,i+1} p_{avg,i+1})} x_i + \frac{p_{avg,i+1}}{(1 - \alpha_{avg,i+1} p_{avg,i+1})} h_{i+1} \quad 3.2.7$$

From equation 3.2.7, sensitivity analysis based on different α_{avg} (averaged passenger boarding time) and $p_{avg,i+1}$ (averaged passenger arrival rate) were performed to test the stability of the number of boarding passengers.

3.2.2 Model Simulation

The purpose of this simulation was to identify the behavior of buses at a stop. Specifically, sensitivity of total number of boarding passengers were investigated under the different sets of passenger arrival rate and boarding time. Multiple simulation runs were conducted by using the PC version of *MATLAB* software.

In order for the perturbation to be damped over successive buses, as suggested by Bell and Cowell, the passenger arrival rate should be less than half of the boarding rate. Simulation proved the above system stability condition. Another issue verified in the simulation was that when the number of boarding passengers equal to the multiplication of headway and passenger arrival rate ($x = ph$), the system reaches at the stable condition. It was also satisfied from the simulation result.

3.2.3 Simulation Results

3.2.3.1 Constant Passenger Arrival Rate and Passenger Boarding Rate

Figure 3.2.2 illustrates the stable behavior of boarding passengers due to the fact that passenger arrival rate (p=0.2 passenger/minute) is less than half of the boarding rate

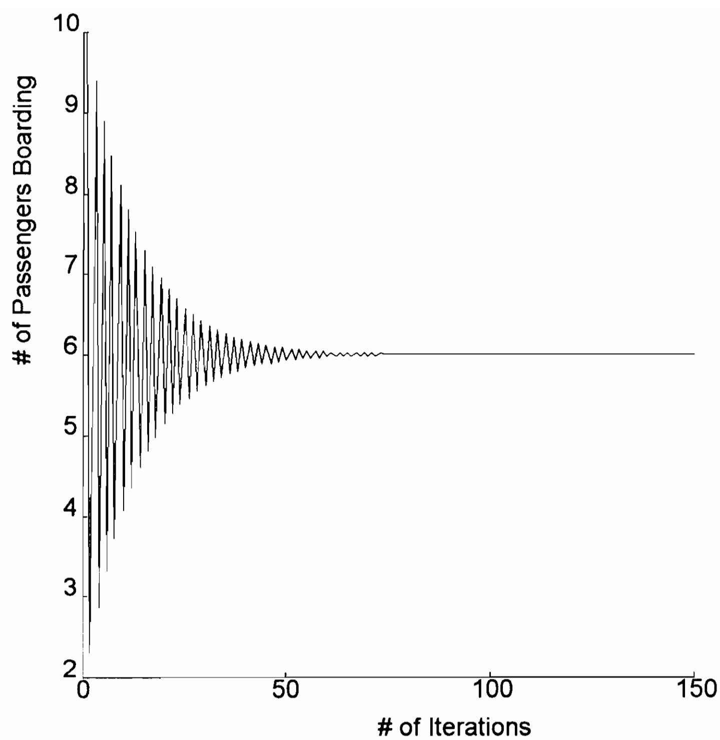


Figure 3.2.2 Number of Passenger Boarding (constant α and p, stable condition)

($1/\alpha=0.417$ passenger/minute). In the simulation, headway was assumed to be 30 minutes, thus, the converged number of passenger is equal to 6 (ph = 0.2*30). When the passenger arrival rate (p=0.2 passenger/minute) is not less than half of the boarding rate ($1/\alpha=0.377$ passenger/minute), system stability is perturbed after certain number of iterations. Figure 3.2.3 shows perturbed stability

3.2.3.2 Varying Passenger Arrival Rate and Passenger Boarding Time

In the next simulation, passenger arrival rate and boarding time were assumed to follow continuous functions. p = .1*(1+.1*sin(1.5*i)) and α = .001*(1+.001*sin(1.5*i)) were applied. The result is shown in Figure 3.2.4. In this case, passenger arrival rate is considerably smaller than the boarding rate, the amplitude of curve is small and number of passenger boarding is in stable condition. When p = .51*(1+.51*sin(1.5*i)) and α = .01*(1+.01*sin(1.5*i)) were applied, the result is illustrated in Figure 3.2.5. Passenger arrival rate is smaller than boarding rate, but it is not significantly smaller than the previous case. Thus, its amplitude is large but still it falls into the stable condition. However, when p = .51*(1+.51*sin(1.5*i)) and α = .90*(1+.90*sin(1.5*i)) were applied, perturbed number of boarding passengers was occurred. In this case, passenger arrival rate is not smaller than half of the boarding rate, thus perturbation occurred. Its result is shown in Figure 3.2.6.

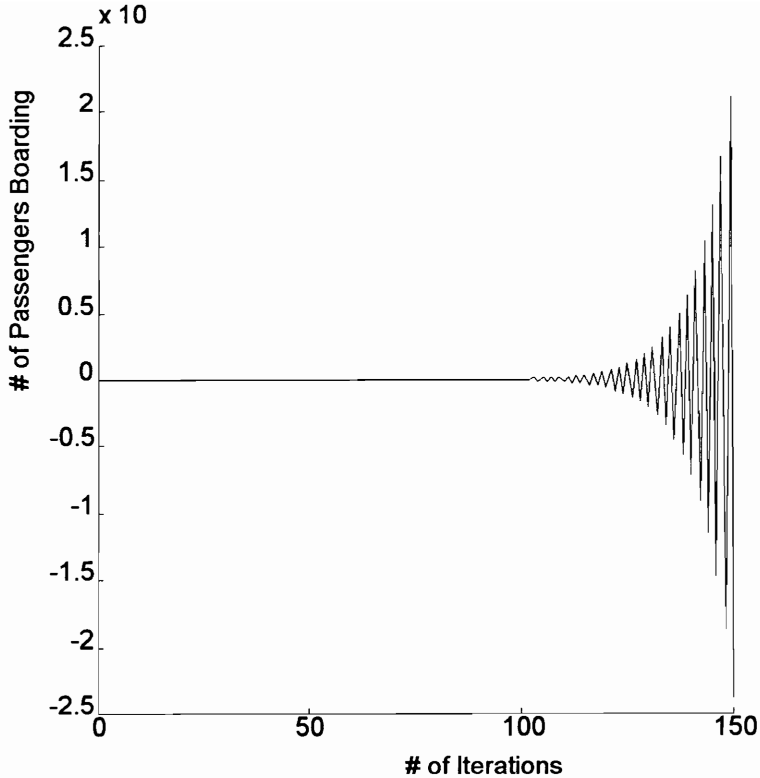


Figure 3.2.3 Number of Passenger Boarding (constant α and p, unstable condition)

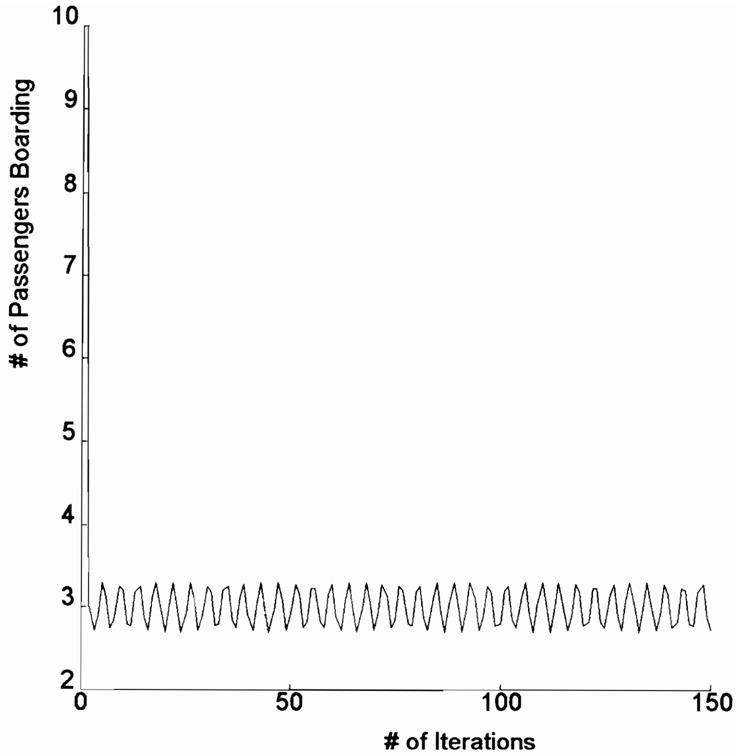


Figure 3.2.4 Number of Passenger Boarding (varying α and p, stable condition)

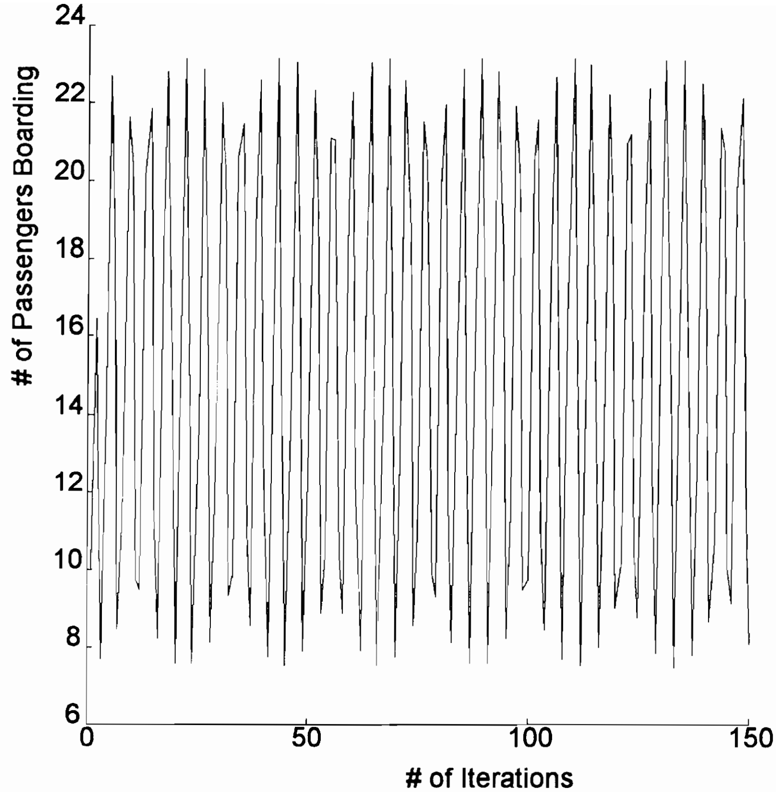


Figure 3.2.5 Number of Passenger Boarding (varying α and p, stable condition)

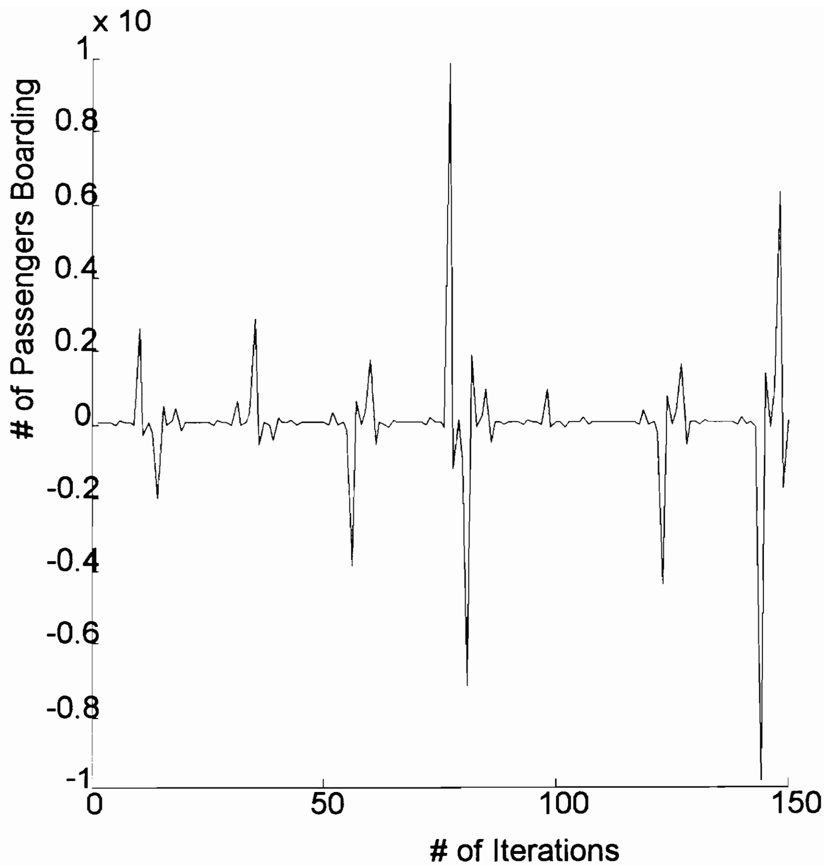


Figure 3.2.6 Number of Passenger Boarding (varying α and p, unstable condition)

3.3 MULTIPLE BUSES AT MULTIPLE BUS STOPS

3.3.1 Model Formulation

From Bell and Cowell's model, travel time between two locations including a single stop would be expressed as

$$t_i = c + \alpha x_i \tag{3.3.1}$$

where, c = departure time headway that excludes passenger loading time at a bus stop

αx_i = passenger loading time at a bus stop.

Since x_i is also defined by Bell and Cowell as

$$x_i = \frac{-\alpha p}{1-\alpha p} x_{i-1} + \frac{ph}{1-\alpha p}, \tag{3.3.2}$$

equation 3.3.2 can be replaced with x_i in equation 3.3.1. Therefore,

$$t_i = c + \alpha\left[\frac{-\alpha p}{1-\alpha p} x_{i-1} + \frac{ph}{1-\alpha p}\right] \tag{3.3.3}$$

Since $t_{i-1} = c + \alpha x_{i-1}$, equation 3.3.3 can be rewritten as

$$t_i = c + \alpha\left[\frac{-(t_{i-1} - c)p}{1 - \alpha p} + \frac{ph}{1 - \alpha p}\right]$$

$$= \frac{-\alpha p}{1 - \alpha p} t_{i-1} + \frac{c + \alpha ph}{1 - \alpha p}. \qquad 3.3.4$$

Based on equation 3.3.4, Bell and Cowell's model calibrates the travel time at a single stop. This equation describes the recursive way of approach in the travel time estimation. The current travel time between the stops rely on the most recent travel time.

Now, let's consider the case of travel time prediction between two locations that include two buses. According to the Bell and Cowell's model, we have to add two consecutive travel times in two stops to forecast bus arrival time for the second bus. However, even though current travel time is based on that of the previous one, it can not be simply summed up for forecasting the next bus. It is explained well in the following statements. If we add two travel times on a route, it can be expressed as

$$t_{i,1} + t_{i,2} = \frac{-\alpha p}{1 - \alpha p} t_{i-1,1} + \frac{c + \alpha p h_{i,1}}{1 - \alpha p} + \frac{-\alpha p}{1 - \alpha p} t_{i-1,2} + \frac{c + \alpha p h_{i,2}}{1 - \alpha p} \qquad 3.3.5$$

where, $t_{i,1}$ = time taken the ith bus to pass the first stop

$t_{i,2}$ = time taken the ith bus from the first stop to the second stop

$h_{i,1}$ = arrival time headway between i-1 and ith bus at the first stop

$h_{i,2}$ = arrival time headway between i-1 and i^th bus at the second stop

Equation 3.3.5, however, can not be expressed as one time variant equation. For example,

$$\alpha_1 x_1 + \alpha_2 x_2 = \alpha_3 (x_1 + x_2)$$

$$\alpha_3 = \frac{\alpha_1 x_1 + \alpha_2 x_2}{x_1 + x_2}$$

$$\alpha_3 = f(x_1, x_2) \qquad 3.3.6$$

from equation 3.3.6, α_3 is the function of two variables (x_1 and x_2). Therefore, travel time on multiple stops must be estimated separately in the stepwise manner, instead of summing them up for one route to calibrate predicted travel time. For this, segment by segment calibration method is introduced. The description of new approach in this research is as follows.

From Figure 3.3.1, estimated arrival time headway in the first stop will be expressed as

$$\hat{h}_{i,1} = \tau + c - \xi_{i-1,1} \qquad 3.3.7$$

where, $\hat{h}_{i,1}$ = estimated arrival time headway for bus i at stop no.1

τ = initial arrival time headway

$\xi_{i-1,1}$ = arrival time of previous bus (i-1) at stop no.1

Then travel times on the s^{th} stop will be estimated as

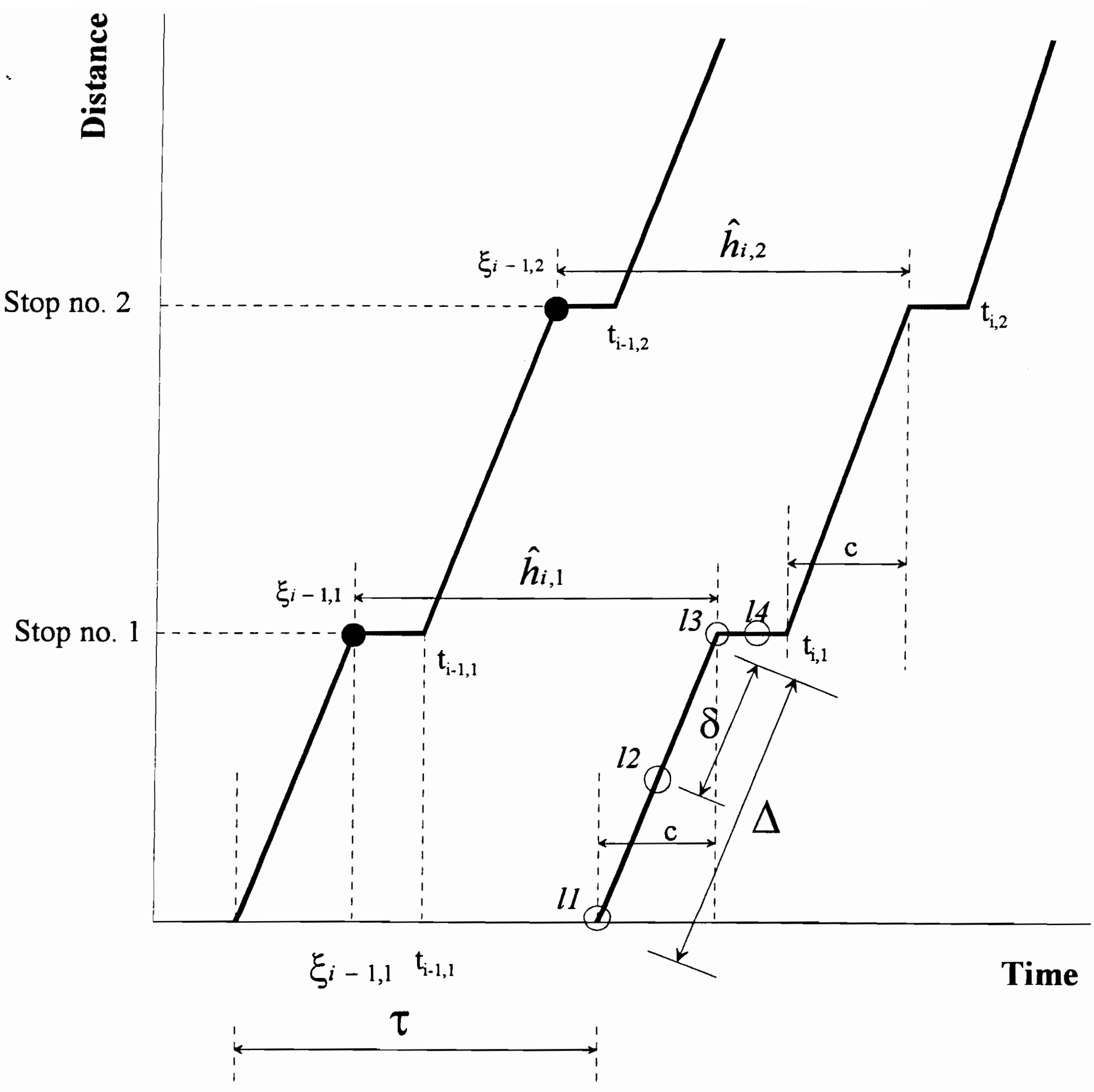


Figure 3.3.1 Time-Space Diagram for Multiple Stops

$$\hat{t}_{i,1} = \frac{-\alpha p}{1-\alpha p} t_{i-1,1} + \frac{c + \alpha p \hat{h}_{i,1}}{1-\alpha p} \quad 3.3.8$$

$$\hat{h}_{i,2} = t_{i,1} + c - \xi_{i-1,2} \quad 3.3.9$$

$$\hat{t}_{i,2} = \frac{-\alpha p}{1-\alpha p} t_{i-1,2} + \frac{c + \alpha p \hat{h}_{i,2}}{1-\alpha p} \quad 3.3.10$$

...

$$\hat{h}_{i,s} = t_{i,s-1} + c - \xi_{i-1,s} \quad 3.3.11$$

$$\hat{t}_{i,s} = \frac{-\alpha p}{1-\alpha p} t_{i-1,s} + \frac{c + \alpha p \hat{h}_{i,s}}{1-\alpha p} \quad 3.3.12$$

where, i = bus number

s = stop number

When a person in the n[th] stop ahead requests arrival time of next bus, there are 4 possible cases of the location of bus. These four cases are designated as white circles in Figure 3.3.1. When the bus is located in l_2, approximation of time estimation would be made, i.e., $c \times \frac{\delta}{\Delta}$. Then, the $\sum_{i=s=1}^{n} \hat{t}_{i,s}$ will be calculated.

3.3.2 Model Simulation

The formulation of simulation model, based on Newell and Potts's model, started from the time independence of passenger arrival rate and bus boarding time. Thus, number of passenger boarding on bus m at stop n equals to the multiplication of arrival time and arrival rate, and the multiplication of loading time and loading rate. This statement leads to the following result.

$$k_{mn} = \frac{loading\ time}{passenger\ arrival\ time} = \frac{passenger\ arrival\ rate}{loading\ rate} \qquad 3.3.13$$

In Figure 3.3.2, m denotes bus number (positive integer value), n stands for stop number (positive integer value), t_{mn} is the time when the bus m leaves the stop n, τ denotes the headway of bus, and T_{mn} is the travel time of bus m between stop n and previous stop n-1.

According to equation 3.3.13 and Figure 3.3.2, the following derivation can be made.

$$t_{mn} - t_{mn-1} - T_{mn} = k_{mn}(t_{mn} - t_{m-1n}) \qquad 3.3.14$$

$$t_{mn} = \frac{t_{mn-1}}{1-k_{mn}} - \frac{k_{mn}}{1-k_{mn}} t_{m-1n} + \frac{T_{mn}}{1-k_{mn}} \qquad 3.3.15$$

Simulation of dynamics of bus behavior at multiple stops was based on equation 3.3.15. Since t_{m-1n} is assumed to be known from the AVL probe vehicle data, and if we assume loading time and time taken from i^{th} to $i+1^{th}$ stop, simulation can be conducted. First, steady state simulation (constant loading time and k) was performed for four stops with twenty buses. Figure 3.3.3 illustrates dynamics of bus behaviors on steady state (constant α and k). Then, simulation based on non-steady were investigated with the same assumptions except state varying α and k by applying random numbers. Figure 3.3.4 illustrates dynamics of bus behavior on non-steady state.

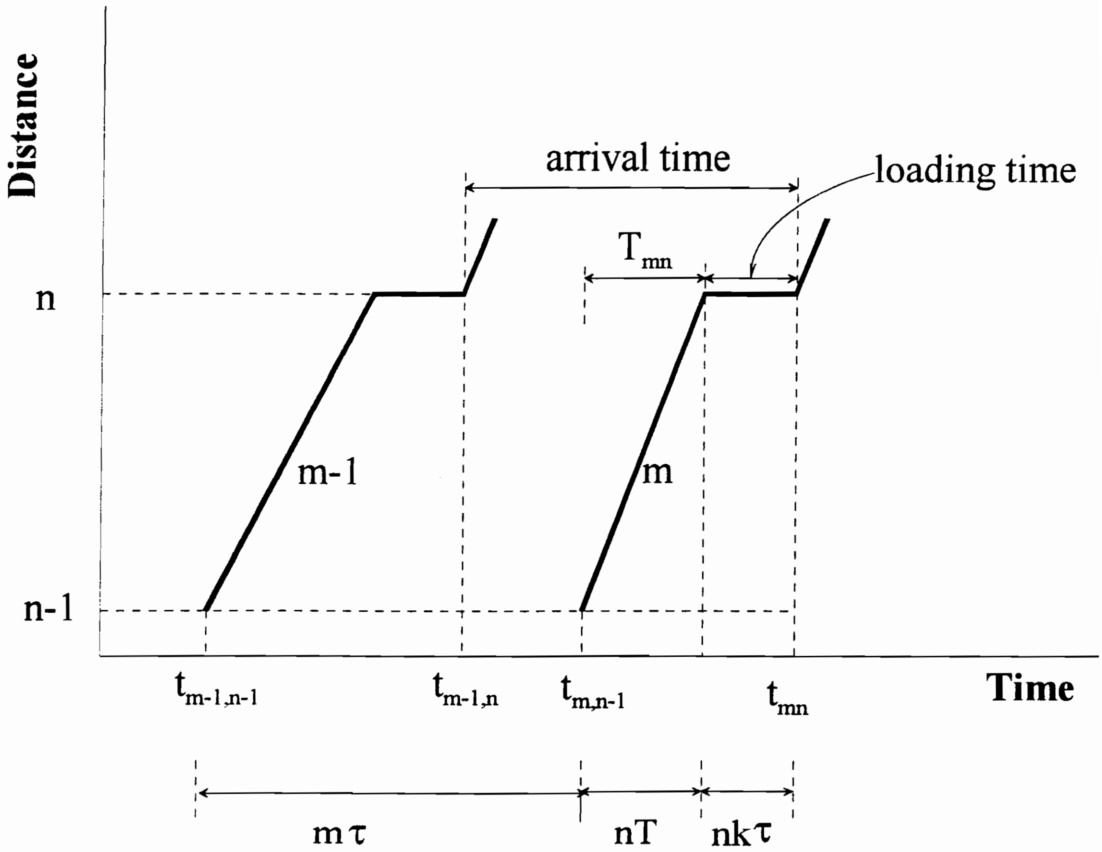


Figure 3.3.2 Time-Space Diagram for Simulation

Source: Newell and Pott's, "Maintaining a Bus Schedule," Paper page 390.

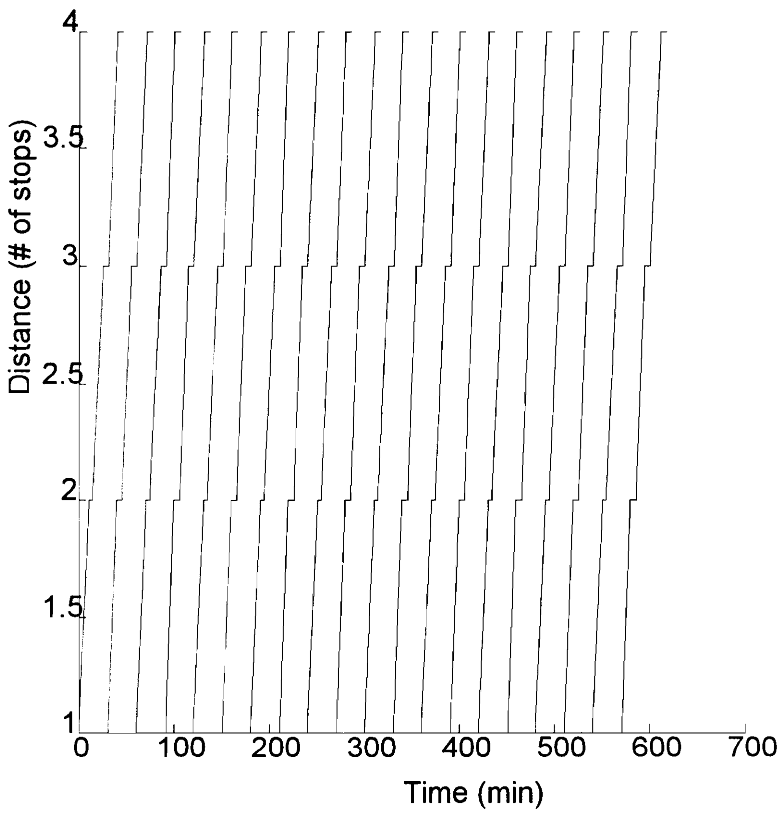


Figure 3.3.3 Dynamics of Bus Behavior on Steady State

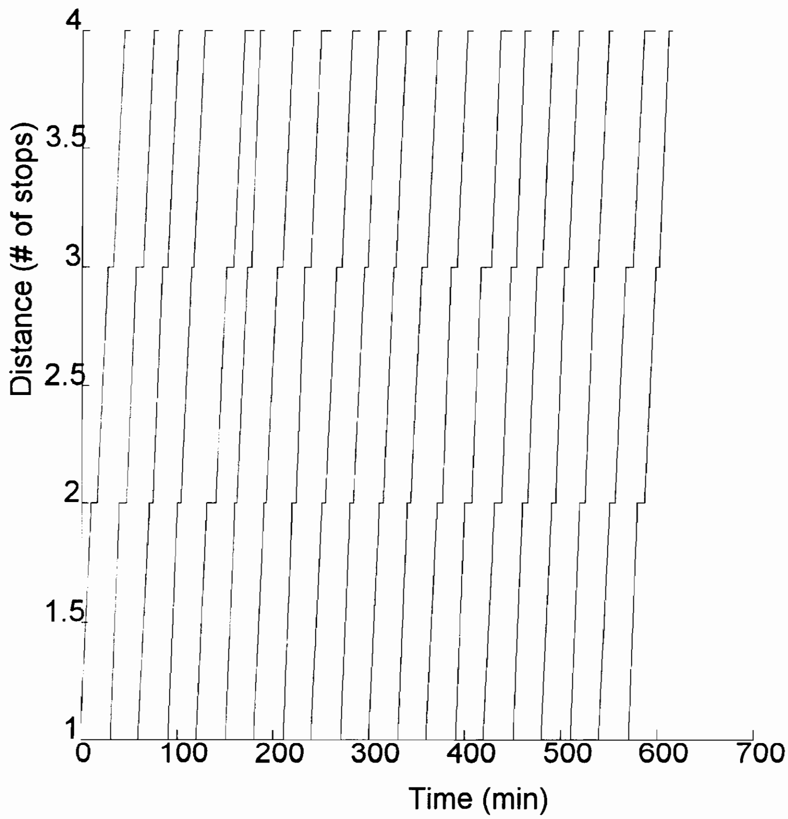


Figure 3.3.4 Dynamics of Bus Behavior on Non-Steady State

In the non-steady state condition, when the loading time of bus gets longer due to delay, then the loading time of following bus at the same stop gets shorter. On the other hand, since the previous bus left early because of less passengers, the following bus at the same stop stay longer, and so on. Because of this fact, perturbation is occurred in the system. The varying loading time of non-steady state is descriptively shown in table 3.3.1.

As simulations of bus behavior were completed, next step investigated was the prediction of bus arrival time at stops. Least square parameter estimation algorithm was adopted for identification of the system.

Table 3.3.1 Non-Steady State Loading Time

No. of Buses	*Loading Time at First Stop*	*Loading Time at Second Stop*	*Loading Time at Third Stop*	*Loading Time at Forth Stop*
1	0	4.0948	3.2352	6.3943
2	0	3.1012	3.2077	5.8573
3	0	3.0400	5.7365	3.4025
4	0	8.3906	8.2457	13.8082
5	0	5.5854	2.544	2.3059
6	0	7.7199	11.6253	13.1391
7	0	2.7395	4.3987	0.9239
8	0	9.1491	13.6022	9.3529
9	0	8.7883	5.1909	3.9725
10	0	2.7438	5.4893	7.7680
11	0	6.2809	9.5145	7.2424
12	0	4.3502	3.8504	2.3321
13	0	9.9517	12.4607	5.2881
14	0	5.0404	3.2215	2.0689
15	0	6.1930	5.0543	4.1420
16	0	3.0860	5.4645	4.9753
17	0	10.6106	8.3153	8.4602
18	0	2.3969	4.5456	3.2545
19	0	3.9459	3.1066	8.3737
20	0	7.6035	10.0346	10.4576

3.3.3 Arrival Time Estimation: A Prototype

3.3.3.1 Parameter Adaptation Algorithm Approach

The underlying assumption in parameter adaptation algorithm is that system structure is known, but parameter values are time varying (unknown). In general, two approaches, off-line and on-line computational methods, are utilized in the system identification in order to identify the parameters. Since parameters are varying during the bus operation, application of on-line estimation (recursive identification, real-time identification) can be appropriate for this research.

Least square algorithm is the most popular parameter estimation algorithms. The function of adaptation mechanism is as follows:

$$y(k) = -ay(k-1) + bu(k) = \theta^T \phi(k-1) \qquad 3.3.16$$

where, $\theta^T = [-a \;\; b]$ and $\phi(k-1) = \begin{bmatrix} y(k-1) \\ u(k) \end{bmatrix}$

Since time taken from *m-1* to m^{th} stop was formulated as

$$t_m = \frac{-\alpha p}{1-\alpha p} t_{m-1} + \frac{c + \alpha p h_m}{1-\alpha p}, \qquad 3.3.17$$

it can be rewritten as

$$t_m = -a t_{m-1} + b h_m + d \qquad 3.3.18$$

where, $\frac{-\alpha p}{1-\alpha p} = -a$, $\frac{\alpha p}{1-\alpha p} = b$, and $\frac{c}{1-\alpha p} = d$.

Thus,

$$\theta^T = [-a \quad b \quad d] \text{ and } \phi(k-1) = \begin{bmatrix} t_{m-1} \\ h_m \\ 1 \end{bmatrix}.$$

The estimated output would be identified by an estimate of parameter vector $\hat{\theta}$. It can be written as

$$\hat{y}(k) = \hat{\theta}^T(k)\phi(k-1) \qquad 3.3.19$$

The estimation output error would be

$$e(k) = y(k) - \hat{y}(k) \qquad 3.3.20$$

Least square estimate (LSE) minimizes the summation of the squared prediction errors, i.e,

$$E = \sum_{k=1}^{n}\left[y(k) - \hat{\theta}^T(n)\phi(k-1)\right]^2 \qquad 3.3.21$$

The LSE can be found by setting the partial derivative of E with respect to $\hat{\theta}$ to zero.

$$\frac{\partial E}{\partial \hat{\theta}} = -2\sum_{k=1}^{n}\left[y(k) - \hat{\theta}^T(n)\phi(k-1)\right]\phi(k-1) \qquad 3.3.22$$

Therefore,

$$\hat{\theta}(k) = F(k)\sum_{k=1}^{n}\phi(k-1)y(k) \qquad 3.3.23$$

where, $F(k) = \left[\sum_{k=1}^{n} \phi(k-1)\phi^T(k-1) \right]^{-1}$ 3.3.24

$\hat{\theta}(k+1)$ in the recursive form will be

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \textit{correction term} \quad 3.3.25$$

Note that

$$F^{-1}(k+1) = F^{-1}(k) + \phi(k)\phi^T(k) \quad 3.3.26$$

and

$$\sum_{k=1}^{n} \phi(k-1)y(k) = \hat{\theta}(k)F^{-1}(k) \quad 3.3.27$$

Therefore,

$$\hat{\theta}(k+1) = F(k+1)[\{F^{-1}(k+1) - \phi(k)\phi^T(k)\}\hat{\theta}(k) + \phi(k)y(k+1)]$$

$$= F(k+1)F^{-1}(k+1)\hat{\theta}(k) + F(k+1)\phi(k)[y(k+1) - \hat{\theta}^T(k)\phi(k)]$$

$$= \hat{\theta}(k) + F(k+1)\phi(k)[y(k+1) - \hat{\theta}^T(k)\phi(k)] \quad 3.3.28$$

Since $\hat{\theta}^T(k)\phi(k)$ represents a priori predicted output based on the parameter estimate vector at time k, it can be replaced with

$$\hat{y}^0(k+1) = \hat{\theta}^T(k)\phi(k) \text{ and } \hat{y}(k+1) = \hat{\theta}^T(k+1)\phi(k). \quad 3.3.29$$

Prediction errors can be expressed as

$$e^0(k+1) = y(k+1) - \hat{y}^0(k+1) \text{ and } e(k+1) = y(k+1) - \hat{y}(k+1). \quad 3.3.30$$

Therefore,

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)e^{0}(k+1) \qquad 3.3.31$$

Equation 3.3.28 is the recursive form of the parameter adaptation algorithm.

From the matrix inversion lemma, gain term would be expressed as

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^{T}(k)F(k)}{1+\phi^{T}(k)F(k)\phi(k)} \qquad 3.3.32$$

Multiplication of $\phi(k)$ in Equation 3.3.29 will yield

$$F(k+1)\phi(k) = \frac{F(k)\phi(k)}{1+\phi^{T}(k)F(k)\phi(k)} \qquad 3.3.33$$

Therefore, parameter updating law will be

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \frac{F(k)\phi(k)}{1+\phi^{T}(k)F(k)\phi(k)}e^{0}(k+1) \qquad 3.3.34$$

Relationship between e^{0} and e can be identified by manipulating Equation 3.3.28 and 3.3.29.

$$e(k+1) = y(k+1) - \hat{\theta}^{T}(k+1)\phi(k) \qquad 3.3.35$$

$$= y(k+1) - [\hat{\theta}(k) + \frac{F(k)\phi(k)}{1+\phi^{T}(k)F(k)\phi(k)}e^{0}(k+1)]^{T}\phi(k)$$

$$= y(k+1) - \hat{\theta}^{T}(k)\phi(k) - \{[F(k)\phi(k)e^{0}(k+1)]^{T}[1+\phi^{T}(k)F(k)\phi(k)]^{-1T}\phi(k)\}$$

$$= e^{0}(k+1) - \frac{e^{0T}(k+1)\phi^{T}(k)F^{T}(k)\phi(k)}{1+\phi^{T}(k)F(k)\phi(k)}$$

$$= \frac{e^0(k+1)}{1+\phi^T(k)F(k)\phi(k)}$$

3.3.3.2 Forgetting Factor

The vector θ was assumed to be time invariant in the derivation of least square formula. However, in the time dependent parameter identification problem the time decreasing adaptation gain factor, F(k), is not suitable because of its weaker and weaker adaptation functionality. Due to this reason, forgetting factor, λ, is introduced to forget the identifier the past data i.e., assigning less weight to older measurements that are no longer representative for the system). Therefore, new performance index will be

$$E = \sum_{k=1}^{n} \lambda^{n-k} [y(k) - \hat{\theta}^T(n)\phi(k-1)]^2 \qquad 3.3.36$$

where, $0 < \lambda < 1$

From adopting Landau and Silveira's general formula, F(k) can be represented as

$$F^{-1}(k+1) = \lambda_1(k)F^{-1}(k) + \lambda_2(k)\phi(k)\phi^T(k) \qquad 3.3.37$$

where, $F(0) > 0$,

$0 < \lambda_1(k) \leq 1$ and $0 < \lambda_2(k) \leq 2$.

According to the matrix inversion lemma, *F(k)* will be updated as

$$F(k+1) = \frac{1}{\lambda 1(k)}[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\lambda 1(k)/\lambda 2(k) + \phi^T(k)F(k)\phi(k)}] \qquad 3.3.38$$

3.3.3.3 Parameter Convergence

In system identification asymptotic stability of parameter adaptation algorithm should satisfy $\lim_{k\to\infty} \hat{\theta}(k) = \theta$ as well as $\lim_{k\to\infty} e(k+1) = 0$.

$$e(k+1) = y(k+1) - \hat{y}(k+1)$$

$$= -\tilde{\theta}^T (k+1)\phi(k)$$

where, $\tilde{\theta} = \hat{\theta} - \theta$

It implies that $\tilde{\theta}(k+1)$ and $\phi(k)$ should be asymptotically orthogonal for the asymptotic stability of parameter adaptation algorithm (PAA). Also, stability of PAA implies that the following equation is asymptotically satisfied:

$$[a - \hat{a}(k)]y(k-1) + [b - \hat{b}(k)]u = 0 \qquad 3.3.39$$

where, $\hat{a} = \lim_{k\to\infty} \{\hat{a}(k)\}$,

$\hat{b} = \lim_{k\to\infty} \{\hat{b}(k)\}$,

$\tilde{a} = a - \hat{a}$ and $\tilde{b} = b - \hat{b}$.

3.3.3.4 Simulation Results

Simulations to test *least square* parameter estimation of bus arrival time prediction consist of two scenarios - constant actual parameters and varying actual parameters. Also, nonsteady-state loading time was assumed for each bus stop. Bus arrival time estimations, parameter convergences, estimation errors for each stops were the main outputs to be analyzed in this simulation. Simulation source code is attached to Appendix A.

1) Constant Actual Parameter

The first scenario was focused on *the constant actual parameter*. Actual parameter was calculated under the assumption that the travel time (t) is an equilibrium value. Hence, equation 3.3.12 will be expressed as

$$t = \frac{-\alpha p}{1-\alpha p} t + \frac{c+\alpha ph}{1-\alpha p} \qquad 3.3.40$$

When equation 3.3.16 is represented according to αp, then

$$\alpha p = \frac{t-c}{h}. \qquad 3.3.41$$

Since average total travel time (t), crusing travel time (c), and headway (h) were assumed as 10, 5, and 30 minutes, respectively, initial actual parameter a, b, and d can be assumed to be calculated as:

$$a = \frac{-\alpha p}{1-\alpha p} = -\frac{1/6}{5/6} = -.2$$

$$b = \frac{\alpha p}{1-\alpha p} = \frac{1/6}{5/6} = .2$$

$$d = \frac{c}{1-\alpha p} = \frac{10}{5/6} = 12\,.$$

First Simulation

Estimation of bus arrival time was simulated. In the simulation plot, x axis represents the clock time (sequential time) and y axis represents absolute time. Twenty buses with four bus stops were simulated so that estimation of arrival time can be predicted for nineteen buses (arrival time of the 1st bus was assumed to be known) at three stops (2nd, 3rd, and 4th stops). Hence, total of nineteen figures could be obtained. Figure 3.3.5 illustrates estimated arrival time of the second bus. In the figure, plot in the top illustrates the arrival time of bus at the second stop, plot in the middle shows arrival time at the third stop, and the plot in the bottom shows arrival time at the forth stop.

The bus was assumed to be located at zero time point (i.e., first bus at the first stop). Hence, estimated arrival time was predicted as 40 minutes (headway 30 minutes + travel time 10 minutes) in absolute time in the top plot in Figure 3.3.5. The plot in the

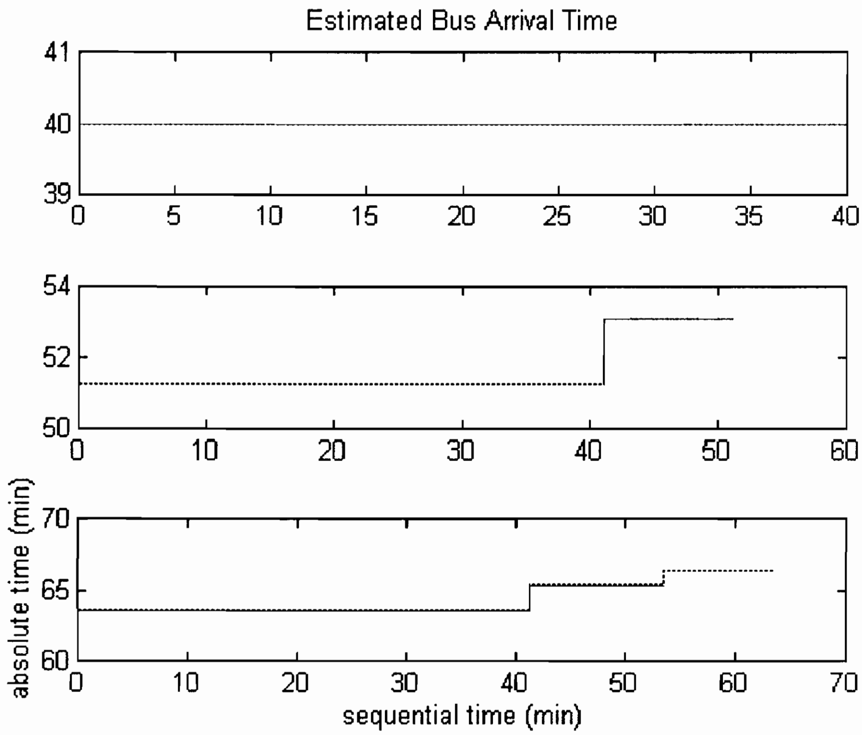


Figure 3.3.5 Estimated Arrival Time of Second Bus (constant actual parameter)

middle, the first hop was made due to acquired information that the time when the bus left the second stop (i.e., predictor is updated). In the bottom plot, two hops were made because of two information obtained from previous two buses (i.e., updated actual travel time information can be obtained as soon as the bus left second and third stop). As the more buses passing through the same bus stops, more accurate bus arrival time can be predicted for the stops.

Second Simulation

The estimated parameters were plotted against constant actual parameter. Three parameters (a, b, and d) are associated in each stops. Hence, total of nine (excluding the first stop) plots can be obtained. Figure 3.3.6 to 3.3.8 illustrate convergence of estimated parameter at the second stop. As it is illustrated in the figures, parameter errors tend to converge to zero with some error rates. In the overall parameter convergence rate, however, the results are acceptable.

Third Simulation

Estimation errors (subtraction from estimated parameter to actual parameter) were plotted for each stop against number of buses. Figure 3.3.9 shows the plotting of estimation errors for all the stops. Even though parameter errors do not converge to zero, estimation errors do converge to the zero as bus goes on. It informs us that least square estimator performs well for predictions of bus arrival time.

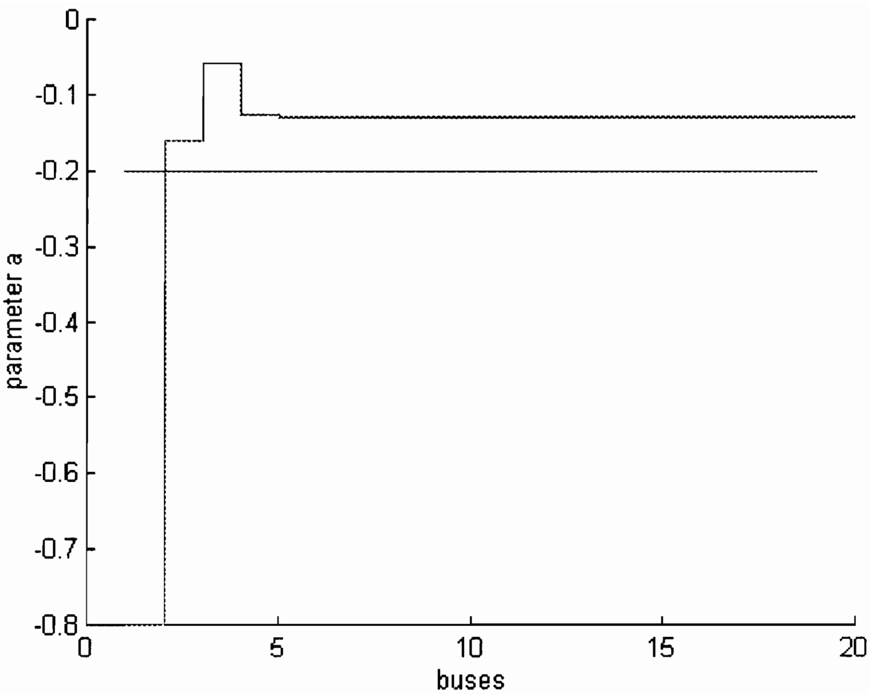


Figure 3.3.6 Convergence of Estimated Parameter **a** at the Second Stop
(constant actual parameter)

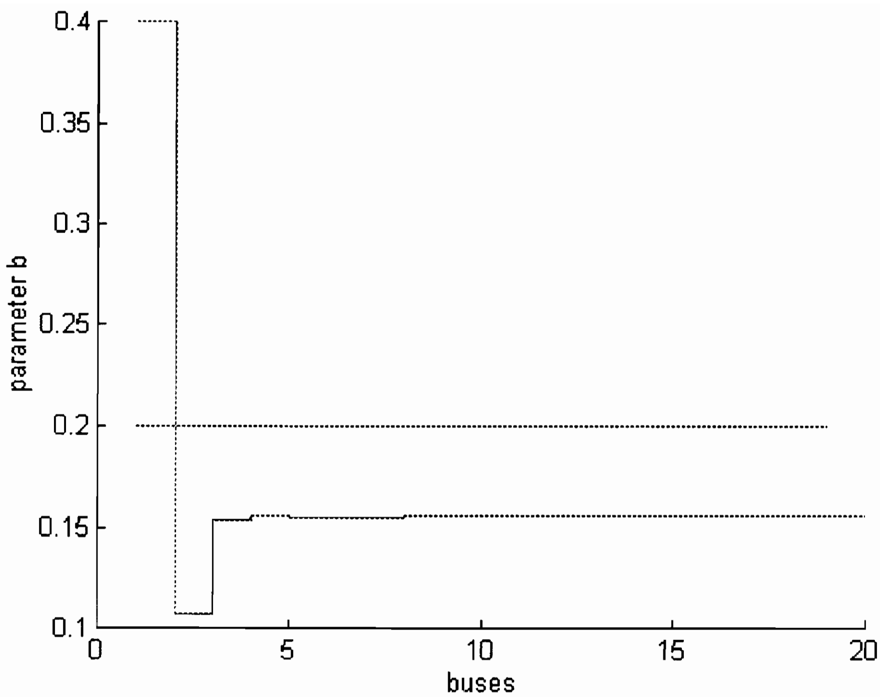


Figure 3.3.7 Convergence of Estimated Parameter **b** at the Second Stop
(constant actual parameter)

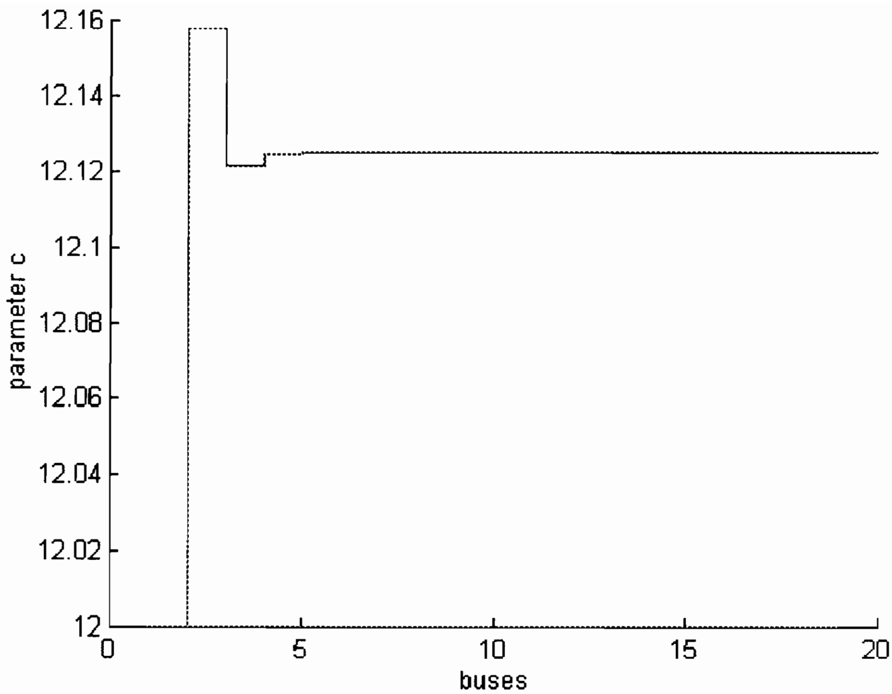


Figure 3.3.8 Convergence of Estimated Parameter **d** at the Second Stop
(constant actual parameter)

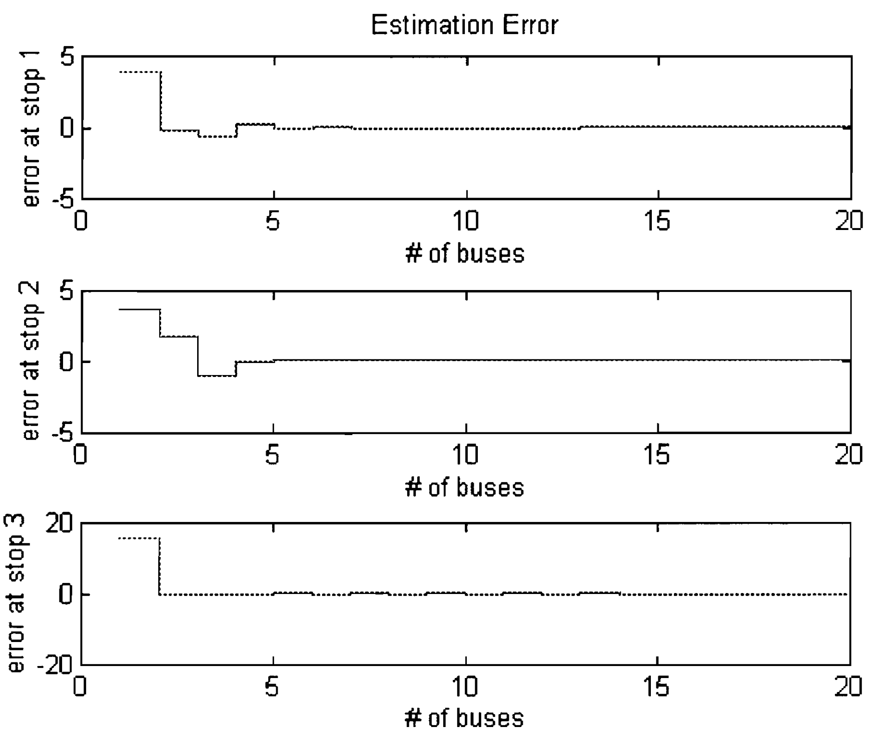


Figure 3.3.9 Estimation Errors for All the Stops (constant actual parameter)

2) Varying Actual Parameter

Actual parameters *a, b,* and *d* were assumed to be following continuous functions.

$$a = -.2+.05*sin(.13*(i-1))$$

$$b = .2+.05*sin(.13*(i-1))$$

$$d = 12+.75*sin(.13*(i-1))$$

where, i = number of simulating buses

First Simulation

Estimated bus arrival time were plotted. All the assumptions were the same as previous conditions (constant actual parameters) except the number of simulated buses (i.e., 100 buses were simulated in this case, hence 99 figures were obtained). All the explanations are equivalent to the previous case. Figure 3.3.10 shows the estimated travel time for the second bus.

Second Simulation

Parameter convergence was plotted. Since actual parameters are changing, convergence rate of estimated parameters to the actual parameters were slower than the previous case. However, least square estimator updated its parameters fairly good enough to be accepted. Figure 3.3.11 to 3.3.13 show the parameter errors at the second stop.

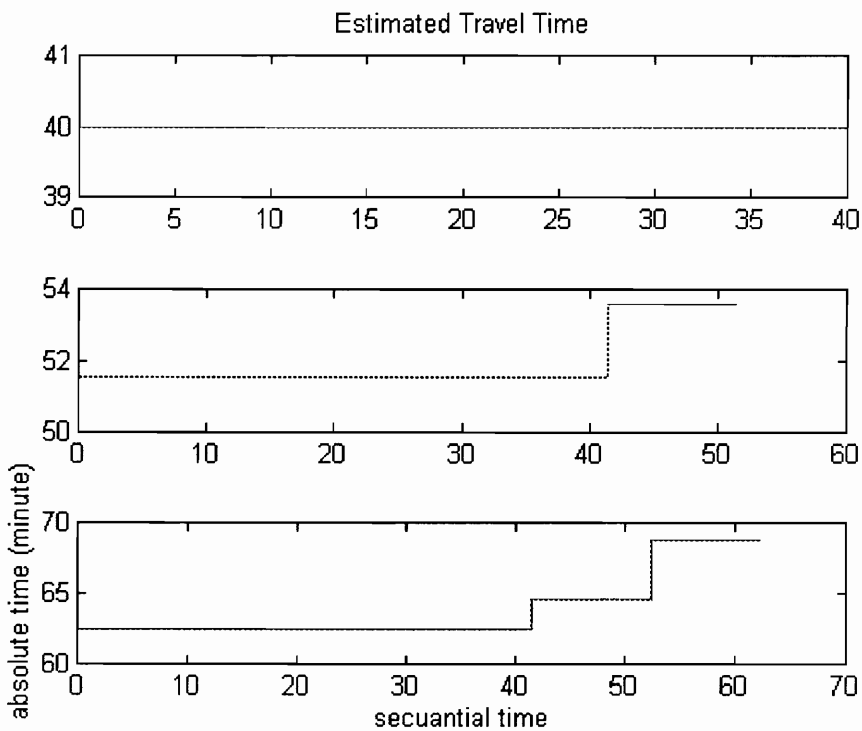


Figure 3.3.10 Estimated Travel Time for the Second Bus (varying actual parameter)

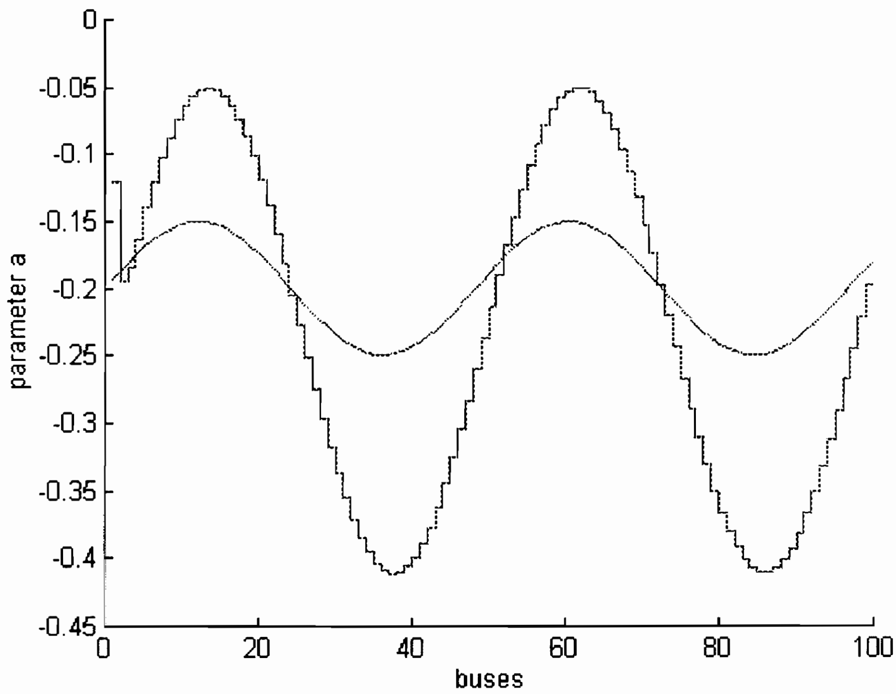


Figure 3.3.11 Convergence of Estimated Parameter **a** at the Second Stop
(varying actual parameter a)

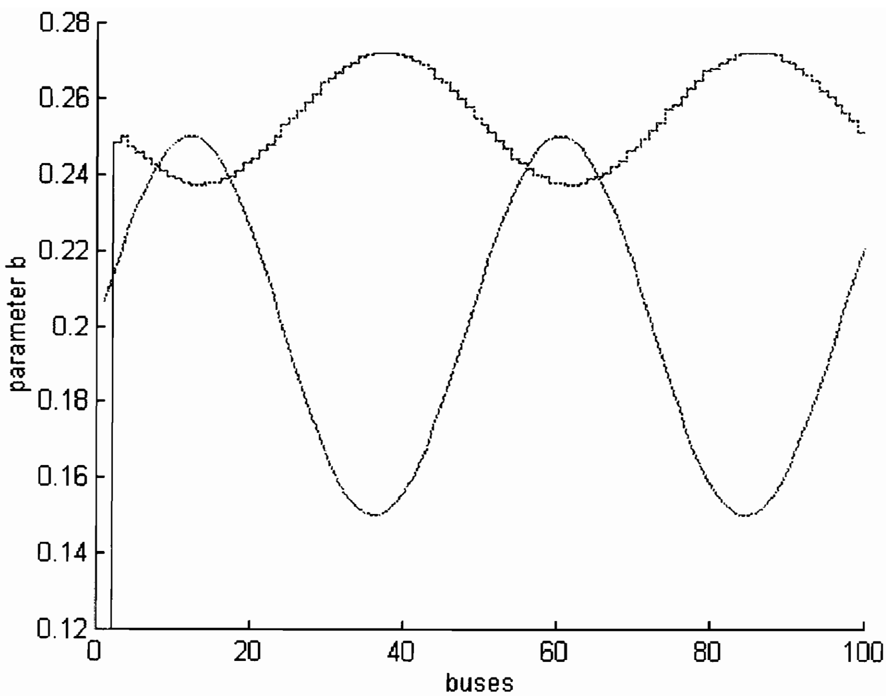


Figure 3.3.12 Convergence of Estimated Parameter **b** at the Second Stop
(varying actual parameter b)

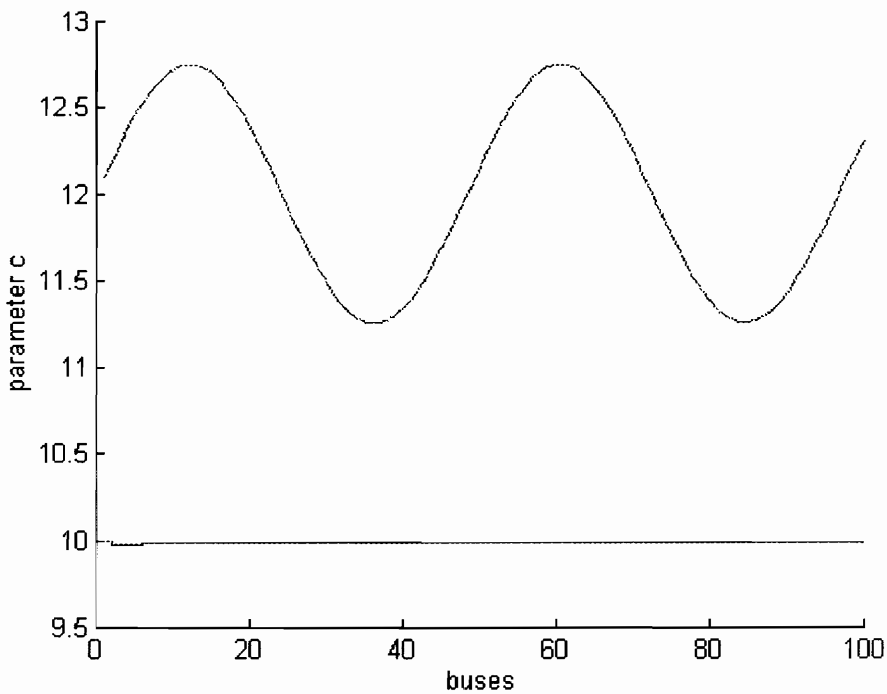


Figure 3.3.13 Convergence of Estimated Parameter **d** at the Second Stop
(varying actual parameter d)

Third Simulation

Estimation errors were plotted for each stop. Estimation error also converge to the zero with the error margin of 3 minutes for three stops. Its results can be acceptable. Figure 3.3.14 to 3.3.16 illustrate estimation error.

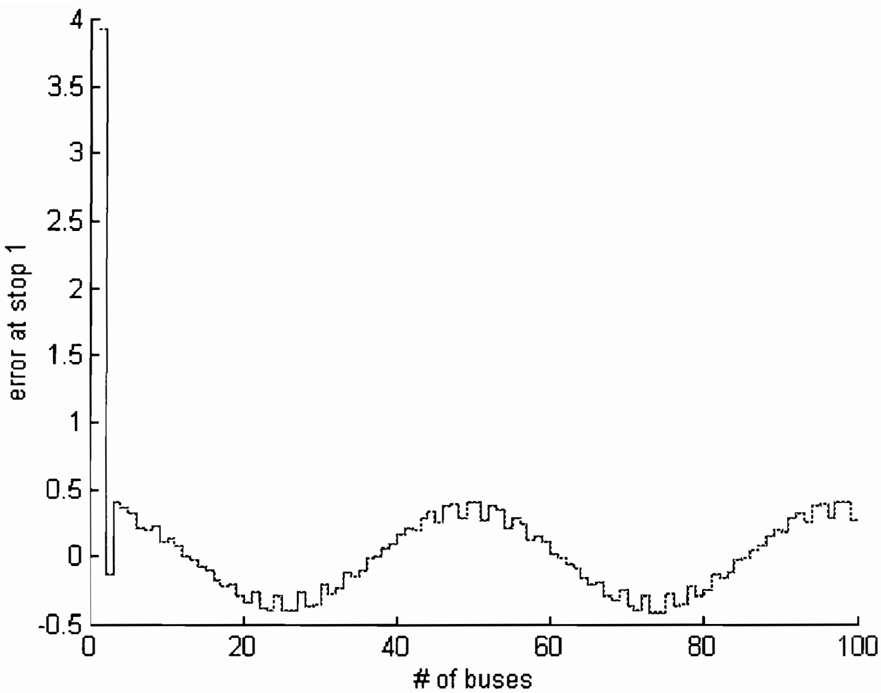


Figure 3.3.14 Estimation Error at the Second Stop (varying actual parameter)

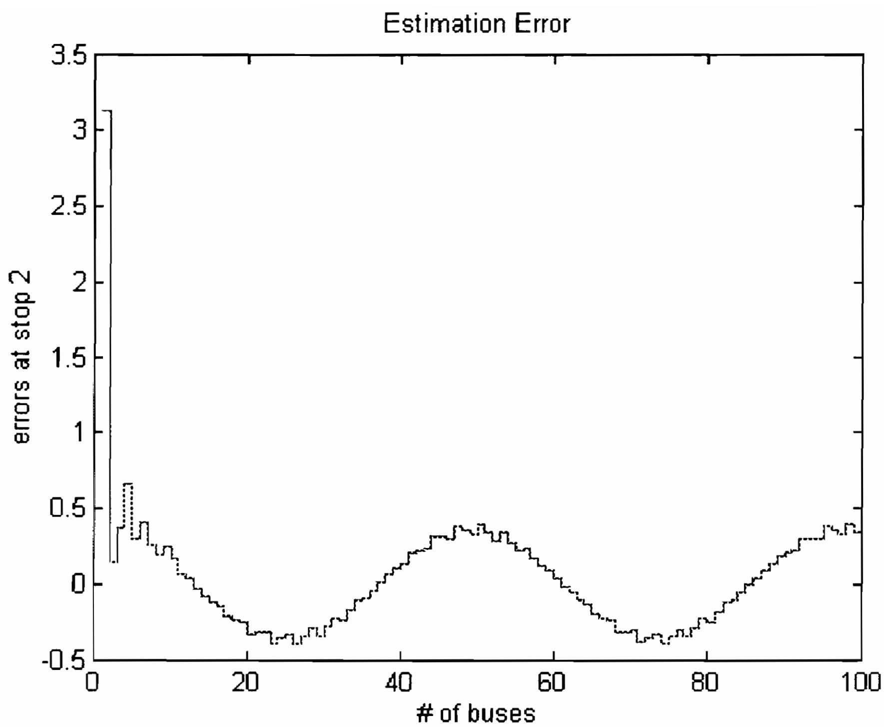


Figure 3.3.15 Estimation Error at the Third Stop (varying actual parameter)

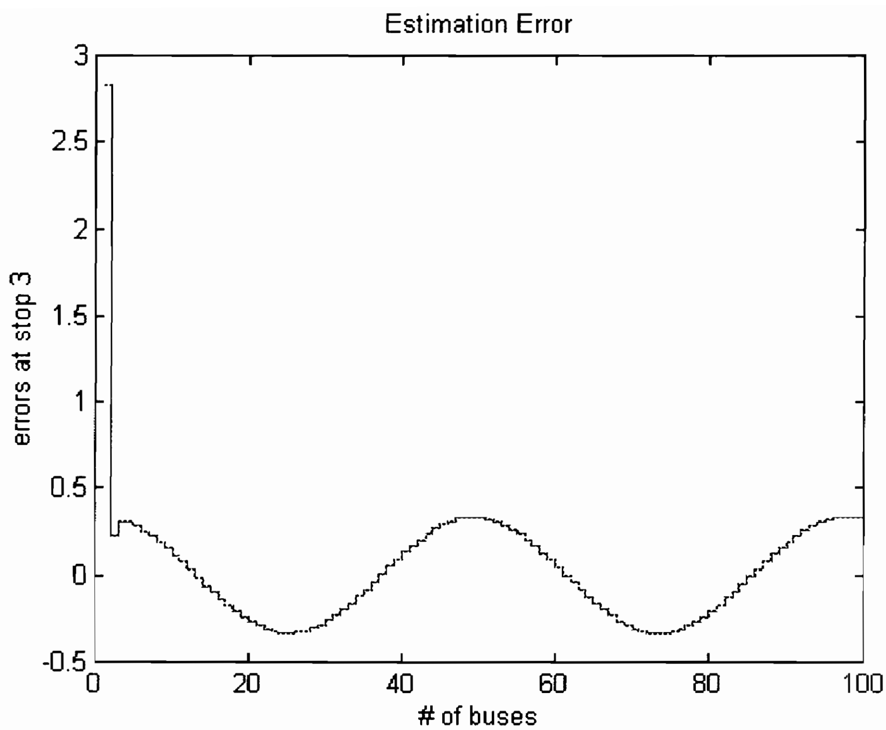


Figure 3.3.16 Estimation Error at the Forth Stop (varying actual parameter)

CHAPTER IV DYNAMIC LINK TRAVEL TIME FUNCTION

4.1 INTRODUCTION

The conventional definition of a link on arterial roads can be considered as a section of roadway from the stop line at upstream intersection to the stop line at immediate downstream intersection. In this study, however, travel time considering intersection delay is based upon the division of the conventional link into two sublinks (uncongested and congested link) as illustrated in Figure 4.1.1. The selection of border of two regions in Figure 4.1.1 is flexible, and determined dynamically by the flow condition and, therefore, the number of vehicle in the queue.

Since the link is further devided into two, the conventional definition of flow (q), which associated in a certain link is also adjusted more detail than the conventional definition of flow at a certain link. Its result makes future arterial study more reliable and concrete in many aspects, specifically in dynamic flow estimation, travel time estimation, etc.

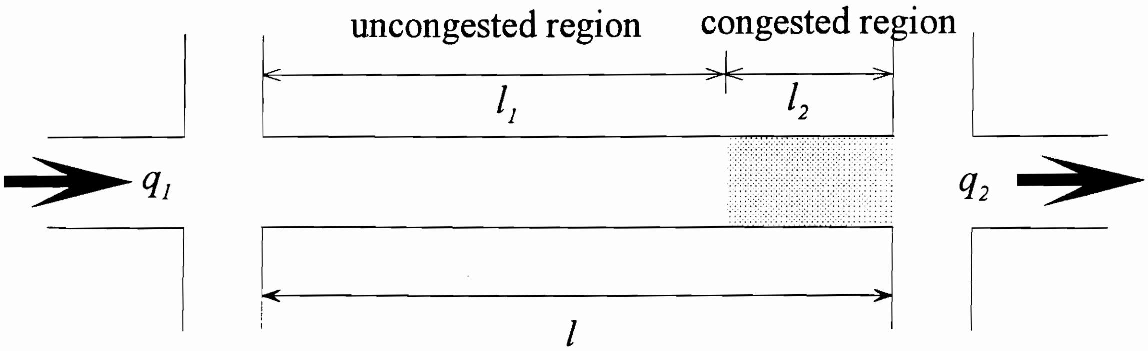


Figure 4.1.1 Link Division

Eventually, four different flows can be measured in a link - inflow, flow in uncongested region, flow in congested region, and outflow, respectively. The application of newly defined flows will be discussed in the following sections. In this study, travel time of congested and uncongested regions are estimated individually and summed for a link travel time, and these link travel times are summed again for a segment travel time.

4.2 FORMULATION OF TRAVEL TIME IN THE UNCONGESTED REGION

For uncongested region, from the density-speed relationship of Greenshield model, speed can be expressed as

$$v = v_f(1 - \frac{k}{k_j}) \qquad 4.2.1$$

where, v = speed

v_f = free flow speed

k = density

k_j = jam density

Equation 4.2.1 can be rewritten as

$$k = k_j(1 - \frac{v}{v_f}) \qquad 4.2.2$$

In addition, density (k) in the uncongested region can be expressed as

$$k = \frac{n - r}{l_1} \qquad 4.2.3$$

where, n = number of vehicle in a link at certain time

r = number of vehicle in the queue

l_1 = link length of uncongested region

In this region, basic concepts on vehicle kinematics are applied to obtain more realistic travel time function. Since this stage of research is focused on the estimation of bus arrival time, the effect of kinematical behaviors of buses into the total travel time would be significant in determining the accurate travel time estimation - especially, due to the facts that not only multiple numbers of intervening intersections where buses have to stop in red phase of signal cycle, but also frequent bus stops are main sources of causing delays in the urban transportation networks. Thus, frequent acceleration and deceleration maneuverings are unavoidable in the operation of buses in such a system.

For the general case, for instance, a vehicle which stopped at the upstream intersection would be discharged at the green signal phase of that intersection. The vehicle will accelerate, then maintain the cruise speed for certain time period, and start decelerating at the downstream intersection. Therefore, simply three regimes of vehicle profile can be obtained in the time and speed domain as illustrated in the Figure 4.2.1

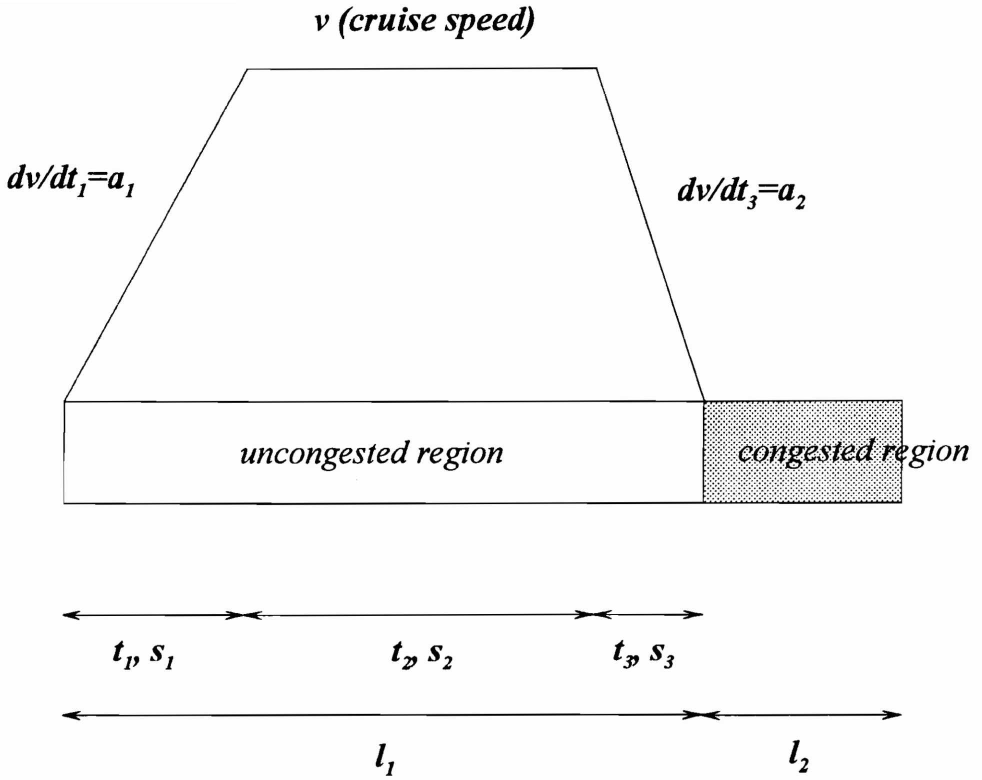


Figure 4.2.1 Vehicle Speed Profile in the Uncongested Region

Since

$$v = v_0 + at\,, \qquad 4.2.4$$

where, v = speed

v_0 = initial speed

From Equation 4.2.4, when $v_0 = 0$

$$t_1 = \frac{v}{a_1} \qquad 4.2.5$$

where, t_1 = acceleration time

a_1 = acceleration rate

and

$$t_3 = \frac{v}{a_2} \qquad 4.2.6$$

where, t_3 = deceleration time

a_2 = deceleration rate

From the relationship of distance, acceleration, and time,

$$s = \frac{1}{2}at^2\,, \qquad 4.2.7$$

When Equation 4.2.5, and 4.2.6 is plugged in Equation 4.2.7,

$$s_1 = \frac{1}{2}a_1\left(\frac{v}{a_1}\right)^2 = \frac{1}{2}\frac{v^2}{a_1} \qquad 4.2.8$$

$$s_3 = \frac{1}{2} a_2 \left(\frac{v}{a_2} \right)^2 = \frac{1}{2} \frac{v^2}{a_2} \qquad 4.2.9$$

Hence,

$$s_2 = l_1 - \frac{1}{2} \frac{v^2}{a_1} - \frac{1}{2} \frac{v^2}{a_2} \qquad 4.2.10$$

The travel time in the cruising velocity regime will be

$$t_2 = \frac{l_1}{v} - \frac{v}{2a_1} - \frac{v}{2a_2} \qquad 4.2.11$$

Therefore, total travel time in the uncongested region which considering acceleration and deceleration behaviors of vehicle will be formulated as

$$t_u = t_1 + t_2 + t_3 = \frac{v}{2} \left(\frac{1}{a_1} + \frac{1}{a_2} \right) + \frac{l_1}{v} \qquad 4.2.12$$

4.3 FORMULATION OF TRAVEL TIME IN THE CONGESTED REGION

On the other hand, vehicle delays (travel time delays) caused by the signalized intersection is modeled in this section. If the average vehicle length is assumed to be a constant value μ, the length of sublink l_2 in Figure 4.1.1 (congested region) can be expressed as

$$l_2 = r \times \mu \qquad 4.3.1$$

where, l_2 = sublink length of congested region

r = number of vehicle in the queue

μ = average vehicle length

Then,

$$l_1 = l - r \times \mu \quad 4.3.2$$

where, l = total link length

Replace Equation 4.3.2 with l_1 in Equation 4.2.3.

$$k(l - r\mu) + r - n = 0 \quad 4.3.3$$

Hence, the number of vehicles in the queue (r) can be expressed as

$$r = \frac{n - kl}{(1 - k\mu)} \quad 4.3.4$$

where, n is the number of vehicle in a link at certain time t, and it can be estimated as

$$n = q_{1,cum} - q_{2,cum} \quad 4.3.5$$

where, $q_{1,cum}$ = cumulative input flow

$q_{2,cum}$ = cumulative output flow

The number of vehicle in a link at certain time t can be obtained by measuring cumulative inflow and outflow of the link. It can be explained graphically in Figure 4.3.1.

According to the conservation law and under the assumption of uniform arrival and departure rate, cumulative curve of inflow (q_1) is always higher than that of outflow (q_2) at a certain time. Thus the difference between cumulative inflow and outflow at certain time t will be the number of vehicles in a link at certain time.

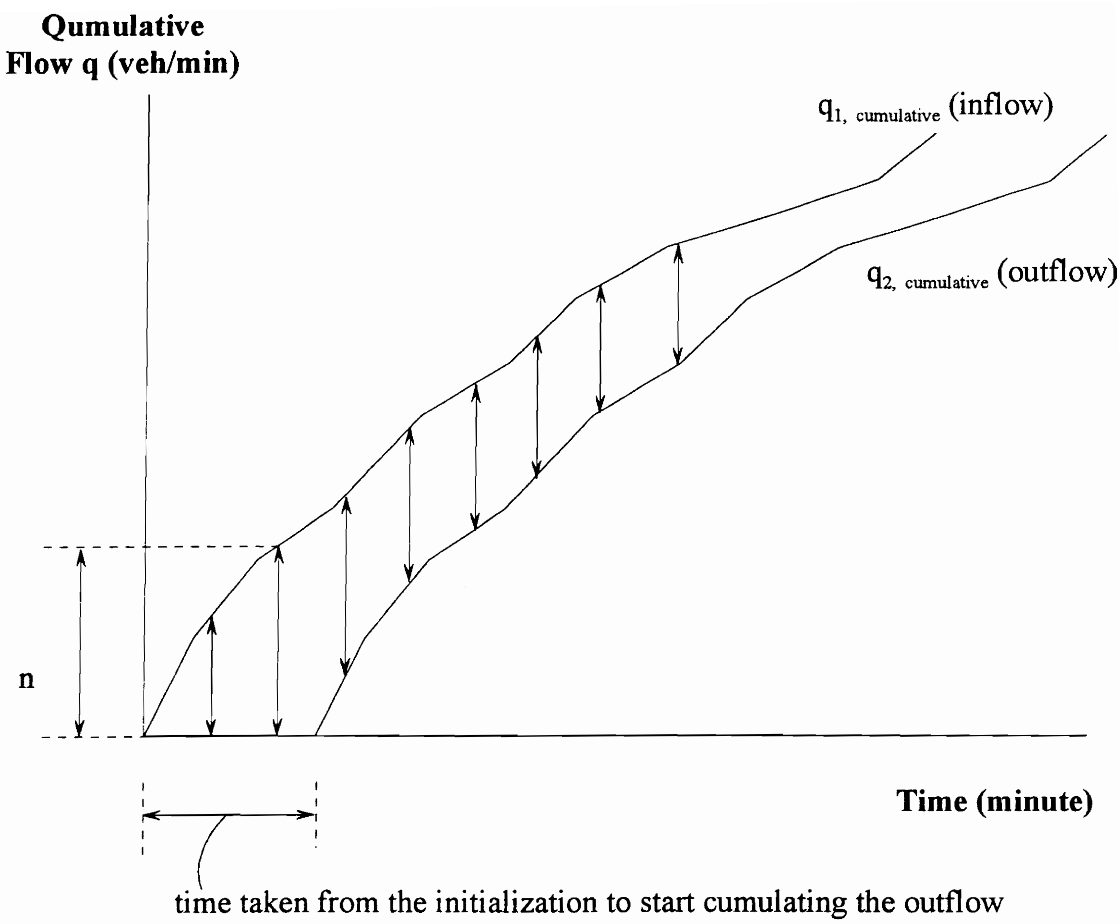


Figure 4.3.1 Numbers of Vehicles in a Link

Therefore, travel time in congested region can be formulated as

$$t_c = \frac{r}{q} \qquad 4.3.6$$

where, t_c = travel time in congested region

r = number of vehicle in congested region

q = flow which represents the link (midblock flow)

4.4 FORMULATION OF DYNAMIC LINK TRAVEL TIME FUNCTION

Based on the previous sections, travel time considering the intersection delay can be formulated as

$$t = t_c + t_u$$

$$t = \frac{r}{q} + \frac{v}{2}\left(\frac{1}{a_1} + \frac{1}{a_2}\right) + \frac{l_1}{v} \qquad 4.4.1$$

where, q = flow in uncongested region

$l_1 = l$ - rμ

v = space mean speed in uncongested region

The first term in Equation 4.41 expresses the travel time in the congested region. The second term represents the delay effect caused by acceleration and deceleration of the vehicles in the uncongested region. The last term represents cruising travel time in the uncongested region.

As it is noticed, flow variable (q), which represents the flow of the link l in the first term of Equation 4.4.1, can be obtained from the mid-section of the uncongested region of the sublink l_1. Since four different flows (inflow, outflow, flow in uncongested region, flow in congested region) are defined according to Figure 4.1.1, more detail analysis on arterial roads can be possible than before - the conventional definition of flow on arterial link used to represent an entire link with a single flow. Speed variables (v) in the second and the third term represent the space mean speed which can also be estimated in the middle of the cruise regime (mid-block) in the uncongested region for the mixed traffic.

In the second and the third term of Equation 4.4.1, speed (v) can be alternatively obtained by utilizing the linear macroscopic flow model of Greenshield.

Since

$$\text{flow (q)} = \text{density (k)} \times \text{speed (v)} \qquad 4.4.2$$

If we plug Equation 4.2.2 into Equation 4.4.2, we can obtain

$$q = \left[k_j\left(1 - \frac{v}{vf}\right)\right] \times v$$

$$= k_j v - \frac{k_j v^2}{v_f} \qquad 4.4.3$$

Hence, we can solve Equation 4.4.3 with respect to speed (v). Then,

$$v = \frac{v_f\left(k_j \pm \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j} \qquad 4.4.4$$

From Equation 4.4.4, since we are dealing with the stable regime (under capacity regime) of the Greenshield model, only the positive sign can be applied in this research. It can be explained graphically in Figure 4.4.1.

Hence, speed can be defined as

$$v = \frac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j} \qquad 4.4.5$$

If Equation 4.3.4 and Equation 4.4.4 are replaced with the number of vehicle in the queue (r) and mean speed (v) in Equation 4.4.1, then total travel time in a link (Equation 4.4.1) can be expressed as

$$t = \frac{\frac{n-kl}{(1-k\mu)}}{q} + \frac{\frac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j}}{2}\left(\frac{1}{a_1} + \frac{1}{a_2}\right) + \frac{l - \frac{n-kl}{(1-k\mu)}\mu}{\frac{v_f\left(k_j - \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j}} \qquad 4.4.6$$

The density (k) in Equation 4.4.6 can be rewritten by adopting the same procedure of Equation 4.4.2 to 4.4.4. Hence, density would be represented as

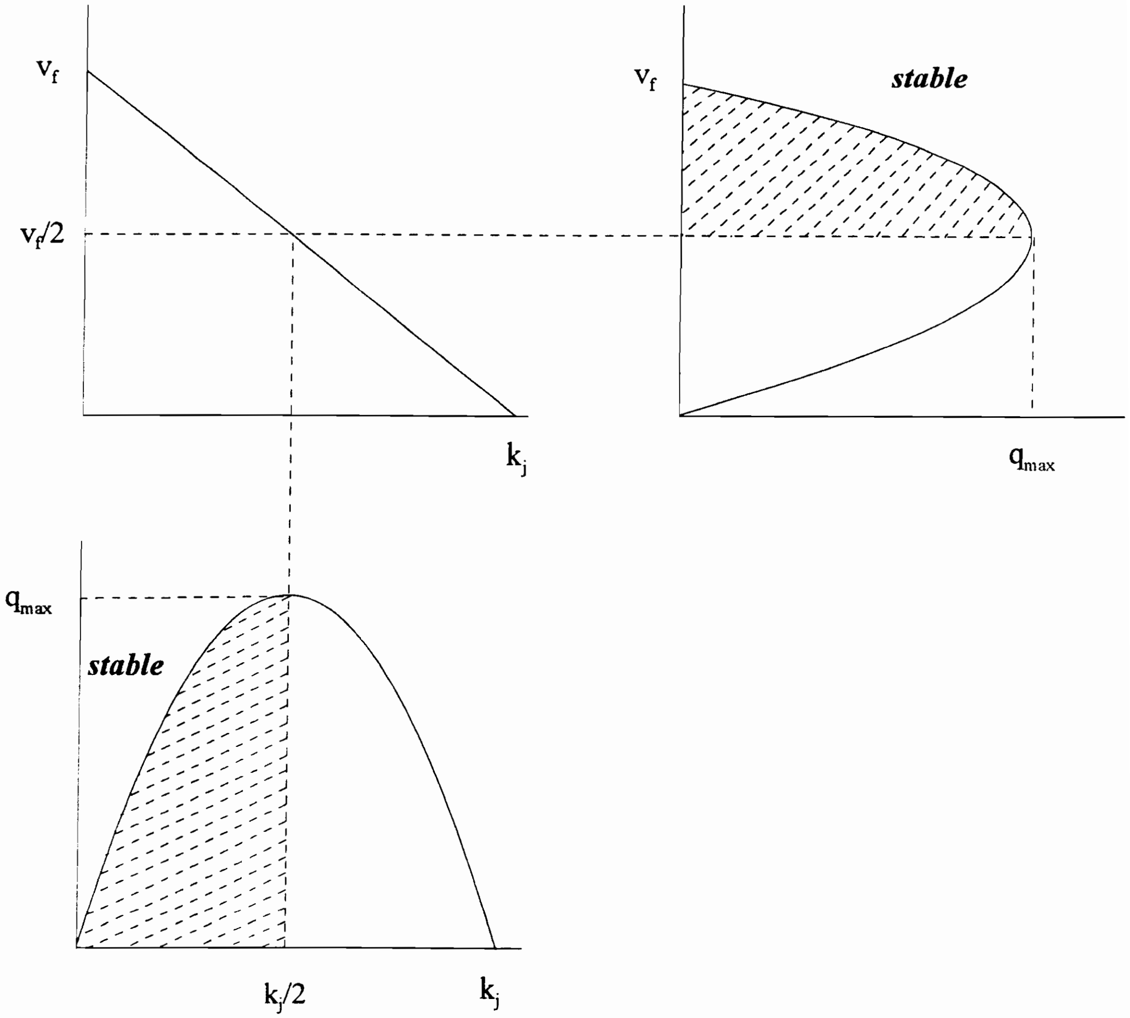


Figure 4.4.1 Flow-Density-Speed Relationship

$$k = \frac{k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f} \qquad 4.4.7$$

Equation 4.4.7 can be plugged in the variable density (k) in Equation 4.4.6. Then, the dynamic link travel time will be represented as

$$t = \frac{\dfrac{n - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)}{2v_f}l}{\left(1 - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)}{2v_f}\mu\right)}}{q} + \frac{\dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j}}{2}\left(\frac{1}{a_1} + \frac{1}{a_2}\right) + \frac{l - \dfrac{n - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)}{2v_f}l}{\left(1 - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)}{2v_f}\mu\right)}\mu}{\dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j}} \qquad 4.4.8$$

In Equation 4.4.8, the time dependent variables which will be measured in practice are flow (q), and number of vehicle in a link (n). As it was explained in Equation 4.3.5, the number of vehicle in a link can be obtained by measuring cumulative inflow and outflow. Therefore, as long as we have real-time data for flow in each link, Equation 4.4.8 can estimate travel time simply and dynamically because the rest of variables are all constants.

4.5 FURTHER RESEARCH

In order to be more reasonable and accurate arterial link travel time estimation model, following issues should be studied further in the research. First of all, it was assumed in this research that the queue length in a link never exceeded the total link length at any time of the day, i.e., spill-back problem was not considered. Downtown in metropolitan areas, however, spill back problems are not so uncommon.

Secondly, the location of vehicle in the queue was not specifically considered in the link travel time estimation model considering intersection delay. Because of this reason, when the bus is located at the back of the queue, and if the green phase of the signal cycle is not long enough to accommodate all queued vehicles, a bus can not pass through the intersection. However, in this approach it was assumed that each green phase will accommodates all the queued vehicles including the bus at the back of the queue.

Thirdly, signal progressions are in effect in most of the arterial roads in the urbanized transportation networks. Hence, more advanced algorithm which deals with stochastic analysis of arriving vehicles at intersections at specific phase (green and red) of signal better be considered for more realistic approach.

CHAPTER V INTEGRATED TRAVEL TIME ESTIMATION

5.1 LINEAR PARAMETERIZATION OF DYNAMIC LINK TRAVEL TIME MODEL

The first step in parameter estimation process is the identification of system structure. Then, formulate analytic model which represents the system. The analytic model can perform as a nominal model so that its parameters will be adjusted in on-line process. The separation of known (measurable) variables and unknown parameters (immeasurable), i.e., linear parameterization, are the next process in parameter adaptation algorithm.

In the dynamic link travel time estimation model in Chapter IV, flow (q), number of vehicles in a link (n), and link length (l) are the known variables. Among these variables, the flow and the number of vehicles in a link can be assumed to be obtained by conventional vehicle detectors. On the other hand, combination of free flow speed (v_f), jam density (k_j), average vehicle length (μ), acceleration rate (a_1), and deceleration rate (a_2) are considered to be unknown parameters.

Now, Equation 4.4.8 can be simplified as follows

$$t = \frac{\dfrac{n - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}l}{\left(1 - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}\mu\right)}}{q} + \frac{\dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)}{2k_j}}{2}\left(\frac{1}{a_1} + \frac{1}{a_2}\right) + \frac{l - \dfrac{n - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}l}{\left(1 - \dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}\mu\right)}\mu}{\dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)}{2k_j}}$$

$$= \frac{4nv_f^2 - v_f k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)l}{2qv_f\left[2v_f - k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)\mu\right]} + \frac{v_f\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)(a_1 + a_2)}{4a_1a_2k_j} + \frac{4lv_fk_j^2\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)(1-\mu) + 8k_jv_f^2(l-n)}{2v_f^2\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)\left[2v_f - k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)\right]} \quad 5.1.1$$

Thus, Equation 5.1.1 can be rewritten in terms of known and unknown variables.

$$t=\frac{v_f}{2v_f-k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)\mu}\cdot\frac{2n}{q}-\frac{k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)}{2v_f-k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)\mu}\cdot\frac{l}{q}$$

$$+\frac{v_f\left(k_j+\sqrt{k_j^2-4q\frac{k_j}{v_f}}\right)(a_1+a_2)}{4a_1a_2k_j}$$

$$+\frac{1}{\frac{v_f\left(k_j+\sqrt{k_j^2-4q\frac{k_j}{v_f}}\right)}{2k_j}\left[1-\frac{\mu k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)}{2v_f}\right]}l-\frac{\mu}{\frac{v_f\left(k_j+\sqrt{k_j^2-4q\frac{k_j}{v_f}}\right)}{2k_j}\left[1-\frac{\mu k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)}{2v_f}\right]}n \qquad 5.1.2$$

Equation 5.1.2 can be simplified as

$$t=a\left(\frac{2n}{q}\right)-b\left(\frac{l}{q}\right)+c(1)+d(l)-e(n) \qquad 5.1.3$$

where, $a=\dfrac{v_f}{2v_f-k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)\mu}$

$$b=-\frac{k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)}{2v_f-k_j\left(v_f-\sqrt{v_f^2-4q\frac{v_f}{k_j}}\right)\mu}$$

$$c = \frac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)(a_1 + a_2)}{4a_1a_2k_j}$$

$$d = \frac{1}{\frac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j}\left[1 - \frac{\mu k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)}{2v_f}\right]}$$

$$e = -\frac{\mu}{\frac{v_f\left(k_j + \sqrt{k_j^2 - 4q\frac{k_j}{v_f}}\right)}{2k_j}\left[1 - \frac{\mu k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)}{2v_f}\right]}$$

In Equation 5.1.3, first two terms are associated with the congested region, whereas the rest of three terms are associated with the uncongested region in the link travel time estimation. Therefore, as it was reviewed in the *Section 3.3.3 Parameter Estimation*, for the adaptive parameter estimation, parameter vector (θ) and independent variable (signal) vector (ϕ) can be defined for the dynamic link travel time model as

$$\theta = \begin{bmatrix} \dfrac{v_f}{2v_f - k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)\mu} \\[2ex] -\dfrac{k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f - k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)\mu} \\[2ex] \dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)(a_1 + a_2)}{4a_1a_2k_j} \\[2ex] \dfrac{1}{\dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)}{2k_j}\left[1 - \dfrac{\mu k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}\right]} \\[2ex] -\dfrac{\mu}{\dfrac{v_f\left(k_j + \sqrt{k_j^2 - 4q\dfrac{k_j}{v_f}}\right)}{2k_j}\left[1 - \dfrac{\mu k_j\left(v_f - \sqrt{v_f^2 - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}\right]} \end{bmatrix}$$

For θ^T, if the relationship of Equation 5.1.3 is considered, then θ^T will be

$$\theta^T = [\mathrm{a} \quad \text{-b} \quad \mathrm{c} \quad \mathrm{d} \quad \text{-e}] \qquad 5.1.4$$

$$\phi(k-1) = \begin{bmatrix} \frac{2n}{q} \\ \frac{1}{q} \\ \\ 1 \\ l \\ n \end{bmatrix} \qquad 5.1.5$$

Therefore, output *y(k)* will be simply represented as

$$y(k) = \theta^T \phi(k-1)$$

$$= \begin{bmatrix} a & -b & c & d & -e \end{bmatrix} \begin{bmatrix} \frac{2n}{q} \\ \frac{1}{q} \\ \\ 1 \\ l \\ n \end{bmatrix}$$

Since we are not interested in the value of the specific unknown variables, but the travel time of a link, some of the known variables in the unknown parameter groups will not cause any problem. In fact, if we are interested in the specific unknown variable, it can be obtained from the theta (θ) matrices. There are four unknown variables and five equations are defined. Hence, all these variables can be obtained.

In Equation 5.1.2, very complicated nonlinear relationships are associated in the parameters. However, since the relationship of known (measurable) and unknown variables are defined linearly, linear parameterization was able to be conducted even though there is a redundancy (five equations and four unknown variables).

5.2 INTEGRATED TRAVEL TIME ESTIMATION MODEL

In order for the integration of the dynamic link travel time model with the bus arrival time prediction model, described in Equation 3.3.1, the dynamic link travel time considering the intersection delay should be an input to the integrated travel time prediction model. In the first parameter estimation, which was already conducted and discussed in the Chapter 3, the link travel time was considered as a constant value. However, it varies depending on the configuration of the route. Thus, the integration process is essential and it is discussed in this chapter.

Since we are considering varying link travel times (c_{ij}), Equation 3.3.1 should be modified for the integrated model. Thus, Equation 3.3.1 can be rewritten as

$$t_{ij} = c_{ij} + \alpha x_{ij} \qquad 5.2.1$$

where, t_{ij} = departure time headway (travel time)

αx_{ij} = passenger loading time

c_{ij} = departure time headway that excludes passenger loading time

i = stops

j = buses

Equation 5.2.1 can be developed as

$$t_{ij} = c_{ij} + \alpha\left[\frac{-\alpha p}{1-\alpha p}x_{ij-1} + \frac{ph_{ij}}{1-\alpha p}\right]$$

$$= c_{ij} + \alpha\left[\frac{-(t_{ij-1} - c_{ij-1})p}{1-\alpha p} + \frac{ph_{ij}}{1-\alpha p}\right]$$

$$= \frac{-\alpha p}{1-\alpha p}t_{ij-1} + \frac{\alpha p}{1-\alpha p}(hij + c_{ij-1} - c_{ij}) + \frac{1}{1-\alpha p}c_{ij} \qquad 5.2.2$$

Later in this chapter, Equation 5.2.2 will be adopted as the nominal model for the parameter estimation in the integrated travel time model.

In the integrated model, it is also defined that bus stop to stop as a link, and a link is further divided into sublinks based on the configuration of route. Figure 5.2.1 illustrates definition of link and sublink.

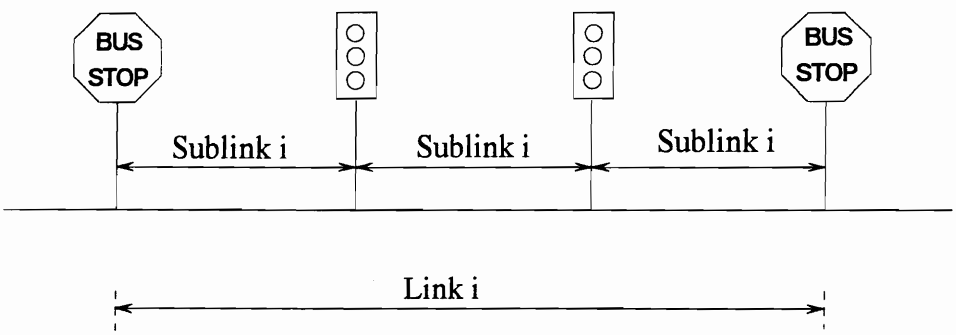


Figure 5.2.1 Definition of Link and Sublink

Variations of variable c_{ij} (link travel time) in the bus arrival time model heavily rely on the location and number of intervening signalized intersections in between the stops. Two main scenarios can be assumed (Refer to Figure 5.2.1) for testing and validation of integrated travel time model. These are:

- No intervening signalized intersection in between stops

  In this case, the second and the third terms of Equation 4.4.8, which deal with acceleration and deceleration, and uncongested region's link travel time, can only be applied to calculate travel time on a link in the bus arrival time estimation model. Thus, the first term for link travel time in congested region in Equation 4.4.8 can be omitted.

- One or more intervening intersections in between stops

  a) If there is only one intersection located in between the stops, adopt whole Equation 4.4.8 to bus stop to intersection link, and apply only the second and the third term of Equation 4.4.8 for the intersection to bus stop link. Then, sum up these two links as link travel times.

  b) If there exists more than one intervening intersections, adopt the whole Equation 4.4.8 for the bus stop to intersection sublinks, and all the intersection to intersection links. The second and the third term of Equation 4.4.8 will be applied to the intersection to bus stop link. Then, add all the travel time of sublinks for bus travel time estimation for downstream stops.

5.3 THREE STEPS-AHEAD LINK TRAVEL TIME ESTIMATION

The estimation of link travel time can be conducted in two ways. The first approach is to conduct the parameter estimations on each individual sublinks independently, and add all the sublink travel times which coincide with the link in between the first and the second bus stops, and repeat this procedure for other sublinks in between two downstream bus stops. The other approach is to consider link configurations and conduct parameter estimation for bus stop to stop base without considering each individual sublinks. Two of these approach have their own advantages and disadvantages in terms of the accuracy and the complexity. Since the reliability of prediction model is a critical issue in arrival time estimation, the first approach was selected because of accuracy and ample data availability when the Global Positioning System is utilized as the bus detection technology in AVL. The drawback of this approach, however, is the computational complexity due to the estimation on each individual sublinks.

The hypothetical network segment for parameter estimation consist of four bus stops including five intervening intersections. Figure 5.3.1 illustrates the configuration of the hypothetical segment of an arterial route which is utilized for the link travel time parameter estimation considering intersection delay.

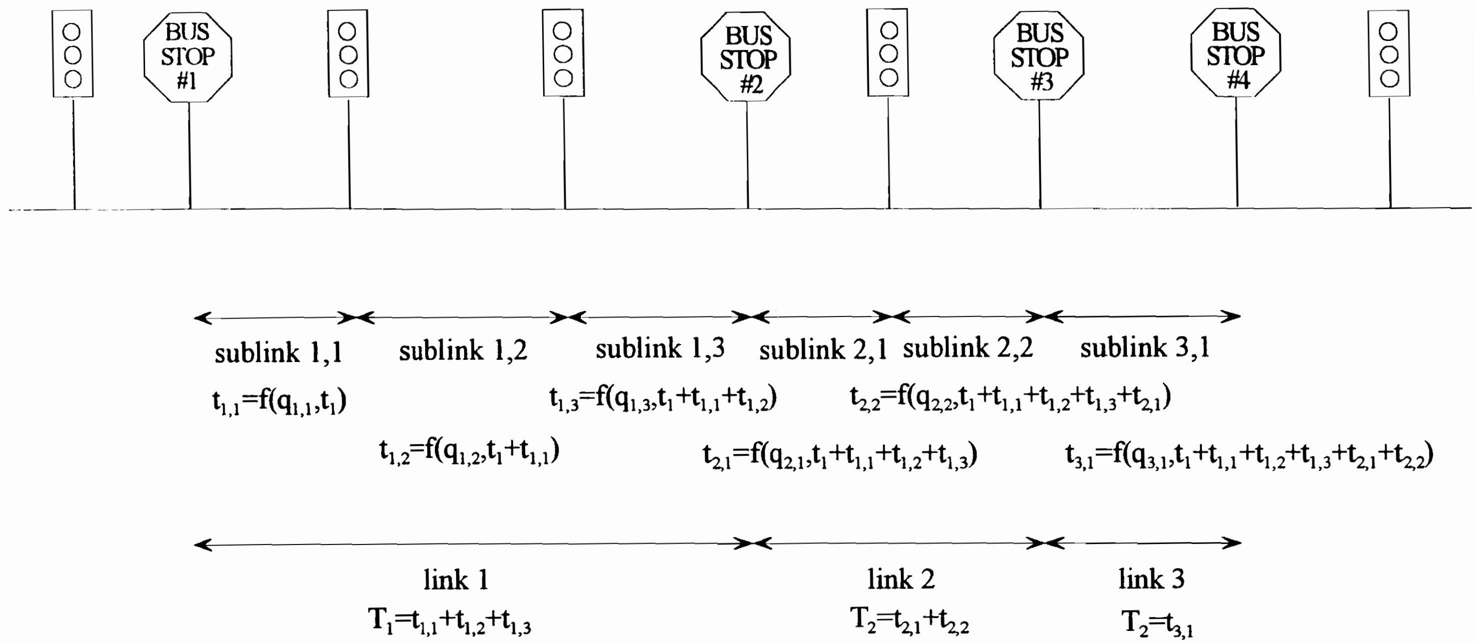


Figure 5.3.1 Configuration of Hypothetical Network Segment

In order to be more realistic estimation, time varying flow in each links is to be adopted as an input to the least square parameter estimation. For the time varying link flow data, commercial software INTEGRATION, developed in Queen's university, Canada, was adopted and tested. The main purposes of the INTEGRATION simulation package are to obtain both the minimum path among the arterial roads, combined with freeways for route guidance under the incident condition, and signal optimization. In the INTEGRATION simulation, however, non-incident traffic condition is assumed and main interest is focused on the time varying link flows on arterial road only. Then, the real flow data obtained from the town of Blacksburg, conducted by Virginia Department of Transportation (VDOT) was utilized for the parameter adaptation algorithm testing. A 24 hour of 15 minute data were available. The overall flow rate is low, but still Blacksburg data has unique characteristics of representing am and pm peak periods. The plot of flow is shown in Figure 5.3.2. As it was stated in the previous section, a link is further subdivided into any number of sublinks wherever interruptions are made on the stream of traffic flow.

5.3.1 INTEGRATION Simulation Model

Initialization for Master File

- Simulation duration was assumed as 28200 seconds (7.83 hours) because when the bus headway is 30 minutes, the simulation should cover at least fifteen buses.

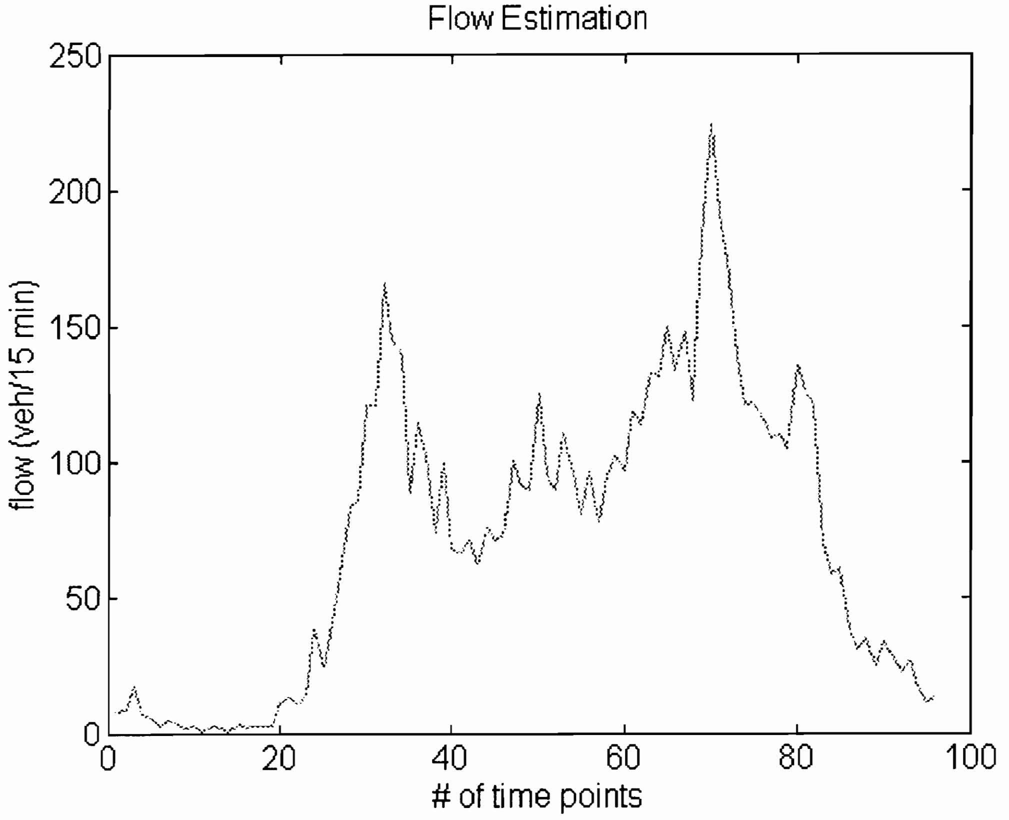


Figure 5.3.2 Blacksburg Flow Data Plot

- Every 1800 second, output which contains accumulated flow information in each link can be obtained.

Node Specification

- For the analysis, nineteen nodes (nine origin/destination and ten intermediate) were specified.
- To reduce the computer memory and execution time, seven macro cluster zones were defined.
- It was assumed that none of changeable message sign (CMS) and route guidance system (RGS) beacon exist at nodes.

Link Specification

- Free flow speed on arterial road was defined as 45 mph.
- All the lanes were assumed to be two lanes.
- Saturation flow rate was equally assumed as 2000 vph.
- Speed at capacity on arterial road was defined as 28 mph.
- Jam density was assumed as 240 vpm.
- No turn prohibition was assumed.
- All the link had surveillance systems so that each vehicle can access real time link information.

Signal Specification

- Five signals were located at each of the intermediate intersection nodes.
- Two control plans were assumed for signal optimization.
- 80 and 60 second were assumed as initial cycle lengths, and two signal phases were assumed.
- Lost time was assumed as four second for each control plan.

Flow data generated by INTEGRATION is illustrated in Figure 5.3.3. As it is illustrated in the figure, INTEGRATION data don't show the general trend of peak periods.

5.3.2 Three Steps-Ahead Link Travel Time Estimation

For the dynamically predicted travel time (arrival time), time dependent flow and travel time of the previous link are the main variables affecting the three step-ahead link travel time prediction on arterial roads. In the prediction, at the first bus stop (current location), arrival time (travel time) of bus for downstream bus stops are expected. Current flow (q) data on each link at time t will be, however, changed by the time when the bus arrives at the downstream sublink (intersection) or link (bus stop). Hence, newly estimated flow at downstream links (sublinks) should be applied for the travel time prediction update for further downstream links or bus stops.

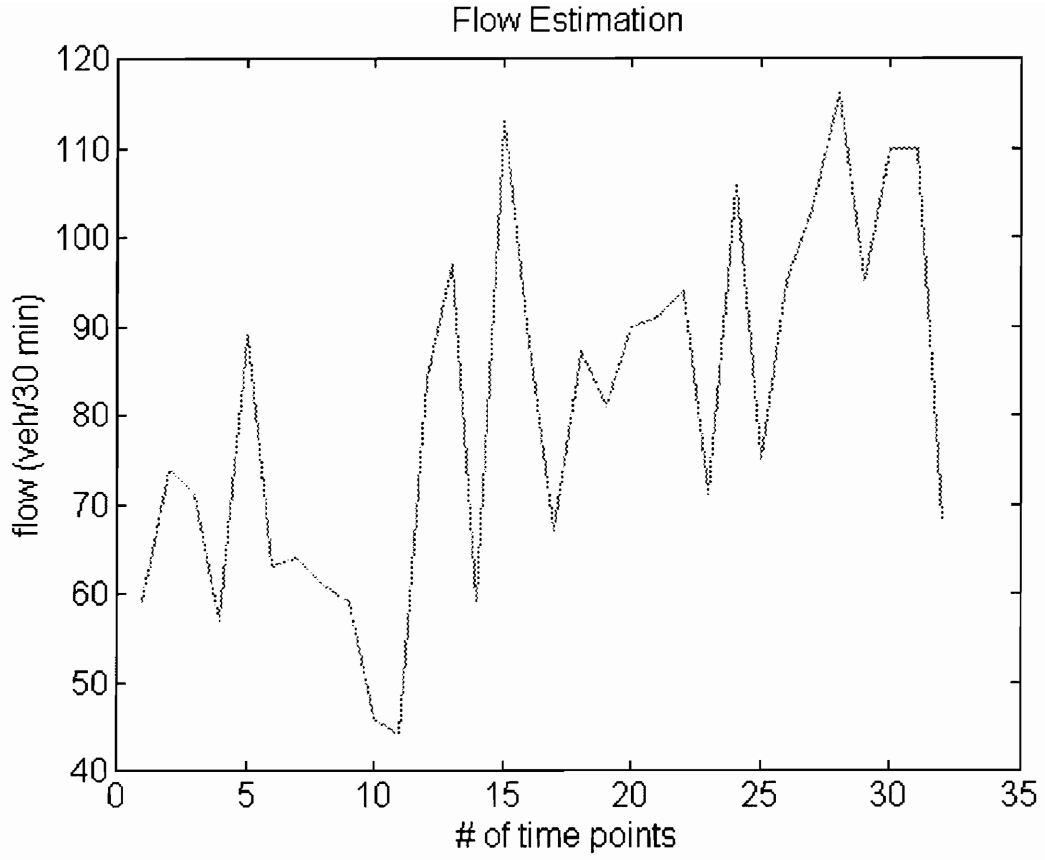


Figure 5.3.3 INTEGRATION Flow Data Plot

From Figure 5.3.1, travel time from bus stop 1 to bus stop 2, 3 and 4 will be expressed as

$$T_1 = t_{1,1} + t_{1,2} + t_{1,3} \qquad 5.3.1$$

$$T_2 = t_{2,1} + t_{2,2} \qquad 5.3.2$$

$$T_3 = t_{3,1} \qquad 5.3.3$$

where, T_1 = travel time of bus stop 1 to bus stop 2

T_2 = travel time of bus stop 2 to bus stop 3

T_3 = travel time of bus stop 3 to bus stop 4

$t_{i,j}$ = sublink travel time

i = link number

j = sublink number

Further, the sublink travel time can be estimated as

$$t_{1,1} = f(q_{1,1}, t_1)$$
$$t_{1,2} = f(q_{1,2}, t_1 + t_{1,1})$$
$$t_{1,3} = f(q_{1,3}, t_1 + t_{1,1} + t_{1,2})$$

$$t_{2,1} = f(q_{2,1}, t_1 + t_{1,1} + t_{1,2} + t_{1,3})$$
$$t_{2,2} = f(q_{2,2}, t_1 + t_{1,1} + t_{1,2} + t_{1,3} + t_{2,1})$$
$$t_{3,1} = f(q_{3,1}, t_1 + t_{1,1} + t_{1,2} + t_{1,3} + t_{2,1} + t_{2,2})$$

where, $t_{1,1}$ = travel time of sublink 1,1 at time t_1

$t_{1,2}$ = travel time of sublink 1,2 at time $t_1+t_{1,1}$

...

$t_{3,1}$ = travel time of sublink 3,1 at time $t_1+t_{1,1}+t_{1,2}+t_{1,3}+t_{2,1}+t_{2,2}$

$q_{1,1}$ = link flows between bus stop 1 to the first intersection

$q_{2,2}$ = link flows between the third intersection to bus stop 2

...

$q_{3,1}$ = link flows between bus stop 3 to bus stop 4

In summary, the first sublink travel time is estimated based on the flow at that time, then for the immediate downstream sublink or link travel time, the flow, which will be estimated as summation of initial time and estimated time of previous sublink or link will be used to estimate the travel time of that (downstream) link. The mechanism of estimation follows this rule, and in order to have as reliable as possible estimation of arrival time, the estimation of flow at different time points are essential. Therefore, dynamic flow estimation should be devised at the initial stage of the travel time estimation. Detail descriptions on dynamic flow estimation is made in the following section.

5.4 DYNAMIC FLOW ESTIMATION

5.4.1 The Approach

In the estimation of integrated travell time estimation, which was described in section 5.2, obtaining the predicted time-dependent link flow data is essential in order to adopt the dynamic characteristics of network so that it can generate more reliable prediction model. As it was specifically noted in the dynamic link travel time model, which heavily depends on the time varying link flow condition, predicting the link flow at different time *t* plays an important role.

The approach of time dependent flow prediction model formulation, however, mostly rely on the sampling rate of data. If the sampling rate is less frequent than average travel time of that link, then the formulation of flow prediction model would be based on the flow at upstream link on previous time. It can be explained from the basic fluid mechanics that the car traveling on upstream link will pass that link before the data update, and influencing directly on the current link.

On the other hand, if the data sampling rate is short so that its time is less than that of the link travel time, current link flow at previous time point better be considered for model formulation. In this case, for instance, link travel time is comparatively long, after the update of the flow data majority of vehicles will still remain on the same link. Therefore, different approaches should be established in formulating the flow model which depending on the sampling rate. In this research, now that it is assumed that link flow sampling rate is every 5 minute (both from the Integration simulation and reflecting the best reality of flow data collection on arterials under ITS) and link travel time is at least 5 to 10 minutes (considering the rather long link length and intersection delay), the formulation follows the latter case.

Once again, recursive least square (LS) parameter adaptation algorithm (PAA) is adopted. In LS algorithm, all the parameters are assumed to be the time invariant. However, for the real time application, these parameters are time varying and it should be adapted for the time varying situation, too. In this matter, time decreasing gain term (*F*) makes the situation even worse. In order to avoid this situations, forgetting factor (λ), which can make the LS identifier forget the past data, should be introduced. More detail descriptions on forgetting factor were made in section 3.3.3.2.

Gain term which considering the forgetting factor can be defined by Landau and Silveira (1979) as

$$F(k+1) = \frac{1}{\lambda_1(k)}\left[F(k) - \frac{F(k)\phi^T(k)F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)} + \phi^T(k)F(k)\phi(k)}\right]$$

and typical range of $\lambda_1(k)$ and $\lambda_2(k)$ is 0.5 to 1.0 and 1.0, respectively. Meanwhile, parameter updating law remains as the same as Equation 3.3.34 in Chapter 3.

Another suggestion for coping with the decreasing gain is made by Ljung (1987). He discussed about the normalizing the gain, and formulated gain term and uptating law. These were:

$$\bar{\theta}(t) = \bar{\theta}(t-1) + (1-\lambda)R^{-1}(t)\phi(t)\varepsilon(t) \qquad 5.4.1$$

$$R(t) = R(t-1) + (1-\lambda)[\phi(t)\phi^T(t) - R(t-1)] \qquad 5.4.2$$

where, $\varepsilon(t) = y(t) - \phi^T(t)\bar{\theta}(t-1)$

λ = forgetting factor

An issue, which should be discussed in this section is the parameter estimation against nonparameteric estimation. If the system structure is unknown, and only inputs and outputs data are available, nonparameteric approach, such as ANN adopting supervised learning, is more appropriate. However, nominal model can be structured for parameter estimation as long as main interests are focused on the dependent variable instead of parameter itself. Thus, PAA is adopted in this section.

Three different structure of PAA scenarios were developed.

- a model based on the same link flow of one step-previous sampling time and a constant
- a model based on the same link flow of one step and two step-previous sampling time and a constant

• a model based on immediate upstream link flow at previous sampling time, the same link flow of one step-previous sampling time, and a constant

Simulation of parameter estimation was done in PC Matlab. The problem in LS with forgetting factor did not predict the time varying flow. Mean Square Error was tested to check the predictability. It failed. The reason was in decreasing gain factor.

Dynamic flow estimation plays a key role in integrated travel time estimation. Based on the scenario, three approaches were identified, formulated, and tested. The commonalties in these approaches are that these models were based on the autoregressive model.

5.4.2 One Step Lagging Model

Formulation of recursive model based on the flow in one step before was attempted in order to estimate future flow. The autoregressive (AR) model was formulated as

$$q(k) = \sum_{k=1}^{p} aq(k-1) + b \tag{5.4.1}$$

where, p = number of time points

a and b= parameter vector

This model is tested and validated by using simulated INTEGRATION output and Blacksburg data. PC version of Matlab was utilized as the simulation tool. The code is attached in the Appendix B. The outputs of PAA in this task are to obtain the estimated flow, and estimation errors for each link. At the initial simulation, two parameters (a and b) were estimated.

The initial results for Blacksburg data showed that it follows the same trend as flow variation in prediction with time lagging, and mean square error (MSE) was fairly big (926). Figure 5.4.1 shows an output of validation among the six links. For INTEGRATION data, one of the results is shown in Figure 5.4.2. In this validation, least square (LS) algorithm with forgetting factor was tested. One of the known drawback in LS algorithm is that parameter tracking is slow in quickly time varying parameters. In the result, as it is stated, LS algorithm couldn't catch up with the time varying flow condition very actively. Also, decreasing gain term in LS algorithm decelerates its sensitivity in quickly time varying situation. Better approach proposed by Ljung (1987) was also tested. Although the result is slightly better than previous one, its slow response in actively changing flow condition is not able to be accepted. Its result is also illustrated in Figure 5.4.3 for Blacksburg and Figure 5.4.4 for INTEGRATION, and code is attached in Appendix C.

5.4.3 Two Step Lagging Model

The formulation is as follows

$$q(k) = \sum_{k=1}^{p} aq(k-1) + bq(k-2) + c$$

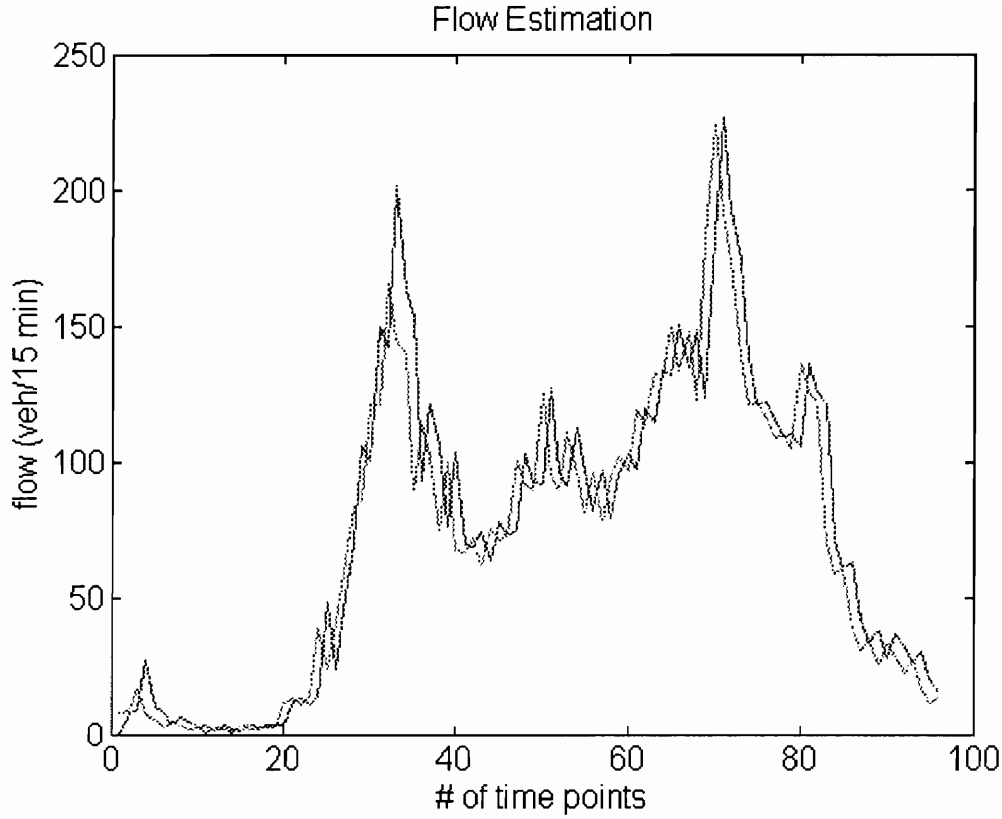


Figure 5.4.1 One Step Lagging Flow Prediction (Blacksburg, generic w/forgetting Factor)

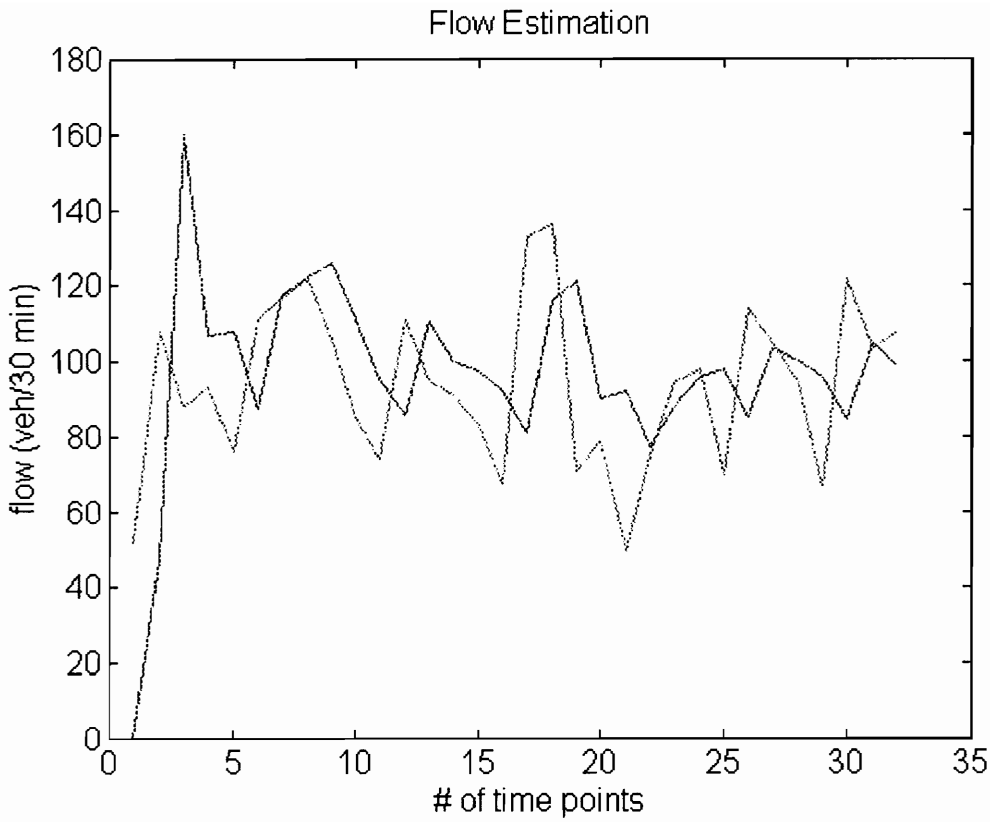


Figure 5.4.2 One Step Lagging Prediction (INTEGRATION, generic w/forgetting Factor)

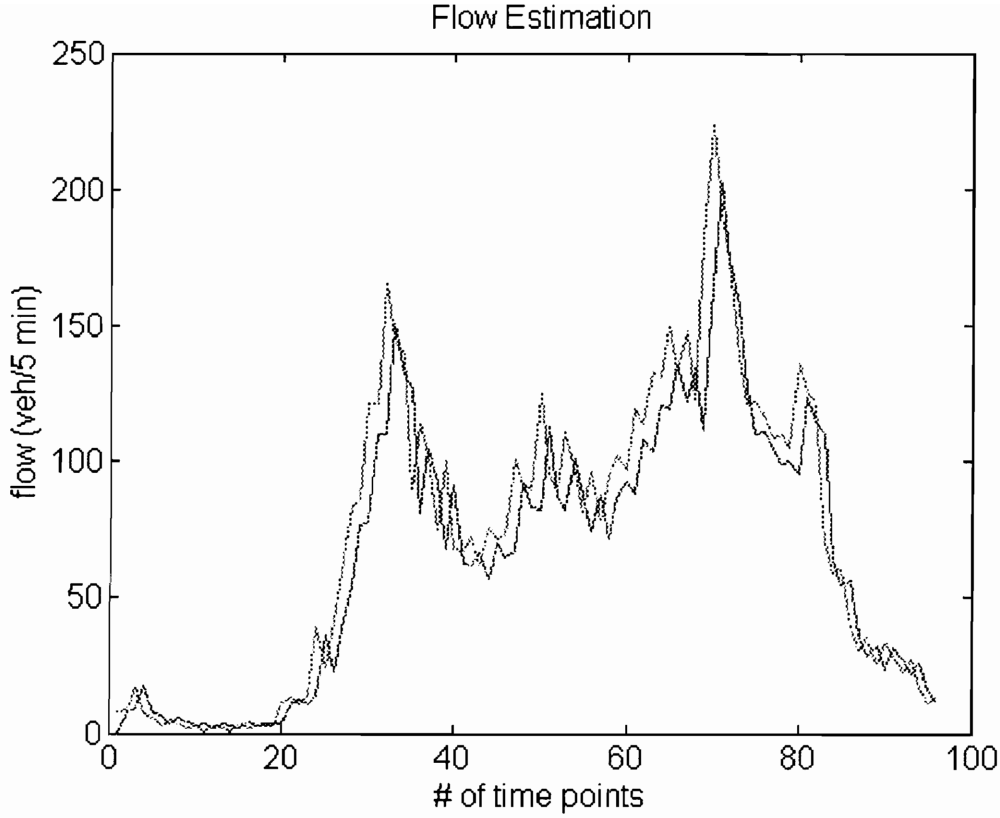


Figure 5.4.3 One Step Lagging Flow Prediction (Blacksburg, Ljung's Approach)

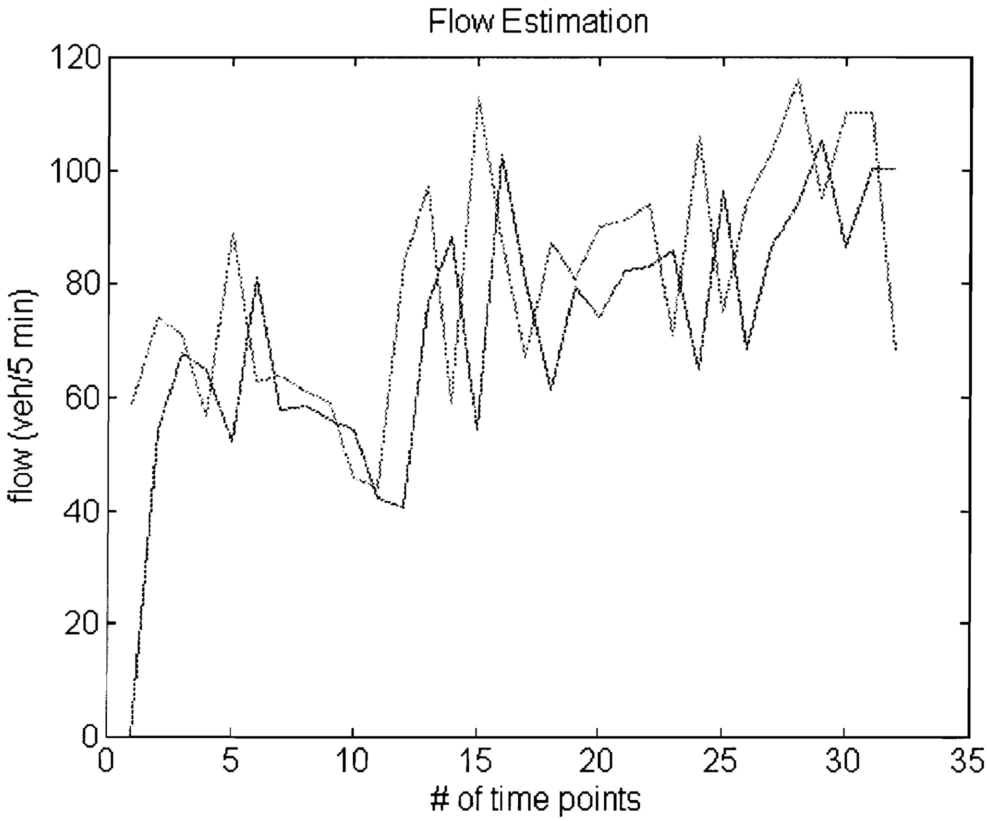


Figure 5.4.4 One Step Lagging Flow Prediction (INTEGRATION, Ljung's Approach)

For this model, generic LS model with forgetting factor was applied and tested on INTEGRATION output and Blacksburg data. Also, Ljung's approach was tested. The results of Blacksburg data are shown in Figure 5.4.5 and Figure 5.4.6. INTEGRATION results are shwn in Figure 5.4.7 and Figure 5.4.8.

5.4.4 One Step Lagging Considering Space Domain Model

The formulation is as follows

$$q_l(k) = \sum_{k=1}^{p} a q_l(k-1) + b q_{l-1}(k-2) + c$$

where, l = link

One step lagged flow at the same link at previous time point, and two steps lagged flow with the immediate upstream link was introduced in this formulation. Again, generic LS algorithm with forgetting factor was applied for the first test, and Ljung's model was also applied. The results of Blacksburg data are demonstrated in Figure 5.4.9 and Figure 5.4.10. INTEGRATION results are illustrated in Figure 5.4.11 and Figure 5.4.12.

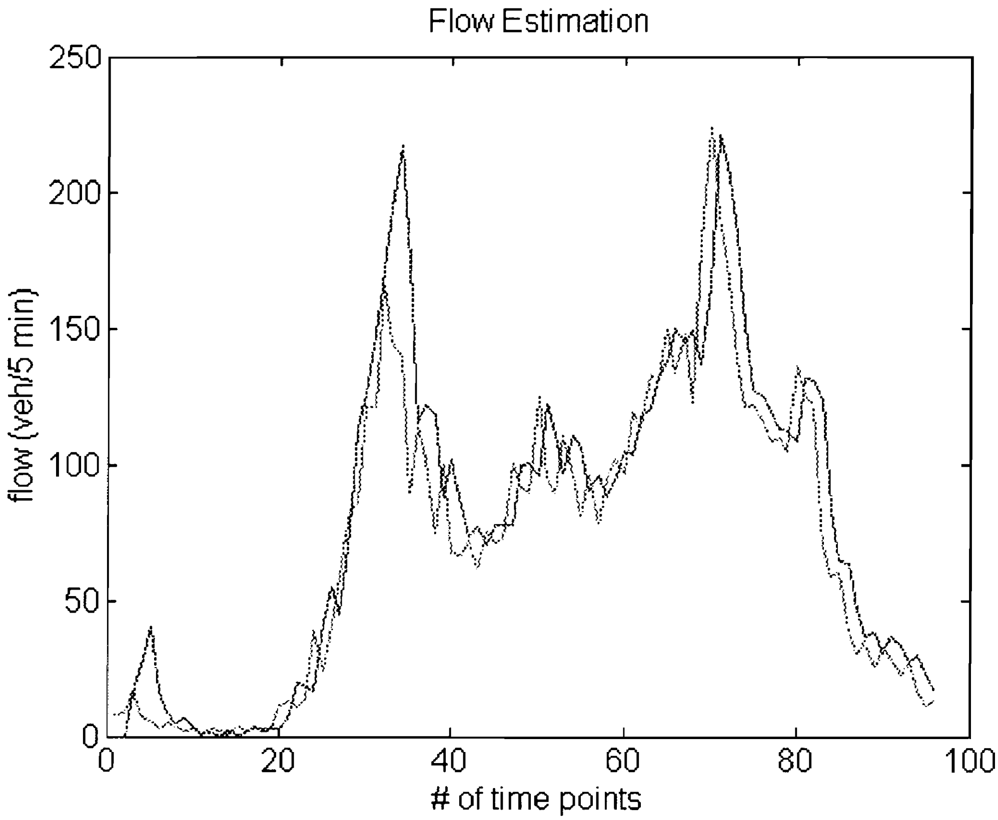


Figure 5.4.5 Two Step Lagging Flow Prediction (Blacksburg, LS w/ Forgetting Factor)

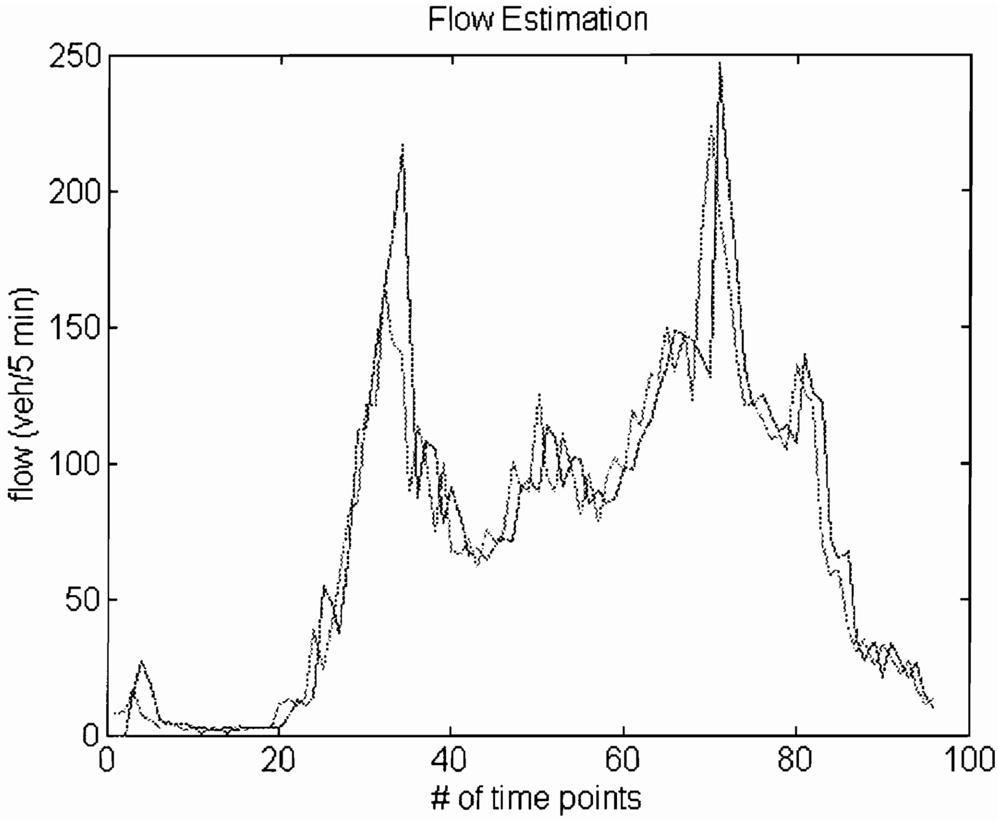


Figure 5.4.6 Two Step Lagging Flow Prediction (Blacksburg, Ljung's Approach)

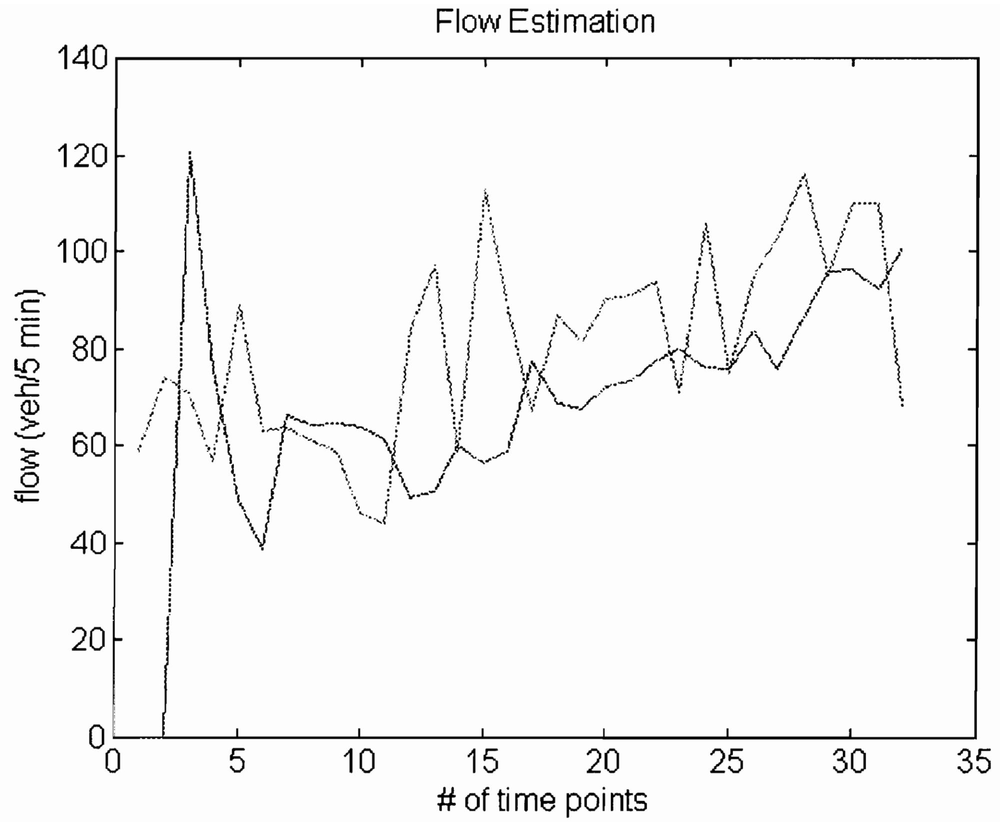


Figure 5.4.7 Two Step Lagging Prediction (INTEGRATION, LS w/ Forgetting Factor)

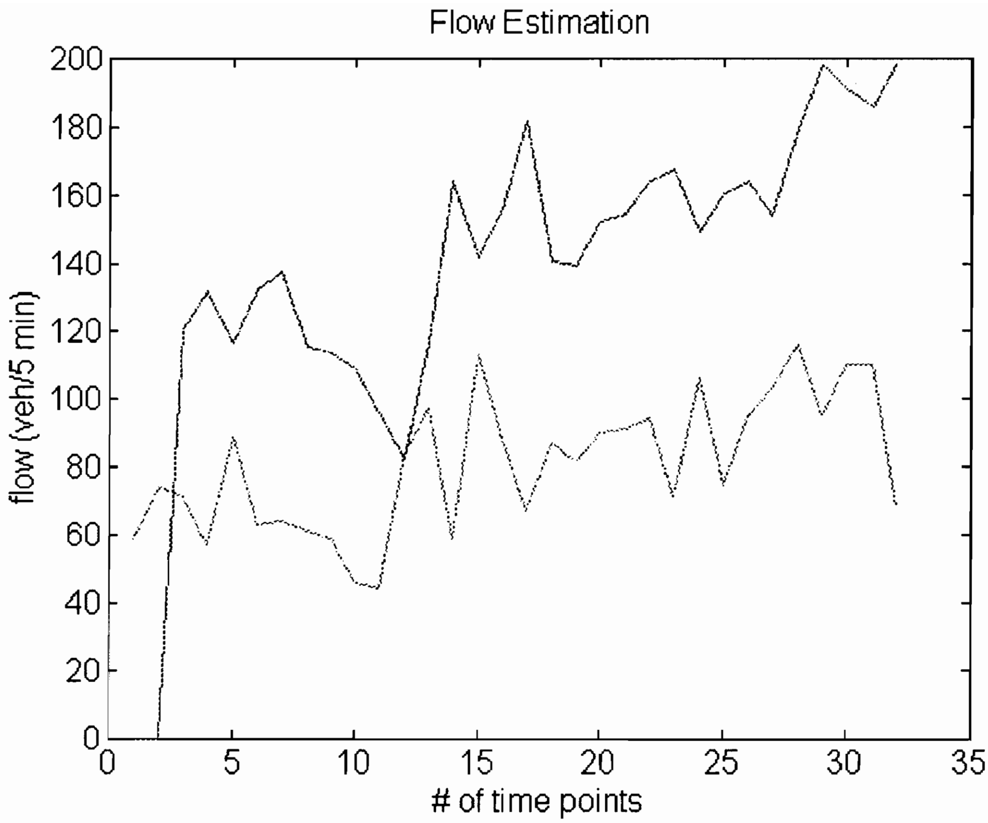


Figure 5.4.8 Two Step Lagging Flow Prediction (INTEGRATION, Ljung's Approach)

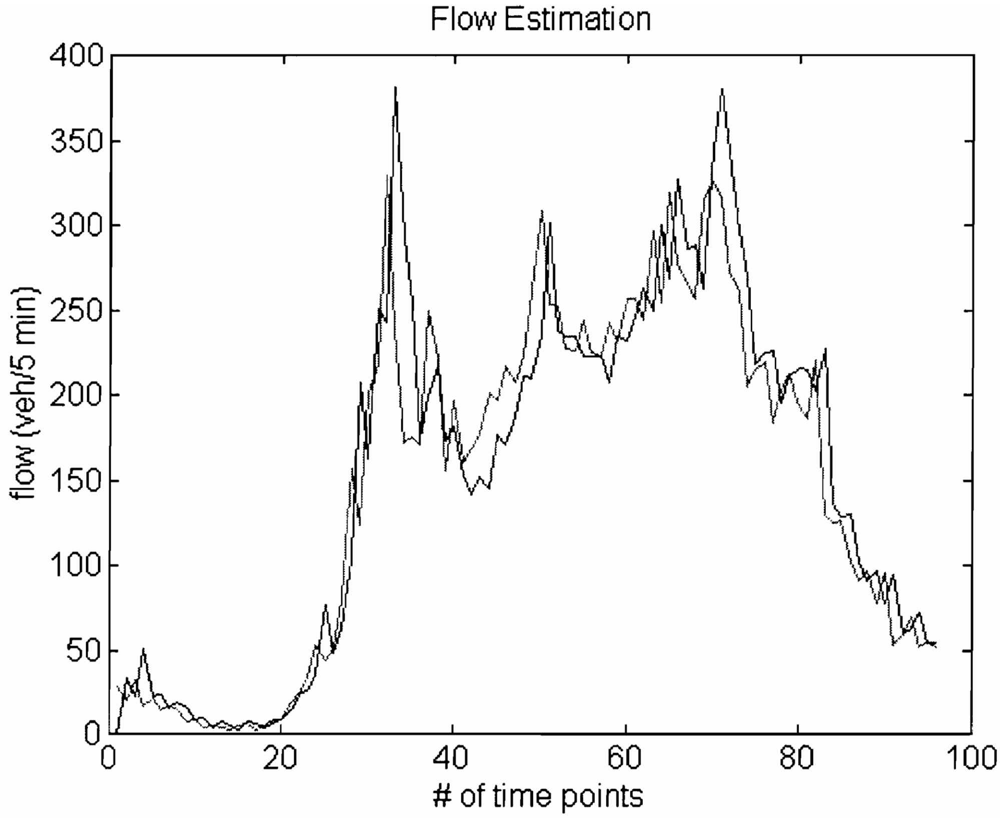


Figure 5.4.9 One Step Lagging Considering Space Domain Flow Prediction Model

(Blacksburg, LS with Forgetting Factor)

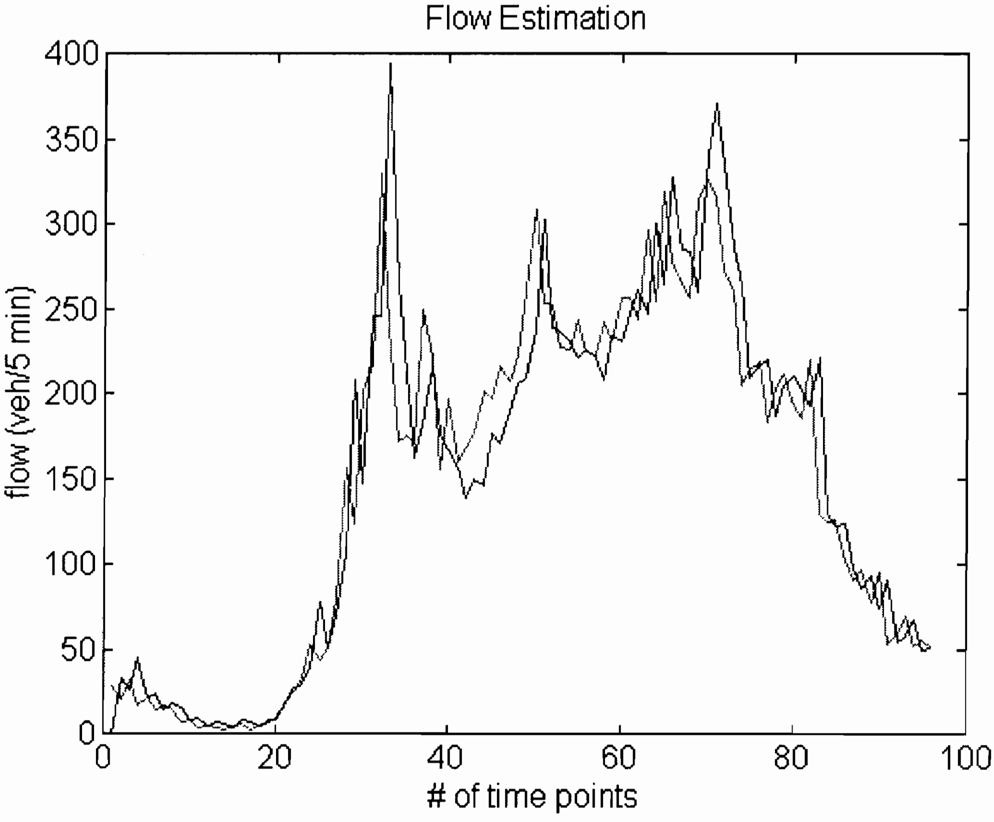


Figure 5.4.10 One Step Lagging Considering Space Domain Flow Prediction Model

(Blacksburg, Ljung's Approach)

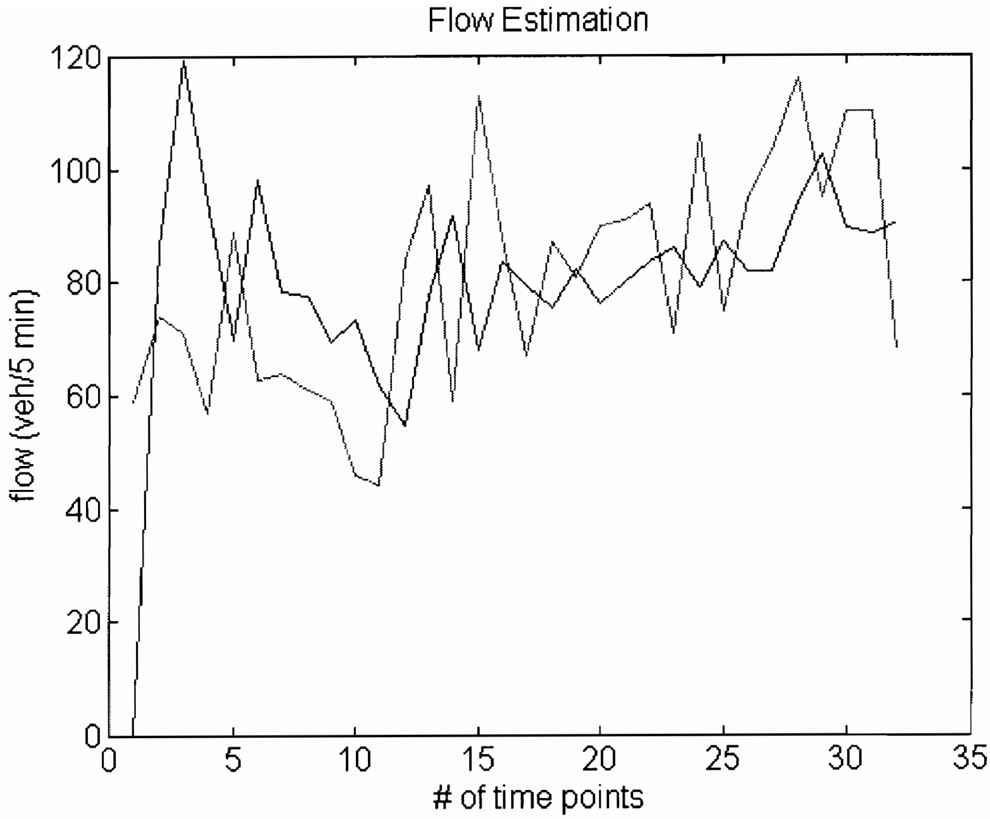


Figure 5.4.11 One Step Lagging Considering Space Domain Flow Prediction Model

(INTEGRATION, LS with Forgetting Factor)

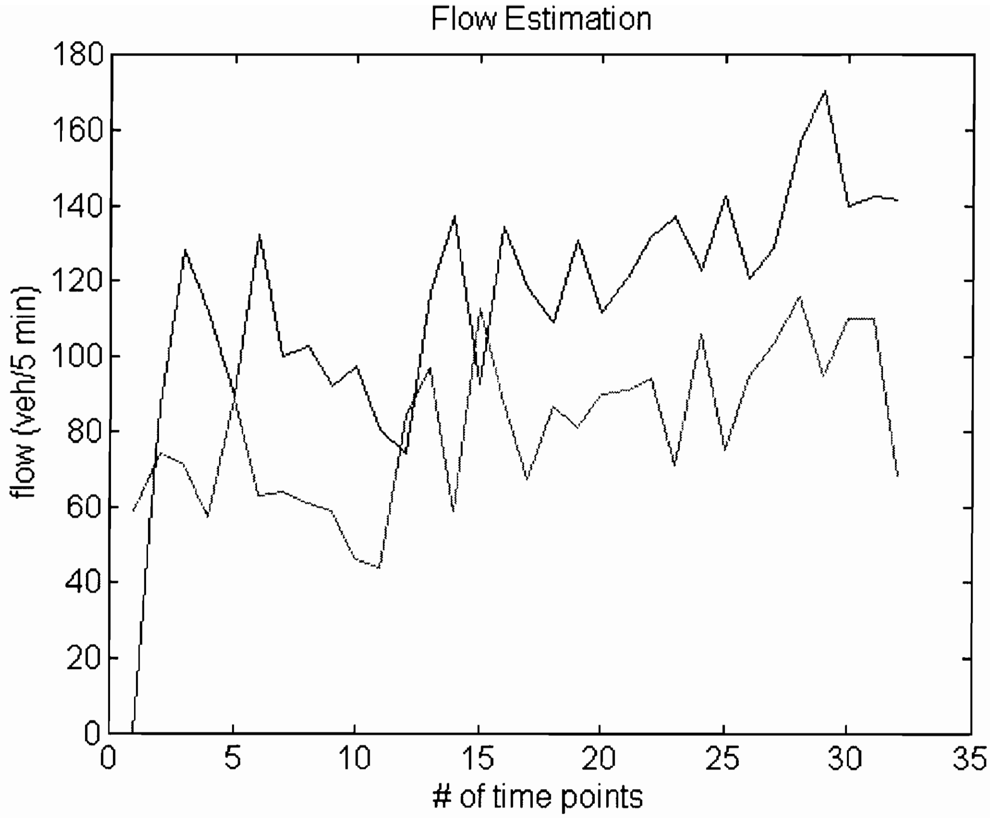


Figure 5.4.12 One Step Lagging Considering Space Domain Flow Prediction Model

(INTEGRATION, Ljung's Approach)

5.5 PARAMETER ESTIMATION

5.5.1 Initialization

The first step in the initialization for the parameter estimation is to obtain the estimated parameter vector (θ^T). Since, free flow speed (v_f), jam density (k_j), acceleration (a_1), deceleration (a_2), and time varying link flow (q) are associated in the vector, initial values are assigned for all these values. These are 55 mph for free flow speed, 240 vpm for jam density, 3.0 ft/sec^2 for acceleration and deceleration rates for buses.

One consideration made before the calculation of initial actual parameter vector was the unification in their units. Since final output better be represented in terms of minute based, all the initial values are converted into *feet/minute*. For the jam density (vehicle/mile), conversion factor (1/5280) was multiplied, and for the free flow speed (mile/hr), conversion factor (5280/60) was multiplied. Conversion factor (3600) was multiplied for the acceleration and deceleration. Therefore, from the Equation 5.1.3 initial values of parameter vector can be obtained. These are

$$a = \frac{v_f}{2v_f - k_j\left(v_f - \sqrt{v_f^2 - 4q\frac{v_f}{k_j}}\right)\mu} = \frac{55\times 88}{2(55\times 88) - 240/5280\left(55\times 88 - \sqrt{(55\times 88)^2 - 4\times 10\frac{45\times 88}{240/5280}}\right)15} = 0.5135$$

$$b = -\frac{k_j\left(v_f - \sqrt{v_f^{\,2} - 4q\dfrac{v_f}{k_j}}\right)}{2v_f - k_j\left(v_f - \sqrt{v_f^{\,2} - 4q\dfrac{v_f}{k_j}}\right)\mu} = \frac{240/5280\left(55\times 88 - \sqrt{(55\times 88)^2 - 4\times 10\dfrac{55\times 88}{240/5280}}\right)}{2(55\times 88) - 240/5280\left(55\times 88 - \sqrt{(55\times 88)^2 - 4\times 10\dfrac{55\times 88}{240/5280}}\right)15} = -0.00223$$

$$c = \frac{v_f\left(k_j - \sqrt{k_j^{\,2} - 4q\dfrac{k_j}{v_f}}\right)(a_1 + a_2)}{4a_1a_2k_j} = \frac{55\times 88\left(240/5280 - \sqrt{(240/5280)^2 - 4\times 10\dfrac{240/5280}{55\times 88}}\right)(3\times 3600 + 3\times 3600)}{4(3\times 3600)(3\times 3600)240/5280} = 0.0214$$

$$d = \frac{1}{\dfrac{v_f\left(k_j + \sqrt{k_j^{\,2} - 4q\dfrac{k_j}{v_f}}\right)}{2k_j}\left[1 - \dfrac{\mu k_j\left(v_f - \sqrt{v_f^{\,2} - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}\right]}$$

$$= \frac{1}{\dfrac{55\times 88\left(240/5280 + \sqrt{(240/5280)^2 - 4\times 10\times\dfrac{240/5280}{55\times 88}}\right)}{2(240/5280)}\left[1 - \dfrac{15\times 240/5280\left(55\times 88 - \sqrt{(55\times 88)^2 - 4\times 10\dfrac{55\times 88}{240/5280}}\right)}{2v_f}\right]} = 0.00011$$

$$e = -\frac{\mu}{\dfrac{v_f\left(k_j + \sqrt{k_j^{\,2} - 4q\dfrac{k_j}{v_f}}\right)}{2k_j}\left[1 - \dfrac{\mu k_j\left(v_f - \sqrt{v_f^{\,2} - 4q\dfrac{v_f}{k_j}}\right)}{2v_f}\right]}$$

$$= -\frac{15}{\dfrac{55\times 88\left(240/5280 + \sqrt{(240/5280)^2 - 4\times 10\times\dfrac{240/5280}{55\times 8}}\right)}{2(240/5280)}\left[1 - \dfrac{15\times 240/5280\left(55\times 88 - \sqrt{(55\times 88)^2 - 4\times 10\dfrac{55\times 88}{240/5280}}\right)}{2v_f}\right]} = -0.00168$$

5.5.2 Simulation

Physical layout of the simulation consists of four bus stops including three intersections. 96 buses were simulated based on this configuration. Time headway of buses are assumed to be fifteen minutes. Sublinks and link lengths are assumed as 0.7, 0.8, 0.9, 1.8, 1.6, and 2.5 miles (refer to Figure 5.3.1). At the first stage of the simulation, bus travel time profiles were investigated according to the ratio of passenger arrival rate and passenger boarding rate. The result was plotted in the time-space domain in Figure 5.5.1.

Later in the simulation, whenever a bus leaves either the intersection or the bus stop, the model has capability to update its arrival time for the downstream bus stops. Thus, when the arrival time of the last (forth) bus stop is estimated, there should be seven updates. It means estimation of arrival time is getting better and better when the bus traveling toward the downstream (forth) bus stop. At the same time, when the bus leaves the bus stop, parameters will be updated, and updated parameter vectors will be passed onto the next bus. Hence, the next bus at the same bus stop will adopt the recently updated parameter values in their travel time estimation. It implies as buses pass by the bus stops, the estimation is getting better and better toward the actual travel time. Matlab source code for integrated arrival time estimation is attached to Appendix D.

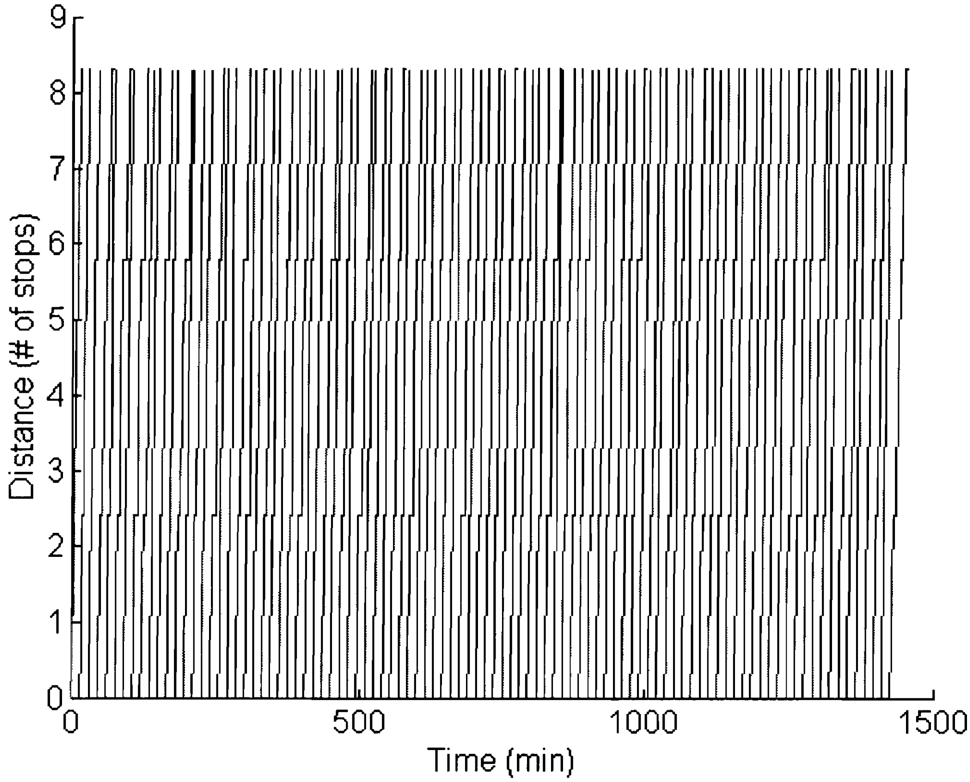


Figure 5.5.1 Bus Travel Time Simulation

An assumption made in this simulation is that when arrival time is updated, the estimation should use estimated flow at that time. In this simulation, however, this concept was not fully incorporated due to data unavailability as well as further refinement for the dynamic flow estimation. Instead, the real flow data obtained from Blacksburg was used as if it is dynamically estimated flow. Another assumption made in this simulation is that the number of vehicle in a link is assumed to be the 95 percent of flow. It should be either measured or estimated dynamically, if cumulated flow data are available.

5.5.3 Results

One of the main outputs in the simulation is the arrival time of the bus at each bus stops. The outputs of arrival time estimation for the second bus at the second, third, and forth bus stops are illustrated in the Figure 5.5.2. As it was discussed, apparently three updates were made for the second bus stop due to two intersections in between the stops. Parameter convergences are the second output, and these are illustrated in the Figures of 5.5.3 through 5.5.5. Parameter convergences were tested for the bus stops only, because five parameter vector components are associated with each intersections and bus stops. The results show that parameter convergence is also acceptable for all the bus stops. Finally, estimation error at each intersections and bus stops are shown in Figures of 5.5.6 through 5.5.14. Majority of estimation errors fall into the very small error margin. Therefore, it can be said that the estimation is acceptable.

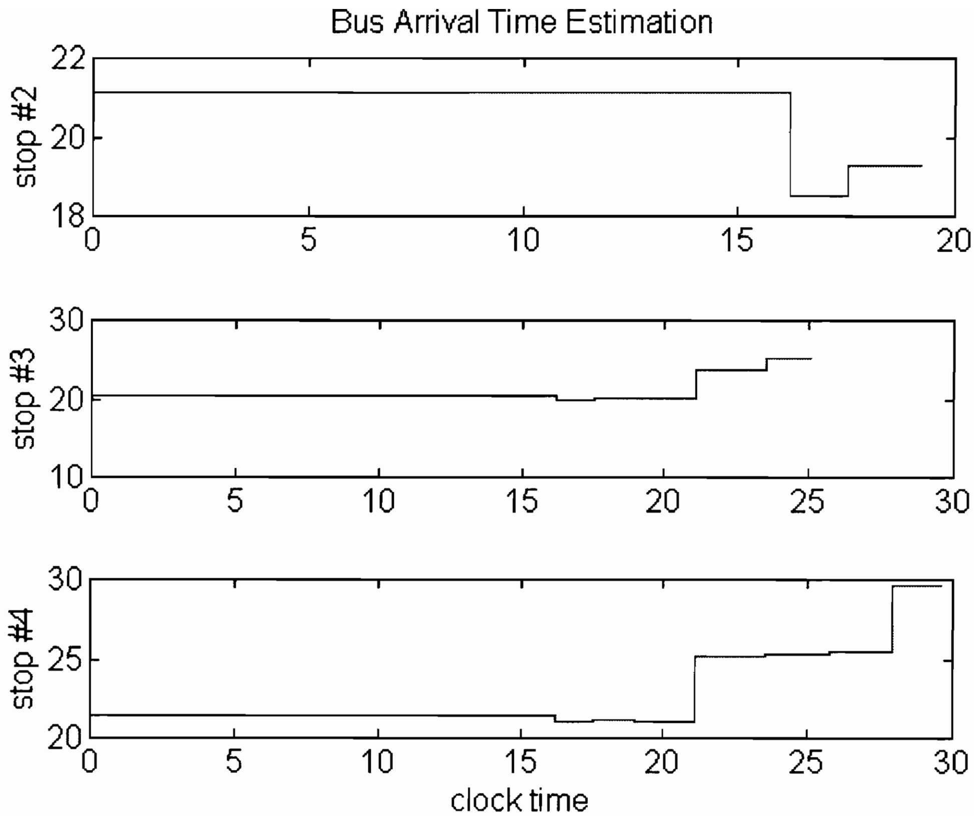


Figure 5.5.2 Estimated Arrival Time (Integrated Model)

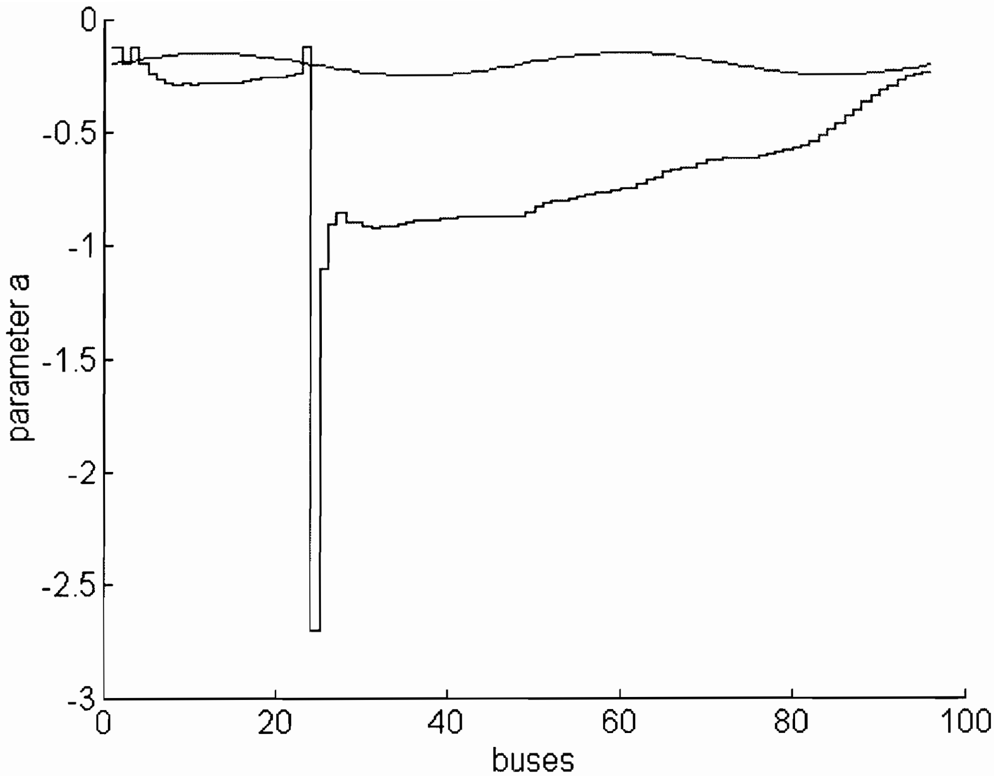


Figure 5.5.3 Parameter Convergence (Parameter 'a' at 2nd Bus Stop)

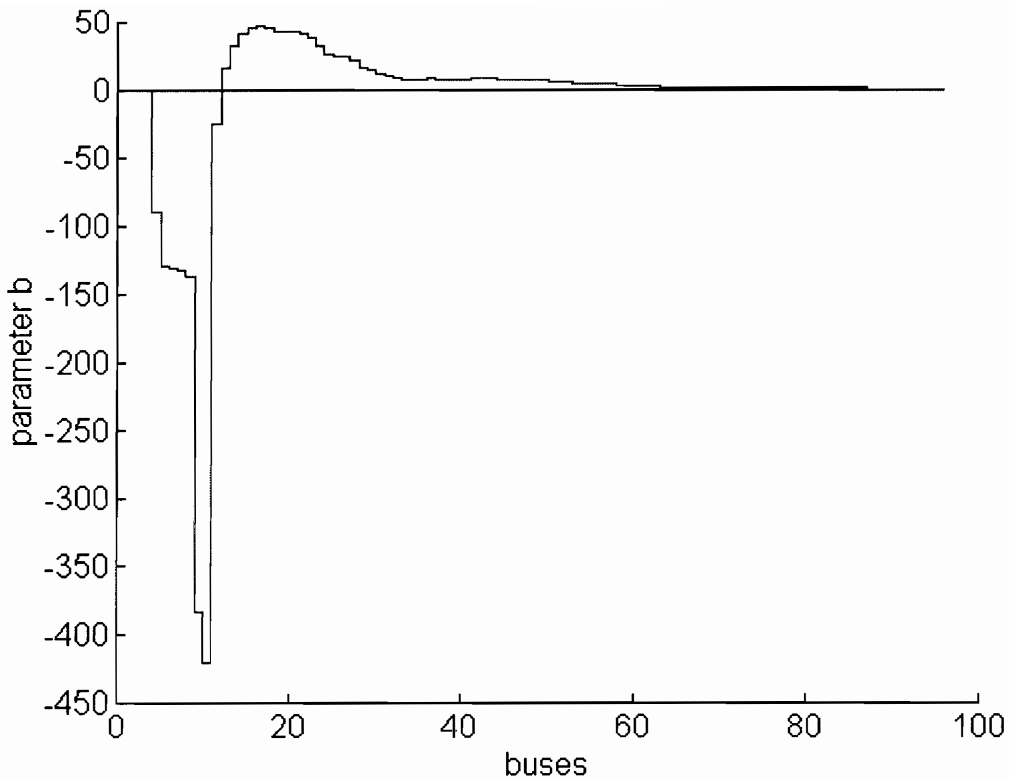


Figure 5.5.4 Parameter Convergence (Parameter 'b' at 2nd Bus Stop)

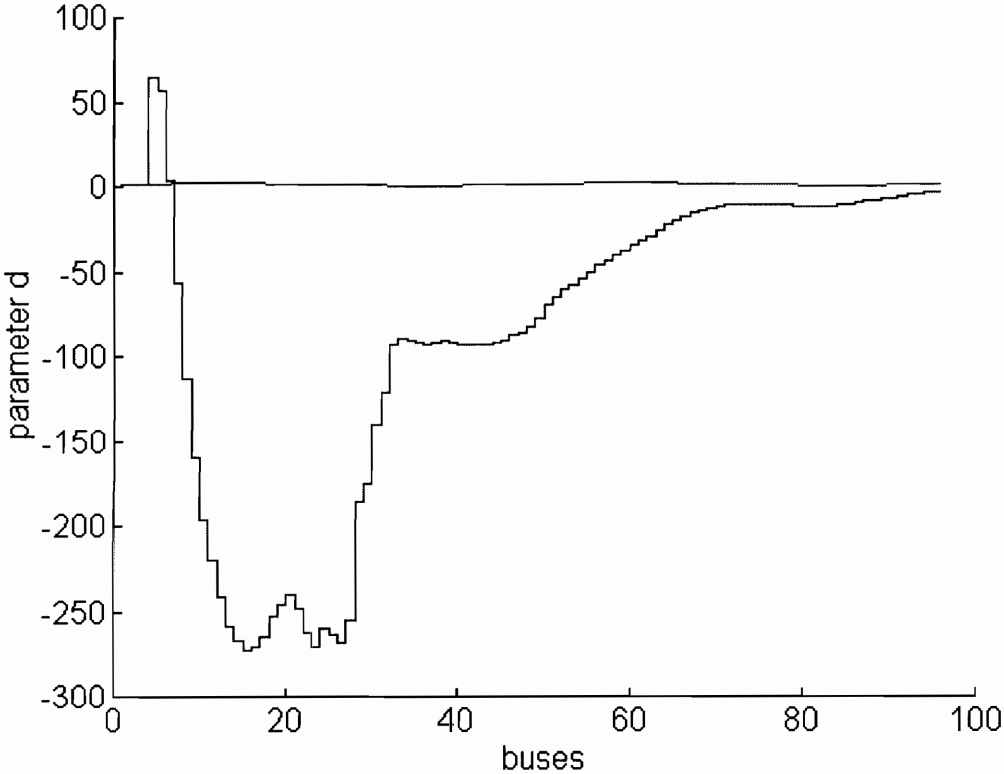


Figure 5.5.5 Parameter Convergence (Parameter ‘d’ at 2nd Bus Stop)

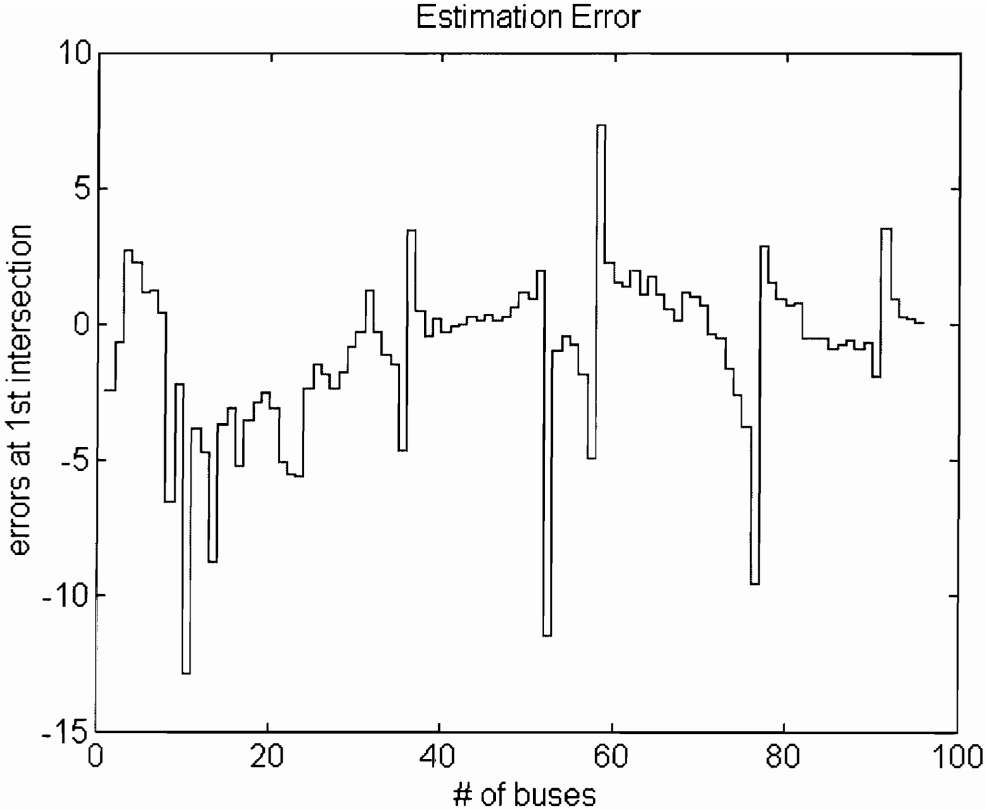


Figure 5.5.6 Estimation Error for Intersection #1

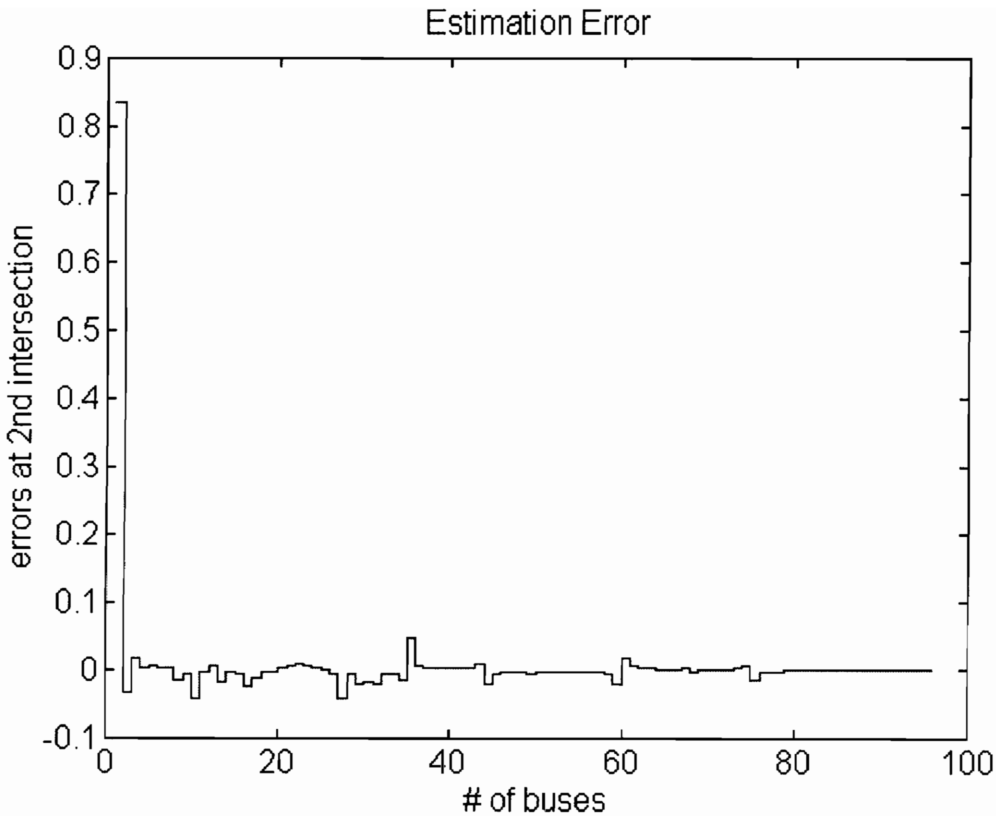


Figure 5.5.7 Estimation Error for Intersection #2

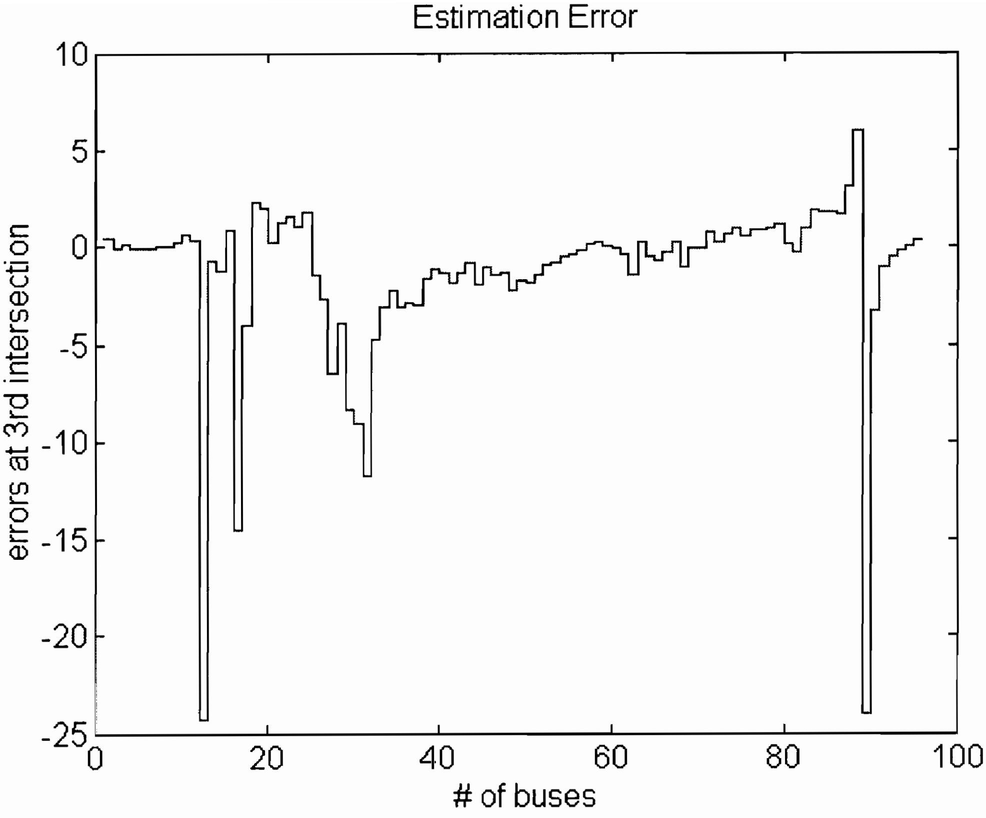


Figure 5.5.8 Estimation Error for Intersection #3

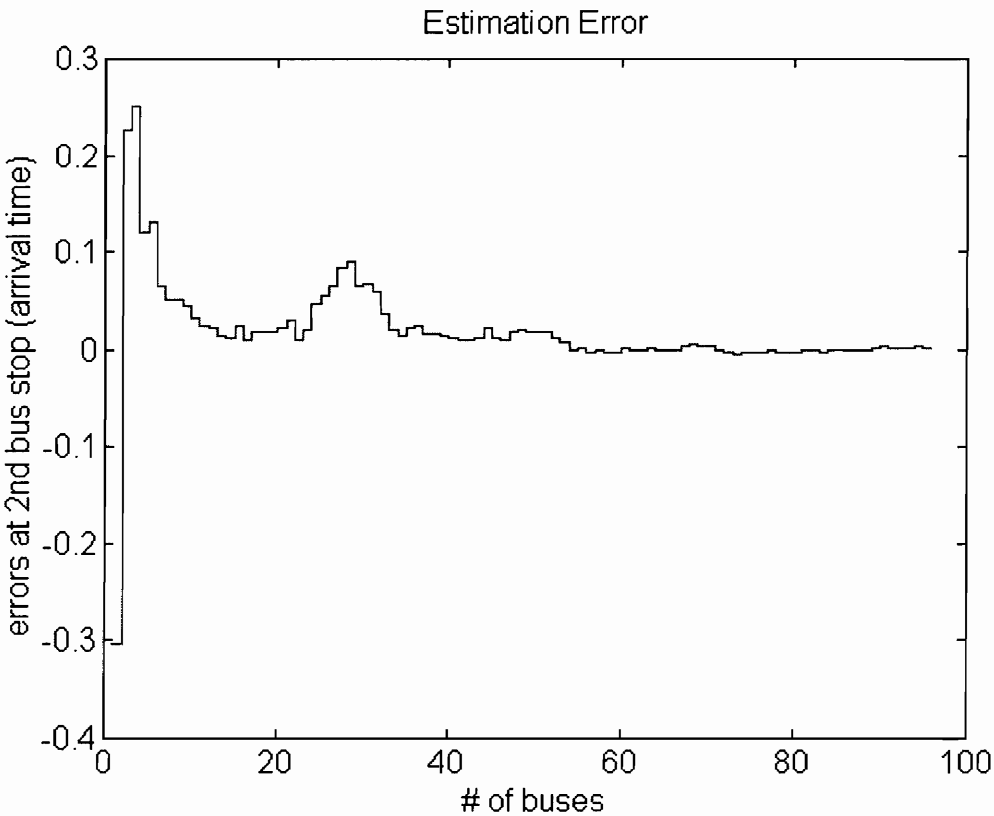


Figure 5.5.9 Estimation Error for Bus Stop #2 (Arrival Time)

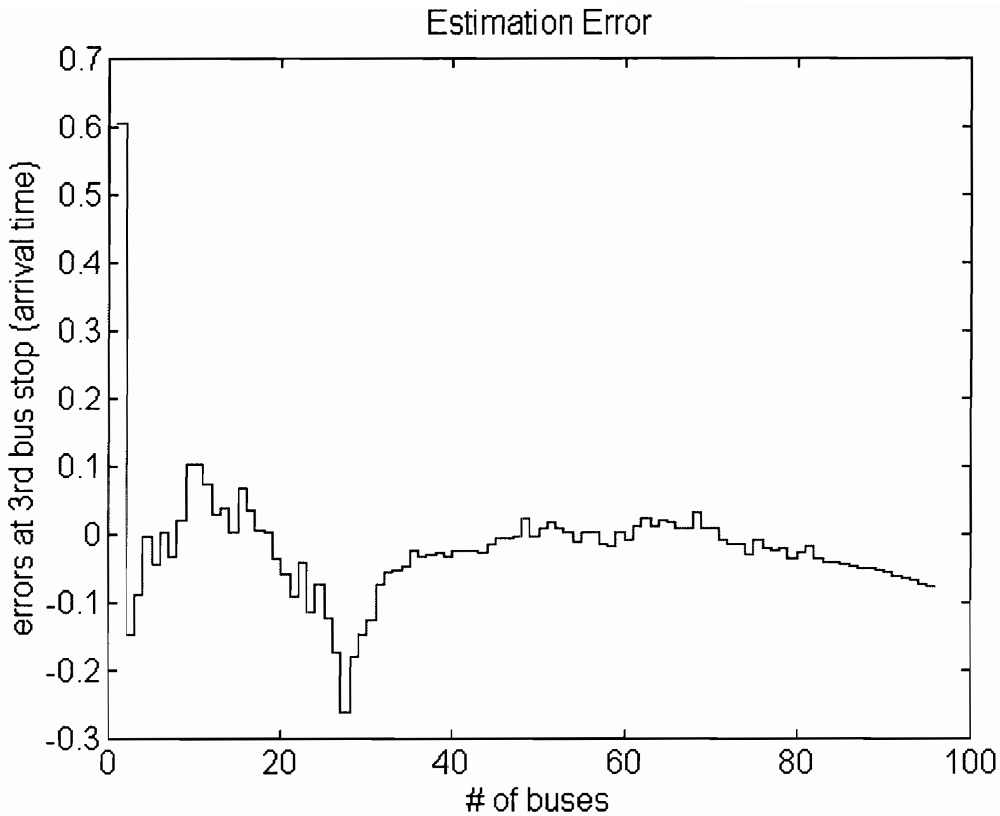


Figure 5.5.10 Estimation Error for Bus Stop #3 (Arrival Time)

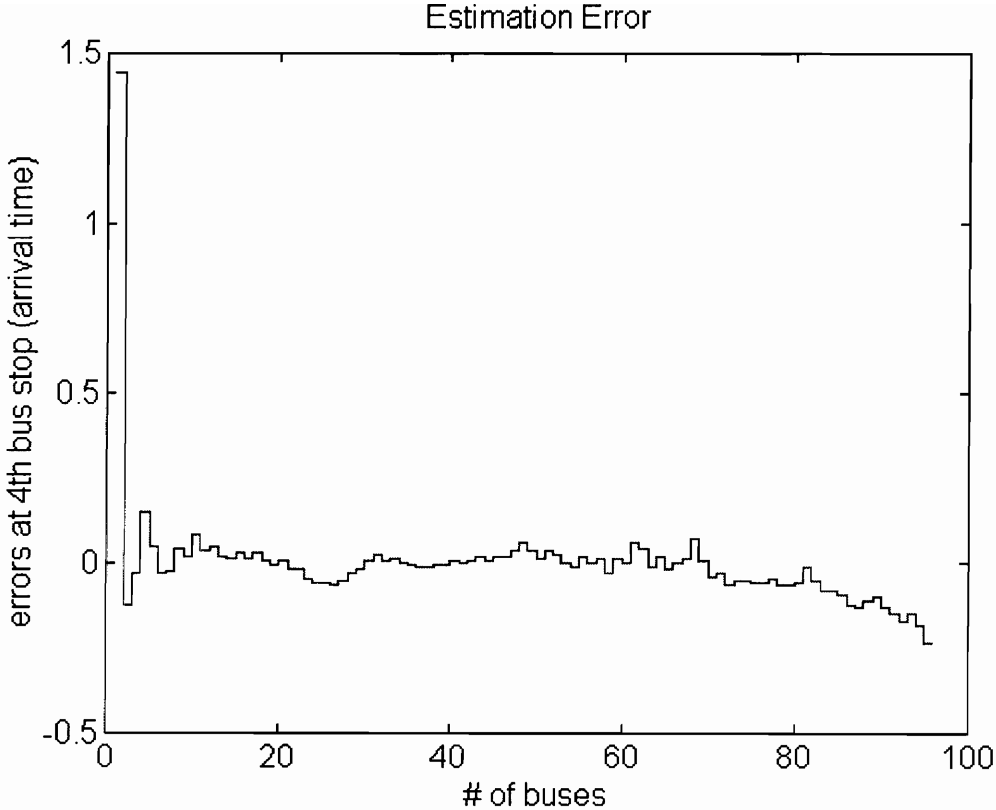


Figure 5.5.11 Estimation Error for Bus Stop #4 (Arrival Time)

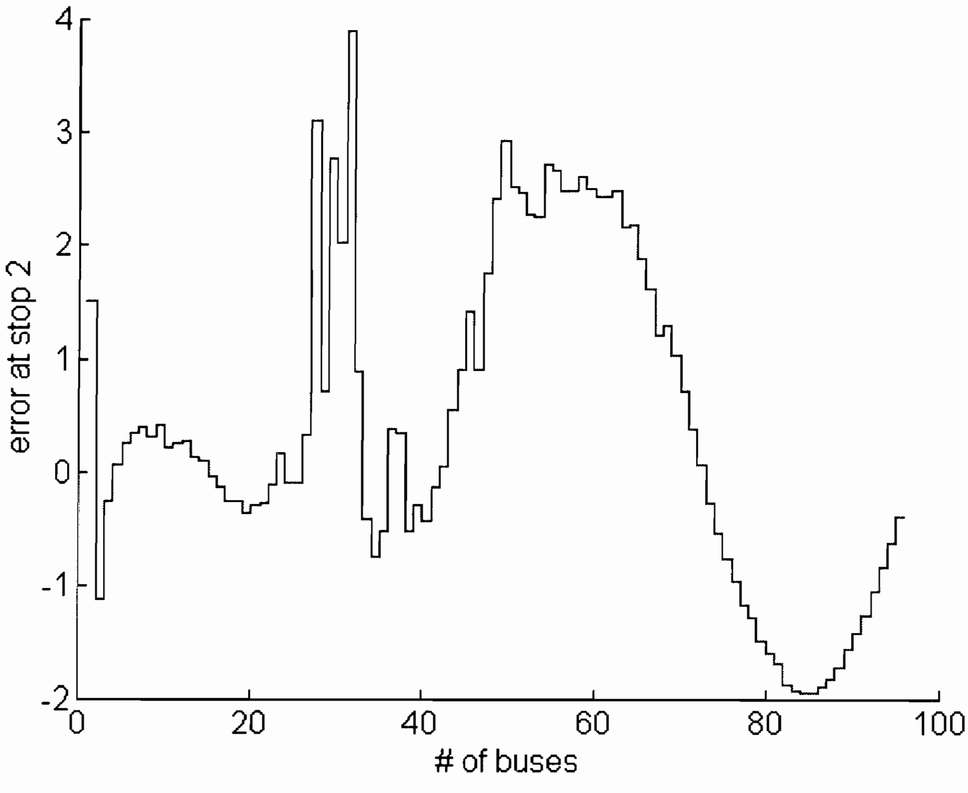


Figure 5.5.12 Estimation Error for Bus Stop #2

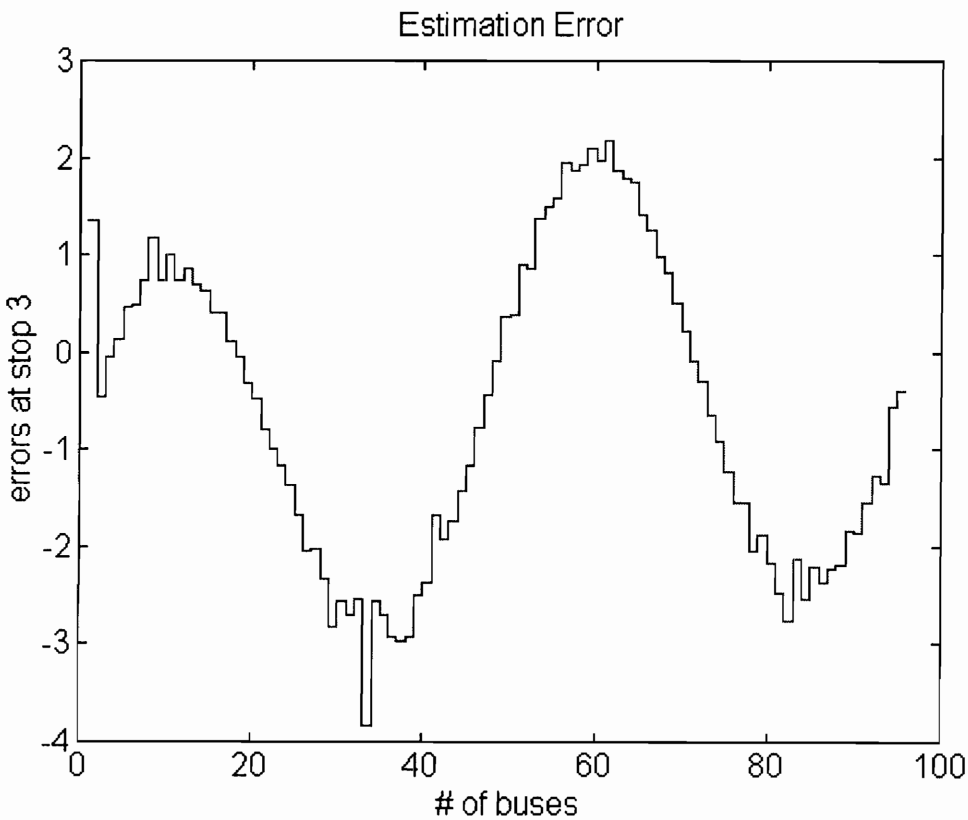


Figure 5.5.13 Estimation Error for Bus Stop #3

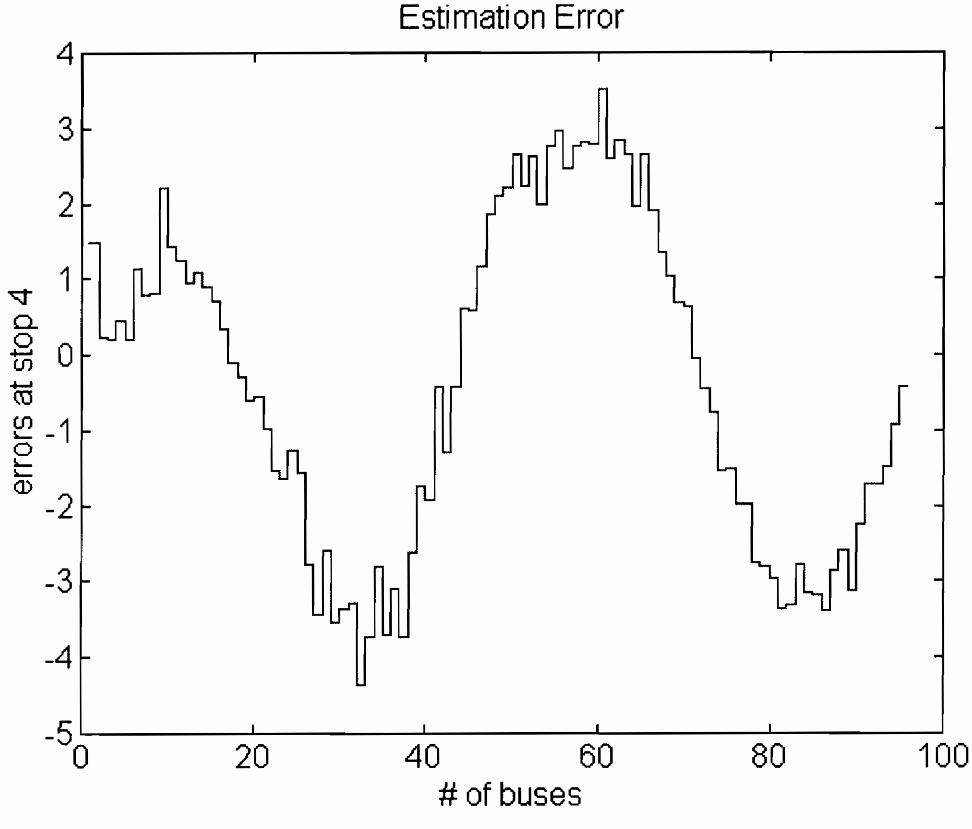


Figure 5.5.14 Estimation Error for Bus Stop #4

In the mean time, forgetting factor was introduced, and tested in the second simulation. Also, Ljung's algorithm which dealing with the decreasing gain was also introduced, and tested. The test results of both of these approach generates slightly better results than the generic least square parameter adaptation algorithm but not much, and for some cases these approaches generated worse results than LS algorithm. Thus, for the conclusion of estimation, it can be said that generic least square algorithms is acceptable.

CHAPTER VI AUTO TRAVEL TIME FROM PROBE VEHICLE

6.1 INTRODUCTION

Maximizing the utility of current Automatic Vehicle Location (AVL) bus system is one of the main goals of this research. Under this circumstances, the use of AVL system equipped bus as a probe vehicle to collect the data and estimate its arrival time was one of the important task better to be incorporated in the future AVL system. On the other hand, obtaining the information of non-transit (auto) travel time from the AVL system equipped probe bus is the other goal that is proposed and must be fulfilled in this phase of the research. For the Intelligent Transportation Systems, real-time travel time data plays such an important role in establishing the various real-time decision making and mid to long-term planning processes. Since both buses and cars are operated on the same roadway at the same time under the same traffic condition, there should be some correlation between these two modes.

In a sense, the travel time of car might be expressed as a function of travel time of bus for a given route. According to this initial understanding, this phase of the research is initiated to identify the direct functional relationship between these two variables, such as

$$\textit{car travel time} = f(\textit{bus travel time}) \qquad 6.1.1$$

Mapping of bus and car travel time can be made by excluding the dwelling time at a bus stop, considering the difference of acceleration and deceleration rates of bus and car at a stop and a intersection, and flow condition of link at certain time interval. Therefore, simple conceptual model which addresses this concept can be reformulated as:

$$\textit{car travel time} = \textit{bus travel time model} - \sum \textit{bus dwelling time} - \sum \textit{accel. and decel.}$$

Mathematical approach was attempted to prove this hypothesis. Simply, bus travel time (BTT) is assumed to be a nonlinear function of two different independent variables, and car travel time (CTT) also has the same relationship as the CTT. For instance,

$$CTT = 4x^2 + 6y \qquad 6.1.2$$

$$BTT = 2x^2 + 3y \qquad 6.1.3$$

In this case, we can simply identify the relationship as

$$CTT = 2BTT \qquad 6.1.4$$

For the general case, CTT and BTT can be expressed as

$$CTT = f_1(x, y) \qquad 6.1.5$$

$$BTT = f_2(x, y) \qquad 6.1.6$$

Then, we try to obtain one dimensional function ψ such that

$$CTT = \psi(BTT) \qquad 6.1.7$$

$$CTT = \psi \cdot f_2(x,y) \qquad 6.1.8$$

If CTT and BTT are one dimensional form, and BTT is a strictly increasing (or decreasing) function, then we can get inverse function, and identify the one dimensional function ψ. However, if CTT and BTT have more than one dimensional functional relationships, except the special case (e.g., example in Equation 6.1.4), there is no guarantee to have such a direct relationship. For this reason, calibration is a way to solve more than two dimensional function. The main mechanism embedded in the calibration is to minimize the sum of square error (SSE). Regression, and Artificial Neural Networks are another alternatives to solve this problem.

6.2 DATA ACQUISITION

In order to fulfill this task, data collection on bus travel time was assumed to be obtained from the AVL bus probe, however, in reality existing systems do not generate the sufficient data for this task, specifically arrival time and departure time at each bus stops. Also, there is no guarantee that conventional AVL systems generate reliable data for arrival time and departure time. In this viewpoint, real data were decided to be acquired by field observation. Basically, an observer was on board, and measured the arrival time and departure time of the bus at the major bus stops. At the same time, the observer counted the number of passengers loading and unloading while the bus travels bus stop to bus stop.

Car travel time, which is measured synchronously with the bus travel time, was also collected for this task in order to identify the coreliarity with the bus travel time. In order to compare the travel time of these two modes at the same flow condition, which was assumed to be an important factor influencing the travel time of vehicle, departure time of bus and car at each bus stops was synchronized. Later at the major bus stop, the car which left at the same time with the bus, but arrived at the bus stop earlier than the bus (usual case), waited for the bus, then car and bus can depart at the same time again.

Two study areas were identified. Blacksburg, Virginia was the first study areas for this study, and Norfolk, Virginia was the other study area. The reason for choosing Blacksburg was due to the facts that it is cost effective in data collection and Blacksburg has fairly good bus transit system. On the other hand, city of Norfolk has a lot bigger bus transit system, and its characteristics almost represent the metropolitan traffic flow condition, especially during the am and pm peak hours.

Data acquisition for Blacksburg was conducted for two days, Wednesday and Thursday. It was started at 7:15 am, and ended at 5:45 in the afternoon. Blacksburg transit operates

24 buses on its eight routes during the peak hours. Daily ridership is 8,5000 passengers while the school is in session, and it covers approximately 642,000 miles per year, including demand-respond system. Among its eight routes, Yellow Line and Blue Line were selected as study routes. In fact, these two routes cover all the major road network in Blacksburg. Yellow Line (Winsor Hills to Hethwood route) covers east and west side of the town. Whereas Blue Line (North Main to South Main route) accommodates the riders in north and south region of the town. Since the running time for round trip of Yellow and Blue Lines takes only half an hour, data collection on entire routes were conducted. In total, 133 man-hours were invested to obtain 189 data points (link data) for the study. Data entry forms for bus and car were prepared, and these are attached to Appendix E. Area map of Blacksburg is attached to Appendix F.

Meanwhile, the city of Norfolk was selected as another study area. Tidewater Regional Transit (TRT) in Norfolk operates 92 buses on 30 routes in the areas of Hampton, Virginia Beach, and Chesapeake in Virginia, covering a total distance of 12,800 miles per day. It transports 25,000 passengers daily. In terms of its transit functionality, Norfolk is able to represent one of the major cities in the United States. Hence, in order to be contrast to the light flow condition in Blacksburg, Norfolk was selected as an alternative study area. Among TRT's 30 routes, one of the most congested routes, Route #15, was chosen as study route. Route #15 starts from the Military Circle (big shopping mall) near the downtown and ends at the Naval Air Station (northwest region of Norfolk which attracts lots of trips throughout the day). Since round trip takes 2 hours, some of its major bus stops were extracted. Then, data collection was conducted on the extracted links, which starts from the junction of Little Creek road and Granby to Military Circle. In total, six major bus stops (five links) were chosen as data collection locations. Route map is attached to Appendix G. The data collection started 6:15 in the morning till 7:45 in the evening for two days (Thursday and Friday). 176 link data were collected, and 81 man-hours were invested for Norfolk data collection.

6.3 A STATISTICAL APPROACH

Data for Blacksburg and Norfolk were coded in the PC MS-EXCEL (version 5.0). Finding out the correlation between bus travel time and car travel time was the main interest in this task, all the paired link data (bus and car travel time) were plotted in one graph. Then, simple linear regression analysis on all the data sets for Blacksburg and Norfolk was conducted separately. First, the main variable which was interested in was bus and car travel time. Discussions on this analysis are as follows:

6.3.1 Analysis on Blacksburg Data

As illustrated in the Table 6.3.1, the result of regression reveals statistically sound (high R-square value, 0.79) relationship between two variables. Figure 6.3.1 also illustrates plots

Table 6.3.1 ANOVA Table for Bus vs. Auto Travel Time in Blacksburg

SUMMARY OUTPUT

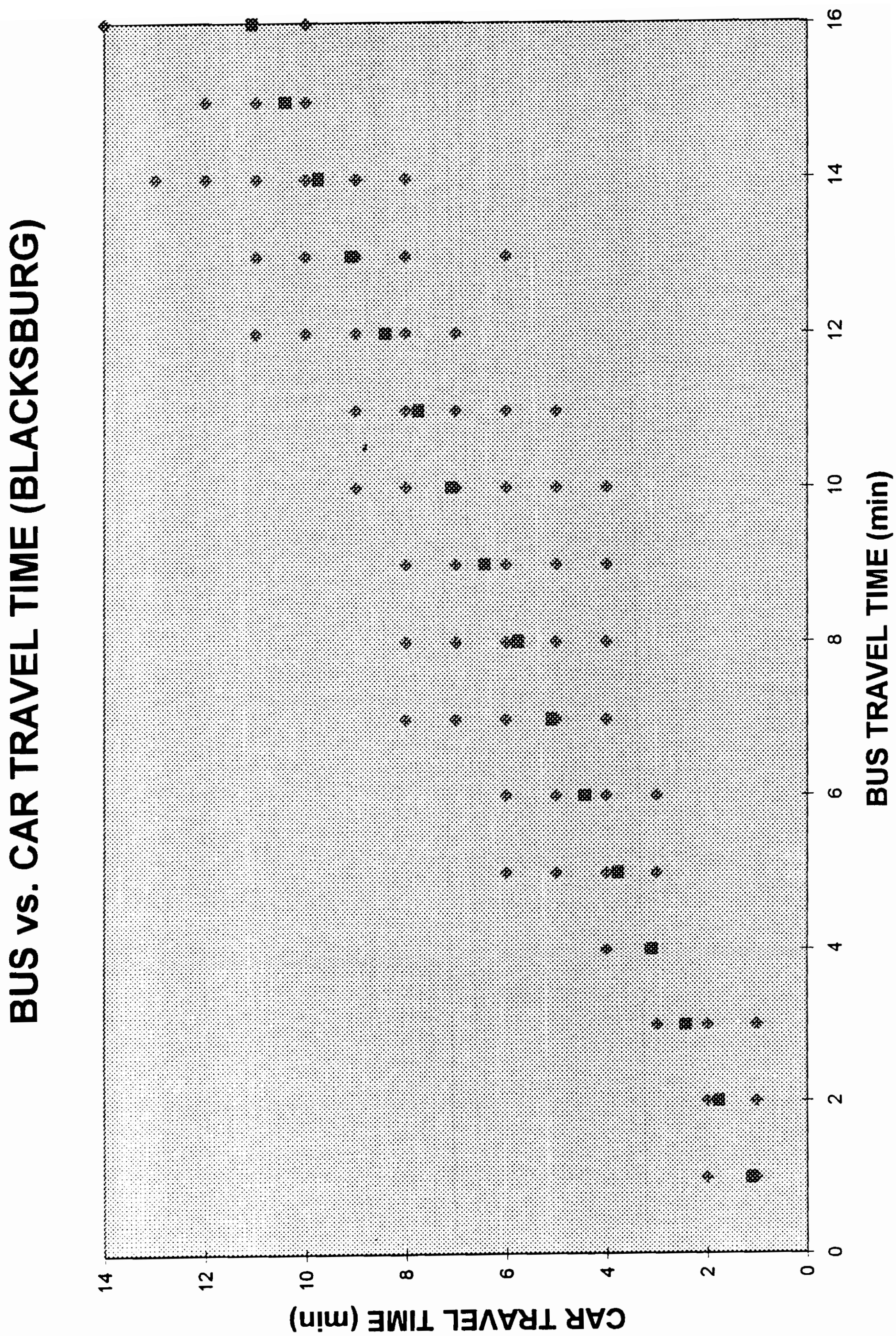
Regression Statistics	
Multiple R	0.889915393
R Square	0.791949407
Adjusted R Square	0.790818698
Standard Error	1.265618493
Observations	186

ANOVA

	df	*MS*	*F*	*Significance F*
Regression	1	1121.894	700.400269	1.23176E-64
Residual	184	1.60179		
Total	185			

	Coefficients	*t Stat*	*P-value*	*Lower 95%*	*pper 95*	*ower 95.000*	*Upper 95.000%*
Intercept	0.429459842	1.84307	0.06692754	-0.030261368	0.889181	-0.03026137	0.889181052
X Variable 1	0.665057187	26.46508	1.2318E-64	0.615477974	0.714636	0.615477974	0.714636399

Figure 6.3.1 Blacksburg Data and Regression (Bus vs. Auto Travel Time)



of the data and regression line. For this result, some of the outliers (3 data points) were post processed. The residual table and plot (refer to Appendix H and I) also illustrate that the maximum residual value was 3.3 minute.

In conclusion of this analysis, the relationship between the bus travel time and car travel time in Blacksburg can be represented as

$$y = 0.429459842 + 0.665057187x$$

where, y = car travel time

x = bus travel time

Although it indicates the site-specific relationship between car and bus travel time, its results provide the good and concise indication of travel time correlations between two different major modes of transportation.

6.3.2 Analysis on Norfolk Data

During the period of data collection, 3 incidents (one accident and two road maintenance) were occurred. It represents the traffic situation in bigger city well. Among the 176 initial data points, 8 outliers were post processed. The result of regression after the post process is illustrated in Table 6.3.2. From Table 6.3.2, R square value is high enough to be accepted (0.60), and all the test results are statistically significant. The plots of data points and regression line are shown in Figure 6.3.2. As it is also depicted in the residual table

Table 6.3.2 ANOVA Table for Norfolk (Bus vs. Auto Travel Time)

SUMMARY OUTPUT

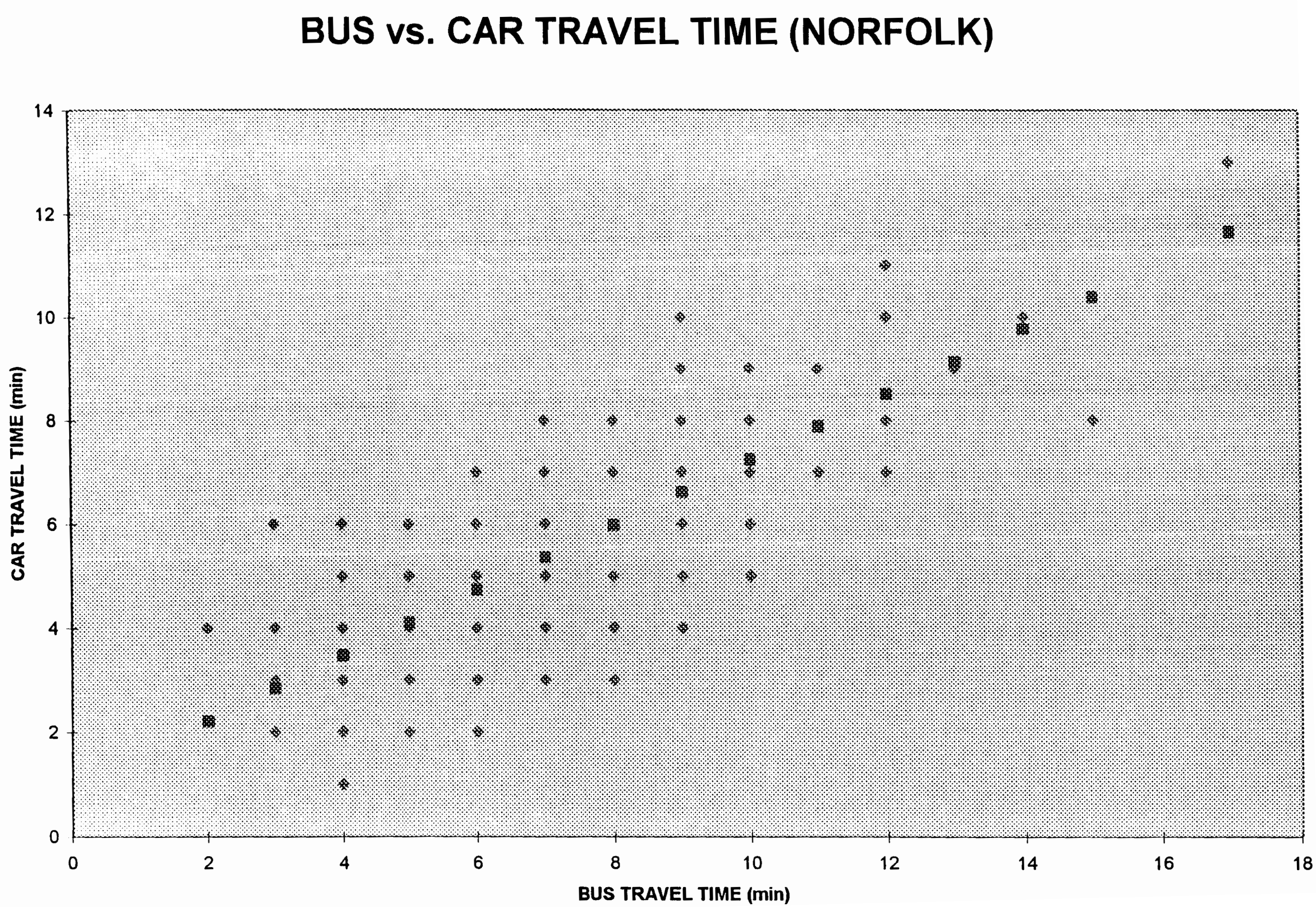
Regression Statistics	
Multiple R	0.779392018
R Square	0.607451918
Adjusted R Square	0.60508717
Standard Error	1.409632409
Observations	168

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	510.4331686	510.4331686	256.8781	1.55E-35
Residual	166	329.8525457	1.987063528		
Total	167	840.2857143			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*ower 95*	*pper 95*	*Lower 95.000%*	*pper 95.000%*
Intercept	0.948274829	0.291661286	3.251287969	0.001391	0.372431	1.524119	0.372430925	1.524118733
X Variable 1	0.628722889	0.039227958	16.02741815	1.55E-35	0.551273	0.706173	0.551272841	0.706172937

Figure 6.3.2 Norfolk Data and Regression (Bus vs. Auto Travel Time)



and plot, its maximum residual is 3.39 minutes. Its result is also acceptable and illustrated in Appendix J and K.

Hence, it can be concluded for this analysis that travel time relationship between bus and car at Route 15 in Norfolk will be

$$y = 0.948274829 + 0.628722889x$$

6.3.3 Further Analysis

Since factors affecting bus travel times are various (e.g., link length, number of stops that bus made while it runs in between the major bus stops, number of intersections, etc), simple linear regression considering travel time only is not enough to capture the complete correlation between these two variables. Thus, link length is introduced in order to be more descriptive analysis than bus travel time versus car travel time.

After considering link lengths (i.e., bus speed vs. car speed) for each data points in Blacksburg and Norfolk, the overall R-square values went down. R-square values for Blacksburg and Norfolk were 0.35 and 0.43, respectively. However, its analysis results are more descriptive than the previous (time only) one. Especially, for the Blacksburg, all the residual values are in the range of ±10 mph. The following two tables (Table 6.3.3 and Table 6.3.4) and two figures (Figure 6.3.3 and Figure 6.3.4) illustrate the bus speed

Table 6.3.3 ANOVA Table for Blacksburg (Nus vs. Auto Speed)

SUMMARY OUTPUT

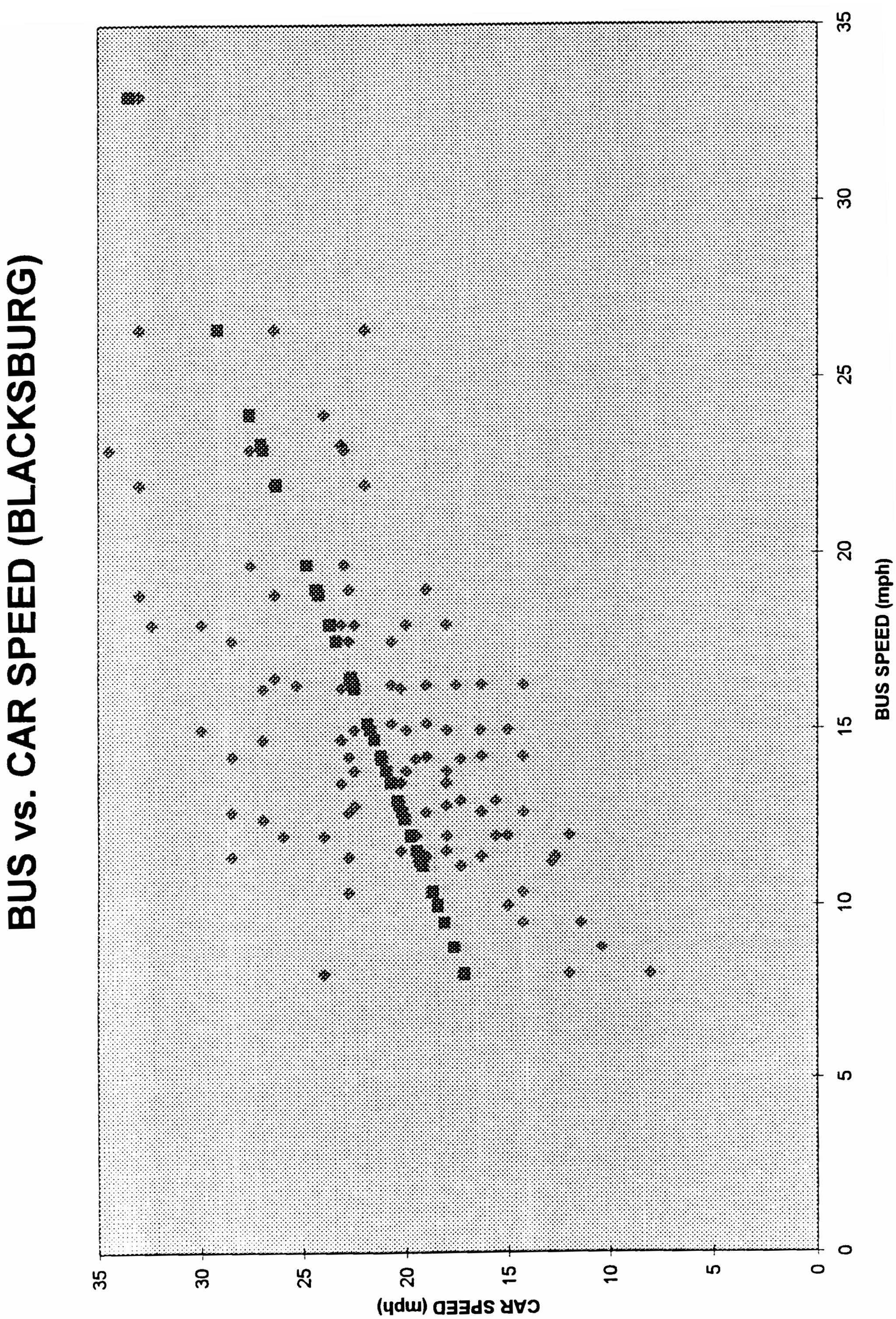
Regression Statistics	
Multiple R	0.587712187
R Square	0.345405614
Adjusted R Square	0.341768979
Standard Error	4.202361824
Observations	182

ANOVA

	df	SS	MS	F	ignificance F
Regression	1	1677.322243	1677.322	94.97944	2.739E-18
Residual	180	3178.772082	17.65984		
Total	181	4856.094325			

	Coefficients	tandard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.000%	Upper 95.000%
Intercept	11.87998825	1.095941244	10.83999	2.15E-21	9.7174434	14.042533	9.717443443	14.04253306
X Variable 1	0.655068817	0.067215915	9.74574	2.74E-18	0.5224363	0.7877013	0.522436316	0.787701318

Figure 6.3.3 Blacksburg Data and Regression (Bus vs. Auto Speed)



Tavle 6.3.4 ANOVA Table for Norfolk (Bus vs. Auto Speed)

SUMMARY OUTPUT

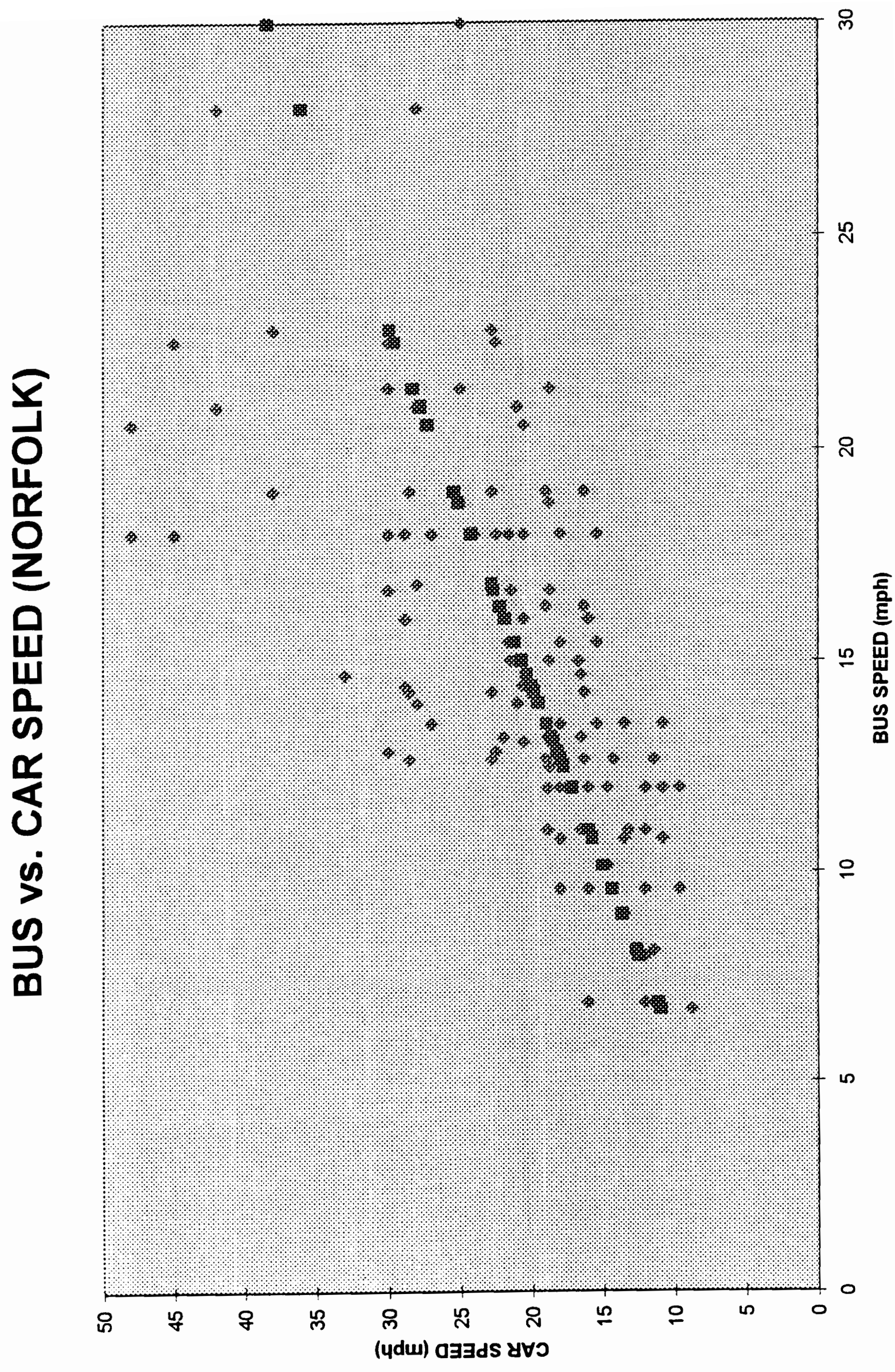
Regression Statistics	
Multiple R	0.65408315
R Square	0.42782477
Adjusted R Square	0.42422618
Standard Error	6.54658761
Observations	161

ANOVA

	df	*SS*	*MS*	*F*	*ignificance F*
Regression	1	5095.232	5095.232	118.8869	5.0693E-21
Residual	159	6814.391687	42.85781		
Total	160	11909.62369			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*pper 95*	*Lower 95.000%*	*Upper 95.000%*
Intercept	2.98700856	1.781107555	1.677051	0.095497	-0.53067001	6.504687	-0.530670008	6.504687127
X Variable 1	1.17951907	0.108177757	10.90353	5.07E-21	0.96586854	1.39317	0.965868542	1.393169604

Figure 6.3.4 Norfolk Data and Regression (Bus vs. Auto Speed)



versus car speed regression results of Blacksburg and Norfolk. Residual table and plot for Norfolk are attached to Appendix L and Appendix M.

For the next step, schedule deviation was also considered as a main factor affecting travel time of these two modes. As illustrated in the Figure 6.3.5 and Figure 6.3.6, however, most of the cases bus arrived at the bus stops ahead the schedule. This was due to the reason that bus schedule time has already considered the enough time for travel in between the stops, and if the bus is ahead the schedule, then the bus will be dragging in the bus stop then leave the bus stop on time. Thus, it was concluded in the schedule deviation approach that it does not influence much on either travel time in light flow condition or heavy traffic flow condition.

Another intuitive approach will be the multiple regression. Multiple regression can be very well applicable in this situation, but the significant drawback in this approach will be the initial functional form assumption. It is not certain to know the accurate relationship between dependent variable and independent variables. Therefore, Artificial Neural Network (ANN) approach, which can handle uncertainties between dependent and independent variables, was applied.

Figure 6.3.5 Bus Schedule Deviation (Blacksburg)

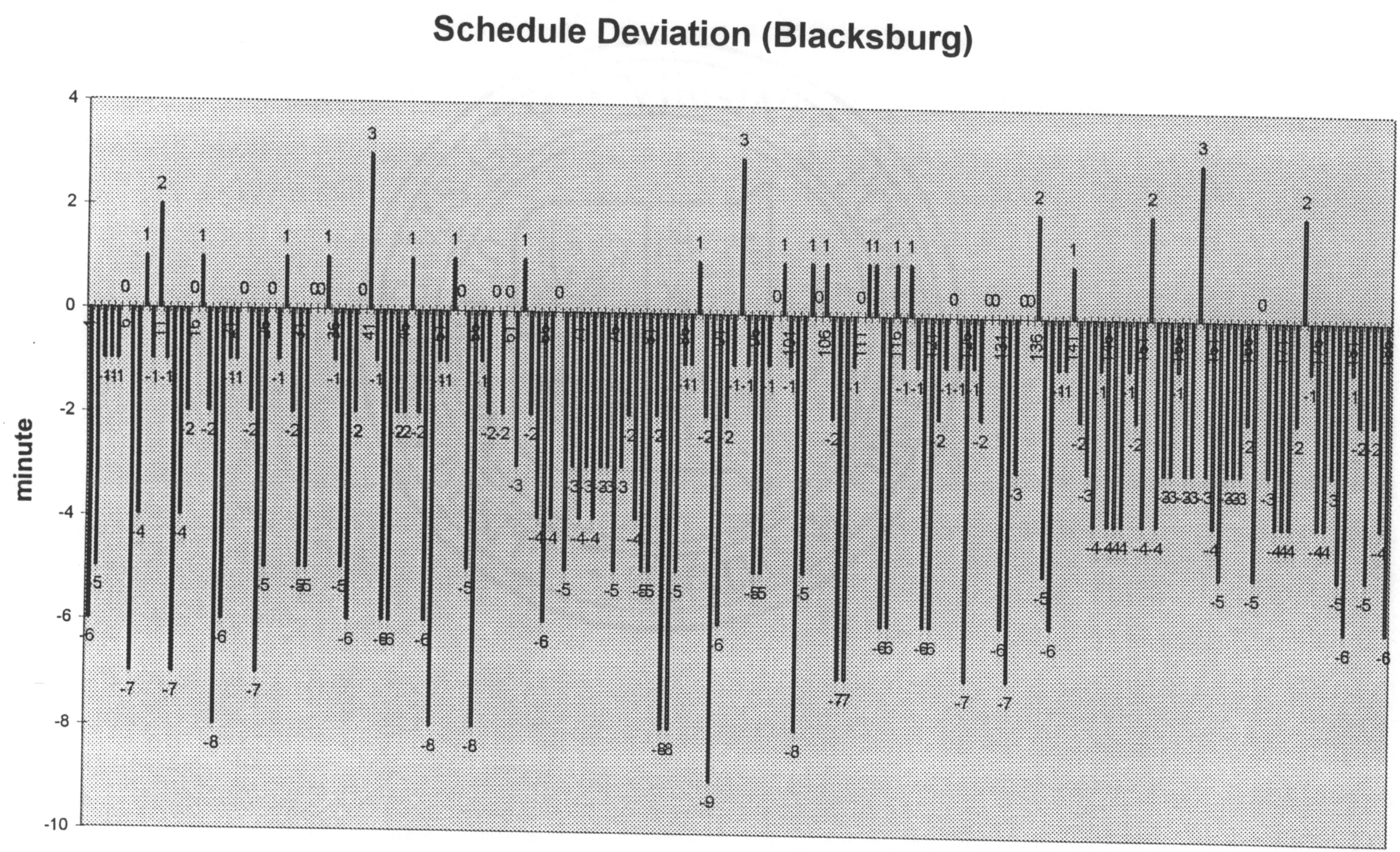
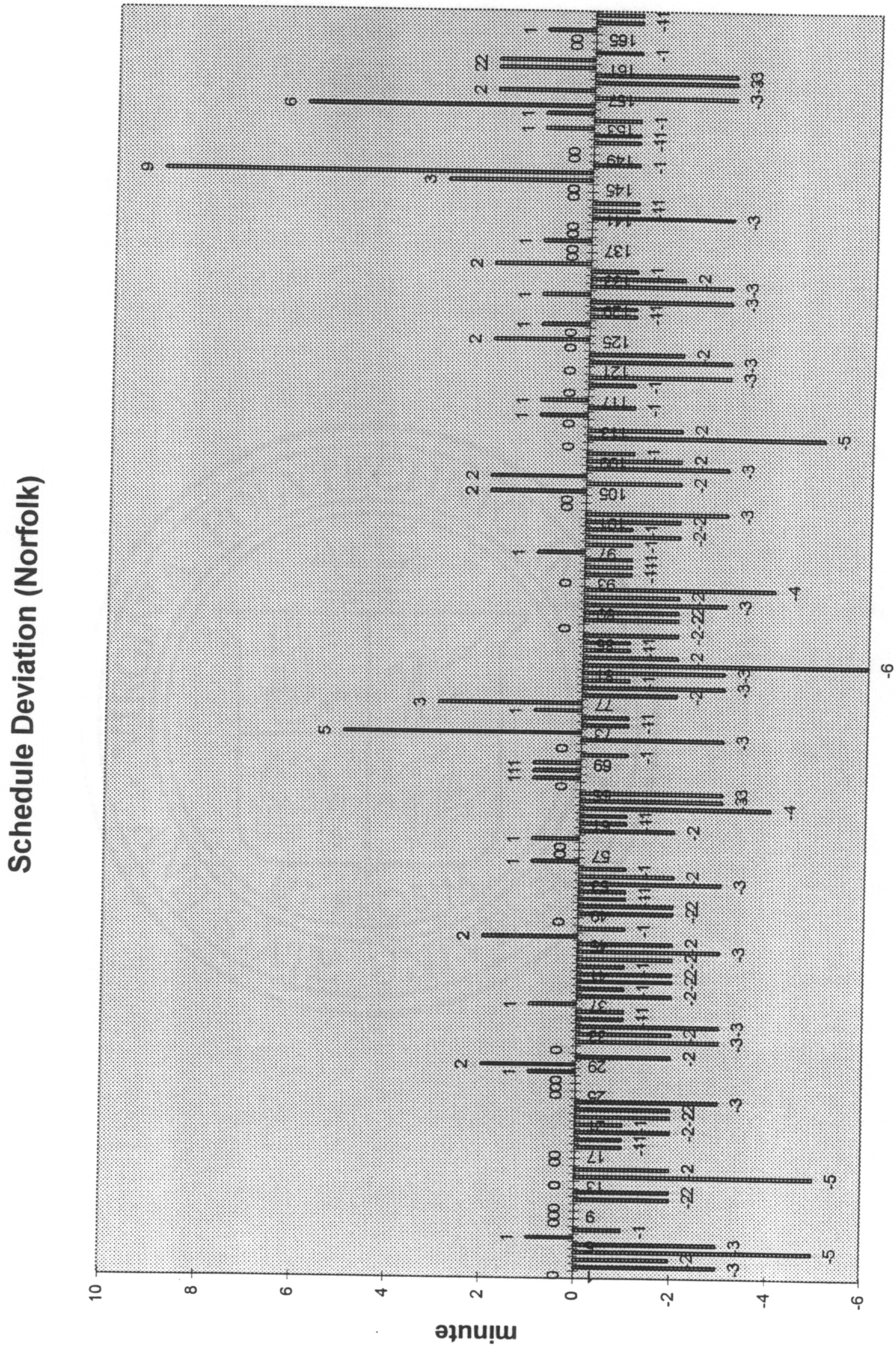


Figure 6.3.6 Bus Schedule Deviation (Norfolk)



6.4 ARTIFICIAL NEURAL NETWORKS APPROACH

As it was reviewed in Chapter 2, Artificial Neural Networks (ANN) is an emerging field in entire engineering and science field. The ANN application areas in transportation engineering were also reviewed in the same chapter. For this application, mapping of bus travel time into car travel time is the main intention. One output (car travel time) was identified, and multilayer perceptron adopting supervised learning (backpropagation) algorithm was the main structure of the ANN applied in the mapping.

6.4.1 Data Input Identification

Each link data collected for Blacksburg and Norfolk consist of 16 data fields (attributes), attached to Appendix N and Appendix O. Data consist of dynamic and static attributes that affect to the travel time of bus in different aspects. Dynamic data are composed of bus travel time, number of stops made in order to load and unload the passengers while the bus travels one major bus stop to another, number of passengers loading and unloading during that journey, incident, and weather condition. Whereas the rest of data, such as link length, number of lanes, parking availability, peak hour factor, number of intersections in between the stops, number of yield signs, number of turns, number of stop signs, speed limit, and turning bay availability constitute the static data attributes. Table 6.4.1 shows classification of the static and the dynamic data attributes.

Table 6.4.1 Artificial Neural Network Input Data Classification

Static Data	**Dynamic Data**
link length	*bus travel time*
number of lanes	*number of stops made*
parking availability	*number of loading passengers*
peak hour factor	*number of unloading passengers*
number of intersections	*number of incidents*
number of yield signs	*weather condition*
number of turns	
number of stop signs	
speed limit	
turning bay availability	

Among the static data, some of data (e.g., number of lanes, parking availability, speed limit, turning bay) require special considerations in preparing the input data set, because a link (bus stop to bus stop) is usually a couple of miles long so that these characteristics are not consistent throughout the link. However, these were coded numerically as accurate as possible based on their proportions.

Next step conducted for input identification was the normalization of the input data set. Since the unit of each attributes are not consistent, and ANN requires value in between 0 and 1 in training phase, normalization is necessary. Then, later in the testing stage of the ANN, it should be denormalized to be easily identified by the users. The commonly adopting normalization algorithm is as follows:

$$\textit{input value} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad 6.4.1$$

Maximum and minimum values for one data field from the entire data set should be obtained. Based on these maximum and minimum values, input values were normalized.

6.4.2. Training

The first task in training phase of ANN is to construct the initial structure of the network based on the configuration of the application problem. Since number of input nodes (data fields) are identified from the data set, constructing preliminary number of hidden layer, hidden nodes, type of transfer function are the next step in this task. Through the multiple testing, 1 hidden layer, 8 hidden nodes, and sigmoidal transfer function were selected. After the structure of networks is settled, the next step for the training process will be the initialization of the weights.

The platform for training and testing the networks was the Matlab ANN toolbox. However, for the efficient and effective training, not all the library function, which was supported by the tool box was used. For example, random numbers in the range of 0.1 to 0.5 were assigned for the initialization of the weights instead of using *initff* function in the ANN toolbox.

One technique that applied in training process was the dividing the input data set into certain number of smaller groups. This technique is extremely efficient when the size of data set is small. For the case of Blacksburg data set, the input vector was only 186 (number of data points) x 16 (attributes). It is not known how large the input data set should be, but usually thousands data points are required for good training. Thus, data set was divided into five subgroups for this research, and training was performed in four of these subgroups. Then, test the trained networks with the rest of one subgroup data. Repeat the same process for five times with the different combination of five groups. Eventually, all the five groups will be participated in the training and testing.

The actual sum of square errors (SSE) generated during the training process are attached at Table 6.4.2 and Table 6.4.3 for Blacksburg and Norfolk. The goal of SSE was defined as 0.01 for both cases, after 500 epochs the training never reached at the goal SSE, however evidently it fell into the local minima. For Blacksburg data SSE after 500 epochs later was 0.4, whereas Norfolk was 0.5, slightly higher than that of Blacksburg. Even though SSE didn't reach at the goal, still it can be said that training is successful because SSE is in the allowable error range of mapping purpose.

6.4.3 Testing

As stated at the previous section, data set was divided into five groups and testings were conducted for the 5th group for five times. One of each results for Blacksburg and Norfolk are illustrated in Figure 6.4.1 and Figure 6.4.2. The results of testing on Blacksburg are slightly better than that of Norfolk due to the reason that travel times for bus and car are more stable in light flow condition in Blacksburg. Averaged SSE in Blacksburg and Norfolk data were 0.4 and 0.5.

6.4.4 Conclusion

ANN provides better result for interpreting car travel time from the bus travel time then the regression. As it was discussed, ANN is a functional approximator. With the sound

Table 6.4.2 Sum of Square Error of Mapping (Blacksburg)

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 12.1669.
TRAINBPX: 100/500 epochs, lr = 0.0883083, SSE = 0.448434.
TRAINBPX: 200/500 epochs, lr = 0.25879, SSE = 0.409068.
TRAINBPX: 300/500 epochs, lr = 0.0685185, SSE = 0.405441.
TRAINBPX: 400/500 epochs, lr = 0.181208, SSE = 0.377874.
TRAINBPX: 500/500 epochs, lr = 0.0609239, SSE = 0.362713.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 5.84184.
TRAINBPX: 100/500 epochs, lr = 0.117744, SSE = 0.444378.
TRAINBPX: 200/500 epochs, lr = 0.0295403, SSE = 0.426182.
TRAINBPX: 300/500 epochs, lr = 0.0678287, SSE = 0.41762.
TRAINBPX: 400/500 epochs, lr = 0.180294, SSE = 0.409743.
TRAINBPX: 500/500 epochs, lr = 0.221775, SSE = 0.408899.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 17.1855.
TRAINBPX: 100/500 epochs, lr = 0.0662312, SSE = 0.441953.
TRAINBPX: 200/500 epochs, lr = 0.0269298, SSE = 0.41484.
TRAINBPX: 300/500 epochs, lr = 0.261892, SSE = 0.388926.
TRAINBPX: 400/500 epochs, lr = 0.0570461, SSE = 0.37344.
TRAINBPX: 500/500 epochs, lr = 0.0210386, SSE = 0.356992.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 3.95289.
TRAINBPX: 100/500 epochs, lr = 0.0600737, SSE = 0.386079.
TRAINBPX: 200/500 epochs, lr = 0.15968, SSE = 0.372557.
TRAINBPX: 300/500 epochs, lr = 0.34919, SSE = 0.364655.
TRAINBPX: 400/500 epochs, lr = 0.062576, SSE = 0.359299.
TRAINBPX: 500/500 epochs, lr = 0.158411, SSE = 0.35094.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 4.9538.
TRAINBPX: 100/500 epochs, lr = 0.0883083, SSE = 0.439193.
TRAINBPX: 200/500 epochs, lr = 0.0638335, SSE = 0.413036.
TRAINBPX: 300/500 epochs, lr = 0.169674, SSE = 0.396689.
TRAINBPX: 400/500 epochs, lr = 0.0798645, SSE = 0.390458.
TRAINBPX: 500/500 epochs, lr = 0.080822, SSE = 0.366466.

Table 6.4.3 Sum of Square Error of Mapping (Norfolk)

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 5.55064.
TRAINBPX: 100/500 epochs, lr = 0.0600737, SSE = 0.492117.
TRAINBPX: 200/500 epochs, lr = 0.15968, SSE = 0.470907.
TRAINBPX: 300/500 epochs, lr = 0.0789185, SSE = 0.461027.
TRAINBPX: 400/500 epochs, lr = 0.0724395, SSE = 0.450798.
TRAINBPX: 500/500 epochs, lr = 0.166332, SSE = 0.444789.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 2.81167.
TRAINBPX: 100/500 epochs, lr = 0.0450553, SSE = 0.658307.
TRAINBPX: 200/500 epochs, lr = 0.0990268, SSE = 0.602841.
TRAINBPX: 300/500 epochs, lr = 0.0236612, SSE = 0.584069.
TRAINBPX: 400/500 epochs, lr = 0.0543296, SSE = 0.575748.
TRAINBPX: 500/500 epochs, lr = 0.0231951, SSE = 0.581578.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 10.683.
TRAINBPX: 100/500 epochs, lr = 0.106798, SSE = 0.613075.
TRAINBPX: 200/500 epochs, lr = 0.312974, SSE = 0.57269.
TRAINBPX: 300/500 epochs, lr = 0.0618347, SSE = 0.567485.
TRAINBPX: 400/500 epochs, lr = 0.148328, SSE = 0.558904.
TRAINBPX: 500/500 epochs, lr = 0.0307707, SSE = 0.545666.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 4.2994.
TRAINBPX: 100/500 epochs, lr = 0.0518939, SSE = 0.626201.
TRAINBPX: 200/500 epochs, lr = 0.131369, SSE = 0.598543.
TRAINBPX: 300/500 epochs, lr = 0.0259549, SSE = 0.586363.
TRAINBPX: 400/500 epochs, lr = 0.0514814, SSE = 0.581217.
TRAINBPX: 500/500 epochs, lr = 0.130325, SSE = 0.576919.

Training in Progress
TRAINBPX: 0/500 epochs, lr = 0.1, SSE = 2.15333.
TRAINBPX: 100/500 epochs, lr = 0.21979, SSE = 0.689465.
TRAINBPX: 200/500 epochs, lr = 0.0527824, SSE = 0.680577.
TRAINBPX: 300/500 epochs, lr = 0.115425, SSE = 0.654225.
TRAINBPX: 400/500 epochs, lr = 0.306808, SSE = 0.642995.
TRAINBPX: 500/500 epochs, lr = 0.0698171, SSE = 0.630542

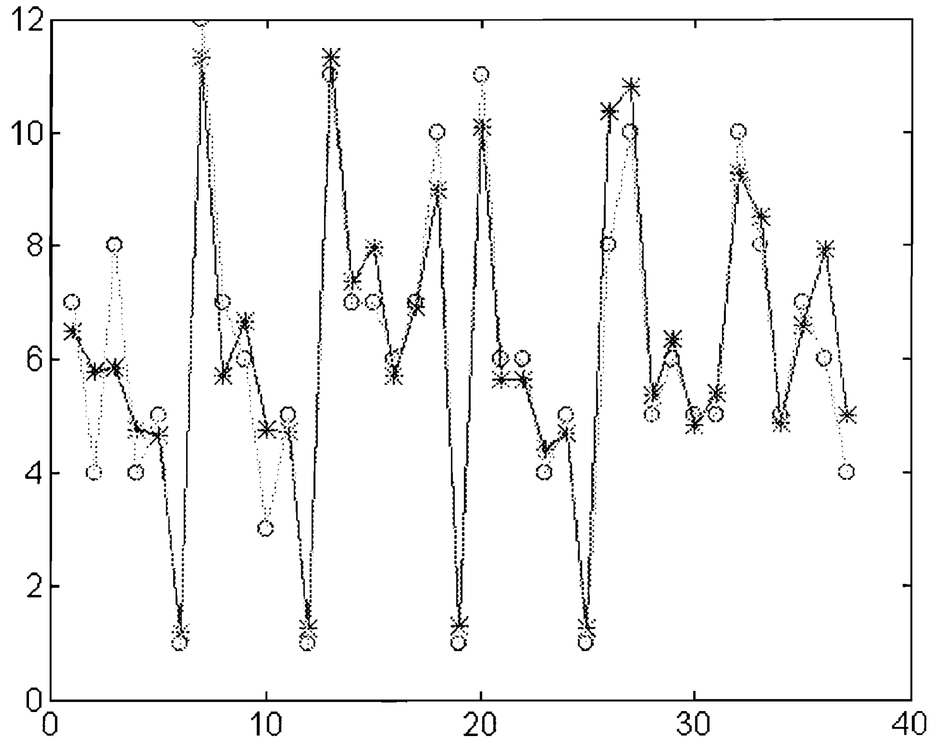


Figure 6.4.1 Artificial Neural Networks Mapping Result (Blacksburg)

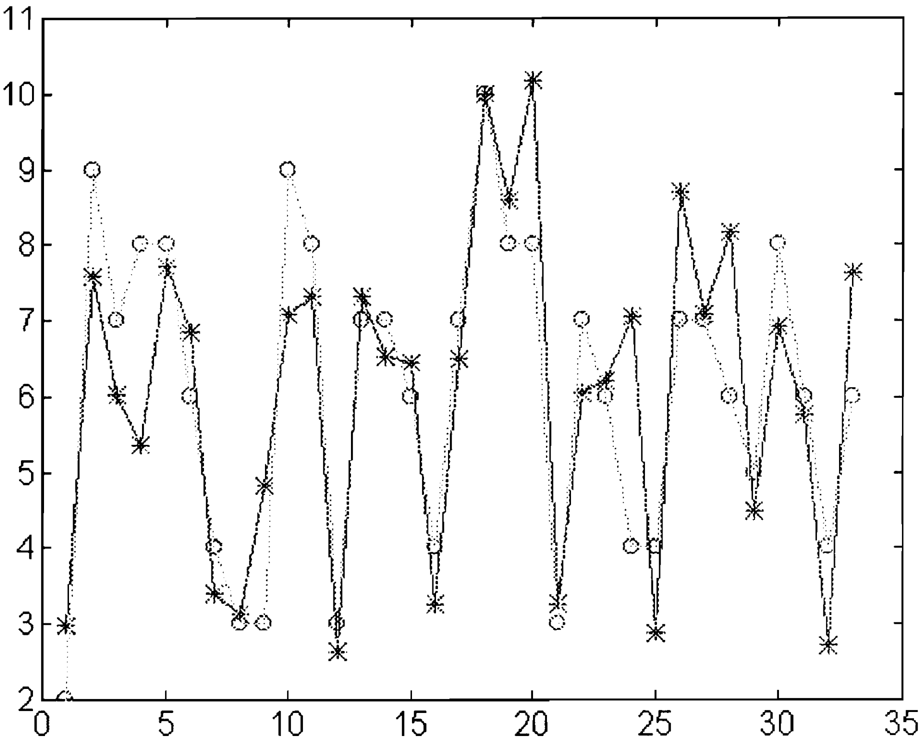


Figure 6.4.2 Artificial Neural Networks Mapping Result (Norfolk)

input data, the weights will be adjusted in order to represent the characteristics of the input data set. However, it is not explicitly known that what kind of ANN structure maps the input data set the best to the output. For instance, number of hidden layers, number of hidden nodes, type of transfer function, etc. are not consistent application by application.

Among the input data sets, some of them, such as weather, time varying peak hour factors, are hard to be expressed in numerical terms. Hence, one recommendation that can be made for the future research of this phase of research is the fuzzy logic approach.

CHAPTER VII CONCLUSION AND RECOMMENDATIONS

7.1 CONCLUSION

This research can be concluded with three major dedications in transportation engineering. Firstly, a new concept of using AVL system-equipped bus as a probe vehicle to estimate arrival time of bus is introduced in this research. The use of the bus probe results in cost effectiveness in real-time data collection.

Secondly, developed analytical model for estimating the arrival time of bus, which integrates passenger arrival rate, passenger loading time and dynamic link travel time was developed by utilizing parameter adaptation algorithm with forgetting factors. In the process of arrival time model development, dynamic behaviors of buses at single and multiple bus stops were researched, and then a prototype arrival time model was formulated and evaluated. The results showed that the prototype arrival time performed better and better as the estimation went on.

Dynamic link travel time estimation model based on detector output (flow data) was developed. Conventional link was divided into congested and uncongested region. Formulation for the link travel time on congested region was based on measuring the number of vehicle in the queue. Whereas link travel time in uncongested region was based on the Greenshield's linear macroscopic model.

Then, linear parameterization of dynamic link travel time, development of algorithm for three stops-ahead arrival time estimation, and dynamic flow estimation models were developed. Among the flow estimation models, the result of first order model provided better performance than the other approaches (second order model and first order with two input model). Finally, integrated arrival time estimation model which combines a prototype arrival time estimation model and dynamic link travel time model was developed. The main algorithm used in the integrated arrival time estimation model was the parameter adaptation algorithm. Since entire arrival time estimation algorithm was based on the on-line estimation, the model tended to adjust its predictability as the estimation went on, and it was validated in the simulation.

Thirdly, newly proposed concept of estimating nontransit travel time from travel time of bus probe was developed and validated through the real world data from the Blacksburg and Norfolk, Virginia. First regression analysis was used as a statistical approach. Then, an Artificial Neural Network was adopted for mapping of bus and car travel time. The results showed in the testing phase of the ANN approach outperformed the regression approach with the sum of square errors of 0.4 for Blacksburg and 0.5 for Norfolk.

In terms of achievement in the first goal, developed arrival time estimation model has a high potential to be incorporated into Automatic Vehicle Location system as a new module. Thus, as long as up to date link flow data are available by real time, the current development can enhance its usage for better customer services. For the second goal, since an algorithm to interpret from the bus travel time to auto travel time was devised and tested with the small sum of square errors in the Artificial Neural Networks approach, it also has a potential to be incorporated in a Traffic Management Center to provide the real-time travel time information for the transit and non-transit users.

7.2 RECOMMENDATIONS

Refinements on dynamic flow estimation will be an essential and the first task which should be researched in the near future. Based on development of this task, accurate data for estimated number of vehicles in a certain link can also be obtained. Another task which should be conducted in the future research is the model validation by using real-time AVL data to interpret the auto travel time. Since data for bus travel time was collected manually and tested in the simulation, conversion model as well as arrival time model can be tested in real-time basis when the AVL generated data are available.

BIBLIOGRAPHY

I. TRAVEL TIME FUNCTIONS

1. Ahmed, Samir A. (1978), "Stochastic Processes in Freeway Traffic," Traffic Engineering and Control, Vol. 24, No. 6-7.

2. Balke, Kevin N., McCasland, William R., Levine, Steven Z., Dudek, Conrad L. (1991), "Collection and Dissemination of Real-Time Travel Time and Incident Information with In-vehicle Communication Technologies," Proceedings of the 2nd Vehicle Navigation and Information System, No. 912737, SAE.

3. Bell, M.G.H., Cowell, M. (1988), "Journey Time Prediction for Real Time Bus Monitoring and Passenger Information Systems," Transport Operations Research Group, University of Newcaltle Upon Tyne, Research Report No. 74.

4. Berinzon, Marian (1993), "Development of a Transportation Data Processing System for Metropolitan Toronto," Proceedings of the 3nd Vehicle Navigation and Information System, No. 93xxxxx, SAE.

5. Daganzo, Carlos F. (1993), "Properties of Link Travel Time Functions Under Dynamic Loads," UCB-ITS-PWP-93-5.

6. Daily, D.J. (1993), "Travel-Time Estimation Using Cross-Correlation Techniques," Transportation Research Part B, Vol. 27B, No. 2, pp 97-107.

7. Fisk, C.S. (1991), "Link Travel Time Functions for Traffic Assignment," Transportation Research part B, Vol. 25B, Nos. 2/3, pp 103-113.

8. Hamerslag, Rudi, Kribbe, Willeke J. (1989), "A Proposal for a Travel-Time-Predicting System," pp 329-335.

9. Heti, Gabriel (1994), "Summary of IVHS R&D and Implementation Projects at the Ontario Ministry of Transportation", Ontario Ministry of Transportation.

10. Heti, Gabriel (1991), "TravElGuide - Ontario's Route Guidance Concept, Ontario Ministry of Transportation," Proceedings of the 2nd Vehicle Navigation and Information System, No. 912778, SAE.

11. Longfoot, John (1991), "An Automatic Network Travel Time System - ANTTS," Proceedings of the 2nd Vehicle Navigation and Information System, No. 912862, SAE.

12. Newell, G.F. and Potts, R.B. (1964), "Maintaining a Bus Schedule," Autralian Road Research Board, Paper No. 170, Vol.2, Part1.

13. Polus, Abishai (1979), "A study of Travel Time and Reliability on Arterial Routes," Transportation, Vol. 8, pp 141-151.

14. Ran, Bin, Rouphail, Nagui, Tarko, Andrzej, Boyce, David E. (1992), "Toward a Set of Dynamic Link Travel Time Functions for Dynamic Traffic Assignment," Paper presented at the 39th Annual Meeting of the Regional Science Association International, Chicago, IL.

15. Rose, G., Taylor, M.A.P., Tisato, P. (1989), "Estimatimg Travel Time Functions for Urban Roads: Options and Issues," Transportation Planning and Technologies, Vol.14, pp.63-82.

16. Sisiopiku, Virginia, Rouphail Nagui M. (1994), "Towards the Use of Detector Output for Arterial Link Travel Time Estimation: A Literature Review," Transportation Research Board, 73rd Annual Meeting, Paper No. 940160.

17. Sisiopiku, Virginia, Rouphail Nagui M., Santiago, Alberto (1994), "Analysis of the Correlation Between Arterial Travel Times and Detector Data From Simulation and Field Studies," Transportation Research Board, 73rd Annual Meeting, Paper No. 940488.

18. Stamatiadis, Chronis, Taylor, Willam, G. (1993), "Travel Time Prediction for Dynamic Route Guidance with Recursive Adaptive Algorithm," Paper Submitted to the Transportation Research Board.

19. Sussman, Joseph M., Wong, Ho-Kwan, Miller, Roger (1974), "Estimating Travel Times on Higeway Networks," ASCE Journal of Transportation, TE1, pp 13-26.

20. Takaba, Sadao, Morita, Takeshi, Hada, Takeshi, Usami, Tsutomu, Yamaguchi, Morie (1991), "Estimation and Measurement of Travel Time by Vehicle Detectors and License Plate Readers," Proceedings of the 2nd Vehicle Navigation and Information System, No. 912760, SAE.

21. Usami, Tsutomu, Ikenoue, Keiichiro, and Miyasako, Tsuneo (1986), "Travel Time Prediction Algorithm and Signal Operations at Critical Intersections for Controlling Travel Time," IEE Second International Conference on Road Traffic Control.

22. Wang, P.T.R., Niedringhaus, Codelli, D.K., McGurrin (1992), Proceedings of the 3rd International Conference on Vehicle Navigation and Information Systems, IEEE, pp 476-479.

23. Westerman, Marcel (1994), "Estimation of Travel Times for ATMS and ATIS," Draft.

24. Westerman, Marcel, Linnartz, Jean-Paul, Hamerslag (1994), "An Advanced Traffic Monitoring System for IVHS Using Probe Vehicles," Draft.

25. Wunderlich, Karl E. (1992), "An Asynchronous Distributed Algorithm for Real-Time Travel-Time Prediction in Vehicular Traffic Networks," IVHS Technical Report #92-11, University of Michigan.

26. Wunderlich, Karl E. (1994), "Link Travel Time Prediction Methods and Convergence for Iterative Anticipatory Route Guidance Methods," University of Michigan, Draft.

II. INTERSECTION DELAY

1. Akcelik, Rahmi (1988), "The Highway Capacity Manual Delay Formula for Signalized Intersections," ITE Journal, March.

2. Buehler, Martin G., Hicks, Thomas J., and Berry, Donald S. (1976), "Measuring Delay bu Sampling Queue Backup," Transportation Research Record No. 612, Transportation Research Board.

3. Drew, Donald R. (1968), "Traffic Characteristics and Flow," McGraw-Hill.

4. Hagen, Lawrence T and Courage, Ken G. (1989), "Comparison of Macroscopic Models for Signalized Intersection Analysis," Transportation Research Record No. 1225, Transportation Research Board.

5. Hurdle, V.F. (1984), "Signalized Intersection Delay Models-A Primer for the Uninitiated," Transportation Research Record No. 971, Transportation Research Board.

6. Legere, Jay F., and Radwan, A. Essam (1984), "Automated Collection of Vehicular Delay Data at Intersections," Transportation Research Record No. 971, Transportation Research Board.

7. Lin, Feng-Bor (1989), "Applications of 1985 Highway Capacity Manual for Estimating Delays at Signalized Intersections," Transportation Research Record No. 1225, Transportation Research Board.

8. May, Adolf D. (1990), "Traffic Flow Fundamentals," Prentice-Hall.

9. McShane, William R. and Roess, Roger P. (1990), "Traffic Engineering," Prentice-Hall.

10. Miller, Alan J. (1963), "Settings for Fixed-Cycle Traffic Signal," Operational Research Quarterly, Vol. 14, No. 4.

11. Sofokidis, H, Tilles, D.L., and Geiger, D.R. (1973), "Evaluation of Intersection-Delay Measurement Techniques," Highway Research Record No. 453.

12. Teply, S. (1989), "Accuracy of Delay Surveys at Signalized Intersections," Transportation Research Record No. 1225, Transportation Research Board.

13. Tarko, Andrzej, Roupahil, Nagui, and Akcelik, Rahmi (1993), "Overflow Delay at a Signalized Intersection Approach Influenced by an Upstream Signal: An Analytical Investigation," Transportation Research Record No. 1398.

14. Transportation Research Board, National Research Council (1985), Highway Capacity Manual, Special Report 209.

III. ARTIFICIAL NEURAL NETWORKS

1. B. Hurdal, J. Hajek (1993), "A Comparison of Rule-Based and Neural Network-based Solutions for a Structured Selection Problem", Presented at 72nd TRB Annual Meeting, January.

2. Bullock, Darcy, Garrett, James Jr., and Hendrickson, Chris (1991), "A Prototype Neural Network for Vehicle Detection", Department of Civil Engineering, Carnegie Mellon University, QA76.87 A77.

3. Caudill, Maureen, and Butler, Charles (1993), “Understanding Neural Networks: Computer Exploration,” Volume I and II, The MIT Press.

4. Chang, Edmond Ching-Ping, and Huarng, Kunhuang (1993), "Incident Detection Using Advanced Technologies", Presented at 72nd TRB Annual Meeting, January.

5. Cheu, Ryeu L., Ritchie, Stephen G., Recker, Wilfred W., and Bavarian, Behnam (1991), "Investigation of a Neural Network Model for Freeway Incident Detection", ITS, UC Irvine.

6. Djukanovic, Miodrag, and Babic, Borivoje (1991), "Unsupervised/Supervised Learning Cocept for 24-hour Load Forecasting", QA76.87 A77.

7. Faghri, Ardeshir and Hua, Jiuyi (1992), "Evaluation of Artificial Neural Network Applications in Transportation Engineering", TRR 1358.

8. Goldstone, J.S., and Myers, C.E. (1991), "Traffic Monitoring With Wisard and Probabilistic Logic Nodes", Department of Electrical Engineering, Emperial College, Artificial Neural Networks, QA76.87 I56.

9. Hua, Jiuyi, and Faghri, Ardeshir (1994), "Applications of ANNs to IVHS", Presented at 73rd TRB Annual Meeting, January.

10. Hua, Jiuyi, and Faghri, Ardeshir (1993), "Dynamic Traffic Pattern Classification Using Artificial Neural Networks", Presented at 72nd TRB Annual Meeting, January.

11. Kalaputapu, Ravi, and Demetsky, Michael J. (1994), "Bus Transit Schedule Behavior Models Using Advanced Techniques and Automatic Vehicle Location Information", Department of Civil Engineering, University of Virginia.

12. Khotanzad, Alireza, and Fowler, Mark (1991), "Neural Network Based Time Series Forecasting", Electrical Engineering Department, Southern Methodist University, QA76.87 A77.

13. Kim, Moon W. (1993), "Neural Networks: What They Are and What They Do?", Naval Research Laboratory, KSEA Letters, Vol. 22, No. 1, August.

14. Kirby, Howard R., Dougherty, Mark S., and Watson, Susan M. (1994), "Should We Use Neural Networks or Statistical Models for Short Term Motorway Traffic Forecasting?", Institute for Transport Studies, University of Leeds, Presented at 73rd TRB Annual Meeting, January.

15. Kornhauser, Alain L. (1991), "Neural Network Approaches for Lateral Control of Autonomous Highway Vehicles", Princeton University, 912871.

16. McNally, Michael G., and Lo, Zhen-Ping (1993), "Prediction of Discrete Choice Via Neural Network", ITS, UC Irvine, Presented at 72nd TRB Annual Meeting, January.

17. Mead, William C., Fisher, Henry N., Jones, Roger D., Bisset, Keith R., and Lee, Leopold A. (1994), "Application of Adaptive and Neural Network Computational Techniques to Traffic Volume and Classification Monitoring", Presented at 73rd TRB Annual Meeting.

18. O'Neill, Michael J., "Neural Network Simulation As A Computer-Aided Design Tool for Predicting Wayfinding Performance", University of Wisconsin-Madison.

19. Smith, Brian L., and Demetsky, Michael J. (1994), "Short-Term Traffic Flow Prediction - A Neural Network Approach," Presented at TRB 1994.

20. Soulie, Fogelman (1991), "Neural Network Architectures and Algorithms: A Perspective", Artificial Neural Networks, QA76.87 I56.

21. Tarr, Gregory L., Rogers, Steven K., and Kabrisky, Matthew (1991), "Effective Neural Network Modeling in C", Artificial Neural Networks, QA76.87 I56.

22. Wasserman, Philip D. (1989), "Neural Computing: Theory and Practice," Van Nostrand Reinhold.

23. Wei, Chien-Hung, and Schonfeld, Paul M., "An Artificial Neural Network Approach for Estimating Multiperiod Travel Times in Transportation Networks", Presented at 72nd TRB Annual Meeting, 1993.

24. Xiong, Yihua, and Schneider, Jerry B. (1992), "Transportation Network Design Using a Cumulative Genetic Algorithm and Neural Network", pp37-44, TRR 1364.

25. Yang, Hai, Kitamura, Ryuichi, Jovanis, Paul P., Vaughn, Kenneth M., and Abdel-Aty, Mohamed A. (1992), "An Analysis of Route Choice Behavior Under Advanced Traveler Information Systems: A Neural Network Approach", ITS, UC Davis.

26. Yang, Hai, Kitamura, Ryuichi, Jovanis, Paul P., Vaughn, Kenneth M., Abdel-Aty, Mohamed A., and Reddy, Prasuna DVG (1993), " Exploration of Driver Route Choice With Advanced Traveler Information Using Neural Network Concepts", ITS, UC Davis.

27. Zhang, Hongjun, Ritchie, Stephen G., and Lo, Zhen-Ping (1993), "Macroscopic Modeling of Freeway Traffic Using an Artificial Neural Network", Presented at 72nd TRB Annual Meeting.

IV. GENERAL

1. Bell, Michale G.H. (1991), The Real Time Estimation of Origin-Destination Flows in the Presence of Platoon Dispersion, Transportation Research Part B, Vol. 25B, Nos. 2/3, pp 115-125.

2. Brannas, Kurt, Laitila, Thomas (1991), Modeling and Prediction of Travel Distance By Car, Transportation Planning and Technology, Vol. 16, pp 129-143.

3. Heidemann, Dirk (1991), Queue Length and Waiting-Time Distributions at Priority Intersections, Transportation Research Part B, Vol. 25B, No. 4, pp 163-174.

4. Hsia, T.C. (1977), “System identification: least-squares methods,” Lexington Books, 1977.

5. Institute of Transportation Engineers (1986), “Urban Traffic Congestion: What Does the Future Hold?,” ITE Publication No. IR-040

6. Institute of Transportation Engineers (1989), “A Tool Box for Alleviating Traffic Congestion,” ITE Publication No. IR-054A

7. Johnson, L.R. (1988), “Lecture on adaptive parameter estimation,” Prentice-Hall.

8. Joyce, Peter, Yagar, Sam (1990), Representing Stochastic Transit Dwell Times in Traffic Signal Optimization, Transportation Research Part A, Vol. 24A, No. 2, pp 87-98.

9. Klein, Lawrence A., MacCalden, M. Scott, Mills, Milton (1992), "Detection Technology for IVHS," IVHS America Annual Meeting.

10. Lindley, J.A. (1987), "Urban Freeway congestion: Quantification of the Problem and Effectiveness of Potential Solutions," ITE Journal, Vol. 57, No1, pp 27-32.

11. Ljung, L. (1987), "System identification: theory for the user," Prentice-Hall.

12. Ljung, L., and Soderstrom, T. (1983), "Theory and practice of recursive identification," MIT Press.

13. Mitoh, Kunihiko, Yumoto, Nobuo (1991), Development of Advanced Dynamic Navigation System, Proceedings of the 2nd Vehicle Navigation and Information System, No. 912810, SAE.

14. Schweiger, Carol L., Kihl, Mary, and Labell, Lawrence N. (1994), "Advanced Public Transportation Systems: The State of the Art Update '94", U.S.D.O.T. Final report, RSPA, January.

15. Sorenson, H.W. (1980), "Parameter estimation: principles and problems," M. Dekker.

APPENDIX A Source Code for a Prototype Arrival Time Estimation

```
%==============================================================
%
% This program is written for Simulation of Bus Arrival Time Estimation.
% In this simulation, time varying parameters are considered.
% The outputs are arrival time estimation, parameter convergence, and estimation error.
%
%==============================================================

clf
hold on

%-------------------------------------
%           Initialization
%-------------------------------------
clear

buses = 20;
stops = 4;

for i = 1:4,
        c(i) = 10;                          % travel time
end

lt = zeros(buses,stops);                    % initialization for loading time
lt(1,2:stops) = 5*rand(1,3)+3;

for i = 1:buses,

        tau(i) = 30;                        % initial departure time headway
end

for i = 1:buses,
        for j =1:stops,
                k(i,j) = .2*rand+.1;        % assignment for pass. arrvl rate/loading rate
        end
end

%---------------------------------------------------
%           Calculation of departure time
%---------------------------------------------------

t(1, 1) = 0;
for j = 2:stops,

        t(1, j) = t(1, j-1) + c(j-1) + lt(1, j);
end

for i = 2:buses
        t(i, 1) = t(i-1, 1) + tau(i-1);
        for j = 2:stops
                t(i, j) = (t(i, j-1)/(1-k(i, j)))-(k(i, j)/(1-k(i, j))*t(i-1, j)) + (c(j-1)/(1-k(i,j)));

        end
end
```

```
%-----------------------------------------------
%           plotting of simulation
%-----------------------------------------------

for i = 1:buses
            for j = 1:stops
                        t1(j,i) = t(i,j);
            end
end

t2=zeros(7,6);

for kk=1:buses,
            v=t1(1,kk);
            for kk2=2:stops,
                        v=[v; t1(kk2-1,kk)+c(kk2-1); t1(kk2,kk)];
                        if kk>1,
                                    lt(kk,kk2) =t1(kk2,kk)-c(kk2-1)-t1(kk2-1,kk);
                        end
            end
            t2(:,kk)=v;
end

plot(t2,[1,2,2,3,3,4,4]);                    % plotting for buses at multiple bus stops

ylabel ('Distance (# of stops)');
xlabel ('Time (min)');
title ('Multiple Stops Simulation');

pause
hold off
clc

%-----------------------------------------------------
%           Least Squares Parameter Estimate
%-----------------------------------------------------

hold on

F1 = 1e+15*eye(3,3);                        % initialization for gain term
F2 = 1e+15*eye(3,3);
F3 = 1e+15*eye(3,3);

atheta1 = [ -.25
            .25
            12.05];

atheta2 = [ -.26
            .26
            12.06];

atheta3 = [ -.27
            .27
            12.07];

etheta1 = [ -.12
            .12
            10.];

etheta2 = [ -.12
            .12
            10.0];

etheta3 = [ -.12
            .12
            10.];
```

```
eat(2,2) = tau(1)+c(1);

h(2,2) = eat(2,2)-(t(1,1)+c(1));
y=t(1,2);
u=h(2,2);
phi1 = [    y
            u
            1];
edt(2,2) = tau(1)+etheta1'*phi1;

eat(2,3) = edt(2,2)+c(2);
h(2,3) = eat(2,3)-(t(1,2)+c(2));
y=t(1,3)-t(1,2);
u=h(2,3);
phi2 = [    y
            u
            1];

edt(2,3) = edt(2,2)+etheta2'*phi2;

eat(2,4) = edt(2,3)+c(3);
h(2,4) = eat(2,4)-(t(1,3)+c(3));

y=t(1,4)-t(1,3);
u=h(2,4);
phi3 = [    y
            u
            1];
edt(2,4) = edt(2,3)+etheta3'*phi3;

%-----------------------------------------------
%           Update at 2nd Stop
%-----------------------------------------------
-
up_edt1(2,2) = t(2,2);
up_eat1(2,3) = up_edt1(2,2)+c(2);

y = t(1,3)-t(1,2);
h1(2,3) = up_eat1(2,3)-(t(1,2)+c(2));
u = h1(2,3)
up_edt1(2,3) = up_edt1(2,2)+etheta2'*phi2;

up_eat1(2,4) = up_edt1(2,3)+c(3);

y = t(1,4)-t(1,3);
h1(2,4) = up_eat1(2,4)-(t(1,3)+c(3));
u = h1(2,4)
up_edt1(2,4) = up_edt1(2,3)+etheta3'*phi3;

%-----------------------------------------
% Update at 3rd Stop
%-----------------------------------------

up_edt2(2,3) = t(2,3);
up_eat2(2,4) = up_edt2(2,3)+c(3);

y = t(1,4)-t(1,3);
h2(2,4) = up_eat2(2,4)-(t(1,3)+c(3));
u = h2(2,4);
up_edt2(2,4) = up_edt2(2,3)+etheta3'*phi3;

%---------------------------------------------------------------------------
%           Plotting for Estimated Bus Arrival Time
%---------------------------------------------------------------------------

subplot(3,1,1)
```

```
qq = [0		eat(2,2)];
rr = [eat(2,2) eat(2,2)];

plot(qq, rr)						% plotting for estimated arrival time at the second bus stop

subplot(3,1,2)
qq2 = [0			edt(2,2)	edt(2,2)		eat(2,3)];
rr2 = [eat(2,3)		eat(2,3)	up_eat1(2,3)		up_eat1(2,3)];

plot(qq2, rr2)						% plotting for estimated arrival time at the third bus stop

subplot(3,1,3)
qq3 = [0			edt(2,2)	edt(2,2)		edt(2,3)		edt(2,3)		eat(2,4)];
rr3 = [eat(2,4)		eat(2,4)	up_eat1(2,4)		up_eat1(2,4)		up_eat2(2,4)		up_eat2(2,4)];
plot(qq3, rr3)						% plotting for estimated arrival time at the forth bus stop

pause

mm(1,1) = etheta1(1);
nn(1,1) = etheta1(2);
oo(1,1) = etheta1(3);

mm(2,1) = etheta2(1);
nn(2,1) = etheta2(2);
oo(2,1) = etheta2(3);

mm(3,1) = etheta3(1);
nn(3,1) = etheta3(2);
oo(3,1) = etheta3(3);

err1(1) = atheta1'*phi1-etheta1'*phi1;			% estimation error calculation
err2(1) = atheta1'*phi2-etheta2'*phi2;
err3(1) = atheta1'*phi3-etheta3'*phi3;

[F1, etheta1] = update1(F1, phi1, etheta1, atheta1);	% update the parameters
[F2, etheta2] = update2(F2, phi2, etheta2, atheta2);
[F3, etheta3] = update3(F3, phi3, etheta3, atheta3);

hold off

%==================================================
% Third to Nth Bus Estimation and Parameter Update
%==================================================

for i =3:buses,

	atheta1 = [ -.2+.05*sin(.13*(i-1))
		     .2+.05*sin(.13*(i-1))
		     12+.75*sin(.13*(i-1))];

	atheta2 = [ -.2+.05*sin(.13*(i-1))
		     .2+.05*sin(.13*(i-1))
		     12+.75*sin(.13*(i-1))];

	atheta3 = [ -.2+.05*sin(.13*(i-1))
		     .2+.05*sin(.13*(i-1))
		     12+.75*sin(.13*(i-1))];

	clf
	hold on

	sum_tau = 0;
	for j = 1:i-1
		sum_tau = sum_tau+tau(j);
	end

	eat(i,2) = sum_tau+c(1);
```

```
y = up_edt1(i-1,2)-t(i-1, 1);
h(i,2) = eat(i,2)-eat(i-1,2);
u = h(i,2);
edt(i,2) = sum_tau+etheta1'*phi1;

eat(i,3) = edt(i,2)+c(2);

y = up_edt2(i-1,3)-up_edt1(i-1,2);
h(i,3) = eat(i,3)-up_eat1(i-1,3);
u=h(i,3);
edt(i,3) = edt(i,2)+etheta2'*phi2;

eat(i,4) = edt(i,3)+c(3);

y = up_edt2(i-1,4)-up_edt2(i-1,3);
h(i,4) = eat(i,4)-up_eat2(i-1,4);
u = h(i,4);
edt(i,4) = edt(i,3)+etheta3'*phi3;

%-----------------------------------------
% Update at the 2nd Stop
%-----------------------------------------

up_edt1(i,2) = t(i,2);
up_eat1(i,3) = up_edt1(i,2)+c(2);

y = up_edt2(i-1,3)-up_edt1(i-1,2);
h1(i,3) = up_eat1(i,3)-up_eat1(i-1,3);
u = h1(i,3)
up_edt1(i,3) = up_edt1(i,2)+etheta2'*phi2;;

up_eat1(i,4) = up_edt1(i,3)+c(3);

y = up_edt2(i-1,4)-up_edt2(i-1,3);
h1(i,4) = up_eat1(i,4)-up_eat2(i-1,4);
u = h1(i,4)
up_edt1(i,4) = up_edt1(i,3)+etheta3'*phi3;

%-----------------------------------------
% Update at 3rd Stop
%-----------------------------------------

up_edt2(i,3) = t(i,3);
up_eat2(i,4) = up_edt2(i,3)+c(3);

y = up_edt2(i-1,4)-up_edt2(i-1, 3);
h2(i,4) = up_eat2(i,4)-up_eat2(i-1,4);
u = h2(i,4);
up_edt2(i,4) = up_edt2(i,3)+etheta3'*phi3;

%------------------------------------------------------------
% Plotting
%------------------------------------------------------------

subplot(3,1,1)
qq1 = [0            eat(i,2)];
rr1 = [eat(i,2)     eat(i,2)];
plot(qq1, rr1)

subplot(3,1,2)
qq2 = [0            edt(i,2)    edt(i,2)            eat(i,3)];
rr2 = [eat(i,3)     eat(i,3)    up_eat1(i,3)        up_eat1(i,3)];
plot(qq2, rr2)


subplot(3,1,3)
```

```
        qq3 = [0        edt(i,2)    edt(i,2)        edt(i,3)        edt(i,3)        eat(i,4)];
        rr3 = [eat(i,4)     eat(i,4)    up_eat1(i,4)        up_eat1(i,4)        up_eat2(i,4)        up_eat2(i,4) ];
        plot(qq3, rr3);

        pause

        mm(1,i-1) = etheta1(1);
        nn(1,i-1) = etheta1(2);
        oo(1,i-1) = etheta1(3);

        mm(2,i-1) = etheta2(1);
        nn(2,i-1) = etheta2(2);
        oo(2,i-1) = etheta2(3);

        mm(3,i-1) = etheta3(1);
        nn(3,i-1) = etheta3(2);
        oo(3,i-1) = etheta3(3);

        err1(i-1) = atheta1'*phi1-etheta1'*phi1;
        err2(i-1) = atheta2'*phi2-etheta2'*phi2;
        err3(i-1) = atheta3'*phi3-etheta3'*phi3;

        [F1, etheta1] = update1(F1, phi1, etheta1, atheta1);
        [F2, etheta2] = update2(F2, phi2, etheta2, atheta2);
        [F3, etheta3] = update3(F3, phi3, etheta3, atheta3);

end
clf
hold off

%======================================
%       Plotting for Parameter Convergence
%======================================

for i = 1:stops-1,
        clf
        hold on

        ttt = -.2+.05*sin(.13*[1:buses]);
        plot(ttt);

        xx=[];
        for j = 1:buses-1,
                xx=[xx mm(i,j) mm(i,j)];
        end

        yy = [1];
        for vv = 2: buses-1
                yy = [yy;vv;vv];
        end
        yy = [yy;buses];

        plot(yy,xx);

        ylabel ('parameter a');
        xlabel ('buses');
        title ('Parameter Convergency');
        pause

        hold off

        clf
        hold on

        ttt = .2+.05*sin(.13*[1:buses]);
        plot(ttt);
```

```
		xx=[];
		for j = 1:buses-1,
			xx=[xx nn(i,j) nn(i,j)];
		end

		yy = [1];
		for vv = 2: buses-1
			yy = [yy;vv;vv];
		end
		yy = [yy;buses];

		plot(yy,xx);

		ylabel ('parameter b');
		xlabel ('buses');
		title ('Parameter Convergency');

		pause
		hold off

		clf
		hold on

		ttt = 12+.75*sin(.13*[1:buses]);
		plot(ttt);

		xx=[];
		for j = 1:buses-1,
			xx=[xx oo(i,j) oo(i,j)];
		end
		yy = [1];
		for vv = 2: buses-1
			yy = [yy;vv;vv];
		end
		yy = [yy;buses];

		plot(yy,xx);

		ylabel ('parameter c');
		xlabel ('buses');
		title ('Parameter Convergency');

		pause
		hold off

end  %plotting for loop

%======================================
%		Plotting for Estimation Errors
%======================================

clf
hold on

xx1 = [];
for j = 1:buses-1,
		xx1 = [xx1 err1(j) err1(j)];
end

yy = [1];
for vv = 2:buses-1,
		yy = [yy; vv; vv];
end

yy = [yy; buses];

plot(yy, xx1);
```

```
ylabel( 'error at stop 1');
xlabel( '# of buses');
title( 'Estimation Error');

pause

clf
xx2 = [];
for j = 1:buses-1,
        xx2 = [xx2 err2(j) err2(j)];
end

plot(yy, xx2);

ylabel( 'errors at stop 2');
xlabel( '# of buses');
title('Estimation Error');

pause

clf

xx3 = [];
for j = 1:buses-1,
        xx3 = [xx3 err3(j) err3(j)];
end

plot(yy, xx3);

ylabel( 'errors at stop 3');
xlabel( '# of buses');
title( 'Estimation Error');
```

APPENDIX B Source Code for Flow Estimation (Least Square with Forgetting Factor)

```
%==========================================================================
% Main function for this program is to estimate the link flow by adoting generic least square parameter
% estimation algorithm with forgetting factor.
% This code is an example for the first order model for Blacksburg data.
% The formulation of one step lagging model is
%         q(k+1) = aq(k) + b.
%==========================================================================

clf

%------------------------------
%         Initialization
%------------------------------

num_of_links = 6;
num_of_time_pt = 32;

load c:\matlab\bin\simflow1.dat

sum_mse = 0;

err = zeros(num_of_links, num_of_time_pt);
e_flow = zeros(num_of_links,num_of_time_pt);

for i = 1:num_of_links,
        F = 1e+1*eye(2,2);
        etheta = [  0.9
                    1.1];

        err_sum = 0;

        for j = 2:num_of_time_pt,
                phi = [     simflow1(i, j-1)
                            1];
                e_flow(i, j) = etheta'*phi;

                err(i, j) = simflow1(i,j) - e_flow(i, j);

                err_sum = err_sum + err(i,j)^2;

                F = 0.99*(F - (F*phi*phi'*F)/(0.99+phi'*F*phi));
                etheta = etheta+F*phi*(simflow1(i,j)-etheta'*phi)/(1+phi'*F*phi);

        end

        mse(i) = err_sum/num_of_time_pt-1;

        plot (simflow1(i,:),'g');
        hold on;
        plot ( e_flow(i, :),'r');
        ylabel('flow (veh/30 min)');
        xlabel('# of time points');
        title('Flow Estimation');

        pause
```

```
		hold off

		clf
		hold on

		k =0;
		xx1 = [];
		for k = 1:num_of_time_pt-1,
				xx1 = [xx1 err(i, k) err(i, k)];
		end

		yy = [1];
		for vv = 2:num_of_time_pt-1,
				yy = [yy; vv; vv];
		end

		yy = [yy; num_of_time_pt];

		plot(yy, xx1);

		ylabel('error');
		xlabel('# of time points');
		title('Estimation Error');

		pause
		hold off

sum_mse = sum_mse+mse(i);

end

amse = sum_mse/num_of_links;
```

APPENDIX C Source Code for Flow Estimation (Ljung's Approach)

```
%==========================================================================
% Main function for this program is to estimate the link flow by adoting Ljung's approach.
% This code is an example for the first order model for Blacksburg data.
% The formulation of one step lagging model is
%           q(k+1) = aq(k) + b.
%==========================================================================

clf

%------------------------------
%           Initialization
%------------------------------

num_of_links = 7;
num_of_time_pt = 96;

load c:\matlab\bin\bb_flow.txt

sum_mse = 0;

err = zeros(num_of_links, num_of_time_pt);
e_flow = zeros(num_of_links,num_of_time_pt);

for i = 1:num_of_links,

        lamda = 0.99;
        R = 1e+20*eye(2,2);

        etheta = [   0.9
                     1.1];

        err_sum = 0;

        for j = 2:num_of_time_pt,
                phi = [      bb_flow(i, j-1)
                             1];
                e_flow(i, j) = phi'*etheta;

                err(i, j) = bb_flow(i,j) - e_flow(i, j);

                err_sum = err_sum + err(i,j)^2;

                R = R + (1-lamda)*(phi*phi' - R);
                etheta = etheta + (1-lamda)*R^(-1)*(phi*err(i, j));
        end

        mse(i) = err_sum/num_of_time_pt-1;

        plot (bb_flow(i,:),'g');
        hold on;
        plot ( e_flow(i, :),'r');
        ylabel('flow (veh/5 min)');
        xlabel('# of time points');
        title('Flow Estimation');
```

```
            pause

            hold off

            clf
            hold on

            k =0;
            xx1 = [];
            for k = 1:num_of_time_pt-1,
                        xx1 = [xx1 err(i, k)/phi(1)*100 err(i, k)/phi(1)*100];
            end

            yy = [1];
            for vv = 2:num_of_time_pt-1,
                        yy = [yy; vv; vv];
            end

            yy = [yy; num_of_time_pt];

            plot(yy, xx1);

            ylabel('percentage error');
            xlabel('# of time points');
            title('Estimation Error');

            pause
            hold off

sum_mse = sum_mse+mse(i);

end

amse = sum_mse/num_of_links;
```

APPENDIX D Source Code for Integrated Arrival Time Estimation

```
%==========================================================================
% In this code, integrated arrival time estimation which combines the dynamic link travel time considering
% intersection delay and the prototype arrival time estimation model is estimated.
%==========================================================================

clf
hold on

%-------------------------------------------
%            Definition
%-------------------------------------------
clear

buses = 96;
stops = 4;
links = 6;

load flow.txt

jd = 240/5280;                    % jam density: veh/ft (when jd=240 veh/mile)
ffs = 55*5280/60;                 % free flow speed: ft/min
mu = 15;                          % average car length: feet
acc1 = 3.0*60*60;                 % acceleration rate: ft/min^2
acc2 = 3.0*60*60;                 % decelleration rate: ft/min^2

l = 5280*[.7 .8 .9  1.8 1.6 2.5];   %link length: ft

d1 =.7; d2=d1+.8; d3=d2+.9; d4=d3+1.8; d5=d4+1.6; d6=d5+2.5;

veh = zeros(links, buses);

for i = 1:links,
        for j = 1:buses,

                op(i, j) = ffs - sqrt(ffs^2-4*(1/15)*flow(i, j)*ffs/jd);
                qr(i, j) = jd + sqrt(jd^2-4*(1/15)*flow(i, j)*jd/ffs);

                veh(i, j) = .95*(1/15)*flow(i, j) ;

                if (links <= 2),
                        c(i, j) = ffs/(2*ffs-jd*op(i, j)*mu)*(2*veh(i, j)/(1/15)*flow(i, j)) - jd*op(i, j)/(2*ffs-jd*op(i,
j)*mu)*l(i)/(1/15)*flow(i, j) + ffs*qr(i, j)*(acc1+acc2)/(4*jd*acc1*acc2) + 1/((ffs*qr(i, j)/(2*jd))*(1-mu*jd*op(i, j)/(2*ffs)))*l(i) -
mu/((ffs*qr(i, j)/(2*jd))*(1-mu*jd*op(i, j)/(2*ffs)))*veh(i, j);
                elseif (links == 4),
                        c(i, j) = ffs/(2*ffs-jd*op(i, j)*mu)*(2*veh(i, j)/(1/15)*flow(i, j)) - jd*op(i, j)/(2*ffs-jd*op(i,
j)*mu)*l(i)/(1/15)*flow(i, j) + ffs*qr(i, j)*(acc1+acc2)/(4*jd*acc1*acc2) + 1/((ffs*qr(i, j)/(2*jd))*(1-mu*jd*op(i, j)/(2*ffs)))*l(i) -
mu/((ffs*qr(i, j)/(2*jd))*(1-mu*jd*op(i, j)/(2*ffs)))*veh(i, j);
                else
                        c(i, j) = ffs*qr(i, j)*(acc1+acc2)/(4*jd*acc1*acc2) + 1/((ffs*qr(i, j)/(2*jd))*(1-mu*jd*op(i,
j)/(2*ffs)))*l(i) - mu/((ffs*qr(i, j)/(2*jd))*(1-mu*jd*op(i, j)/(2*ffs)))*veh(i, j);
                end
        end
end

lt = zeros(buses,stops);
```

```
lt(1,2:stops) = rand(1,3)+1;

for i = 1:buses,

        tau(i) = 15;            % initial headway
end


for i = 1:buses,
        for j =1:stops,
                k(i,j) = .3*rand+.01; % pass. arrvl rate/loading rate;
        end
end

%------------------------------
%       Initialization
%------------------------------

t(1, 1) = 0;
sum_tt = zeros(buses, stops-1);

c=c';

for j = 2:stops,
        if (j == 2),
                sum_tt(1,j-1) = c(1, 1)+c(1, 2)+c(1, 3);
        elseif (j==3),
                sum_tt(1,j-1) = c(1, 4)+c(1, 5);
        elseif (j==4),
                sum_tt(1,j-1) = c(1,6);
        end

        t(1, j) = t(1, j-1) + sum_tt(1, j-1) + lt(1, j);
end

for i = 2:buses
        t(i, 1) = t(i-1, 1) + tau(i-1);
        for j = 2:stops
                %t(i, j) = t(i, j-1) + c(j-1) + lt(i, j);

                if (j == 2),
                        sum_tt(i,j-1) = c(i, j-1) +c(i, j)+c(i, j+1);
                elseif (j == 3)
                        sum_tt(i,j-1) = c(i, j+1)+c(i, j+2);
                elseif (j == 4)
                        sum_tt (i,j-1) = c(i, j+2);
                end
                t(i, j) = (t(i, j-1)/(1-k(i, j)))-(k(i, j)/(1-k(i, j))*t(i-1, j)) + (sum_tt(i,j-1)/(1-k(i,j)));

        end
end

%------------------------------
%       plotting of simulation
%------------------------------

for i = 1:buses
        for j = 1:stops
                t1(j,i) = t(i,j);
        end
end

depart(:,1) = t(:,1);
depart(:,5) = t(:,2);
depart(:,8) = t(:,3);
depart(:,10) = t(:,4);
depart(:,2) = t(:,1)+c(:,1);
```

```
depart(:,3) = depart(:,2)+c(:,2);
depart(:,4) = depart(:,3)+c(:,3);
depart(:,6) = depart(:,5)+c(:,4);
depart(:,7) = depart(:,6)+c(:,5);
depart(:,9) = depart(:,8)+c(:,6);

for j = 2:buses,

        lt(j, 2) = depart(j, 5)-depart(j, 4);
        lt(j, 3) = depart(j, 8)-depart(j, 7);
        lt(j, 4) = depart(j, 10)-depart(j, 9);

end

%while 1>2,

dist = [0 d1 d2 d3 d3 d4 d5 d5 d6 d6];

hold on
for i=1:buses,
        plot(depart(i,:),dist)
end

ylabel ('Distance (# of stops)');
xlabel ('Time (min)');
title ('Multiple Stops Simulation');

disp('paused')
pause

clc

%end %while end

%-----------------------------------------------
%        Parameter Adaptation Algorithm
%-----------------------------------------------

hold on

F1 = 1e+15*eye(3,3);
F2 = 1e+15*eye(3,3);
F3 = 1e+15*eye(3,3);

Fl1 = 1e+15*eye(5,5);
Fl2 = 1e+15*eye(5,5);
Fl3 = 1e+15*eye(3,3);
Fl4 = 1e+15*eye(5,5);
Fl5 = 1e+15*eye(3,3);
Fl6 = 1e+15*eye(3,3);

atheta1 = [ -.25
         .25
         1.21];

atheta2 = [ -.26
         .26
         1.22];

atheta3 = [ -.27
         .27
         1.23];

atheta_l1 = [ ffs/(2*ffs-jd*op(1,2)*mu)
          -jd*op(1,2)/(2*ffs-jd*op(1,2)*mu)
          (ffs*qr(1,2)*(acc1+acc2))/(4*acc1*acc2*jd)
          1/((ffs*qr(1,2)/(2*jd))*(1-mu*jd*op(1,2)/(2*ffs)))
```

```
            -mu/((ffs*qr(1,2)/(2*jd))*(1-mu*jd*op(1,2)/(2*ffs))) ];

atheta_l2 = [ ffs/(2*ffs-jd*op(1,2)*mu)
            -jd*op(2,2)/(2*ffs-jd*op(2,2)*mu)
             (ffs*qr(2,2)*(acc1+acc2))/(4*acc1*acc2*jd)
            1/((ffs*qr(2,2)/(2*jd))*(1-mu*jd*op(2,2)/(2*ffs)))
            -mu/((ffs*qr(2,2)/(2*jd))*(1-mu*jd*op(2,2)/(2*ffs))) ];

atheta_l3 = [ (ffs*qr(3,2)*(acc1+acc2))/(4*acc1*acc2*jd)
            1/((ffs*qr(3,2)/(2*jd))*(1-mu*jd*op(3,2)/(2*ffs)))
            -mu/((ffs*qr(3,2)/(2*jd))*(1-mu*jd*op(3,2)/(2*ffs))) ];


atheta_l4 = [ ffs/(2*ffs-jd*op(4,2)*mu)
              -jd*op(4,2)/(2*ffs-jd*op(4,2)*mu)
              (ffs*qr(4,2)*(acc1+acc2))/(4*acc1*acc2*jd)
              1/((ffs*qr(4,2)/(2*jd))*(1-mu*jd*op(4,2)/(2*ffs)))
              -mu/((ffs*qr(4,2)/(2*jd))*(1-mu*jd*op(4,2)/(2*ffs))) ];


atheta_l5 = [ (ffs*qr(5,2)*(acc1+acc2))/(4*acc1*acc2*jd)
              1/((ffs*qr(5,2)/(2*jd))*(1-mu*jd*op(5,2)/(2*ffs)))
              -mu/((ffs*qr(5,2)/(2*jd))*(1-mu*jd*op(5,2)/(2*ffs))) ];


atheta_l6 = [ (ffs*qr(6,2)*(acc1+acc2))/(4*acc1*acc2*jd)
              1/((ffs*qr(6,2)/(2*jd))*(1-mu*jd*op(6,2)/(2*ffs)))
              -mu/((ffs*qr(6,2)/(2*jd))*(1-mu*jd*op(6,2)/(2*ffs))) ];

etheta_l1 = [ .001
            -.0001
             .4
             .001
            -.001];

etheta_l2 = [ .001
            -.0001
            .4
            .0001
           -.0001];


etheta_l3 = [ .001
             .0001
             .4];

etheta_l4 = [ .001
            -.0001
             .4
            .0001
             -.0001];

etheta_l5 = [ .001
             .0001
             .4];

etheta_l6 = [ .001
             .0001
             .4];


etheta1 = [ -.12
               .12
              1.0];

etheta2 = [ -.12
            .12
```

```
                1.0];

etheta3 = [ -.12
               .12
                1.0 ];

flow=1/15*flow;

phi_l1 = [2*veh(1,2)/flow(1,2)
                l(1)/flow(1,2)
                1
                l(1)
                veh(1,2)];

eat_l(2,1) = tau(1)+etheta_l1'*phi_l1;
edt_l(2,1) = eat_l(2,1);

phi_l2 = [      2*veh(2,2)/flow(2,2)
                l(2)/flow(2,2)
                1
                l(2)
                veh(2,2)];

eat_l(2,2) = edt_l(2,1) + etheta_l2'*phi_l2;
edt_l(2,2) = eat_l(2,2);

phi_l3 = [      1
                l(3)
                veh(3,2)];

eat(2,2) = edt_l(2,2) + etheta_l3'*phi_l3;

h(2,2) = eat(2,2)-(t(1,1)+sum_tt(1,1));                 % arrival time headway

y=t(1,2);
u=h(2,2)+c(1,2)-c(2,2);
z=c(2,2);
phi1 = [        y
                u
                z];

edt(2,2) = tau(1) + etheta1'*phi1;


phi_l4 = [      2*veh(4,2)/flow(4,2)
                l(4)/flow(4,2)
                1
                l(4)
                veh(4,2)];

eat_l(2,4) = edt(2,2) + etheta_l4'*phi_l4;
edt_l(2,4) = eat_l(2,4);

phi_l5 = [      1
                l(5)
                veh(5,2)];

eat(2,3) = edt_l(2,4)+etheta_l5'*phi_l5;

h(2,3) = eat(2,3)-(t(1,2)+sum_tt(1,2));
y=t(1,3)-t(1,2);
u=h(2,3)+c(1,3)-c(2,3);
z=c(2,3);
phi2 = [        y
                u
                z];
```

```
edt(2,3) = edt(2,2)+etheta2'*phi2;

phi_l6 = [    1
              l(6)
              veh(6,2)];

eat(2,4) = edt(2,3)+etheta_l6'*phi_l6;

h(2,4) = eat(2,4)-(t(1,3)+sum_tt(1,3));
y=t(1,4)-t(1,3);
u=h(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [     y
             u
             z];
edt(2,4) = edt(2,3)+etheta3'*phi3;

%------------------------------------------------------------------
% Update at the 1st intersection (bus just past the 1st intersection)
%------------------------------------------------------------------

up_edt_l1(2,1) = tau(1) + c(2, 1);                    %depart(2,2) = tau(1) + c(2, 1)
up_eat_l1(2,1)=up_edt_l1(2,1);

up_eat_l1(2,2) = up_edt_l1(2,1) + etheta_l2'*phi_l2;     % for good, updated flow should be used in here
up_edt_l1(2,2) = up_eat_l1(2,2);

up_eat1(2,2) = up_edt_l1(2,2) + etheta_l3'*phi_l3;

y=t(1,2);
z=c(2,2);
h1(2,2) = up_eat1(2,2)-(t(1,1)+sum_tt(1,1));
u=h1(2,2)+c(1,2)-c(2,2);
phi1 = [     y
             u
             z];
up_edt1(2,2) = tau(1) + etheta1'*phi1;

up_eat_l1(2,4) = up_edt1(2,2) + etheta_l4'*phi_l4;
up_edt_l1(2,4) = up_eat_l1(2,4);

up_eat1(2,3) = up_edt_l1(2,4)+etheta_l5'*phi_l5;

y=t(1,3)-t(1,2);
z=c(2,3);
h1(2,3) = up_eat1(2,3)-(t(1,2)+sum_tt(1,2));
u=h1(2,3)+c(1,3)-c(2,3);
phi2 = [     y
             u
             z];
up_edt1(2,3) = up_edt1(2,2)+etheta2'*phi2;

up_eat1(2,4) = up_edt1(2,3)+etheta_l6'*phi_l6;

y=t(1,4)-t(1,3);
z=c(2,4);
h1(2,4) = up_eat1(2,4)-(t(1,3)+sum_tt(1,3));
u=h1(2,4)+c(1,4)-c(2,4);
phi3 = [     y
             u
             z];

up_edt1(2,4) = up_edt1(2,3)+etheta3'*phi3;

%------------------------------------------------------------------
% Update at the 2nd intersection (bus just past the 2nd intersection)
%------------------------------------------------------------------
```

```
up_edt_l2(2,2) = tau(1)+c(2,1)+c(2,2);                    %depart(2,3)=tau(1)+c(2,1)+c(2,2)
up_eat_l2(2,2)=up_edt_l2(2,2);

up_eat2(2,2) = up_edt_l2(2,2) + etheta_l3'*phi_l3;


z=c(2,2);
h2(2,2) = up_eat2(2,2)-(t(1,1)+sum_tt(1,1));
y=t(1,2);
u=h2(2,2)+c(1,2)-c(2,2);
phi1 = [     y
             u
             z];
up_edt2(2,2) = tau(1) + etheta1'*phi1;

up_eat_l2(2,4) = up_edt2(2,2) + etheta_l4'*phi_l4;
up_edt_l2(2,4) = up_eat_l2(2,4);

up_eat2(2,3) = up_edt_l2(2,4)+etheta_l5'*phi_l5;

z=c(2,3);
h2(2,3) = up_eat2(2,3)-(t(1,2)+sum_tt(1,2));
y=t(1,3)-t(1,2);
u=h2(2,3)+c(1,3)-c(2,3);
phi2 = [     y
             u
             z];
up_edt2(2,3) = up_edt2(2,2)+etheta2'*phi2;

up_eat2(2,4) = up_edt2(2,3)+etheta_l6'*phi_l6;
h2(2,4) = up_eat2(2,4)-(t(1,3)+sum_tt(1,3));

y=t(1,4)-t(1,3);
u=h2(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [     y
             u
             z];
up_edt2(2,4) = up_edt2(2,3)+etheta3'*phi3;



%-----------------------------------------------------------------
% Update when the bus arrives at the 2nd bus stop
%-----------------------------------------------------------------

up_eat3(2,2) = tau(1)+sum_tt(2,1);

z=c(2,2);
h3(2,2) = up_eat3(2,2)-(t(1,1)+sum_tt(1,1));
y=t(1,2);
u=h3(2,2)+c(1,2)-c(2,2);
phi1 = [     y
             u
             z];
up_edt3(2,2) = tau(1) + etheta1'*phi1;

up_eat_l3(2,4) = up_edt3(2,2) + etheta_l4'*phi_l4;
up_edt_l3(2,4) = up_eat_l3(2,4);

up_eat3(2,3) = up_edt_l3(2,4)+etheta_l5'*phi_l5;


z=c(2,3);
h3(2,3) = up_eat3(2,3)-(t(1,2)+sum_tt(1,2));
y=t(1,3)-t(1,2);
```

```
u=h3(2,3)+c(1,3)-c(2,3);
phi2 = [     y
             u
             z];
up_edt3(2,3) = up_edt3(2,2)+etheta2'*phi2;

up_eat3(2,4) = up_edt3(2,3)+etheta_l6'*phi_l6;

h3(2,4) = up_eat3(2,4)-(t(1,3)+sum_tt(1,3));
y=t(1,4)-t(1,3);
u=h3(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [     y
             u
             z];
up_edt3(2,4) = up_edt3(2,3)+etheta3'*phi3;


%----------------------------------------------------
% Update at 2nd Stop (when the bus just left the 2nd stop)
%----------------------------------------------------

up_edt4(2,2) = t(2,2);              % depart(2,5)=t(2,2)=tau(1)+sum_tt(2,1)+lt(2,2)

up_eat_l4(2,4) = up_edt4(2,2) + etheta_l4'*phi_l4;
up_edt_l4(2,4) = up_eat_l4(2,4);

up_eat4(2,3) = up_edt_l4(2,4)+etheta_l5'*phi_l5;
h4(2,3) = up_eat4(2,3)-(t(1,2)+sum_tt(1,2));
y=t(1,3)-t(1,2);
u=h4(2,3)+c(1,3)-c(2,3);
z=c(2,3);
phi2 = [     y
             u
             z];
up_edt4(2,3) = up_edt4(2,2)+etheta2'*phi2;

up_eat4(2,4) = up_edt4(2,3)+etheta_l6'*phi_l6;

h4(2,4) = up_eat4(2,4)-(t(1,3)+sum_tt(1,3));
y=t(1,4)-t(1,3);
u=h4(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [     y
             u
             z];
up_edt4(2,4) = up_edt4(2,3)+etheta3'*phi3;


%----------------------------------------------------------
% Update at the 3rd intersection (bus just past the 3rd intersection)
%----------------------------------------------------------

up_edt_l5(2,4) = t(2,2)+c(2,4);    %depart(2,6)=t(2,2)+c(2,4)

up_eat5(2,3) = up_edt_l5(2,4)+etheta_l5'*phi_l5;
h5(2,3) = up_eat5(2,3)-(t(1,2)+sum_tt(1,2));
y=t(1,3)-t(1,2);
u=h5(2,3)+c(1,3)-c(2,3);
z=c(2,3);
phi2 = [     y
             u
             z];
up_edt5(2,3) = up_edt4(2,2)+etheta2'*phi2; % up_edt4(2,2), because there is no up_edt5(2,2) in this update

up_eat5(2,4) = up_edt5(2,3)+etheta_l6'*phi_l6;
```

```
h5(2,4) = up_eat5(2,4)-(t(1,3)+sum_tt(1,3));
y=t(1,4)-t(1,3);
u=h5(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [    y
            u
            z];
up_edt5(2,4) = up_edt5(2,3)+etheta3'*phi3;


%-----------------------------------------------------------------
%  Update when the bus just arrives at the 3rd stop
%-----------------------------------------------------------------

up_eat6(2,3) = t(2,2)+sum_tt(2,2);

h6(2,3) = up_eat6(2,3)-(t(1,2)+sum_tt(1,2));
y=t(1,3)-t(1,2);
u=h6(2,3)+c(1,3)-c(2,3);
z=c(2,3);
phi2 = [    y
            u
            z];
up_edt6(2,3) = up_edt4(2,2)+etheta2'*phi2; % up_edt4(2,2), because there is no up_edt6(2,2)

up_eat6(2,4) = up_edt6(2,3)+etheta_l6'*phi_l6;

h6(2,4) = up_eat6(2,4)-(t(1,3)+sum_tt(1,3));
y=t(1,4)-t(1,3);
u=h6(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [    y
            u
            z];
up_edt6(2,4) = up_edt6(2,3)+etheta3'*phi3;


%---------------------------------------------------------
%  Update at 3rd Stop (when the bus just left the 3rd bus stop)
%---------------------------------------------------------

up_edt7(2,3) = t(2,3);

up_eat7(2,4) = up_edt7(2,3)+etheta_l6'*phi_l6;

h7(2,4) = up_eat7(2,4)-(t(1,3)+sum_tt(1,3));
y=t(1,4)-t(1,3);
u=h7(2,4)+c(1,4)-c(2,4);
z=c(2,4);
phi3 = [ y
            u
            z];
up_edt7(2,4) = up_edt7(2,3)+etheta3'*phi3;



%=================================================================
%                              Plotting
%=================================================================

subplot(3,1,1)

qq1 = [0    up_edt_l1(2,1) up_edt_l1(2,1) up_edt_l2(2,2) up_edt_l2(2,2) up_eat2(2,2)];
rr1 = [eat(2,2) eat(2,2)     up_eat1(2,2)  up_eat1(2,2)  up_eat2(2,2)  up_eat2(2,2)];
plot(qq1, rr1)
title('Bus Arrival Time Estimation')
```

```
ylabel('stop #2')
subplot(3,1,2)
qq2 = [0     up_edt_l1(2,1) up_edt_l1(2,1) up_edt_l2(2,2) up_edt_l2(2,2) up_eat3(2,2) up_eat3(2,2) up_edt4(2,2) up_edt4(2,2)
up_edt_l5(2,4) up_edt_l5(2,4) up_eat5(2,3)];
rr2 = [eat(2,3) eat(2,3)     up_eat1(2,3)  up_eat1(2,3)  up_eat2(2,3)  up_eat2(2,3) up_eat3(2,3) up_eat3(2,3) up_eat4(2,3)
up_eat4(2,3)  up_eat5(2,3)  up_eat5(2,3)];
plot(qq2, rr2)
ylabel('stop #3')
subplot(3,1,3)
qq3 = [0     up_edt_l1(2,1) up_edt_l1(2,1) up_edt_l2(2,2) up_edt_l2(2,2) up_eat3(2,2) up_eat3(2,2) up_edt4(2,2) up_edt4(2,2)
up_edt_l5(2,4) up_edt_l5(2,4) up_eat6(2,3) up_eat6(2,3) up_edt7(2,3) up_edt7(2,3) up_eat7(2,4)];
rr3 = [eat(2,4) eat(2,4)     up_eat1(2,4)  up_eat1(2,4)  up_eat2(2,4)  up_eat2(2,4) up_eat3(2,4) up_eat3(2,4) up_eat4(2,4)
up_eat4(2,4)  up_eat5(2,4)  up_eat5(2,4) up_eat6(2,4) up_eat6(2,4) up_eat7(2,4) up_eat7(2,4)];
plot(qq3, rr3)
ylabel('stop #4')
xlabel('clock time')


pause

mm(1,1) = etheta1(1);
nn(1,1) = etheta1(2);
oo(1,1) = etheta1(3);

mm(2,1) = etheta2(1);
nn(2,1) = etheta2(2);
oo(2,1) = etheta2(3);

mm(3,1) = etheta3(1);
nn(3,1) = etheta3(2);
oo(3,1) = etheta3(3);

err1(1) = atheta1'*phi1-etheta1'*phi1;
err2(1) = atheta1'*phi2-etheta2'*phi2;
err3(1) = atheta1'*phi3-etheta3'*phi3;

err_l1(1) = atheta_l1'*phi_l1-etheta_l1'*phi_l1;
err_l2(1) = atheta_l2'*phi_l2-etheta_l2'*phi_l2;
err_l3(1) = atheta_l3'*phi_l3-etheta_l3'*phi_l3;
err_l4(1) = atheta_l4'*phi_l4-etheta_l4'*phi_l4;
err_l5(1) = atheta_l5'*phi_l5-etheta_l5'*phi_l5;
err_l6(1) = atheta_l6'*phi_l6-etheta_l6'*phi_l6;

 [F1, etheta1] = update1(F1, phi1, etheta1, atheta1);
[F2, etheta2] = update2(F2, phi2, etheta2, atheta2);
[F3, etheta3] = update3(F3, phi3, etheta3, atheta3);

[Fl1, etheta_l1] = updatel1(Fl1, phi_l1, etheta_l1, atheta_l1);
[Fl2, etheta_l2] = updatel2(Fl2, phi_l2, etheta_l2, atheta_l2);
[Fl3, etheta_l3] = updatel3(Fl3, phi_l3, etheta_l3, atheta_l3);
[Fl4, etheta_l4] = updatel4(Fl4, phi_l4, etheta_l4, atheta_l4);
[Fl5, etheta_l5] = updatel5(Fl5, phi_l5, etheta_l5, atheta_l5);
[Fl6, etheta_l6] = updatel6(Fl6, phi_l6, etheta_l6, atheta_l6);

hold off

disp('finished 2nd bus');
pause

%==================================================
% Third to Nth Bus Estimation and Parameter Update
%==================================================

for i =3:buses,

        atheta1 = [ -.2+.05*sin(.13*(i-1))
                    .2+.05*sin(.13*(i-1))
```

```
                1.2+.75*sin(.13*(i-1))];

atheta2 = [ -.2+.05*sin(.13*(i-1))
                .2+.05*sin(.13*(i-1))
                1.2+.75*sin(.13*(i-1))];

atheta3 = [ -.2+.05*sin(.13*(i-1))
                .2+.05*sin(.13*(i-1))
                1.2+.75*sin(.13*(i-1))];

atheta_l1 = [ ffs/(2*ffs-jd*op(1,i)*mu)
                -jd*op(1,i)/(2*ffs-jd*op(1,i)*mu)
                (ffs*qr(1,i)*(acc1+acc2))/(4*acc1*acc2*jd)
                1/((ffs*qr(1,i)/(2*jd))*(1-mu*jd*op(1,i)/(2*ffs)))
                -mu/((ffs*qr(1,i)/(2*jd))*(1-mu*jd*op(1,i)/(2*ffs))) ];

atheta_l2 = [ ffs/(2*ffs-jd*op(2,i)*mu)
                -jd*op(2,i)/(2*ffs-jd*op(2,i)*mu)
                (ffs*qr(2,i)*(acc1+acc2))/(4*acc1*acc2*jd)
                1/((ffs*qr(2,i)/(2*jd))*(1-mu*jd*op(2,i)/(2*ffs)))
                -mu/((ffs*qr(2,i)/(2*jd))*(1-mu*jd*op(2,i)/(2*ffs))) ];

atheta_l3 = [ (ffs*qr(3,i)*(acc1+acc2))/(4*acc1*acc2*jd)
                1/((ffs*qr(3,i)/(2*jd))*(1-mu*jd*op(3,i)/(2*ffs)))
                -mu/((ffs*qr(3,i)/(2*jd))*(1-mu*jd*op(3,i)/(2*ffs))) ];

atheta_l4 = [ ffs/(2*ffs-jd*op(4,i)*mu)
                -jd*op(4,i)/(2*ffs-jd*op(4,i)*mu)
                (ffs*qr(4,i)*(acc1+acc2))/(4*acc1*acc2*jd)
                1/((ffs*qr(4,i)/(2*jd))*(1-mu*jd*op(4,i)/(2*ffs)))
                -mu/((ffs*qr(4,i)/(2*jd))*(1-mu*jd*op(4,i)/(2*ffs))) ];

atheta_l5 = [ (ffs*qr(5,i)*(acc1+acc2))/(4*acc1*acc2*jd)
                1/((ffs*qr(5,i)/(2*jd))*(1-mu*jd*op(5,i)/(2*ffs)))
                -mu/((ffs*qr(5,i)/(2*jd))*(1-mu*jd*op(5,i)/(2*ffs))) ];

atheta_l6 = [ (ffs*qr(6,i)*(acc1+acc2))/(4*acc1*acc2*jd)
                1/((ffs*qr(6,i)/(2*jd))*(1-mu*jd*op(6,i)/(2*ffs)))
                -mu/((ffs*qr(6,i)/(2*jd))*(1-mu*jd*op(6,i)/(2*ffs))) ];

clf
hold on

sum_tau = 0;
for j = 1:i-1
            sum_tau = sum_tau+tau(j);
end

phi_l1 = [2*veh(1,i)/flow(1,i)
        l(1)/flow(1,i)
            1
            l(1)
            veh(1,i)];

eat_l(i,1) = sum_tau+etheta_l1'*phi_l1;
edt_l(i,1) = eat_l(i,1);

phi_l2 = [ 2*veh(2,i)/flow(2,i)
            l(2)/flow(2,i)
            1
            l(2)
            veh(2,i)];

eat_l(i,2) = edt_l(i,1) + etheta_l2'*phi_l2;
edt_l(i,2) = eat_l(i,2);

phi_l3 = [ 1
```

```
        l(3)
        veh(3,i)];

eat(i,2) = edt_l(i,2) + etheta_l3'*phi_l3;

h(i,2) = eat(i,2)-(t(i-1,1)+sum_tt(i-1,1));
y=up_edt4(i-1,2)-t(i-1,1);
u=h(i,2)+c(i-1,2)-c(i,2);
z=c(i,2);
phi1 = [ y
         u
         z];
edt(i,2) = sum_tau + etheta1'*phi1;

phi_l4 = [ 2*veh(4,i)/flow(4,i)
           l(4)/flow(4,i)
           1
           l(4)
           veh(4,i)];

eat_l(i,4) = edt(i,2) + etheta_l4'*phi_l4;
edt_l(i,4) = eat_l(i,4);

phi_l5 = [ 1
           l(5)
           veh(5,i)];

eat(i,3) = edt_l(i,4)+etheta_l5'*phi_l5;
h(i,3) = eat(i,3)-(t(i-1,2)+sum_tt(i-1,2)); %t(2,3)
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
edt(i,3) = edt(i,2)+etheta2'*phi2;

phi_l6 = [ 1
           l(6)
           veh(6,i)];

eat(i,4) = edt(i,3)+etheta_l6'*phi_l6;

h(i,4) = eat(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);            % wrong!! ??
u=h(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
edt(i,4) = edt(i,3)+etheta3'*phi3;


%-----------------------------------------------------
% Update at the 1st intersection (bus just past the 1st stop)
%-----------------------------------------------------

up_edt_l1(i,1) = sum_tau + c(i, 1);           %depart(2,2) = tau(1) + c(2, 1)
up_eat_l1(i,1)=up_edt_l1(i,1);

up_eat_l1(i,2) = up_edt_l1(i,1) + etheta_l2'*phi_l2;  % for good, updated flow should be used in here
up_edt_l1(i,2) = up_eat_l1(i,2);

up_eat1(i,2) = up_edt_l1(i,2) + etheta_l3'*phi_l3;

y=up_edt4(i-1,2)-t(i-1,1);
h1(i,2) = up_eat1(i,2)-(t(i-1,1)+sum_tt(i-1,1));
```

```
u=h1(i,2)+c(i-1,2)-c(i,2);
z=c(i,2);
phi1 = [ y
         u
         z];
up_edt1(i,2) = sum_tau + etheta1'*phi1;

up_eat_l1(i,4) = up_edt1(i,2) + etheta_l4'*phi_l4;
up_edt_l1(i,4) = up_eat_l1(i,4);


up_eat1(i,3) = up_edt_l1(i,4)+etheta_l5'*phi_l5;
h1(i,3) = up_eat1(i,3)-(t(i-1,2)+sum_tt(i-1,2));
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h1(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
up_edt1(i,3) = up_edt1(i,2)+etheta2'*phi2;

up_eat1(i,4) = up_edt1(i,3)+etheta_l6'*phi_l6;

h1(i,4) = up_eat1(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h1(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt1(i,4) = up_edt1(i,3)+etheta3'*phi3;


%-----------------------------------------------------------------
% Update at the 2nd intersection (bus just past the 2nd intersection)
%-----------------------------------------------------------------

up_edt_l2(i,2) = sum_tau+c(i,1)+c(i,2);        %depart(2,3)=tau(1)+c(2,1)+c(2,2)
up_eat_l2(i,2)=up_edt_l2(i,2);

up_eat2(i,2) = up_edt_l2(i,2) + etheta_l3'*phi_l3;

h2(i,2) = up_eat2(i,2)-(t(i-1,1)+sum_tt(i-1,1));
y=up_edt4(i-1,2)-t(i-1,1);  % up_edt4 or 2 ???
u=h2(i,2)+c(i-1,2)-c(i,2);
z=c(i,2);
phi1 = [ y
         u
         z];
up_edt2(i,2) = sum_tau + etheta1'*phi1;

up_eat_l2(i,4) = up_edt2(i,2) + etheta_l4'*phi_l4;
up_edt_l2(i,4) = up_eat_l2(i,4);

up_eat2(i,3) = up_edt_l2(i,4)+etheta_l5'*phi_l5;
h2(i,3) = up_eat2(i,3)-(t(i-1,2)+sum_tt(i-1,2));
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h2(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
up_edt2(i,3) = up_edt2(i,2)+etheta2'*phi2;

up_eat2(i,4) = up_edt2(i,3)+etheta_l6'*phi_l6;

h2(i,4) = up_eat2(i,4)-(t(i-1,3)+sum_tt(i-1,3));
```

```
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h2(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt2(i,4) = up_edt2(i,3)+etheta3'*phi3;

%-----------------------------------------------------
% Update when the bus arrives at the 2nd bus stop
%-----------------------------------------------------

up_eat3(i,2) = sum_tau+sum_tt(i,1);

h3(i,2) = up_eat3(i,2)-(t(i-1,1)+sum_tt(i-1,1));
y=up_edt4(i-1,2)-t(i-1,2);
u=h3(i,2)+c(i-1,2)-c(i,2);
z=c(i,2);
phi1 = [ y
         u
         z];
up_edt3(i,2) = sum_tau + etheta1'*phi1;

up_eat_l3(i,4) = up_edt3(i,2) + etheta_l4'*phi_l4;
up_edt_l3(i,4) = up_eat_l3(i,4);

up_eat3(i,3) = up_edt_l3(i,4)+etheta_l5'*phi_l5;
h3(i,3) = up_eat3(i,3)-(t(i-1,2)+sum_tt(i-1,2));
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h3(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
up_edt3(i,3) = up_edt3(i,2)+etheta2'*phi2;

up_eat3(i,4) = up_edt3(i,3)+etheta_l6'*phi_l6;

h3(i,4) = up_eat3(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h3(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt3(i,4) = up_edt3(i,3)+etheta3'*phi3;

%-----------------------------------------------
% Update at 2nd Stop (when the bus just left the 2nd stop)
%-----------------------------------------------

up_edt4(i,2) = t(i,2);          % depart(2,5)=t(2,2)=tau(1)+sum_tt(2,1)+lt(2,2)

up_eat_l4(i,4) = up_edt4(i,2) + etheta_l4'*phi_l4;
up_edt_l4(i,4) = up_eat_l4(i,4);

up_eat4(i,3) = up_edt_l4(i,4)+etheta_l5'*phi_l5;
h4(i,3) = up_eat4(i,3)-(t(i-1,2)+sum_tt(i-1,2));
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h4(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
up_edt4(i,3) = up_edt4(i,2)+etheta2'*phi2;

up_eat4(i,4) = up_edt4(i,3)+etheta_l6'*phi_l6;
```

```
h4(i,4) = up_eat4(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h4(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt4(i,4) = up_edt4(i,3)+etheta3'*phi3;


%-----------------------------------------------------------------
% Update at the 3rd intersection (bus just past the 3rd intersection)
%-----------------------------------------------------------------

up_edt_l5(i,4) = t(i,2)+c(i,4);      %depart(2,6)=t(2,2)+c(2,4)

up_eat5(i,3) = up_edt_l5(i,4)+etheta_l5'*phi_l5;

h5(i,3) = up_eat5(i,3)-(t(i-1,2)+sum_tt(i-1,2));
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h5(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
up_edt5(i,3) = up_edt4(i,2)+etheta2'*phi2; % up_edt4(2,2), because there is no up_edt5(2,2) in this update

up_eat5(i,4) = up_edt5(i,3)+etheta_l6'*phi_l6;

h5(i,4) = up_eat5(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h5(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt5(i,4) = up_edt5(i,3)+etheta3'*phi3;

%-----------------------------------------------------------------
% Update when the bus just arrives at the 3rd stop
%-----------------------------------------------------------------

up_eat6(i,3) = t(i,2)+sum_tt(i,2);

h6(i,3) = up_eat6(i,3)-(t(i-1,2)+sum_tt(i-1,2));
y=up_edt7(i-1,3)-up_edt4(i-1,2);
u=h6(i,3)+c(i-1,3)-c(i,3);
z=c(i,3);
phi2 = [ y
         u
         z];
up_edt6(i,3) = up_edt4(i,2)+etheta2'*phi2; % up_edt4(2,2), because there is no up_edt6(2,2)

up_eat6(i,4) = up_edt6(i,3)+etheta_l6'*phi_l6;

h6(i,4) = up_eat6(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h6(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt6(i,4) = up_edt6(i,3)+etheta3'*phi3;

%-----------------------------------------------
% Update at 3rd Stop (when the bus just left the 3rd bus stop)
```

```
%------------------------------------------------------------

up_edt7(i,3) = t(i,3);

up_eat7(i,4) = up_edt7(i,3)+etheta_l6'*phi_l6;

h7(i,4) = up_eat7(i,4)-(t(i-1,3)+sum_tt(i-1,3));
y=up_edt7(i-1,4)-up_edt7(i-1,3);
u=h7(i,4)+c(i-1,4)-c(i,4);
z=c(i,4);
phi3 = [ y
         u
         z];
up_edt7(i,4) = up_edt7(i,3)+etheta3'*phi3;

%------------------------------------------------------------
% Plotting
%------------------------------------------------------------

subplot(3,1,1)
qq1=[0      up_edt_l1(i,1) up_edt_l1(i,1) up_edt_l2(i,2) up_edt_l2(i,2) up_eat2(i,2)];
rr1=[eat(i,2) eat(i,2)    up_eat1(i,2)  up_eat1(i,2)  up_eat2(i,2)  up_eat2(i,2)];
plot(qq1, rr1)

subplot(3,1,2)
qq2 = [0     up_edt_l1(i,1) up_edt_l1(i,1) up_edt_l2(i,2) up_edt_l2(i,2) up_eat3(i,2) up_eat3(i,2) up_edt4(i,2) up_edt4(i,2)
up_edt_l5(i,4) up_edt_l5(i,4) up_eat5(i,3)];
rr2 = [eat(i,3) eat(i,3)    up_eat1(i,3)  up_eat1(i,3)  up_eat2(i,3)  up_eat2(i,3) up_eat3(i,3) up_eat3(i,3) up_eat4(i,3)
up_eat4(i,3)  up_eat5(i,3)  up_eat5(i,3)];
plot(qq2, rr2)

subplot(3,1,3)
qq3 = [0     up_edt_l1(i,1) up_edt_l1(i,1) up_edt_l2(i,2) up_edt_l2(i,2) up_eat3(i,2) up_eat3(i,2) up_edt4(i,2) up_edt4(i,2)
up_edt_l5(i,4) up_edt_l5(i,4) up_eat6(i,3) up_eat6(i,3) up_edt7(i,3) up_edt7(i,3) up_eat7(i,4)];
rr3 = [eat(i,4) eat(i,4)    up_eat1(i,4)  up_eat1(i,4)  up_eat2(i,4)  up_eat2(i,4) up_eat3(i,4) up_eat3(i,4) up_eat4(i,4)
up_eat4(i,4)  up_eat5(i,4)  up_eat5(i,4) up_eat6(i,4) up_eat6(i,4) up_eat7(i,4) up_eat7(i,4)];
plot(qq3, rr3)


pause

mm(1,i-1) = etheta1(1);
nn(1,i-1) = etheta1(2);
oo(1,i-1) = etheta1(3);

mm(2,i-1) = etheta2(1);
nn(2,i-1) = etheta2(2);
oo(2,i-1) = etheta2(3);

mm(3,i-1) = etheta3(1);
nn(3,i-1) = etheta3(2);
oo(3,i-1) = etheta3(3);

err1(i-1) = atheta1'*phi1-etheta1'*phi1;
err2(i-1) = atheta2'*phi2-etheta2'*phi2;
err3(i-1) = atheta3'*phi3-etheta3'*phi3;

err_l1(i-1) = atheta_l1'*phi_l1-etheta_l1'*phi_l1;
err_l2(i-1) = atheta_l2'*phi_l2-etheta_l2'*phi_l2;
err_l3(i-1) = atheta_l3'*phi_l3-etheta_l3'*phi_l3;
err_l4(i-1) = atheta_l4'*phi_l4-etheta_l4'*phi_l4;
err_l5(i-1) = atheta_l5'*phi_l5-etheta_l5'*phi_l5;
err_l6(i-1) = atheta_l6'*phi_l6-etheta_l6'*phi_l6;

[F1, etheta1] = update1(F1, phi1, etheta1, atheta1);
[F2, etheta2] = update2(F2, phi2, etheta2, atheta2);
[F3, etheta3] = update3(F3, phi3, etheta3, atheta3);
```

```
        [Fl1, etheta_l1] = updatel1(Fl1, phi_l1, etheta_l1, atheta_l1);
        [Fl2, etheta_l2] = updatel2(Fl2, phi_l2, etheta_l2, atheta_l2);
        [Fl3, etheta_l3] = updatel3(Fl3, phi_l3, etheta_l3, atheta_l3);
        [Fl4, etheta_l4] = updatel4(Fl4, phi_l4, etheta_l4, atheta_l4);
        [Fl5, etheta_l5] = updatel5(Fl5, phi_l5, etheta_l5, atheta_l5);
        [Fl6, etheta_l6] = updatel6(Fl6, phi_l6, etheta_l6, atheta_l6);

end

clf
hold off

%while 1>2,


%================================================================
%                   Parameter Errors
%================================================================

for i = 1:stops-1,
        clf
        hold on

        ttt = -.2+.05*sin(.13*[1:buses]);
        plot(ttt);


        xx=[];
        for j = 1:buses-1,
                xx=[xx mm(i,j) mm(i,j)];
        end

        yy = [1];
        for vv = 2: buses-1
                yy = [yy;vv;vv];
        end
        yy = [yy;buses];

        plot(yy,xx);

        ylabel ('parameter a');
        xlabel ('buses');
        title ('Parameter Convergency');
        pause

        hold off

        clf
        hold on

        ttt = .2+.05*sin(.13*[1:buses]);
        plot(ttt);

        xx=[];
        for j = 1:buses-1,
                xx=[xx nn(i,j) nn(i,j)];
        end

        yy = [1];
        for vv = 2: buses-1
                yy = [yy;vv;vv];
        end
        yy = [yy;buses];

        plot(yy,xx);
```

```
        ylabel ('parameter b');
        xlabel ('buses');
        title ('Parameter Convergency');

        pause
        hold off

        clf
        hold on

        ttt = 1.2+.75*sin(.13*[1:buses]);
        plot(ttt);

        xx=[];
        for j = 1:buses-1,
                xx=[xx oo(i,j) oo(i,j)];
        end
        yy = [1];
        for vv = 2: buses-1
                yy = [yy;vv;vv];
        end
        yy = [yy;buses];

        plot(yy,xx);

        ylabel ('parameter d');
        xlabel ('buses');
        title ('Parameter Convergency');

        pause
        hold off

end  %plotting for loop

%end % while

%===========================================================
%                       Estimation Errors Plotting
%===========================================================

clf
hold on

xx1 = [];
xx2 = [];
xx3 = [];
xx_l1=[];
xx_l2=[];
xx_l3=[];
xx_l4=[];
xx_l5=[];
xx_l6=[];


for j = 1:buses-1,

        xx1 = [xx1 err1(j) err1(j)];
        xx2 = [xx2 err2(j) err2(j)];
        xx3 = [xx3 err3(j) err3(j)];

        xx_l1 = [xx_l1 err_l1(j) err_l1(j)];
        xx_l2 = [xx_l2 err_l2(j) err_l2(j)];
        xx_l3 = [xx_l3 err_l3(j) err_l3(j)];
        xx_l4 = [xx_l4 err_l4(j) err_l4(j)];
        xx_l5 = [xx_l5 err_l5(j) err_l5(j)];
        xx_l6 = [xx_l6 err_l6(j) err_l6(j)];
```

```
end

yy = [1];
for vv = 2:buses-1,
            yy = [yy; vv; vv];
end
yy = [yy; buses];

plot(yy, xx1);
ylabel( 'error at stop 2');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx2);
ylabel( 'errors at stop 3');
xlabel( '# of buses');
title('Estimation Error');

figure

plot(yy, xx3);
ylabel( 'errors at stop 4');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx_l1);
ylabel( 'errors at 1st intersection');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx_l2);
ylabel( 'errors at 2nd intersection');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx_l3);
ylabel( 'errors at 2nd bus stop (arrival time)');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx_l4);
ylabel( 'errors at 3rd intersection');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx_l5);
ylabel( 'errors at 3rd bus stop (arrival time)');
xlabel( '# of buses');
title( 'Estimation Error');

figure

plot(yy, xx_l6);
ylabel( 'errors at 4th bus stop (arrival time)');
xlabel( '# of buses');
```

```
title( 'Estimation Error');
```

APPENDIX E Bus Data Entry Form

March 9 1994

Route Name: Foxridge-Winsor Hill

Starting Location: **Squires Eastbound** (Squires Side)　　Starting Time: 7:15 AM

Name of Collector: Mr.Y. Zhang

Weather: dry　　wet

Comments: ______________________________

Stop Number	Arrival Time	# of Passenger Load	# of Passenger Unload	Departure Time	total # of stops made in between the major stop	total # of passengers loading in between the major stop	total # of passengers unloading in between the major stop
1: Squires Eastbound							
2: Winsor Hill (Ascot Lane)							
3. Squires Westbound							
4: Tall Oaks Circle (Foxridge)							
5: Swimming Pool (Foxridge)							
6: University (near duck pond)							
7: Squires Eastbound							
2: Winsor Hill (Ascot Lane)							
3. Squires Westbound							
4: Tall Oaks Circle							

CAR DATA ENTRY FORM

March 8 1994

Route Name: Foxridge-Winsor Hill

Starting Location: **Squires Eastbound** (Squires Side) Starting Time: 7:15 AM

Name of Collector: Mr. S. Bae

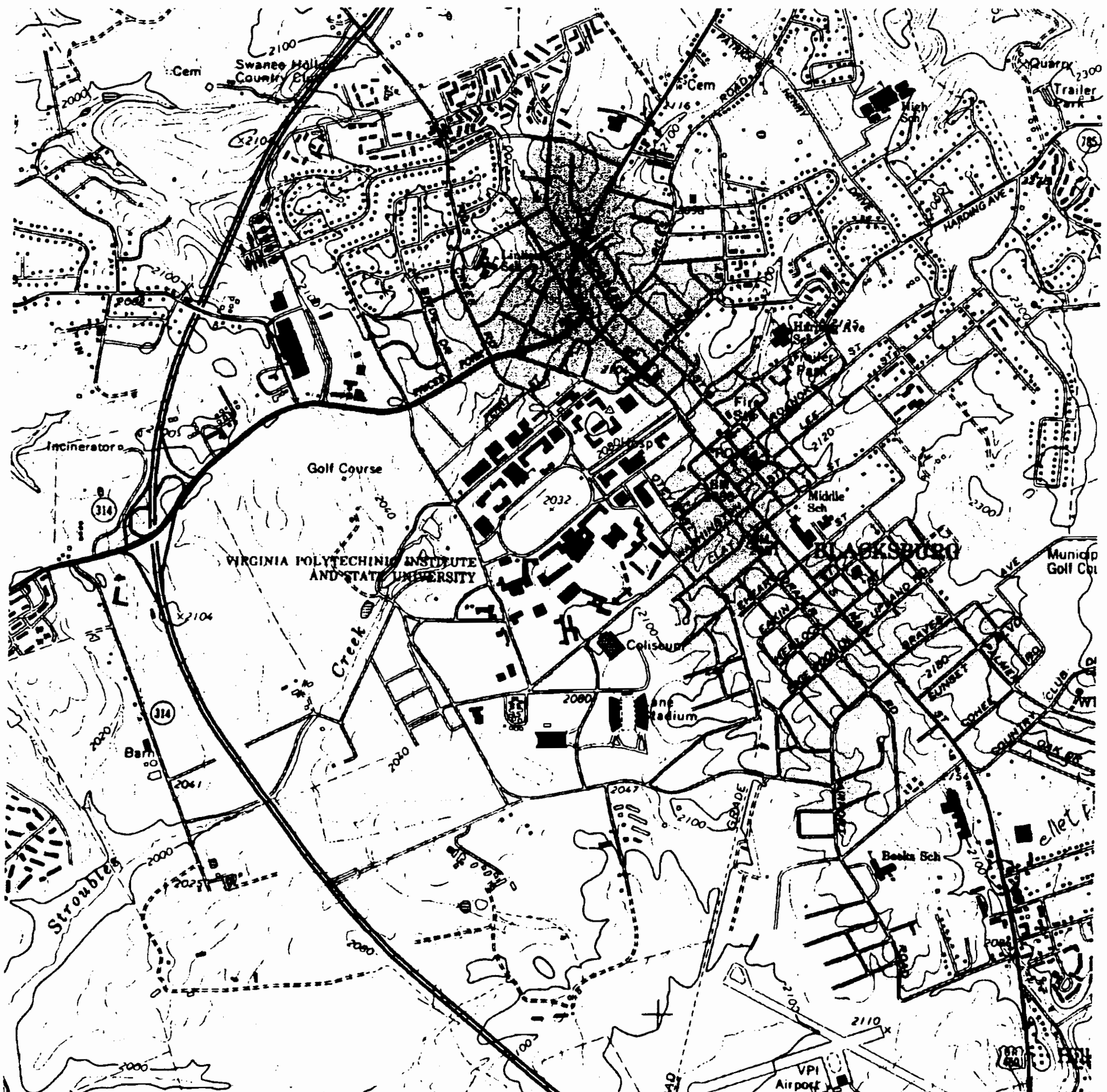
Weather: dry wet

Comments: ____________________

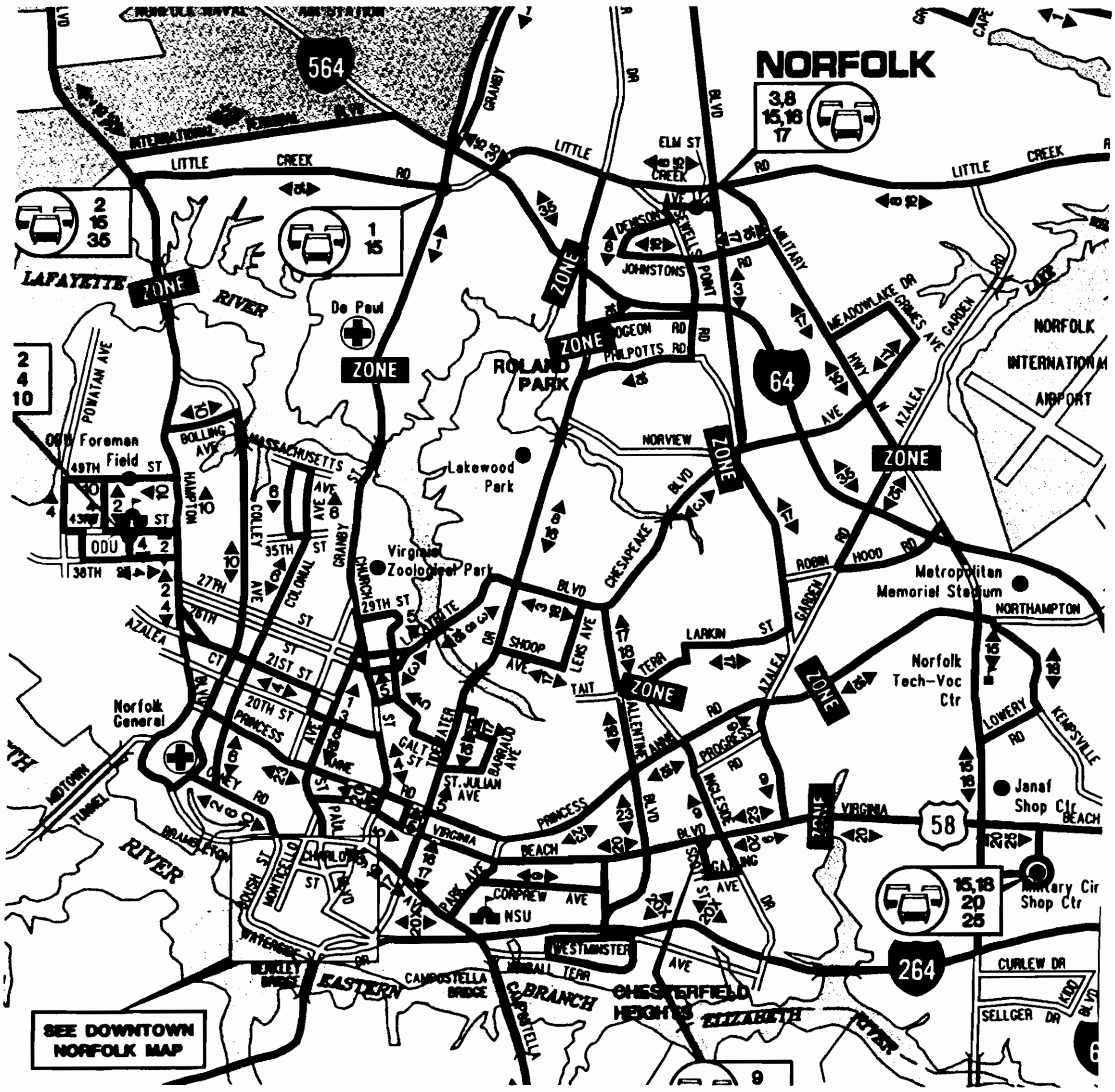
____

Stop Number	Arrival Time	Departure Time
1: Squires Eastbound		
2: Winsor Hill (Ascot Lane)		
3. Squires Westbound		
4: Tall Oaks Circle (Foxridge)		
5: Swimming Pool (Foxridge)		
6: University (near duck pond)		
7: Squires Eastbound		

APPENDIX F Area Map of Blacksburg



APPENDIX G Route Map of Norfolk



APPENDIX H Residual Table for Blacksburg Data (Travel Time)

RESIDUAL OUTPUT

Observation	*Predicted Y*	*Residuals*
1	6.414974522	1.585025478
2	7.080031709	-3.080031709
3	9.740260456	-0.740260456
4	1.759574216	0.240425784
5	3.089688589	0.910311411
6	5.084860149	-0.084860149
7	5.749917336	2.250082664
8	7.745088896	-2.745088896
9	11.07037483	-1.070374829
10	1.759574216	-0.759574216
11	5.084860149	-1.084860149
12	4.419802962	-0.419802962
13	5.749917336	-0.749917336
14	7.745088896	-2.745088896
15	9.075203269	0.924796731
16	2.424631402	-1.424631402
17	4.419802962	0.580197038
18	3.754745776	0.245254224
19	5.084860149	1.915139851
20	6.414974522	-2.414974522
21	9.740260456	-0.740260456
22	1.759574216	-0.759574216
23	3.754745776	1.245254224
24	3.754745776	-0.754745776
25	5.749917336	0.250082664
26	7.080031709	-1.080031709
27	10.40531764	1.594682357
28	1.759574216	-0.759574216
29	4.419802962	1.580197038
30	3.754745776	0.245254224
31	7.080031709	-3.080031709
32	7.080031709	-0.080031709
33	10.40531764	0.594682357
34	2.424631402	0.575368598
35	4.419802962	-0.419802962
36	4.419802962	-1.419802962
37	7.080031709	-1.080031709

MAX Residual

3.25973954

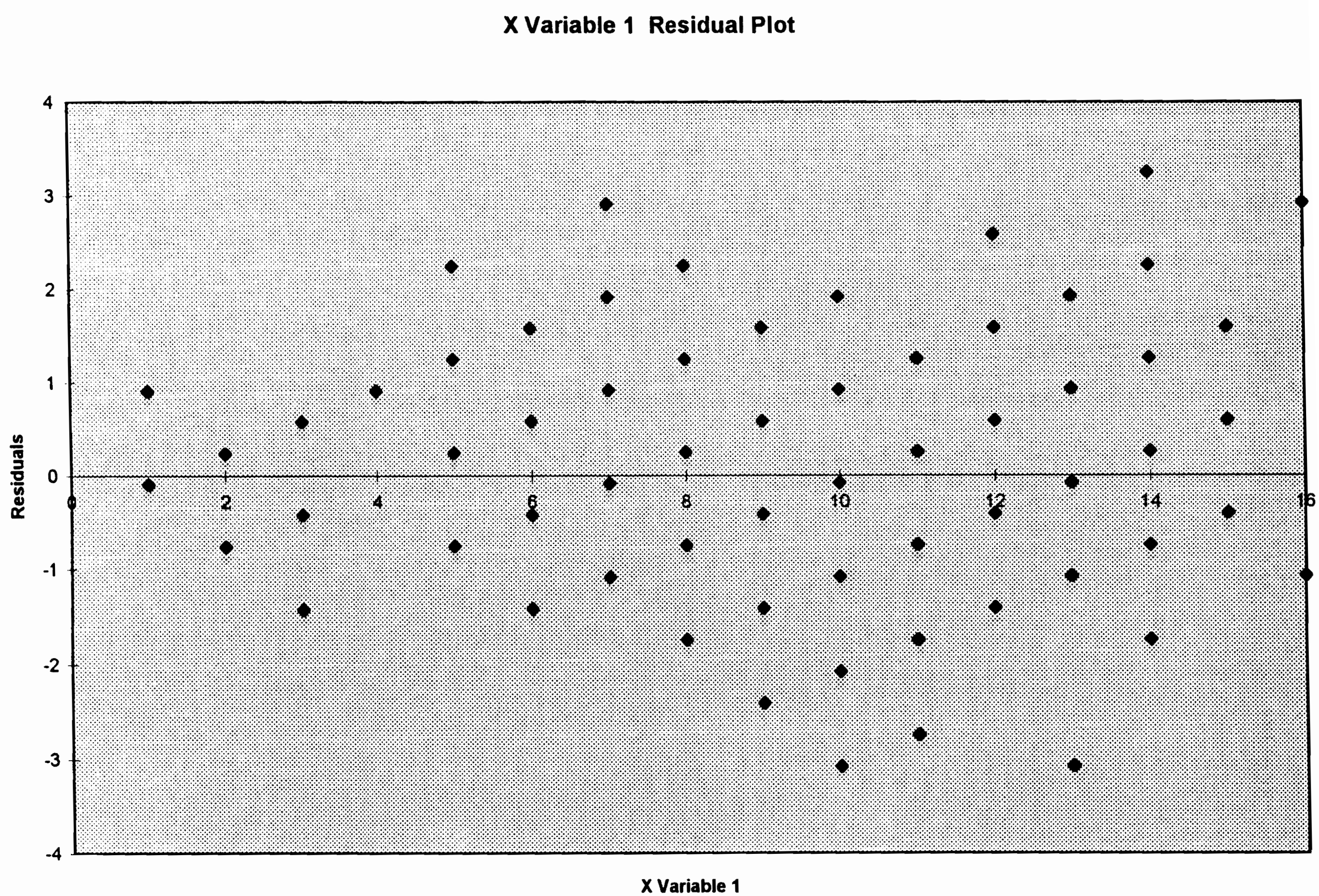
38	6.414974522	-1.414974522
39	9.075203269	1.924796731
40	2.424631402	-0.424631402
41	5.749917336	-0.749917336
42	4.419802962	-1.419802962
43	6.414974522	-1.414974522
44	6.414974522	-1.414974522
45	9.075203269	1.924796731
46	1.094517029	0.905482971
47	4.419802962	0.580197038
48	3.754745776	-0.754745776
49	6.414974522	-0.414974522
50	5.084860149	0.915139851
51	9.740260456	2.259739544
52	1.759574216	-0.759574216
53	4.419802962	1.580197038
54	5.084860149	-1.084860149
55	7.080031709	-1.080031709
56	5.084860149	-0.084860149
57	9.740260456	1.259739544
58	1.094517029	0.905482971
59	3.754745776	0.245254224
60	3.754745776	0.245254224
61	10.40531764	-0.405317643
62	8.410146083	-1.410146083
63	7.745088896	0.254911104
64	9.075203269	0.924796731
65	4.419802962	1.580197038
66	3.089688589	0.910311411
67	7.745088896	-0.745088896
68	10.40531764	1.594682357
69	7.080031709	-0.080031709
70	5.084860149	1.915139851
71	7.745088896	0.254911104
72	5.084860149	0.915139851
73	4.419802962	0.580197038
74	8.410146083	-1.410146083
75	8.410146083	2.589853917
76	7.080031709	-1.080031709
77	5.084860149	1.915139851
78	9.075203269	-1.075203269
79	4.419802962	0.580197038
80	3.754745776	1.245254224
81	7.080031709	0.919968291
82	9.075203269	0.924796731
83	5.084860149	1.915139851
84	5.084860149	2.915139851
85	7.080031709	-0.080031709
86	9.740260456	3.259739544

87	1.759574216	-0.759574216
88	4.419802962	0.580197038
89	3.754745776	1.245254224
90	4.419802962	1.580197038
91	6.414974522	-0.414974522
92	9.075203269	1.924796731
93	1.759574216	-0.759574216
94	5.749917336	-0.749917336
95	4.419802962	1.580197038
96	7.080031709	-2.080031709
97	7.080031709	-1.080031709
98	9.740260456	0.259739544
99	2.424631402	-1.424631402
100	4.419802962	-0.419802962
101	4.419802962	0.580197038
102	5.084860149	0.915139851
103	7.080031709	-1.080031709
104	11.07037483	2.929625171
105	2.424631402	-1.424631402
106	4.419802962	0.580197038
107	3.754745776	0.245254224
108	5.749917336	0.250082664
109	5.749917336	0.250082664
110	9.740260456	1.259739544
111	2.424631402	-0.424631402
112	4.419802962	0.580197038
113	5.749917336	1.250082664
114	6.414974522	-0.414974522
115	6.414974522	-0.414974522
116	11.07037483	-1.070374829
117	1.759574216	-0.759574216
118	4.419802962	-0.419802962
119	4.419802962	-0.419802962
120	6.414974522	1.585025478
121	6.414974522	-1.414974522
122	9.075203269	-1.075203269
123	1.759574216	-0.759574216
124	3.754745776	-0.754745776
125	4.419802962	-0.419802962
126	5.749917336	-1.749917336
127	9.740260456	0.259739544
128	1.094517029	-0.094517029
129	3.754745776	-0.754745776
130	5.084860149	-0.084860149
131	6.414974522	-0.414974522
132	5.749917336	-0.749917336
133	8.410146083	1.589853917
134	2.424631402	-1.424631402
135	3.754745776	2.245254224

136	6.414974522	-0.414974522
137	7.080031709	-1.080031709
138	6.414974522	-1.414974522
139	9.740260456	-0.740260456
140	1.759574216	-0.759574216
141	4.419802962	-0.419802962
142	3.754745776	1.245254224
143	8.410146083	-0.410146083
144	7.745088896	-1.745088896
145	6.414974522	1.585025478
146	7.745088896	1.254911104
147	4.419802962	0.580197038
148	4.419802962	1.580197038
149	9.740260456	-0.740260456
150	9.075203269	-1.075203269
151	7.745088896	-0.745088896
152	8.410146083	1.589853917
153	7.745088896	0.254911104
154	5.084860149	-0.084860149
155	5.084860149	-0.084860149
156	9.740260456	-1.740260456
157	8.410146083	-0.410146083
158	8.410146083	-1.410146083
159	9.075203269	1.924796731
160	8.410146083	1.589853917
161	4.419802962	0.580197038
162	3.754745776	1.245254224
163	8.410146083	0.589853917
164	8.410146083	0.589853917
165	8.410146083	-0.410146083
166	9.075203269	-1.075203269
167	7.080031709	-0.080031709
168	7.080031709	1.919968291
169	8.410146083	0.589853917
170	4.419802962	0.580197038
171	4.419802962	0.580197038
172	7.745088896	-1.745088896
173	9.075203269	-0.075203269
174	8.410146083	-0.410146083
175	9.740260456	-0.740260456
176	4.419802962	-0.419802962
177	4.419802962	-0.419802962
178	8.410146083	-1.410146083
179	7.080031709	1.919968291
180	6.414974522	-1.414974522
181	6.414974522	0.585025478
182	9.075203269	-3.075203269
183	3.754745776	0.245254224
184	9.075203269	-3.075203269

185	7.745088896	0.254911104
186	6.414974522	0.585025478

APPENDIX I Residual Plot for Blacksburg Data (Travel Time)



APPENDIX J Residual Table for Norfolk Data (Travel Time)

RESIDUAL OUTPUT

Observation	*Predicted Y*	*Residuals*
1	3.463166386	1.536833614
2	4.091889276	-1.091889276
3	2.834443497	-0.834443497
4	4.091889276	1.908110724
5	6.606780833	-2.606780833
6	7.864226612	-0.864226612
7	3.463166386	-1.463166386
8	5.349335054	0.650664946
9	4.091889276	-1.091889276
10	3.463166386	0.536833614
11	4.720612165	1.279387835
12	2.834443497	-0.834443497
13	7.235503722	1.764496278
14	5.349335054	0.650664946
15	5.978057944	-0.978057944
16	4.091889276	-2.091889276
17	5.349335054	-0.349335054
18	3.463166386	0.536833614
19	2.834443497	0.165556503
20	4.720612165	0.279387835
21	3.463166386	-1.463166386
22	5.978057944	2.021942056
23	7.235503722	0.764496278
24	6.606780833	-1.606780833
25	7.235503722	-2.235503722
26	4.091889276	-0.091889276
27	5.349335054	1.650664946
28	4.720612165	-2.720612165
29	4.720612165	-0.720612165
30	4.720612165	0.279387835
31	4.091889276	-1.091889276
32	5.349335054	0.650664946
33	7.235503722	-1.235503722
34	6.606780833	-2.606780833
35	6.606780833	-1.606780833
36	3.463166386	0.536833614

MAX Rsiduals

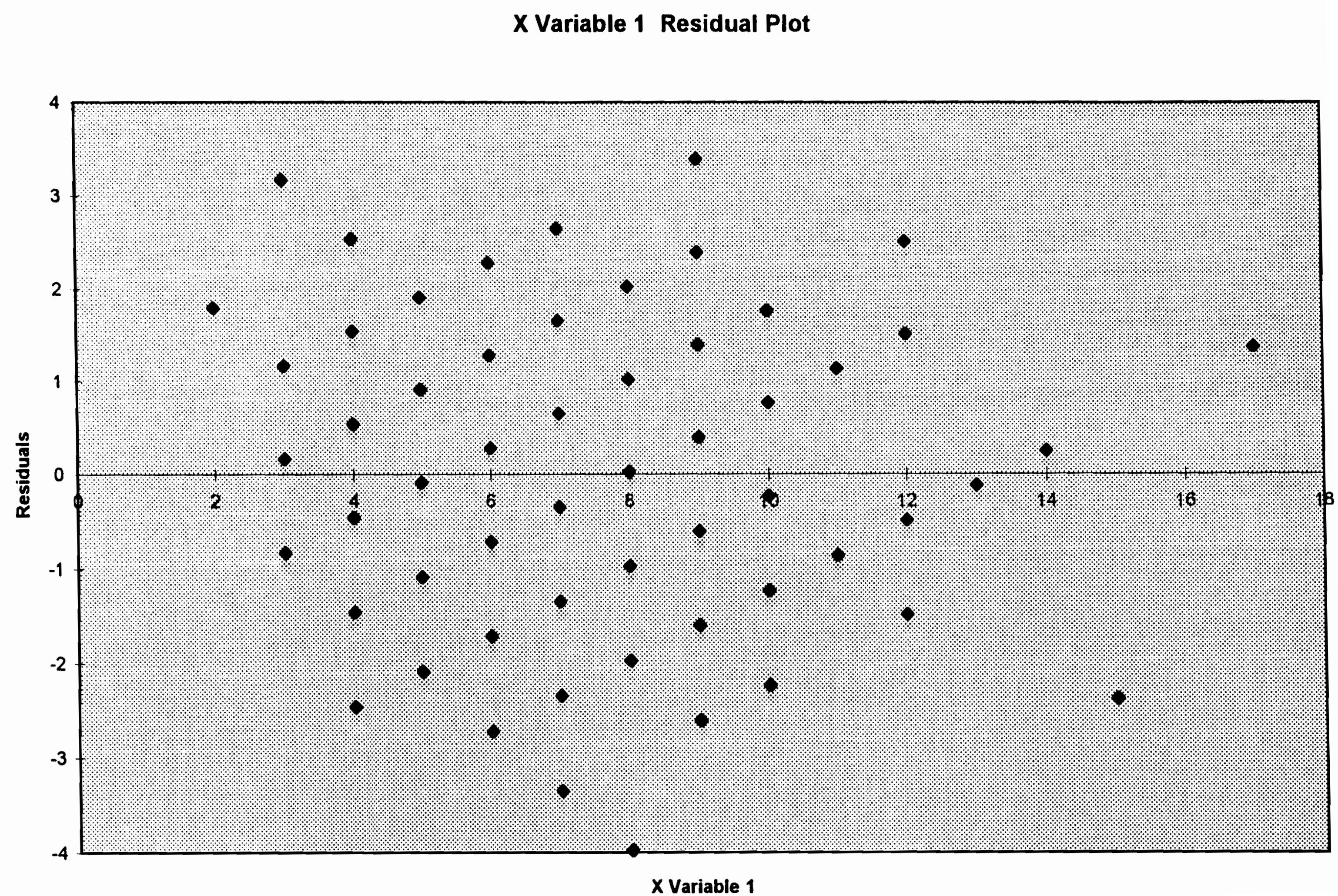
3.3932192

37	5.978057944	-1.978057944
38	2.834443497	3.165556503
39	2.834443497	0.165556503
40	4.720612165	1.279387835
41	2.834443497	1.165556503
42	6.606780833	0.393219167
43	7.235503722	-0.235503722
44	6.606780833	-0.606780833
45	5.978057944	-3.978057944
46	5.349335054	-2.349335054
47	3.463166386	-0.463166386
48	3.463166386	-0.463166386
49	4.720612165	2.279387835
50	2.834443497	0.165556503
51	6.606780833	0.393219167
52	7.864226612	-0.864226612
53	6.606780833	-2.606780833
54	5.978057944	0.021942056
55	3.463166386	-1.463166386
56	5.978057944	0.021942056
57	4.091889276	0.908110724
58	3.463166386	-0.463166386
59	6.606780833	-2.606780833
60	2.834443497	0.165556503
61	6.606780833	0.393219167
62	7.864226612	1.135773388
63	5.978057944	1.021942056
64	5.349335054	1.650664946
65	3.463166386	2.536833614
66	4.091889276	-0.091889276
67	4.091889276	0.908110724
68	6.606780833	0.393219167
69	4.720612165	-1.720612165
70	6.606780833	1.393219167
71	8.492949501	1.507050499
72	6.606780833	-0.606780833
73	10.37911817	-2.379118169
74	3.463166386	-0.463166386
75	4.720612165	2.279387835
76	4.720612165	-0.720612165
77	5.349335054	-1.349335054
78	4.720612165	-1.720612165
79	2.205720608	1.794279392
80	6.606780833	0.393219167
81	6.606780833	1.393219167
82	4.720612165	1.279387835
83	5.978057944	2.021942056
84	3.463166386	-0.463166386
85	4.720612165	0.279387835

86	2.834443497	-0.834443497
87	3.463166386	-0.463166386
88	4.720612165	-0.720612165
89	2.834443497	0.165556503
90	5.349335054	-0.349335054
91	7.235503722	0.764496278
92	5.978057944	-1.978057944
93	7.235503722	-0.235503722
94	3.463166386	-1.463166386
95	4.720612165	1.279387835
96	3.463166386	0.536833614
97	4.091889276	-0.091889276
98	5.349335054	0.650664946
99	2.834443497	0.165556503
100	6.606780833	-1.606780833
101	7.235503722	-1.235503722
102	6.606780833	-1.606780833
103	7.235503722	-0.235503722
104	4.091889276	-1.091889276
105	6.606780833	-0.606780833
106	2.834443497	0.165556503
107	4.720612165	-0.720612165
108	4.091889276	0.908110724
109	2.834443497	0.165556503
110	6.606780833	0.393219167
111	8.492949501	-1.492949501
112	5.349335054	0.650664946
113	5.978057944	1.021942056
114	4.091889276	-1.091889276
115	5.978057944	1.021942056
116	3.463166386	0.536833614
117	4.091889276	-1.091889276
118	5.978057944	-0.978057944
119	3.463166386	-0.463166386
120	5.349335054	2.650664946
121	8.492949501	2.507050499
122	6.606780833	1.393219167
123	5.978057944	0.021942056
124	4.091889276	-0.091889276
125	6.606780833	3.393219167
126	4.091889276	-1.091889276
127	4.091889276	0.908110724
128	5.349335054	1.650664946
129	3.463166386	0.536833614
130	5.349335054	0.650664946
131	9.12167239	-0.12167239
132	6.606780833	0.393219167
133	5.978057944	1.021942056
134	3.463166386	-1.463166386

135	6.606780833	2.393219167
136	4.091889276	-1.091889276
137	3.463166386	0.536833614
138	4.720612165	-0.720612165
139	7.235503722	-0.235503722
140	8.492949501	-0.492949501
141	6.606780833	3.393219167
142	6.606780833	0.393219167
143	3.463166386	-0.463166386
144	5.349335054	1.650664946
145	4.091889276	0.908110724
146	5.349335054	-2.349335054
147	11.63656395	1.363436052
148	3.463166386	-1.463166386
149	7.235503722	0.764496278
150	8.492949501	-1.492949501
151	6.606780833	2.393219167
152	3.463166386	-0.463166386
153	5.978057944	2.021942056
154	3.463166386	1.536833614
155	4.091889276	-0.091889276
156	9.75039528	0.24960472
157	2.205720608	1.794279392
158	8.492949501	-0.492949501
159	6.606780833	1.393219167
160	5.349335054	-3.349335054
161	5.349335054	-1.349335054
162	6.606780833	2.393219167
163	3.463166386	-2.463166386
164	3.463166386	0.536833614
165	5.978057944	-0.978057944
166	4.720612165	-1.720612165
167	6.606780833	1.393219167
168	7.864226612	1.135773388

APPENDIX K Residual Table for Norfolk Data (Travel Time)



APPENDIX L Residual Table for Blacksburg Data (Speed)

RESIDUAL OUTPUT

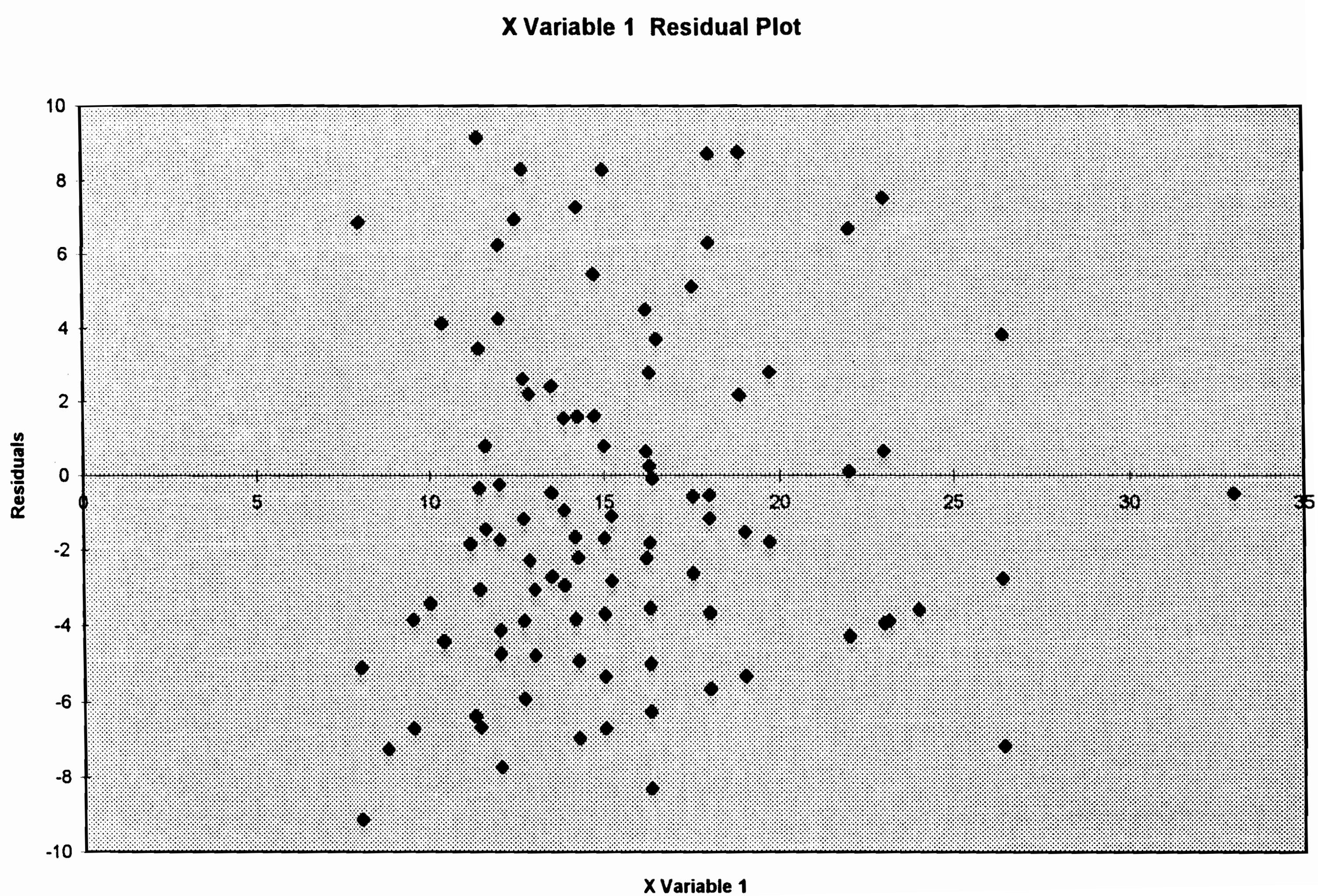
Observation	*Predicted Y*	*Residuals*
1	20.17753	-5.92753
2	19.34777	9.152227
3	22.54825	2.785081
4	19.74081	-7.74081
5	33.49726	-0.49726
6	20.3023	-2.3023
7	21.21472	-6.96472
8	18.66888	4.131117
9	21.21472	1.585281
10	19.74081	4.259186
11	24.23271	8.767285
12	21.70602	0.793979
13	21.21472	1.585281
14	18.66888	4.131117
15	23.36889	-0.56889
16	17.12054	6.879461
17	26.2915	0.108498
18	23.67123	-1.17123
19	22.54825	-6.26254
20	20.17753	8.322473
21	22.54825	2.785081
22	19.74081	4.259186
23	29.17381	-2.77381
24	23.67123	6.328773
25	21.21472	-2.21472
26	19.34777	-0.34777
27	21.83703	-2.83703
28	19.74081	4.259186
29	26.2915	-4.2915
30	23.67123	-1.17123
31	19.34777	9.152227
32	19.34777	-3.06206
33	21.83703	-1.10976
34	17.12054	-9.12054
35	26.2915	6.708498
36	21.70602	8.293979
37	19.34777	-0.34777
38	20.17753	2.622473

39	23.36889	-2.64161
40	17.12054	-5.12054
41	22.68862	3.711376
42	21.70602	8.293979
43	20.17753	2.622473
44	20.17753	2.622473
45	23.36889	-2.64161
46	26.2915	0.108498
47	23.67123	6.328773
48	20.17753	-1.17753
49	22.54825	-3.54825
50	22.54825	-3.54825
51	19.74081	4.259186
52	26.2915	-4.2915
53	20.3023	2.197698
54	19.34777	-0.34777
55	22.54825	0.251748
56	22.54825	-1.82098
57	29.17381	3.826195
58	23.67123	-1.17123
59	19.74081	-1.74081
60	20.72342	2.41944
61	18.66888	-4.41888
62	19.74081	-4.14081
63	26.94657	-3.94657
64	33.49726	-0.49726
65	21.52737	1.615492
66	19.74081	-4.74081
67	22.4921	0.650754
68	22.54825	-6.26254
69	21.17006	-1.67006
70	24.7942	-1.7942
71	26.2915	0.108498
72	20.72342	2.41944
73	21.70602	-5.34238
74	22.4921	4.507897
75	22.54825	-6.26254
76	19.74081	-0.24081
77	26.94657	0.653429
78	29.17381	-2.77381
79	22.4921	-2.2421
80	20.95017	-2.95017
81	27.04015	-3.8973
82	22.54825	-8.29825
83	19.34777	-3.06206
84	22.54825	-5.00979
85	19.74081	4.259186
86	26.2915	0.108498
87	23.67123	-5.67123

88	24.3263	-5.3263
89	20.17753	-1.17753
90	23.36889	-2.64161
91	19.74081	4.259186
92	22.68862	3.711376
93	21.70602	-6.70602
94	19.34777	3.452227
95	19.34777	-0.34777
96	22.54825	0.251748
97	17.12054	6.879461
98	26.2915	6.708498
99	21.70602	-3.70602
100	22.54825	-3.54825
101	19.34777	-0.34777
102	21.21472	-4.929
103	17.12054	6.879461
104	26.2915	0.108498
105	23.67123	-1.17123
106	21.21472	-2.21472
107	21.21472	-2.21472
108	22.54825	-1.82098
109	17.12054	-5.12054
110	26.2915	0.108498
111	19.24951	-6.39237
112	20.17753	-1.17753
113	20.17753	-1.17753
114	21.21472	1.585281
115	19.74081	4.259186
116	26.2915	6.708498
117	21.70602	0.793979
118	20.17753	-5.92753
119	20.17753	2.622473
120	23.36889	5.131112
121	19.74081	4.259186
122	21.70602	0.793979
123	21.21472	7.285281
124	22.54825	0.251748
125	27.60164	-3.60164
126	20.3023	-2.3023
127	20.17753	-1.17753
128	21.21472	1.585281
129	24.3263	-1.5263
130	17.12054	6.879461
131	29.17381	-7.17381
132	18.43068	-3.43068
133	19.34777	-0.34777
134	20.17753	2.622473
135	22.54825	2.785081
136	19.74081	4.259186

137	26.2915	6.708498
138	23.67123	-5.67123
139	21.70602	0.793979
140	21.52737	5.472635
141	20.17753	-5.92753
142	21.17006	-3.83672
143	26.94657	0.653429
144	26.2915	-4.2915
145	19.46007	-1.46007
146	20.95017	1.549828
147	21.52737	1.615492
148	18.10314	-6.70314
149	21.17006	-1.67006
150	24.7942	2.805798
151	24.23271	2.167285
152	19.46007	0.78993
153	21.70602	0.793979
154	20.72342	2.41944
155	17.62444	-7.2608
156	20.39588	-4.79588
157	26.94657	0.653429
158	29.17381	-2.77381
159	20.72342	-2.72342
160	21.70602	-1.70602
161	20.72342	-0.47342
162	20.95017	1.549828
163	22.4921	0.650754
164	19.34777	-6.68111
165	20.39588	-3.06255
166	26.94657	0.653429
167	26.2915	0.108498
168	21.52737	5.472635
169	20.95017	-0.95017
170	18.10314	-3.85314
171	19.17933	-1.84599
172	26.94657	7.553429
173	26.2915	6.708498
174	20.72342	2.41944
175	23.67123	-3.67123
176	23.67123	8.728773
177	20.17753	-3.89181
178	19.74081	6.259186
179	29.17381	3.826195
180	20.04315	6.956846
181	22.5993	-0.0993
182	23.67123	-0.52837

APPENDIX M Residual Plot for Blacksburg Data (Speed)



APPENDIX N ANN Input Data for Blacksburg

bus	*B. Stp*	*Leng.*	*Lns*	*Wea*	*Prk*	*PK*	*Inters.*	*Yld*	*Trn*	*Stp*	*Ld*	*Uld*	*Sp.L*	*Bay*
9	0	1.9	1	0.2	0.1	0	2	0	4	0	0	0	30	0.1
10	5	1.9	1	0.2	0.1	0.2	2	1	4	2	15	0	30	0.1
14	8	3.8	1.8	0.2	0.2	0.2	4	2	5	3	9	10	30	0.6
2	3	0.4	1.5	0.2	0	0.2	0	0	1	1	7	0	25	0.8
4	2	2.2	2	0.2	0	0.2	4	0	2	0	3	0	35	0.8
7	2	1.5	1	0.5	0.5	0.2	0	0	3	2	0	13	15	0.8
8	0	1.9	1	0.5	0.1	0.2	2	0	4	0	0	0	30	0.1
11	6	1.9	1	0	0.1	0.2	2	1	4	2	37	1	30	0.1
16	4	3.8	1.8	0	0.2	0.2	4	2	5	3	6	8	30	0.6
2	1	0.4	1.5	0	0	0.2	0	0	1	1	5	0	25	0.8
7	2	2.2	2	0	0	0.2	4	0	2	0	2	10	35	0.8
6	3	1.5	1	0.2	0.5	0.2	0	0	3	2	1	10	15	0.8
8	0	1.9	1	0	0.1	0.2	2	0	4	0	0	0	30	0.1
11	6	1.9	1	0	0.1	0	2	1	4	2	30	2	30	0.1
13	3	3.8	1.8	0	0.2	0	4	2	5	3	9	6	30	0.6
3	1	0.4	1.5	0	0	0	0	0	1	1	4	1	25	0.8
6	3	2.2	2	0	0	0	4	0	2	0	1	7	35	0.8
5	3	1.5	1	0	0.5	0	0	0	3	2	1	7	15	0.8
7	1	1.9	1	0.2	0.1	0	2	0	4	0	0	1	30	0.1
9	4	1.9	1	0.2	0.1	0	2	1	4	2	21	0	30	0.1
14	7	3.8	1.8	0.5	0.2	0	4	2	5	3	10	1	30	0.6
2	1	0.4	1.5	0.5	0	0	0	0	1	1	1	0	25	0.8
5	1	2.2	2	0.5	0	0	4	0	2	0	0	2	35	0.8
5	3	1.5	1	0.5	0.5	0	0	0	3	2	6	11	15	0.8
8	4	1.9	1	0.5	0.1	0	2	0	4	0	0	7	30	0.1
10	4	1.9	1	0.4	0.1	0	2	1	4	2	9	0	30	0.1
15	8	3.8	1.8	0.4	0.2	0	4	2	5	3	18	23	30	0.6
2	1	0.4	1.5	0.4	0	0	0	0	1	1	3	3	25	0.8
6	1	2.2	2	0.4	0	0	4	0	2	0	1	4	35	0.8
5	1	1.5	1	0.4	0.5	0	0	0	3	2	2	5	15	0.8
10	5	1.9	1	0.4	0.1	0	2	0	4	0	1	14	30	0.1
10	4	1.9	1	0.4	0.1	0	2	1	4	2	11	0	30	0.1
15	5	3.8	1.8	0.4	0.2	0	4	2	5	3	10	16	30	0.6
3	1	0.4	1.5	0.2	0	0	0	0	1	1	2	1	25	0.8
6	1	2.2	2	0.2	0	0	4	0	2	0	0	2	35	0.8
6	3	1.5	1	0.2	0.5	0	0	0	3	2	3	5	15	0.8
10	5	1.9	1	0.2	0.1	0	2	0	4	0	1	18	30	0.1
9	4	1.9	1	0.2	0.1	0	2	1	4	2	5	1	30	0.1
13	6	3.8	1.8	0.2	0.2	0	4	2	5	3	9	18	30	0.6
3	1	0.4	1.5	0.2	0	0	0	0	1	1	0	1	25	0.8
8	3	2.2	2	0.2	0	0	4	0	2	0	2	1	35	0.8

6	2	1.5	1	0.2	0.5	0	0	0	3	2	1	6	15	0.8
9	4	1.9	1	0.2	0.1	0	2	0	4	0	0	17	30	0.1
9	3	1.9	1	0.2	0.1	0	2	1	4	2	5	0	30	0.1
13	4	3.8	1.8	0.2	0.2	0	4	2	5	3	4	10	30	0.6
1	0	0.4	1.5	0.2	0	0.2	0	0	1	1	0	0	25	0.8
6	1	2.2	2	0.2	0	0.2	4	0	2	0	0	2	35	0.8
5	3	1.5	1	0.2	0.5	0.2	0	0	3	2	4	5	15	0.8
9	4	1.9	1	0.2	0.1	0.2	2	0	4	0	1	15	30	0.1
7	3	1.9	1	0.2	0.1	0.2	2	1	4	2	3	0	30	0.1
14	7	3.8	1.8	0.2	0.2	0.2	4	2	5	3	7	15	30	0.6
2	1	0.4	1.5	0.2	0	0.2	0	0	1	1	2	2	25	0.8
6	1	2.2	2	0.2	0	0.2	4	0	2	0	0	1	35	0.8
7	3	1.5	1	0.2	0.5	0.2	0	0	3	2	2	6	15	0.8
10	4	1.9	1	0.2	0.1	0.2	2	0	4	0	2	18	30	0.1
7	2	1.9	1	0.2	0.1	0	2	1	4	2	3	0	30	0.1
14	7	3.8	1.8	0.2	0.2	0	4	2	5	3	8	24	30	0.6
1	0	0.4	1.5	0.2	0	0	0	0	1	1	0	0	25	0.8
5	1	2.2	2	0.2	0	0	4	0	2	0	0	1	35	0.8
5	1	1.5	1	0.2	0.5	0	0	0	3	2	1	5	15	0.8
15	9	3	2	0.4	0.2	0	4	2	9	4	10	17	30	0.7
12	9	2.7	1.5	0.4	0.2	0	5	0	5	1	15	14	35	0.7
11	4	1.9	1	0.4	0.5	0	0	2	4	2	6	14	15	0.8
13	8	2.6	1	0.2	0.1	0	5	1	4	3	14	17	25	0.9
6	2	2.3	2	0.2	0	0	2	3	8	2	0	2	40	0.9
4	0	2.2	2	0.2	0	0	3	2	4	0	0	0	45	0.9
11	5	2.7	1.5	0.2	0.1	0	5	1	4	3	8	3	25	0.8
15	9	3	2	0.2	0.2	0	4	2	9	4	17	18	30	0.7
10	3	2.7	1.5	0.2	0.2	0	5	0	5	1	5	8	35	0.7
7	2	1.9	1	0.2	0.5	0	0	2	4	2	3	4	15	0.8
11	5	2.6	1	0.2	0.1	0	5	1	4	3	3	17	25	0.9
7	1	2.3	2	0.2	0	0	2	3	8	2	0	1	40	0.9
6	1	2.2	2	0.2	0	0	3	2	4	0	1	0	45	0.9
12	6	2.7	1.5	0.2	0.1	0	5	1	4	3	4	3	25	0.8
12	7	3	2	0.2	0.2	0	4	2	9	4	2	13	30	0.7
10	5	2.7	1.5	0.2	0.2	0.2	5	0	5	1	4	3	35	0.7
7	2	1.9	1	0.2	0.5	0.2	0	2	4	2	3	1	15	0.8
13	8	2.6	1	0.2	0.1	0.2	5	1	4	3	1	19	25	0.9
6	0	2.3	2	0.2	0	0.2	2	3	8	2	0	0	40	0.9
5	1	2.2	2	0.2	0	0.2	3	2	4	0	1	0	45	0.9
10	6	2.7	1.5	0.2	0.1	0	5	1	4	3	6	3	25	0.8
13	7	3	2	0.2	0.2	0.2	4	2	9	4	4	7	30	0.7
7	4	2.7	1.5	0.2	0.2	0	5	0	5	1	4	3	35	0.7
7	0	1.9	1	0	0.1	0	2	0	4	0	0	0	30	0.1
10	7	1.9	1	0	0.1	0.2	2	1	4	2	37	1	30	0.1
14	7	3.8	1.8	0	0.2	0.2	4	2	5	3	12	11	30	0.6
2	0	0.4	1.5	0	0	0.2	0	0	1	1	0	0	25	0.8
6	3	2.2	2	0	0	0.2	4	0	2	0	2	8	35	0.8
5	2	1.5	1	0	0.5	0.2	0	0	3	2	0	8	15	0.8
6	0	1.9	1	0	0.1	0.2	2	0	4	0	0	0	30	0.1

9	4	1.9	1	0	0.1	0.2	2	1	4	2	11	1	30	0.1
13	8	3.8	1.8	0	0.2	0.2	4	2	5	3	11	6	30	0.6
2	1	0.4	1.5	0	0	0.2	0	0	1	1	2	0	25	0.8
8	1	2.2	2	0	0	0.2	4	0	2	0	0	6	35	0.8
6	2	1.5	1	0	0.5	0.2	0	0	3	2	1	3	15	0.8
10	4	1.9	1	0	0.1	0.2	2	0	4	0	1	3	30	0.1
10	7	1.9	1	0	0.1	0	2	1	4	2	7	3	30	0.1
14	7	3.8	1.8	0	0.2	0	4	2	5	3	10	13	30	0.6
3	1	0.4	1.5	0	0	0	0	0	1	1	2	1	25	0.8
6	2	2.2	2	0	0	0	4	0	2	0	2	8	35	0.8
6	1	1.5	1	0	0.5	0	0	0	3	2	1	1	15	0.8
7	1	1.9	1	0	0.1	0	2	0	4	0	0	1	30	0.1
10	4	1.9	1	0	0.1	0	2	1	4	2	23	0	30	0.1
16	9	3.8	1.8	0	0.2	0	4	2	5	3	22	24	30	0.6
3	1	0.4	1.5	0	0	0	0	0	1	1	2	1	25	0.8
6	1	2.2	2	0	0	0	4	0	2	0	0	5	35	0.8
5	3	1.5	1	0	0.5	0	0	0	3	2	0	6	15	0.8
8	3	1.9	1	0	0.1	0	2	0	4	0	0	1	30	0.1
8	4	1.9	1	0	0.1	0	2	1	4	2	3	2	30	0.1
14	6	3.8	1.8	0	0.2	0	4	2	5	3	23	14	30	0.6
3	1	0.4	1.5	0	0	0	0	0	1	1	4	1	25	0.8
6	2	2.2	2	0	0	0	4	0	2	0	1	5	35	0.8
8	3	1.5	1	0	0.5	0	0	0	3	2	0	7	15	0.8
9	7	1.9	1	0	0.1	0	2	0	4	0	3	5	30	0.1
9	6	1.9	1	0	0.1	0	2	1	4	2	10	1	30	0.1
16	8	3.8	1.8	0	0.2	0	4	2	5	3	11	15	30	0.6
2	1	0.4	1.5	0	0	0	0	0	1	1	0	2	25	0.8
6	0	2.2	2	0	0	0	4	0	2	0	0	0	35	0.8
6	3	1.5	1	0	0.5	0	0	0	3	2	3	2	15	0.8
9	4	1.9	1	0	0.1	0	2	0	4	0	0	8	30	0.1
9	3	1.9	1	0	0.1	0	2	1	4	2	2	3	30	0.1
13	6	3.8	1.8	0	0.2	0	4	2	5	3	5	8	30	0.6
2	1	0.4	1.5	0	0	0	0	0	1	1	0	1	25	0.8
5	0	2.2	2	0	0	0	4	0	2	0	0	0	35	0.8
6	3	1.5	1	0	0.5	0	0	0	3	2	2	8	15	0.8
8	5	1.9	1	0	0.1	0	2	0	4	0	0	8	30	0.1
14	7	3.8	1.8	0	0.2	0	4	2	5	3	3	13	30	0.6
1	1	0.4	1.5	0	0	0.2	0	0	1	1	0	1	25	0.8
5	2	2.2	2	0	0	0.2	4	0	2	0	2	1	35	0.8
7	4	1.5	1	0	0.5	0.2	0	0	3	2	2	4	15	0.8
9	5	1.9	1	0	0.1	0.2	2	0	4	0	0	11	30	0.1
8	3	1.9	1	0	0.1	0.2	2	1	4	2	3	0	30	0.1
12	6	3.8	1.8	0	0.2	0.2	4	2	5	3	6	7	30	0.6
3	1	0.4	1.5	0	0	0.2	0	0	1	1	1	7	25	0.8
5	0	2.2	2	0	0	0.2	4	0	2	0	0	0	35	0.8
9	2	1.5	1	0	0.5	0.2	0	0	3	2	3	1	15	0.8
10	7	1.9	1	0	0.1	0.2	2	0	4	0	1	14	30	0.1
9	1	1.9	1	0	0.1	0	2	1	4	2	0	1	30	0.1
14	7	3.8	1.8	0	0.2	0	4	2	5	3	19	17	30	0.6

2	1	0.4	1.5	0	0	0	0	0	1	1	2	8	25	0.8
6	4	2.2	2	0	0	0	4	0	2	0	2	4	35	0.8
5	2	1.5	1	0	0.5	0	0	0	3	2	1	11	15	0.8
12	5	3	2	0	0.2	0	4	2	9	4	4	3	30	0.7
11	4	2.7	1.5	0	0.2	0.2	5	0	5	1	5	0	35	0.7
9	3	1.9	1	0	0.5	0.2	0	2	4	2	3	9	15	0.8
11	4	2.6	1	0	0.1	0.2	5	1	4	3	3	4	25	0.9
6	1	2.3	2	0	0	0.2	2	3	8	2	0	1	40	0.9
6	1	2.2	2	0	0	0.2	3	2	4	0	1	0	45	0.9
14	12	2.7	1.5	0	0.1	0.2	5	1	4	3	21	3	25	0.8
13	5	3	2	0	0.2	0.2	4	2	9	4	5	3	30	0.7
11	2	2.7	1.5	0	0.2	0.2	5	0	5	1	2	2	35	0.7
12	4	1.9	1	0	0.5	0.2	0	2	4	2	3	14	15	0.8
11	5	2.6	1	0	0.1	0	5	1	4	3	1	5	25	0.9
7	1	2.3	2	0	0	0	2	3	8	2	0	2	40	0.9
7	3	2.2	2	0	0	0	3	2	4	0	3	0	45	0.9
14	11	2.7	1.5	0	0.1	0	5	1	4	3	15	6	25	0.8
12	8	3	2	0	0.2	0	4	2	9	4	5	3	30	0.7
12	5	2.7	1.5	0	0.2	0	5	0	5	1	7	4	35	0.7
13	4	1.9	1	0	0.5	0	0	2	4	2	9	11	15	0.8
12	4	2.6	1	0	0.1	0	5	1	4	3	2	6	25	0.9
6	0	2.3	2	0	0	0	2	3	8	2	0	0	40	0.9
5	0	2.2	2	0	0	0	3	2	4	0	0	0	45	0.9
12	7	2.7	1.5	0	0.1	0	5	1	4	3	5	4	25	0.8
12	6	3	2	0	0.2	0	4	2	9	4	6	9	30	0.7
12	6	2.7	1.5	0	0.2	0	5	0	5	1	13	4	35	0.7
13	7	3	2	0	0.2	0	4	2	9	4	3	7	30	0.8
10	4	2.7	1.5	0	0.2	0	5	0	5	1	12	0	35	0.9
10	3	1.9	1	0	0.5	0	0	2	4	2	1	5	15	0.9
12	9	2.6	1	0	0.1	0	5	1	4	3	3	14	25	0.9
6	0	2.3	2	0	0	0	2	3	8	2	0	0	40	0.8
6	0	2.2	2	0	0	0	3	2	4	0	0	0	45	0.7
11	7	2.7	1.5	0	0.1	0	5	1	4	3	6	3	25	0.7
13	9	3	2	0	0.2	0	4	2	9	4	8	17	30	0.8
12	3	1.9	1	0	0.5	0	0	2	4	2	3	2	15	0.9
14	11	2.6	1	0	0.1	0	5	1	4	3	4	20	25	0.9
6	1	2.3	2	0	0	0	2	3	8	2	0	1	40	0.8
6	1	2.2	2	0	0	0	3	2	4	0	0	1	45	0.7
12	5	2.7	1.5	0	0.1	0	5	1	4	3	4	1	25	0.7
10	2	3	2	0	0.2	0	4	2	9	4	0	5	30	0.8
9	1	2.7	1.5	0	0.2	0.2	5	0	5	1	1	1	35	0.9
9	3	1.9	1	0	0.5	0.2	0	2	4	2	7	3	15	0.9
13	11	2.6	1	0	0.1	0.2	5	1	4	3	9	31	25	0.9
5	0	2.2	2	0	0	0.2	3	2	4	0	0	0	45	0.7
13	7	2.7	1.5	0	0.1	0	5	1	4	3	1	3	25	0.7
11	8	3	2	0	0.2	0	4	2	9	4	9	11	30	0.8
9	5	2.7	1.5	0	0.2	0	5	0	5	1	4	5	35	0.9

APPENDIX O ANN Input Data for Norfolk

TT(B)	B. Stp	Ld	Uld	Leng.	Lns	Wea	Prk	PK	Inters.	Yld	Trn	Stp	Sp.L	Inci.	T.Bay
4	0	0	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
5	0	0	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
3	1	3	0	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
5	4	0	2	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
9	4	4	1	2.20	2.00	0.00	0.10	0.00	9.00	1	5.00	1	30	0.00	0.80
11	9	6	2	2.40	2.50	0.00	0.00	0.50	3.00	1	4.00	0	40	0.00	0.90
4	3	2	2	1.50	1.20	0.00	0.80	0.50	2.00	2	1.50	2	35	0.00	0.40
7	5	0	5	1.80	2.50	0.00	0.00	0.40	5.00	1	2.00	0	40	0.00	0.80
5	1	0	1	0.90	2.00	0.00	0.00	0.20	2.00	1	3.00	7	18	0.00	1.00
4	0	0	0	0.80	2.00	0.00	0.00	0.20	1.00	0	2.00	6	20	0.00	1.00
6	0	0	0	1.90	2.50	0.00	0.00	0.40	5.00	2	2.00	2	45	0.00	0.80
3	1	1	0	1.40	1.20	0.00	0.80	0.40	2.00	1	2.00	0	33	0.00	0.40
10	5	6	1	2.50	2.50	0.00	0.00	0.60	5.00	3	4.00	0	40	0.00	0.90
7	4	3	1	1.90	2.00	0.00	0.10	0.20	8.00	2	3.00	2	30	0.00	0.80
8	8	1	5	2.40	2.50	0.00	0.00	0.10	3.00	1	4.00	0	40	0.00	0.90
5	2	2	1	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
7	3	0	3	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
4	1	0	1	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
3	0	0	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
6	1	0	1	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
4	1	1	1	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
8	3	0	3	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
10	5	4	1	2.20	2.00	0.00	0.10	0.20	9.00	1	5.00	1	30	0.00	0.80
9	5	3	2	1.90	2.00	0.00	0.10	0.00	8.00	2	3.00	2	30	0.00	0.80
10	6	8	4	2.40	2.50	0.00	0.00	0.20	3.00	1	4.00	0	40	0.00	0.90
5	1	2	2	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
7	1	0	0	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
6	1	0	1	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
6	1	0	1	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
6	1	2	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
5	2	2	1	1.40	1.20	0.00	0.80	0.20	2.00	1	2.00	0	33	0.00	0.40
7	3	3	3	2.50	2.50	0.00	0.00	0.30	5.00	3	4.00	0	40	0.00	0.90
10	4	5	4	2.20	2.00	0.00	0.10	0.40	9.00	1	5.00	1	30	0.00	0.80
9	5	3	2	1.90	2.00	0.00	0.10	0.40	8.00	2	3.00	2	30	0.00	0.80
9	4	3	2	2.40	2.50	0.00	0.00	0.10	3.00	1	4.00	0	40	0.00	0.90
4	2	1	1	1.50	1.20	0.00	0.80	0.05	2.00	2	1.50	2	35	0.00	0.40
8	1	0	1	1.80	2.50	0.00	0.00	0.30	5.00	1	2.00	0	40	0.00	0.80
3	1	1	1	0.90	2.00	0.00	0.00	0.20	2.00	1	3.00	7	18	0.00	1.00
3	1	0	1	0.80	2.00	0.00	0.00	0.20	1.00	0	2.00	6	20	0.00	1.00
6	1	2	1	1.90	2.50	0.00	0.00	0.20	5.00	2	2.00	2	45	0.00	0.80
3	2	0	1	1.40	1.20	0.00	0.80	0.20	2.00	1	2.00	0	33	0.00	0.40

9	5	4	4	2.50	2.50	0.00	0.00	0.20	5.00	3	4.00	0	40	0.00	0.90
10	5	6	6	2.20	2.00	0.00	0.10	0.40	9.00	1	5.00	1	30	0.00	0.80
9	3	0	4	1.90	2.00	0.00	0.10	0.20	8.00	2	3.00	2	30	0.00	0.80
8	6	5	3	2.40	2.50	0.00	0.00	0.10	3.00	1	4.00	0	40	0.00	0.90
7	3	1	1	1.50	1.20	0.00	0.80	0.10	2.00	2	1.50	2	35	0.00	0.40
4	0	0	0	0.90	2.00	0.00	0.00	0.10	2.00	1	3.00	7	18	0.00	1.00
4	1	1	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
6	0	0	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
3	2	1	1	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
9	6	4	4	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
11	4	7	9	2.20	2.00	0.00	0.10	0.50	9.00	1	5.00	1	30	0.00	0.80
9	6	6	12	1.90	2.00	0.00	0.10	0.00	8.00	2	3.00	2	30	0.00	0.80
8	6	3	6	2.40	2.50	0.00	0.00	0.00	3.00	1	4.00	0	40	0.00	0.90
4	2	2	2	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
8	2	1	1	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
5	1	1	1	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
4	1	2	2	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
9	2	3	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
3	5	1	1	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
9	1	5	2	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
11	6	8	8	2.20	2.00	0.00	0.10	0.50	9.00	1	5.00	1	30	0.00	0.80
8	5	3	3	1.90	2.00	0.00	0.10	0.20	8.00	2	3.00	2	30	0.00	0.80
7	3	1	3	2.40	2.50	0.00	0.00	0.30	3.00	1	4.00	0	40	0.00	0.90
4	2	0	2	1.80	2.50	0.00	0.00	0.80	5.00	1	2.00	0	40	0.00	0.80
5	3	4	3	0.90	2.00	0.00	0.00	0.40	2.00	1	3.00	7	18	0.00	1.00
5	3	2	2	0.80	2.00	0.00	0.00	0.40	1.00	0	2.00	6	20	0.00	1.00
9	1	2	0	1.90	2.50	0.00	0.00	0.80	5.00	2	2.00	2	45	0.00	0.80
6	3	4	3	1.40	1.20	0.00	0.80	0.40	2.00	1	2.00	0	33	0.00	0.40
9	9	7	8	2.50	2.50	0.00	0.00	0.40	5.00	3	4.00	0	40	0.00	0.90
12	5	1	8	2.20	2.00	0.00	0.10	0.55	9.00	1	5.00	1	30	0.00	0.80
9	8	2	8	1.90	2.00	0.00	0.10	0.40	8.00	2	3.00	2	30	0.00	0.80
15	5	0	4	2.40	2.50	0.00	0.00	0.20	3.00	1	4.00	0	40	0.40	0.90
4	2	0	4	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
6	2	0	3	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
6	3	7	6	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
7	1	2	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
6	3	2	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
2	1	0	1	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
9	3	0	8	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
9	3	1	1	2.20	2.00	0.00	0.10	0.00	9.00	1	5.00	1	30	0.00	0.80
6	2	1	1	1.90	2.00	0.00	0.10	0.00	8.00	2	3.00	2	30	0.00	0.80
8	3	3	1	2.40	2.50	0.00	0.00	0.00	3.00	1	4.00	0	40	0.00	0.90
4	0	1	1	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
6	0	0	0	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
3	1	0	0	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
4	0	0	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
6	1	1	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
3	0	0	0	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
7	3	4	0	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90

10	4	7	3	2.20	2.00	0.00	0.10	0.40	9.00	1	5.00	1	30	0.00	0.80
8	4	3	1	1.90	2.00	0.00	0.10	0.40	8.00	2	3.00	2	30	0.00	0.80
10	7	6	5	2.40	2.50	0.00	0.00	0.50	3.00	1	4.00	0	40	0.00	0.90
4	2	1	1	1.50	1.20	0.00	0.80	0.50	2.00	2	1.50	2	35	0.00	0.40
6	5	0	6	1.80	2.50	0.00	0.00	0.40	5.00	1	2.00	0	40	0.00	0.80
4	1	0	1	0.90	2.00	0.00	0.00	0.20	2.00	1	3.00	7	18	0.00	1.00
5	0	0	0	0.80	2.00	0.00	0.00	0.20	1.00	0	2.00	6	20	0.00	1.00
7	1	0	1	1.90	2.50	0.00	0.00	0.40	5.00	2	2.00	2	45	0.00	0.80
3	2	1	0	1.40	1.20	0.00	0.80	0.40	2.00	1	2.00	0	33	0.00	0.40
9	5	5	0	2.50	2.50	0.00	0.00	0.40	5.00	3	4.00	0	40	0.00	0.90
10	6	10	3	2.20	2.00	0.00	0.10	0.40	9.00	1	5.00	1	30	0.00	0.80
9	6	6	7	1.90	2.00	0.00	0.10	0.20	8.00	2	3.00	2	30	0.00	0.80
10	5	10	6	2.40	2.50	0.00	0.00	0.20	3.00	1	4.00	0	40	0.00	0.90
5	3	2	3	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
9	4	4	3	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
3	3	0	4	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
6	1	0	1	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
5	0	0	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
3	3	0	3	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
9	3	2	4	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
12	7	6	10	2.20	2.00	0.00	0.10	0.60	9.00	1	5.00	1	30	0.20	0.80
7	5	3	7	1.90	2.00	0.00	0.10	0.00	8.00	2	3.00	2	30	0.00	0.80
8	3	3	3	2.40	2.50	0.00	0.00	0.00	3.00	1	4.00	0	40	0.00	0.90
5	3	2	2	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
8	0	0	0	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
4	2	0	2	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
5	2	5	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
8	1	1	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
4	4	0	4	1.40	1.20	0.00	0.80	0.20	2.00	1	2.00	0	33	0.00	0.40
7	1	0	1	2.50	2.50	0.00	0.00	0.30	5.00	3	4.00	0	40	0.00	0.90
12	6	8	3	2.20	2.00	0.00	0.10	0.60	9.00	1	5.00	1	30	0.00	0.80
9	6	4	4	1.90	2.00	0.00	0.10	0.40	8.00	2	3.00	2	30	0.00	0.80
8	2	6	1	2.40	2.50	0.00	0.00	0.30	3.00	1	4.00	0	40	0.00	0.90
5	3	0	3	1.50	1.20	0.00	0.80	0.10	2.00	2	1.50	2	35	0.00	0.40
9	1	0	1	1.80	2.50	0.00	0.00	0.30	5.00	1	2.00	0	40	0.00	0.80
5	1	0	2	0.90	2.00	0.00	0.00	0.20	2.00	1	3.00	7	18	0.00	1.00
5	3	4	0	0.80	2.00	0.00	0.00	0.20	1.00	0	2.00	6	20	0.00	1.00
7	1	0	1	1.90	2.50	0.00	0.00	0.20	5.00	2	2.00	2	45	0.00	0.80
4	3	2	1	1.40	1.20	0.00	0.80	0.20	2.00	1	2.00	0	33	0.00	0.40
7	5	7	5	2.50	2.50	0.00	0.00	0.20	5.00	3	4.00	0	40	0.00	0.90
13	6	6	11	2.20	2.00	0.00	0.10	0.70	9.00	1	5.00	1	30	0.20	0.80
9	3	6	5	1.90	2.00	0.00	0.10	0.20	8.00	2	3.00	2	30	0.00	0.80
8	5	0	5	2.40	2.50	0.00	0.00	0.10	3.00	1	4.00	0	40	0.00	0.90
4	4	4	0	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
9	2	1	1	1.80	2.50	0.00	0.00	0.10	5.00	1	2.00	0	40	0.00	0.80
5	1	0	1	0.90	2.00	0.00	0.00	0.10	2.00	1	3.00	7	18	0.00	1.00
4	1	1	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
6	5	1	4	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
10	5	1	6	2.50	2.50	0.00	0.00	0.60	5.00	3	4.00	0	40	0.00	0.90

12	7	5	8	2.20	2.00	0.00	0.10	0.60	9.00	1	5.00	1	30	0.00	0.80
9	4	1	5	1.90	2.00	0.00	0.10	0.00	8.00	2	3.00	2	30	0.00	0.80
9	5	0	8	2.40	2.50	0.00	0.00	0.00	3.00	1	4.00	0	40	0.00	0.90
4	2	0	2	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
7	0	0	0	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
5	2	1	1	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
7	1	6	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
17	1	1	0	1.90	2.50	0.00	0.00	0.90	5.00	2	2.00	2	45	0.00	0.80
4	3	1	2	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
10	4	1	4	2.50	2.50	0.00	0.00	0.60	5.00	3	4.00	0	40	0.00	0.90
12	8	12	15	2.20	2.00	0.00	0.10	0.80	9.00	1	5.00	1	30	0.00	0.80
9	5	2	6	2.40	2.50	0.00	0.00	0.40	3.00	1	4.00	0	40	0.00	0.90
4	2	1	1	1.50	1.20	0.00	0.80	0.00	2.00	2	1.50	2	35	0.00	0.40
8	2	2	0	1.80	2.50	0.00	0.00	0.50	5.00	1	2.00	0	40	0.00	0.80
4	2	7	6	0.90	2.00	0.00	0.00	0.40	2.00	1	3.00	7	18	0.00	1.00
5	2	0	0	0.80	2.00	0.00	0.00	0.40	1.00	0	2.00	6	20	0.00	1.00
14	1	1	0	1.90	2.50	0.00	0.00	0.80	5.00	2	2.00	2	45	0.00	0.80
2	4	3	1	1.40	1.20	0.00	0.80	0.40	2.00	1	2.00	0	33	0.00	0.40
12	8	11	19	2.50	2.50	0.00	0.00	0.80	5.00	3	4.00	0	40	0.00	0.90
9	4	5	10	2.20	2.00	0.00	0.10	0.40	9.00	1	5.00	1	30	0.00	0.80
7	3	5	0	2.40	2.50	0.00	0.00	0.20	3.00	1	4.00	0	40	0.00	0.90
7	4	3	1	1.50	1.20	0.00	0.80	0.20	2.00	2	1.50	2	35	0.00	0.40
9	1	1	0	1.80	2.50	0.00	0.00	0.00	5.00	1	2.00	0	40	0.00	0.80
4	1	0	1	0.90	2.00	0.00	0.00	0.00	2.00	1	3.00	7	18	0.00	1.00
4	1	5	0	0.80	2.00	0.00	0.00	0.00	1.00	0	2.00	6	20	0.00	1.00
8	3	5	0	1.90	2.50	0.00	0.00	0.00	5.00	2	2.00	2	45	0.00	0.80
6	2	1	1	1.40	1.20	0.00	0.80	0.00	2.00	1	2.00	0	33	0.00	0.40
9	2	2	0	2.50	2.50	0.00	0.00	0.00	5.00	3	4.00	0	40	0.00	0.90
11	5	3	4	2.20	2.00	0.00	0.10	0.30	9.00	1	5.00	1	30	0.00	0.80

VITA

Sanghoon Bae was born in Pusan, Korea, on August 27, 1964. He received his B.S. degree in 1990 in civil engineering department from Pusan National University. In the summer of 1990, he came to United States for graduate study. He enrolled the department of civil engineering (transportation division), University of Florida at Gainsville. After a year of study, he transferred to the Virginia Polytechnic Institute and State University (VPI&SU). In June 1993, he completed his M.S. degree, specialized in Advanced Public Transportation System, in civil engineering (transportation division) from VPI&SU.

Sanghoon Bae is versed in Parameter Adaptation Algorithm, Artificial Neural Networks, transit management information system, Automatic Vehicle Location system, and computer applications in transportation. He is also interested in Intelligent Transportation Systems-related research areas, specifically in Advance Public Transportation system and Advance Transportation Management System.