SPEED ESTIMATION USING SINGLE LOOP DETECTOR

OUTPUTS

A Dissertation

by

ZHIRUI YE

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

DOCTOR OF PHILOSOPHY

December 2007

Major Subject: Civil Engineering

SPEED ESTIMATION USING SINGLE LOOP DETECTOR OUTPUTS

A Dissertation

by

ZHIRUI YE

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

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yunlong Zhang
Committee Members,	Dominique Lord
	Luca Quadrifoglio
	Faming Liang
Head of Department,	David Rosowsky

December 2007

Major Subject: Civil Engineering

ABSTRACT

Speed Estimation Using Single Loop Detector Outputs. (December 2007)

Zhirui Ye, B.S., Southeast University, China;

M.S., Southeast University, China

Chair of Advisory Committee: Dr. Yunlong Zhang

Flow speed describes general traffic operation conditions on a segment of roadway. It is also used to diagnose special conditions such as congestion and incidents. Accurate speed estimation plays a critical role in traffic management or traveler information systems. Data from loop detectors have been primary sources for traffic information, and single loop are the predominant loop detector type in many places. However, single loop detectors do not produce speed output. Therefore, speed estimation using single loop outputs has been an important issue for decades.

This dissertation research presents two methodologies for speed estimation using single loop outputs. Based on findings from past studies and examinations in this research, it is verified that speed estimation is a nonlinear system under various traffic conditions. Thus, a methodology of using Unscented Kalman Filter (UKF) is first proposed for such a system. The UKF is a parametric filtering technique that is suitable for nonlinear problems. Through an Unscented Transformation (UT), the UKF is able to

capture the posterior mean and covariance of a Gaussian random variable accurately for a nonlinear system without linearization.

This research further shows that speed estimation is a nonlinear non-Gaussian system. However, Kalman filters including the UKF are established based on the Gaussian assumption. Thus, another nonlinear filtering technique for non-Gaussian systems, the Particle Filter (PF), is introduced. By combining the strengths of both the PF and the UKF, the second speed estimation methodology—Unscented Particle Filter (UPF) is proposed for speed estimation. The use of the UPF avoids the limitations of the UKF and the PF.

Detector data are collected from multiple freeway locations and the microscopic traffic simulation program CORSIM. The developed methods are applied to the collected data for speed estimation. The results show that both proposed methods have high accuracies of speed estimation. Between the UKF and the UPF, the UPF has better performance but has higher computation cost.

The improvement of speed estimation will benefit real-time traffic operations by improving the performance of applications such as travel time estimation using a series of single loops in the network, incident detection, and large truck volume estimation. Therefore, the work enables traffic analysts to use single loop outputs in a more cost-effective way.

DEDICATION

To my son

To my family and friends for their love and support…

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my advisor, Dr. Yunlong Zhang, for his guidance and support during my study at Mississippi State University and Texas A&M University. Dr. Zhang treats me not only as his student, but also as a younger brother. It has been an honor to work with him and benefit from his religious and passionate attitude toward research, teaching, and life.

I would also like to thank my committee members, Dr. Dominique Lord, Dr. Luca Quadrifoglio, and Dr. Faming Liang for their guidance, and for taking time to work with me and to provide technical expertise to achieve my objectives. Their valuable comments and suggestions to my initial work directly contribute to the completion of my dissertation.

Special thanks to Dr. Danny R. Middleton at Texas Transportation Institute for providing traffic data for the dissertation. Thanks also to my fellow students in the transportation group with whom I shared a unique research environment and student life.

Finally, I would like to thank my family for the support they provided throughout my studies, regardless of the remote distance. Special thanks to my wife, Ziwei Hu, for her endless support, understanding, and love, without which I would not have achieved this milestone.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Speed is one of the most commonly used measures of performance for traffic facilities and networks (McShane et al., 1998). As an indicator of Level of Service (LOS), speed has been used in traffic operational analysis, traffic simulation models, incident detection and analysis, economic studies, and many other areas of transportation engineering and planning. Moreover, some important decision-making variables such as travel time can be further calculated based on the speed information. Speed information is also important for real-time transportation applications. These applications include Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS), which are part of the Intelligent Transportation Systems (ITS). Therefore, providing timely and accurate speed information is very important for improving traffic management and control.

The importance of speed indicates a need to measure speed timely, accurately, and cost effectively. Speed data can be collected manually or automatically, while the manual method is less practical and efficient than the automatic method when a large amount of speed information of a network is needed. Extensive and continuous real-time traffic data are required in modern traffic management and control. Manual speed measurement apparently cannot meet such requirements. A variety of vehicle detectors have been employed on highways to automatically provide real time traffic data. Based

_____

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

on the types of vehicle detectors, speed measurement techniques can be divided into two broad categories, direct methods and indirect methods. Many technologies have been used to detect vehicle speeds, such as ultrasonic, radar, acoustic, piezoelectric, passive and active infrared, magnetic, pair inductance loops, and Video Image Processor (VIP). Detectors using such technologies can directly measure and output speed data. In the case of indirect methods, speed can be obtained via postprocessing. For example, speed can be estimated by using the outputs (occupancy and traffic count) from single loop detectors.

Although many types of vehicle detectors have accurate speed measurements, they are much more expensive than single loop detectors. This prevents those detectors from widespread implementations. Even though dual-loop detectors have the output of speed, the cost of upgrading from a single loop detector to a dual-loop detector is still high, around $750 direct cost for loop placement and $2500-$5000 indirect cost by lane closure (Wang and Nihan, 2003).

Single loop detectors, however, are the most widely used detectors on the America's highways because of the maturity of the inductance technique and low cost. For example, the California Department of Transportation (DOT) estimated that there are approximately 300,000 single loop detectors on California freeways (PATH, 1997). The extensive deployment of single loops is able to provide tremendous amount of baseline data. The utilization of such baseline data is apparently important for managing and controlling traffic in a cost-effective manner.

1.1   STATEMENT OF THE PROBLEM

Given the widespread implementation of single loop detectors and the importance of speed in numerous transportation applications, there is a need to explore and develop methodologies to estimate speed accurately using single loop outputs. Even though many methods have been presented in the literature for speed estimation, the accuracy of the estimation is unsatisfactory. This is caused by several issues regarding this subject. Firstly, traffic flow is a mixture of various classes of vehicles. Traffic compositions vary spatially (from location to location) and temporally (from time to time). Also, different classes of vehicles have different characteristics such as vehicle length, weight, and number of axles. Secondly, traffic conditions on freeways are complex. With the increase of traffic volume, traffic congestion arises and queue forms on freeways during peak hours or even for significant portions of the day, especially within large urban areas. Vehicles don't have the same speed on a freeway section and speeds can be influenced by many factors such as roadway characteristics, traffic volume, incidents, weather, and driver characteristics. Thirdly, assumptions used for simplifying traffic analysis do not meet real traffic conditions and contribute to analytical errors. Finally, some existing methods are developed for limited conditions and have their own drawbacks in dealing with this problem.

The important role of speed requires that proposed methods should be able to generate accurate estimates of speed. The developed methods should have good performance under various traffic conditions. Moreover, they should be easy for

implementation and on-line estimation. Finally, they should be transferable from one detector station to another without much effort.

1.2 RESEARCH OBJECTIVES

To address the above problems, this research will first identify the problem of speed estimation using single loop detector data. The nonlinearity of the speed estimation problem has been addressed in previous studies (Dailey, 1999; Wang and Nihan, 2000; Lin et al., 2004). This research will further identify the nonlinearity of this problem. A nonlinear Kalman filter, the Unscented Kalman Filter (UKF), will be proposed for the nonlinear speed estimation problem. Based on the analysis of traffic data, this research will show that speed estimation is a nonlinear non-Gaussian problem, while the UKF has the limitation of applying to non-Gaussian problems. Hence, a non-parametric filtering method, the Unscented Particle Filter (UPF), will be presented for solving nonlinear and non-Gaussian problems.

The proposed methodologies will be analyzed and applied to both real world data collected from different freeway locations and simulated data from the simulation program CORSOM. Speed will be estimated using the proposed methods as well as some existing approaches. Estimated results will be compared, analyzed, and evaluated. This research will show that both proposed methods have significant improvements on speed estimation methods developed in the past.

Based on the details presented above, the fundamental objectives of this research are listed as follows:

- Review and assessment of the state-of-the-art related to speed estimation using single loop outputs.

- Identification of the speed estimation problem using single loop outputs. In addition to nonlinearity of the speed estimation problem, this research will show that the speed estimation problem is non-Gaussian.

- Development of new methodologies/algorithms to improve the problem of speed estimation.

- Comparison and evaluation of speed estimation results generated from both the proposed methods and some existing methods using both field data collected from freeways and microscopic traffic simulation program.

1.3   RESEARCH METHODOLOGIES

The research methodologies that include literature review, data collection, speed estimation, and performance evaluation are briefly described in this section.

1.3.1 Literature Review

A comprehensive review of the literature regarding speed estimation from single loop outputs was carried out. Methods, algorithms, and theories adopted in previous works were studied and evaluated. Moreover, detectors that use the inductance technology were also reviewed.

1.3.2 Data Collection

In this research, data were collected from multiple sources. Double loop detector data have been extensively used in the literature because such detectors have the outputs of occupancy, count, and speed. The occupancy and count data can be used for speed estimation, while speed data are used for result comparison and evaluation. Thus, double loop detector data were collected from Interstate Highway 35 in the city of San Antonio for the research. Detector data from the shoulder lane were analyzed for continuous 24-hour periods for a week.

Data from the Peak ADR-6000 detector were also collected from Texas Transportation Institute's vehicle detection test beds. ADR-6000 detectors also employ the inductance technology. Accurate individual vehicle record can be detected by such detectors. Occupancy, count, and (average) speed can be obtained from detector outputs via postprocessing. Several days of data were collected from two test beds, which are located on State Highway 6 in College Station and Interstate Highway 35 in Austin, respectively.

In addition to field data, simulated data were generated from the microscopic simulation program CORSIM. A two-lane unidirectional freeway was simulated in this research with the installation of surveillance detectors. Outputs from CORSIM were used for speed estimation as well as performance evaluation for special conditions such as incidents.

1.3.3 Speed Estimation

Two speed estimation methods, the Unscented Kalman Filter (UKF) and the Unscented Particle Filter (UPF), were proposed in this research. The development of the UKF was to overcome the limitations in the existing Extended Kalman Filter (EKF) method (Dailey, 1999), which has been developed for the nonlinear speed estimation problem. The UKF has been proved to be a better solution for nonlinear systems. However, it still has some assumptions (i.e., Gaussian assumption) that do not meet real world conditions. The intent to overcome the shortcoming of the UKF leads to the development of the UPF method that can be applied to nonlinear non-Gaussian systems.

1.3.4 Performance Evaluation

Two performance measures were used for evaluating estimation results from different methods. They are the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). The measures are able to measure the bias of estimations and the variance of errors. In addition to the measures, statistical tests (paired t-tests) were also conducted to test whether or not estimation errors from different methods were significantly different.

1.4   CONTRIBUTION OF THE RESEARCH

Single loop detectors are the most widely used detectors on the U.S. highways and has been the largest source of real-time traffic data. However, vehicle speed information is not available from such detectors. As a result, there is a need for accurate speed

estimation using single loop outputs. This research is a step in this direction to improve speed estimation. The contributions of this dissertation are listed as follows:

- The problem of speed estimation is analyzed more comprehensively.

- Two new methods (the UKF and the UPF) are presented to improve the accuracy of speed estimation. At the same time, the implementations of both methods are less difficult than most existing methods.

- The improvement of speed estimation accuracy has potential benefits for many applications such as travel time estimation, incident detection, and large truck volume estimation. It is able to improve the operating performance of those applications. Moreover, the improvement enables accurate analysis of related traffic problems without expensive vehicle detection systems.

1.5 ORGANIZATION OF THE DISSERTATION

The dissertation is organized into seven chapters. Chapter I is an introduction to the research and discusses the background of the problem, statement of the problem, research objectives, research methodologies, contributions of the research, and the organization of the dissertation. Chapter II presents a comprehensive literature review on loop detectors and existing speed estimation methods. Chapter III describes the details of data collection and preliminary processing of data. Chapter IV presents the first methodology for speed estimation. A UKF method is proposed and applied to the nonlinear speed estimation problem. The results from the UKF are evaluated and further compared with those from the EKF. Chapter V presents the second speed estimation

methodology that can be applied to nonlinear non-Gaussian problems. Estimation results from this method are compared with those from the UKF and the EKF. Chapter VI presents three examples of applications that can be improved with the completion of the dissertation work. Chapter VII summarizes the dissertation, provides major conclusions of the research, and presents the recommendations for future research.

CHAPTER II

LITERATURE REVIEW

2.1   INTRODUCTION

This chapter will first provide a review of vehicle detectors. Specially, three types of detectors adopting the inductance technology will be reviewed. It is then followed by a general description of the speed estimation problem using single loop outputs. Finally, existing speed estimation methods in the literature will be reviewed and discussed.

2.2   VEHICLE DETECTORS

Since the first vehicle detector's installation at a Baltimore intersection in 1928, which was activated when a driver sounded his/her car horn at a specific location (Kell et al., 1990), various vehicle detectors have been developed and used for collecting traffic data. As defined by the National Electrical Manufactures Association (NEMA, 1983), a vehicle detector system is defined as "… a system for indicating the presence or passage of vehicles." Vehicle detectors can be used to provide input for freeway surveillance, traffic control, and data collection systems.

   Based on types of installation, traffic detectors can be broadly categorized into two classes: non-intrusive and intrusive, and the results can be further classified in terms of vehicle detection and surveillance technologies as shown in Table 2.1 (Mimbela and

Klein, 2000; Michalopoulos and Hourdakis, 2001). Among those types of detectors, Inductive Loop Detectors (ILDs) have been the most widely used vehicle detection devices for several decades in the United States because of their low costs and technology maturity (Raj and Rathi, 1994; Kell et al., 1990).

Table 2.1 Vehicle Detector Classification

Based on Installation	Based on Technology
Intrusive (Embedded)	Pneumatic Road Tube
	Inductive Loop Detectors (ILDs)
	Piezoelectric Sensors
	Magnetic Sensors
	Weigh-in Motion (WIM)
Non-intrusive	Video Image Processor (VIP)
	Microwave Radar
	Infrared Sensors
	Ultrasonic Sensors
	Passive Acoustic Array Sensors

In the following sections, three types of detectors adopting the inductive loop technology are reviewed. In previous speed estimation studies, data from both single loop and dual-loop detectors were commonly used. Thus, this part of review will include both single and dual-loop detectors. In addition, another type of vehicle detector, the

Peek ADR-6000 detectors, will also be discussed because data from them were collected and used in this research.

2.2.1 Single Loop Detectors

The evolution of the inductive loop technology can be summarized into 4 stages (Potter, 2005). From 1960's to middle 1970's, loop detector designs were based on the solid-state analog technology using discrete components (transistors, diodes, etc.). Between middle and late 1970's digital design technique was employed, which made single loop detectors capable of detecting small motorcycles and improved the overall detection reliability. From early 1980's to middle 1990's, the Metal Oxide Semiconductor—Large Scale Integration (MOS-LSI) technology significantly reduced manufacturing costs and improved reliability. Designs in this period are also called "hardware-based" designs. In the middle 1990's, the "programmable software based" digital loop detector technology was introduced. Such design significantly reduced the number of switches required in the detector by using Liquid Crystal Display (LCD).

A typical single loop system is shown in Figure 2.1 (Kell et al., 1990). The system consists of three components: a detector oscillator, a lead-in cable and a loop embedded in the pavement. The size and shape of loops largely depend on the specific application (Gordon et al., 1996). The most common loop size is 6 feet by 6 feet. When a vehicle stops on or passes over the loop, the inductance of the loop is decreased. The decreased inductance then increases the oscillation frequency and causes the electronics unit to send a pulse to the controller, indicating the presence or passage of a vehicle (Mimbela

and Klein, 2000). Single loop detectors output occupancy and traffic count data every time interval (20 sec, 30 sec, etc.).
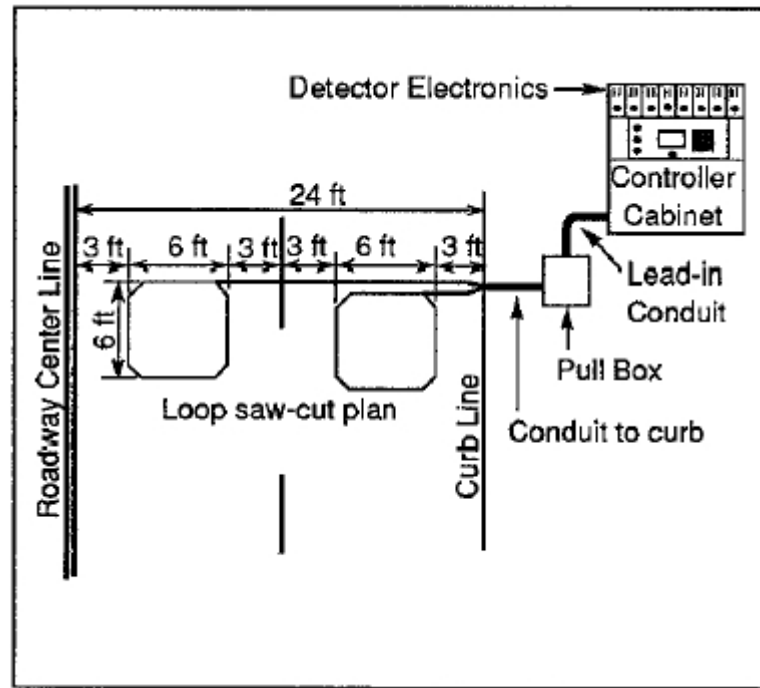
Fig. 2.1 Schematic diagram of single loop detectors

2.2.2 Dual-loop Detectors

Dual-loop detectors are also called speed traps, T loops, or double loop detectors. In a dual-loop system, two consecutive single inductance loops, called "M loop" and "S loop", are embedded a few feet apart. With such a design, when one of them detects a vehicle, a timer is started in the dual-loop system and runs until the same vehicle is detected by the other loop. Thus, in addition to outputs of vehicle count and occupancy

data, individual vehicle speeds can be trapped through the dividend of the distance between those two single loops by the elapsed time (Nihan et al., 2002). Dual-loop detectors can also be used to measure vehicle lengths with extra data extracted from controllers' records (Coifman and Cassidy, 2002).

Speed trap is defined as the measurement of the time that a vehicle requires to travel between two detection points (Woods et al., 1994). Speed is measured by

$$s = \frac{D}{t_{on}^2 - t_{on}^1} \tag{2.1}$$

where

s = is the vehicle speed;

D = is the spacing between loops;

t_{on}^1 = is the time when the first detector turns on;

t_{on}^2 = is the time when the second detector turns on.

In addition to the above speed measurement method, the other method recommends the use of both turn-on and turn-off times for speed measurement (Wilshire et al., 1985). In this method, speed can be calculated by

$$s = \frac{1}{2}\left(\frac{D}{t_{on}^2 - t_{on}^1} + \frac{D}{t_{off}^2 - t_{off}^1}\right) \tag{2.2}$$

where

t_{off}^1 = is the time when the first detector turns off;

t_{off}^{2} = is the time when the second detector turns off.

Figure 2.2 shows an example of the placement of double loops on a freeway section. In this diagram, two loops were installed in the middle of each lane with a few feet apart. The wire-loops ran from the surface to a pull box on the roadside.
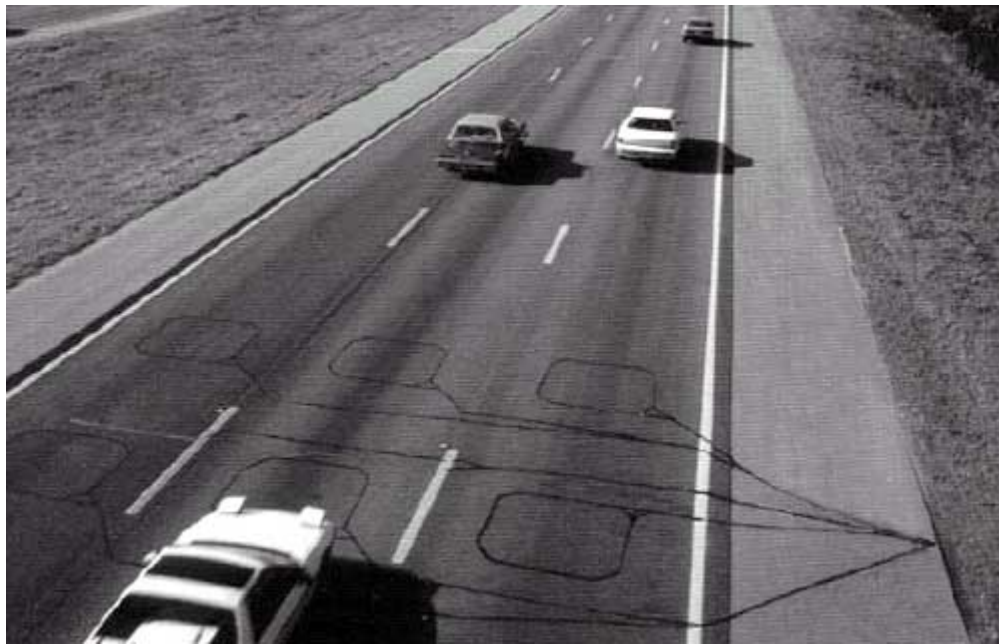
Fig. 2.2 Double loop detector system (Klein, 2003)

2.2.3 Peek ADR-6000 Detectors

A Peek ADR-6000 detector is also known as an Idris or Smart Loop system. The ADR-6000 detector uses state-of-the-art inductive loop technology and the patented Idris technology (Peek Traffic, 2004). Idris is an automatic vehicle detection and

16

classification technology. ADR-6000 detectors are installed under pavement, as shown in Figure 2.3. In each lane, there are two single loops (6.5'×6.5') placed apart with a vehicle axle detector in the middle. Axle detectors consist of two smaller loops (5'×18").

Different from the single and dual-loop detectors, the ADR-6000 detectors detect and output individual vehicle record including vehicle speed, vehicle length, classification, number of axles, and presence time. Individual vehicle speeds are trapped by vehicle signatures generated in the system. Each vehicle passing over the inductive loop will generate a specific shape of signature containing a leading and trailing edge. Thus, each vehicle will have two signatures after passing the detector. The vehicle speed can be trapped by matching two points from these two signatures.

Based on individual vehicle presence time, occupancy can be easily calculated with a specific polling interval. Thus, such detectors are able to provide traffic count and occupancy data, which are typical outputs of single loop detectors.
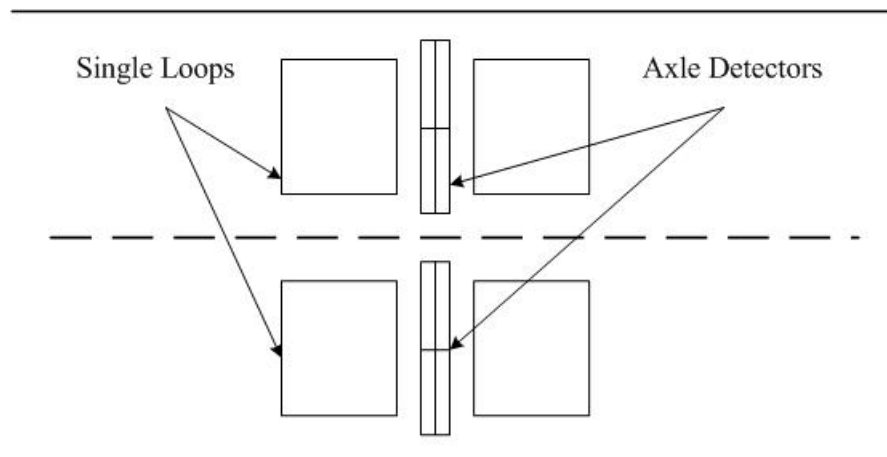
Fig. 2.3 Peek ADR-6000 detectors

2.3   DESCRIPTION OF THE SPEED ESTIMATION PROBLEM

Figure 2.4 shows a two-lane unidirectional freeway segment with single loop detectors installed. Assume that the detection zone length is l_d and is equal to the detector length, the length of the yellow car is l_v, the speed of the vehicle is s, then the presence time (the time period that the red car is over the detector) can be calculated by $t = (l_d + l_v)/s$. Let $L = l_d + l_v$, and L is called the effective vehicle length.

During the time step k within a time period of T, if N_k (count) vehicles passed over the single loop detector, then the total presence time is $t_k = \sum_{i=1}^{N_k} \frac{L_{ki}}{s_{ki}}$. The duration of time interval varies depending on the loop detection systems. The most frequently used durations in practice are 20 seconds, and 30 seconds.

Occupancy is defined as the proportion of time that vehicles occupy the detector in a time period. Based on the definition, the occupancy (O_k) is derived by:

$$O_k = \frac{t_k}{T} = \frac{1}{T} \sum_{i=1}^{N_k} \frac{L_{ki}}{s_{ki}} \tag{2.3}$$

Note that the percent occupancy is usually used in loop detector outputs, that is, $\%O_k = 100 \times O_k$.
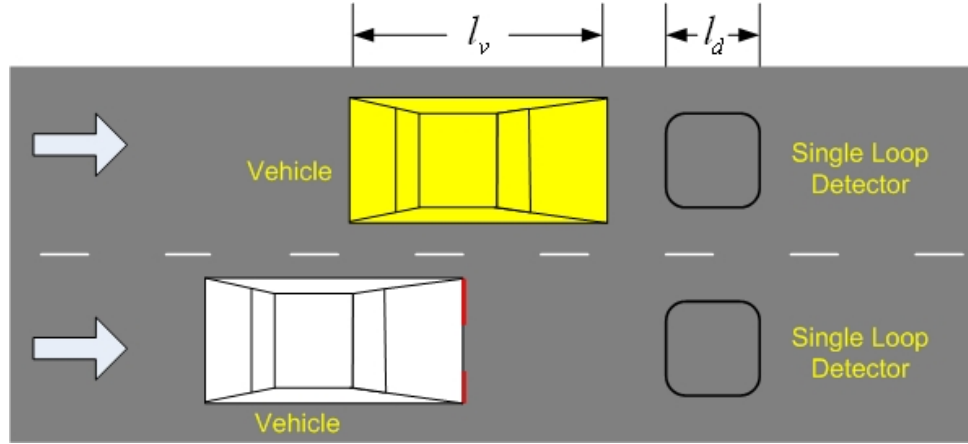
Fig. 2.4 Layout of a freeway segment with single loop detectors

2.4   EXISTING SPEED ESTIMATION METHODS

Since either individual vehicle length or speed cannot be detected by single loops, equation 2.3 is usually aggregated to the average level, which means that the average vehicle length and speed ($\bar{s}_k = \sum_{i=1}^{N_k} s_{ki}$) are used rather than individual values. Therefore, the average speed of vehicles $\bar{s}_k$ during each time period is the value to be estimated in the speed estimation problem.

Many speed estimation methods have been developed in the literature. Different methods may use different aggregation methods and assumptions. These methods are reviewed and presented as follows.

2.4.1   Conventional *g*-Estimator Method

The first method of speed estimation, the conventional *g*-estimator method, was proposed by Athol (Athol, 1965). Base on the definition of occupancy, the author presented the interrelationship between operational traffic flow characteristics, which is shown in equation 2.4.

$$\bar{s}_k = \frac{1}{g} \times \frac{N_k}{T \times O_k} \tag{2.4}$$

where

k = time interval index;

$\bar{s}_k$ = average speed (space mean speed) during *k*th time interval (miles per hour);

T = duration of time intervals (second);

N_k = vehicle count during *k*th time interval (vehicles per time interval per lane);

g = an estimator incorporating site characteristics of average vehicle length and single loop length.

In the calculation of this method, g is an estimate of the reciprocal of Mean Effective Vehicle Length (MEVL), which is denoted by $\bar{L}$ and is equal to the sum of the average vehicle length ($\bar{l} = \sum_{i=1}^{N_k} l_{ki}$) and the single loop length (l_d). In practice, g is set to a constant value. For instance, the Chicago Traffic System Center (TSC) used 1.9 as the constant g value (McDermott, 1980), and the Washington State Department of

Transportation (WSDOT) used $g = 2.4$ with $T = 300$ seconds (Ishimaru and Hallenbeck, 1999).

The study by the WSDOT showed that this constant g-estimator method did not provide satisfactory estimation accuracy. Actually, the interrelationship shown by equation 2.4 is based on two assumptions: 1) vehicle lengths are constant during each time interval; and 2) traffic is uniform (e.g., vehicles have the same speed and the spacing between vehicles is constant). However, as pointed out by Hall and Persuad (1989), those assumptions may not be valid under certain traffic conditions. In reality, the average vehicle length ($\bar{l}$) may have large variations with the presence of long vehicles, such as commercial trucks. Moreover, vehicles on freeways are not steered at a same speed; speed variance sometimes becomes a significant factor due to congestion or other conditions and thus should not be ignored.

2.4.2 Log-linear Regression Method

To account for the variation of vehicle lengths, a dynamic g-estimator method was developed. Wang and Nihan (2000) calculated the g value for each time interval as a function of the MEVL ($\bar{L}$). The relationship of occupancy, count, average vehicle length, and speed is developed by Dailey (1999) and denoted in equation 2.5.

$$\frac{E_k(O_k)}{N_k} = \frac{\bar{L}_k}{T}[\frac{\sigma_k^2 + \bar{s}_k^2}{\bar{s}_k^3}] \tag{2.5}$$

where

$E_k(O_k)$ = expectation of occupancy measurement at kth time interval, equaling

to O_k for perfect measurement;

N_k = count measurement at kth time interval;

$\bar{s}_k^{\ 2}$ = average speed at kth time interval;

$\bar{L}_k$ = mean effective vehicle length at kth time interval;

σ_k^2 = speed variance at kth time interval;

T = duration of the time interval.

Wang and Nihan (2000) also conducted a study on the ratio of $\sigma_k^2 / \bar{s}_k^{\ 2}$ and found that values were very low. Consequently they assumed that speed variance can be ignored, and the following equation was then derived after statistical transformations:

$$\bar{L}_k^2 = \frac{E^2(O_k) \times \sigma_{lk}^2}{V(O_k)} \tag{2.6}$$

where σ_{lk}^2 is the variance of vehicle lengths at kth time interval, and $V(O_k)$ is the occupancy variance. After introducing some additional variables, such as a high-flow dummy, to account for σ_{lk}^2, a regression model of the MEVL at kth time interval was established and is shown in equation 2.7.

$$\ln(\bar{L}_k) = \beta_0 + \beta_1[2\ln(E(O_k)) - \ln(V(O_k))] + \beta_2 \times \ln(N_k) + \beta_3 \times HFD + \beta_4 \times LFD + \varepsilon_k \tag{2.7}$$

where *HFD* is a high-flow dummy, *LFD* is a low-flow dummy, βs are coefficients, and ε_k is a white noise.

However, the correlation coefficient of the regression model might be very low due to the variations of speed and effective vehicle length. Moreover, ignoring speed variance may lead to certain level of inaccuracy. Finally, this method is site-specific and cannot be applied to other locations without recalibration.

2.4.3 Modified g-Estimator Methods

In order to reduce the influences of long vehicles and congested traffic conditions, two studies (Coifman et al., 2003; Lin et al., 2004) modified the *g*-estimator method. They used median values of speed and vehicle passage time respectively, instead of mean values adopted in the *g*-estimator method. The modified median g-estimator methods can reduce the skewnesses of the distributions of speed and pace (the reciprocal of speed). However, additional problems arise with the modified *g*-estimator methods. In the study by Coifman et al. (2003), to estimate the median speed in a single lane, the time unit (length of time intervals) of speed estimation should be long enough (e.g., 5 minutes) to ensure that sufficient sample size (number of vehicles) is achieved according to the sampling criteria. Thus, to obtain good estimates of speed for short time units such as 30 seconds, it is required to combine vehicle data across several lanes. But in doing this, it is impossible to identify speed difference across single lanes. This is because different lanes at a location tend to show different temporal patterns of speed in reality, especially when there exist large differences of traffic flow between lanes.

In the other study (Lin et al., 2004), the median vehicle passage time (l_{median}/s_{median}) at each time interval is used to replace the mean vehicle passage time (l_{mean}/s_{mean}), and the median vehicle passage time is approximated by $(l/s)_{median}$. To implement this method, the information of passage times ($Time_1$ when the vehicle reaches the front part of the loop and $Time_2$ when the rear end of the vehicle leaves the single loop) is required from each vehicle so that the value of $(l/s)_{median} = (Time_2 - Time_1)_{median}$ can be obtained. However, the common outputs (vehicle count and occupancy) of single loops do not include such information.

2.4.4 Extended Kalman Filter (EKF)

Dailey (1999) presented a statistical method, the Extended Kalman Filter (EKF) method, to linearize the measurement equation for speed estimation. A general Kalman Filter (KF) model includes two equations, a state-transition equation and a measurement equation (Bozic, 1994). These two equations are

$$x_k = A\bar{x}_{k-1} + Bu_{k-1} + v_{k-1} \tag{2.8}$$

$$y_k = Hx_k + n_k \tag{2.9}$$

where

x_k = predicted value at kth time interval from previous time interval;

y_k = measurement at kth time interval;

u_{k-1} = control input;

v_k = process noise;

n_k = measurement noise.

The KF method operates with two phases per time interval: the time update phase to "predict" new state, and the measurement update phase to "correct" new state. In a speed estimation application, average speed at kth time interval is the state, and occupancy over count ratio, which can be gathered from single loop detectors, is the measurement. The EKF linearizes the measurement equation, which is established based on equation 2.5 and assumes perfect measurement of occupancy data.

However, there are several issues in the EKF and its speed estimation implementation. As pointed out by Julier and Uhlmann (1997), linearization in the EKF will produce highly unstable filters if assumptions are not met, and the derivation of the Jacobian matrices often lead to significant implementation difficulties. Note that in the EKF, Jacobian matrices are partial derivatives of a nonlinear function with respect to its variables. To better describe the drawbacks of the EKF, assume that x is a random variable and $y = f(x)$, then the mean value of y can be achieved by expecting $f(x)$, this can be shown as

$$\bar{y} = E[y] = E[f(x)] \tag{2.10}$$

Only for linear Gaussian system, we can get $\bar{y} = f(\bar{x})$; for nonlinear systems, this is not the case. While in the EKF, the mean value is calculated as $\bar{y} = f(\bar{x})$, not $\bar{y} = E[f(x)]$.

The EKF only considers the first order of Taylor series (equation 2.11) to perform linearization.

$$f(x_{k-1}) = f(x_{k-1}^a) + f^{'}(x_{k-1}^a)(x_{k-1} - x_{k-1}^a) \tag{2.11}$$

In the implementation of the EKF, the state variable (average speed at time interval k) is calculated based on the two previous states using a state transition matrix

$$G = \begin{pmatrix} a & b \\ 1 & 0 \end{pmatrix}. \tag{2.12}$$

Coefficients a and b represent weights for the two previous states. In the EKF method, these two coefficients are derived using Auto Regression (AR) method with 2 orders based on measured speed data. Theoretically, the accuracy of filtering results largely depends on the number of orders, and the coefficients of AR have a great effect on the results. Since experimentally measured speed data can only represent the variation of some speed change patterns in certain time duration, such AR coefficients may not always lead to good estimation accuracy.

2.4.5 Exponential Smoothing Method

Hellinga (2002) used a volume weighted exponential smoothing method to improve the traditional g-estimator method. This method is applicable to freeway Traffic Management System (TMS) that contains both single and double loop detector stations. Thus, MEVL measured from dual-loop detectors can be applied to nearby single loop detectors. However, it is found that the correlation between the MEVLs measured from

two detectors in a detector station set is very low, which is caused by sampling error. To decrease sampling error, it is needed to choose a longer time period, however this is difficult to do in practice. Therefore, the exponential smoothing method is proposed to avoid the problem of having to select a fixed sampling period duration.

Estimated results using this method are approximately 20% more accurate than the traditional g-estimator method, while estimation errors are still relative high as shown in the study. In addition, the applicability of this method is limited since this method is not applicable when there is not a double loop detector presented in the vicinity of each single loop detector station.

2.4.6   Catastrophe Theory Method

The Catastrophe Theory was originated by French mathematician Rene Thom in the 1960's and developed by Zeeman (1977). Catastrophe means the loss of stability in a dynamic system. As a special branch of dynamical system theory, the Catastrophe theory studies and classifies phenomena characterized by sudden shifts in behavior arising from small changes in circumstances. This theory was used by Hall (1987) and Pushkar (1994) to estimate speed using single loop outputs. The authors established a relationship between traffic variables (occupancy, speed, etc.) and a 3-dimentional folded surface in the Catastrophe Theory. The Catastrophe Theory model is presented in equation 2.13.

$$4x^3 + 2ux + v = 0 \tag{2.13}$$

where x is the state variable associated with speed, and u and v are control variables related to flow and occupancy respectively. To model traffic flow behavior and estimate

speed, the author used two simple linear transformations, as shown in the following, to

establish the relationship between x and speed as well as u and flow.

$$x = speed - speed\_at\_capacity$$
$$u = (flow - capacity)/1000$$

<div align="right">(2.14)</div>

The transformation between v and occupancy is accomplished in *ad hoc* manner and

is shown in Figure 2.5 (Hall, 1987). When occupancy and flow data are available, speed

can be estimated using the Catastrophe Theory model and those three transformations.

Although those transformations simplify the speed estimation, the involvements of

empirical data and results (i.e., capacity, speed at capacity, and arbitrary relationship

between v and occupancy) may introduce significant errors.
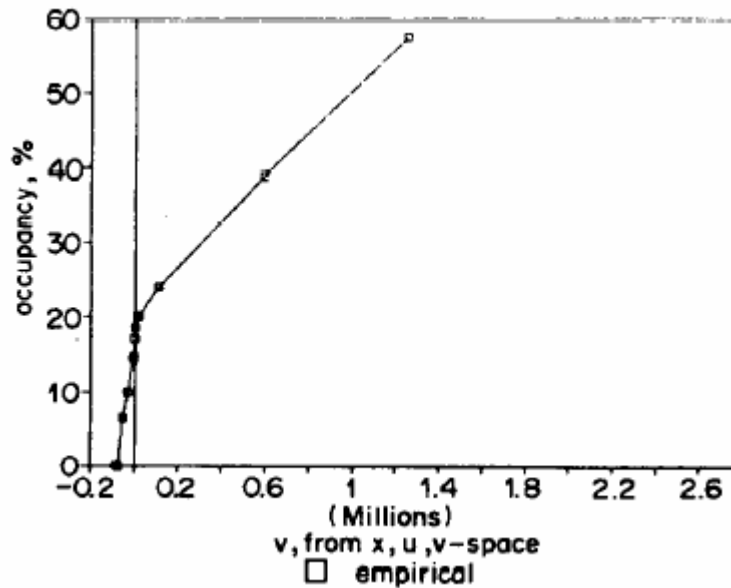
Fig. 2.5 Transformation of occupancy to v based on empirical results (Hall, 1987)

2.4.7 Hybrid Model

Yao et al. (2004) presented a hybrid model, which consists of two sub-models for speed estimation under both free flow and traffic congested conditions. The hybrid model is

$$\bar{s}_k = s_f e^{-O_k} \qquad O_k \leq O_{threshold}$$
$$\bar{s}_k = s_m \ln(1/O_k) \quad O_k \geq O_{threshold}$$

(2.15)

where s_f and s_m represent speed under free flow and congested flow conditions respectively. s_f can be estimated by using single loop data and dropping out congested data. A constant MEVL is also needed to calculate s_f. With s_f obtained, s_m is empirically calculated by $s_m = s_f/e$. A threshold occupancy value $O_{threshold}$ is used to identify whether traffic flow is free or congested.

This method is simple once the initial parameters (i.e., s_f, s_m, $O_{threshold}$, and $\bar{L}$) are calibrated. However, the constant s_m may contribute to large errors because traffic flow and speed is rather unstable under congested conditions. The authors do not provide a sensitive analysis of s_m. Moreover, the threshold value $O_{threshold}$ varies, especially under different weather conditions.

2.4.8 Vehicle Signature

As part of traffic monitoring and surveillance systems, sensor technology has been receiving a lot of attention and many detectors have been developed to obtain more comprehensive and accurate traffic data. In the middle of 1990's, the "programmable

software based" digital loop detector technology was used to upgrade existing "hardware-based" designs, by replacing a very few switches with an active LCD (Potter, 2005). With such a design, more information can be obtained from ILDs besides occupancy and vehicle count data. An ILD system with high speed scanning detector cards is able to capture the "inductive signatures" of different types of vehicles. Each vehicle passing over the inductive loop will generate a different shape of signature containing a leading and trailing edge. The signature information has been used in speed estimation. Sun and Ritchie (1999) proposed a new speed estimation technique using single ILD signatures, with signal processing and linear regression techniques. A simple linear regression model is presented to model the relationship between speed and slew rate. Slew rate is the edges (either leading or trailing) that represent the rate of metallic mass of vehicle passing over the loop magnetic field. Oh et al. (2002) estimated speeds using vehicle signatures through extracting signature feature vectors.

The vehicle signature method is different from the previous methods in that different information (vehicle signature) is used for speed estimation. It should be noted that, except for this method, other speed estimation methods all use count and occupancy data from loop detectors.

2.4.9 Other Methods

Several other methods were also proposed in previous works. They are included in this part of review as those methods are difficult to be classified and given appropriate names. Coifman (2001) stated that under free flow conditions, occupancies from loop

detectors are low. Thus, a threshold value of occupancy was set to identify free flow traffic with a specific free flow speed. This method improves the vehicle length estimates under free flow conditions assuming a linear relationship of MEVL, speed, occupancy and count.

Some other studies have tried to estimate speed by exploring the relationship between speed and occupancy. For instance, An Istanbul study (Ogut, 2004) used data from four locations to analyze and establish a regression model between speed and occupancy, in which occupancy was a function of speed. This method does not take vehicle length and other factors into account.

2.4.10 Summary

This chapter reviewed three types of loop detectors including single loop detectors, dual-loop detectors, and Peek ADR-6000 detectors, all of which adopt the inductance loop technology. Moreover, existing methods for speed estimation using single loop outputs were reviewed.

From the above discussion, there are several issues existing in the problem of speed estimation. First of all, it is difficult to accurately estimate the MEVL for each time interval. Thus, a common MEVL is generally used in practice. Moreover, a simplified linear relationship between speed and other parameters are usually used in past studies. As mentioned before, the linear relationship is based on two assumptions: constant vehicle lengths and uniform traffic. The assumption of constant vehicle lengths is obviously not realistic. Uniform traffic means that all vehicles during a polling interval

have the same speed and spacing. This assumption itself ignores speed variation of those vehicles. Such simplifications will produce estimation errors. Finally, previous methods have their own drawbacks in underlying theories. As a result, the accuracy of estimation results is generally unsatisfactory.

CHAPTER III

DATA COLLECTION AND PRELIMINARY PROCESSING

3.1   INTRODUCTION

As mentioned in Chapter I, the main objective of this dissertation is to develop methodologies to improve speed estimation using single loop outputs. Therefore, the main data of interest in this dissertation are loop detector data collected from the field. This chapter will discuss the details of field data collection and the preprocessing of the data.

In addition to field data, simulated data generated by the microscopic traffic simulation program CORSIM were also used. The advantage of using simulated data is that such data can be easily generated for various conditions including non-recurring congestion that can not be easily obtained in the field. The details of the simulation program, parameter settings, and the details of the simulated data are provided in this chapter.

3.2   DATA SOURCES

3.2.1   Peek ADR-6000 Detector

Peek ARD-6000 detectors are relatively new devices for vehicle detection. Such detectors have not been widely used on the U.S. highways. TTI (Texas Transportation Institute) and Texas DOT are the first agencies that use the Peek ADR-6000 for

evaluating of vehicle detectors (Middleton and Parker, 2000 & 2002). Peek ADR-6000 detectors were installed at two of the TTI's vehicle detection test beds for providing baseline data. One is located on State Highway 6 (SH6) in College Station, Texas; the other is on the south bound of Interstate Highway 35 (IH-35) near the 47th street in Austin, Texas.

A snapshot of the SH6 test bed in College Station is shown in Figure 3.1 (Middleton and Parker, 2000). The freeway section has two lanes in each direction. Several types of intrusive detectors, such as microloops, piezoelectric sensors, and ADR-6000 detectors, were embedded under the pavement. Also, this site has a forty-foot pole with two mast arms, on which non-intrusive detectors are supported. Those non-intrusive detectors include two traffic-monitoring cameras, two vehicle detector cameras, two acoustic detectors, and a microwave radar detector.

Fig 3.1 Test bed in College Station (Middleton and Parker, 2000)

The freeway section at the IH-35 test bed has four through lanes in each direction and an exit lane on the southbound side to Airport Boulevard. This site is located north of the elevated section of IH-35 that contributes to the dispersion of traffic. As a result, an unusually high percentage of trucks use the left two lower lanes of the freeway and avoid the other two elevated lanes (Middleton and Parker, 2002). Usually, the right lanes on multilane highways have a higher truck percentage. This test bed has high traffic volumes during peak hours. Some vehicles even underwent stop-and-go conditions.

Many types of vehicle detectors were installed at this site. Remote Traffic Microwave Sensor (RTMS), SAS-1 acoustic detector, Autoscope Video Image Detector (VID), and other non-intrusive detectors were mounted on light poles. Two types of intrusive detectors were also installed under the pavement surface. They are double loop detectors and Peek ADR-6000 detectors. The ADR-6000 detectors were only installed in the five southbound lanes. Note that data from the four through lanes were collected for this study.

The ADR-6000 detectors are able to store three types of data: raw loop signatures, binned data, and Per Vehicle Records (PVR). TTI has no access to analyze the raw loop signatures and such data take up large amount of disk storage, the feature is hence turned off. In this study, PVR data are used for the purpose of this study. PVR data are saved in PVR files. Each PVR file can store data for around 158 kilobytes. Individual vehicle information in the PVR file include date, time, lane number, vehicle length, vehicle speed, presence time, vehicle classification, and number of axles. Figure 3.2 shows a

sample of PVR data that were imported into Microsoft Excel beforehand for a better view. The first row of the Excel sheet in the figure provides the description for each column.

Fig 3.2 PVR data

TTI carried out a field test on Peek ADR-6000 detectors at the IH-35 test bed (Middleton and Parker, 2002). Tested traffic parameters include count, speed, and vehicle classification. It was found that the Peek ADR-6000 had almost perfect count accuracy. Among the total 1923 vehicles, only one vehicle was missed by the Peek

ADR-6000. The speed accuracy was tested by using a laser device. The Peek ADR-6000 was also found to have high accuracy of speed measurement and had even better performance than RTMS and Autoscope. Moreover, the classification accuracy of the Peek ADR-6000 was close to 99% based on the sample of the 1923 vehicles. The field test demonstrated the good performance of the Peek ADR-6000 in vehicle detection.

From Figure 3.2, it can be seen that the ADR-6000 detectors do not generate occupancy data directly. However, occupancy information can be calculated through postprocessing. An occupancy program was hence developed to generate occupancy data using Matlab (Matrix Laboratory), which is a programming language and a numerical computing environment with powerful capabilities for matrix manipulation, plotting of data, implementation of algorithms, creation of user interfaces, and interfacing with other program languages (Mathworks, 2002). By running the occupancy program, PVR data can be compiled to generate occupancy, traffic count, and speed data for each lane with a specified polling time. This program is shown in Appendix E.

Traffic flows at the SH6 test bed are low to medium. Figure 3.3 shows an example of hourly traffic volumes at this site during a weekday. It can be observed that both morning and afternoon peak hour traffic volumes are between 1000 veh./hr./lane and 1200 veh./hr./lane. The shoulder lane (numbered as lane 1) has a daily traffic volume of 12975 vehicles and 7% trucks with 3-axle or more during the entire day; the median lane (numbered as lane 2) has around 1000 vehicles per hour and 4.5% trucks. Under such traffic volume conditions, vehicles usually drove at free flow speed.
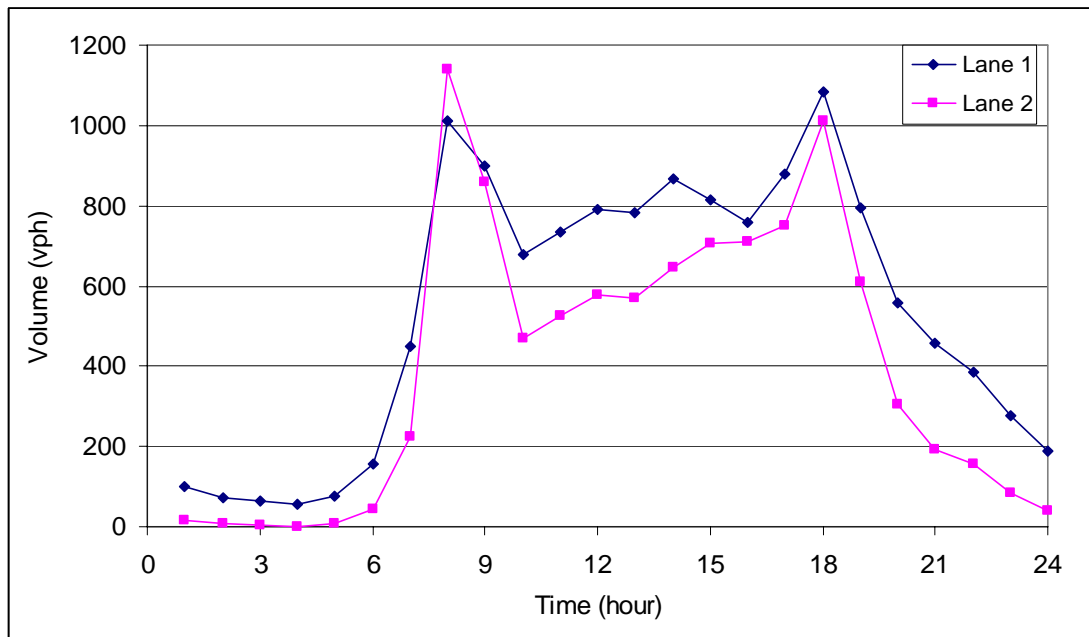
Fig 3.3 Hourly traffic volumes from the SH6 test bed on Jan. 27, 2004.

The IH-35 test bed has heavy traffic loads during the daytime. A plot of hourly traffic volumes on October 27, 2004 is shown in Figure 3.4. Daily traffic volumes from lane 1 (shoulder lane) through lane 4 are 27670, 24936, 20226, and 13850 vehicles, respectively. Correspondingly, the truck percentages are 12.3%, 5.3%, 1.9%, and 2.8%. As mentioned earlier, most trucks are distributed in lanes 1 and 2.
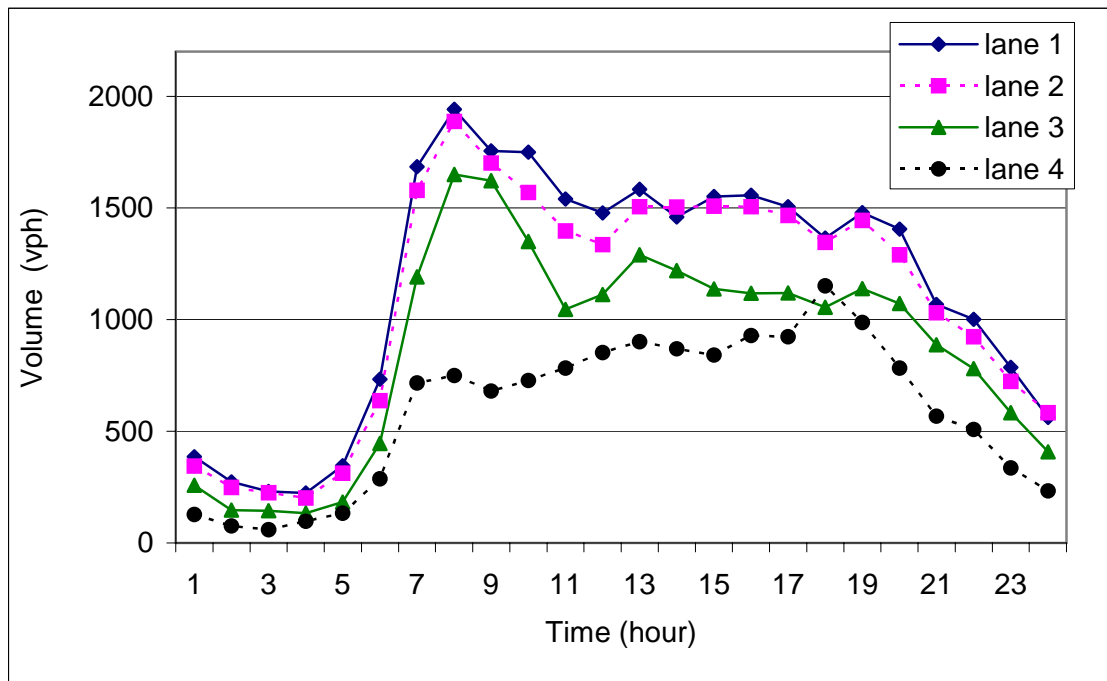
Fig 3.4 Hourly traffic volumes from the IH-35 test bed in Austin on Oct. 27, 2004

3.2.2 Dual-loop Detectors

The city of San Antonio in Texas has an extensive freeway system, with three interstate highways (IH-35, IH-37, and IH-10) passing through the city. Many dual loops were installed on these highways within the city limits and provide data for efficient transportation management. Dual loop data were downloaded from the San Antonio Texas Transportation Institute server for this study. The location of dual loops is on IH-35 at Seguin Road, with 3 lanes in the northbound direction. The duration of time intervals is 20 seconds. Speed, occupancy, and count data were collected at this location.

A sample of dual-loop data is shown in Table 3.1. This sample data include two subsamples separated by dotted lines: one represents data collected under normal traffic conditions, and the other under congested traffic conditions. It can be seen that occupancy data were high during the afternoon peak; speeds sometimes were even lower than 10 mph.

3.2.3   Microscopic Traffic Simulation Using CORSIM

For the test and validation of the methodologies developed in this study as well as existing methods, simulation models can be used to reproduce actual field conditions with reasonable accuracy. Thus, simulated data were generated from the traffic simulation package CORSIM (CORridor SIMulation). CORSIM is one of the most widely used microscopic traffic simulation programs in the United States. CORSIM is able to model complex geometry conditions, simulate different traffic conditions, model time-varying traffic and control conditions, and account for the interactions between different components of networks; its validation, verification, and calibration effort ensures that results from CORSIM reflect real world traffic flow (CORSIM User's Guide, 2001). Under the sponsorship of FHWA (Federal Highway Administration), the CORSIM logic was initially developed in early 1970s. Since then, CORSIM has undergone several technological improvements. TSIS (Traffic Software Integrated System) is a recent development that provides a user-friendly interface environment for running the CORSIM model (Owen et al., 2000).

Table 3.1 Dual-Loop Detector Data

Station ID	Date	Time	Speed (mph)	Volume (veh/period)	Occupancy (%)
L1-0035N-161.405	2/14/2003	12:36:23	66	7	8
L1-0035N-161.405	2/14/2003	12:36:43	68	6	7
L1-0035N-161.405	2/14/2003	12:37:03	68	6	7
L1-0035N-161.405	2/14/2003	12:37:23	68	10	11
L1-0035N-161.405	2/14/2003	12:37:43	62	8	11
L1-0035N-161.405	2/14/2003	12:38:03	63	11	13
L1-0035N-161.405	2/14/2003	12:38:23	62	8	10
L1-0035N-161.405	2/14/2003	12:38:43	63	11	13
L1-0035N-161.405	2/14/2003	12:39:03	62	13	20
L1-0035N-161.405	2/14/2003	12:39:23	62	11	15
L1-0035N-161.405	2/14/2003	12:39:43	65	9	10
L1-0035N-161.405	2/14/2003	16:27:28	19	2	22
L1-0035N-161.405	2/14/2003	16:27:48	9	7	98
L1-0035N-161.405	2/14/2003	16:28:08	17	10	41
L1-0035N-161.405	2/14/2003	16:28:28	23	12	37
L1-0035N-161.405	2/14/2003	16:28:48	27	11	30
L1-0035N-161.405	2/14/2003	16:29:08	33	11	25
L1-0035N-161.405	2/14/2003	16:29:28	38	10	19
L1-0035N-161.405	2/14/2003	16:29:48	39	12	23
L1-0035N-161.405	2/14/2003	16:30:08	40	11	21
L1-0035N-161.405	2/14/2003	16:30:28	39	13	26
L1-0035N-161.405	2/14/2003	16:30:48	38	9	23
L1-0035N-161.405	2/14/2003	16:31:08	31	10	43
L1-0035N-161.405	2/14/2003	16:31:28	4	4	69

CORSIM includes two separate simulation modules: NETSIM (NETwork SIMulation) and FRESIM (FREeway SIMulation). NETSIM is a simulator that describes the performance of vehicles traveling in an urban street network, and FRESIM is a simulator for freeways. TSIS has two processors, the input processor (TRAFED, TRAFfic network EDitor: a graphic input editor) and the output processor (TRAFVU, TRAFfic Visualization Utility: an animation and graphics module). TRAFED includes a translator that can convert a graphically edited network into an input TRF file for CORSIM. TRAFVU is a visualization processor for the CORSIM traffic simulation, so that users can visualize the simulated network and analyze the simulation results.

In this study, a two-lane unidirectional freeway section was created in CORSIM, which is shown in Figure 3.5. The freeway section can be also described by a TRF file shown in Appendix C. A loop detector was placed in the shoulder lane (lane 1) for vehicle detection. CORSIM users can specify *presence* or *passage* for detectors. Since occupancy data are necessary for this study, a *presence* detector was selected.
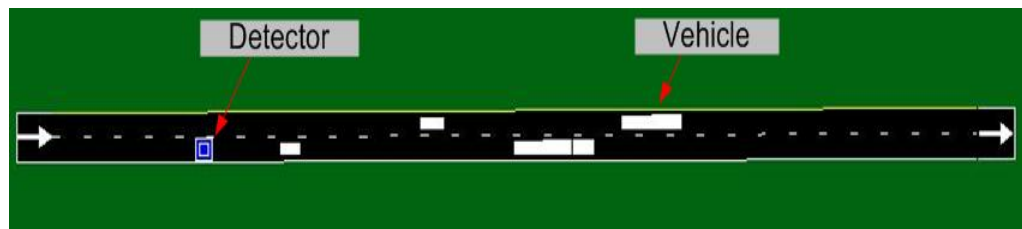
Fig 3.5 Layout of the simulated freeway section

As mentioned in Chapter I, traffic congestion is a common phenomenon and contributes to the difficulty of speed estimation using single loop detectors. High traffic

volumes during peak hours, incidents, construction zones, severe weather conditions, and other situations can cause traffic congestion. Thus, the scenario of congested traffic flow was simulated. An incident was simulated in lane 2 during the simulation and caused lane blockage of this lane for a certain time. With such a configuration, traffic speeds in lane 1 were reduced due to the incident and recovered after the incident and the queue have been cleared. The configurations of some important parameters are as follows:

- Simulation duration: 2 hours.

- Flow rate: 2000 vphpl.

- Free flow speed: 60 mph.

- Start time of the incident: 15 minutes after the beginning of the simulation.

- Duration of the incident: 5 minutes.

- Truck percentage: 10%.

  The percentages of different vehicle types were given by the defaults values in CORSIM and are shown in Table 3.2. Two types of passenger cars and 5 types of trucks were used for the simulation.

- Time period of the detector outputs: 20 seconds.

Other parameters were set as default values in CORSIM since they can give reasonable results. Once the simulation is run, detector data can be read from the CORSIM output file (OUT file). Data include speed, occupancy, vehicle count, and *on* (presence) time. Therefore, the outputs can be directly used for speed estimation. A sample of detector outputs is shown in Appendix D.

Table 3.2 Configuration of Vehicle Types

Vehicle Classifications	Percentage by Classifications (%)	Vehicle Types	Vehicle Length (ft)	Percentage by Types (%)
Passenger Cars	90	1	14	25
		2	16	75
Trucks	10	3	35	20
		4	53	36
		5	53	24
		6	64	9
		7	40	11

3.3 SUMMARY

This chapter described the details of data sources and the pre-processing of data. This study used three different data sources for speed estimation: Peek ADR-6000 detector data, dual-loop data, and simulated data. ADR-6000 detector data were collected from two of TTI's vehicle detection test beds, located in the cities of College Station and Austin; dual-loop data were collected from IH-35 in San Antonio. In addition to field data, simulated data from the microscopic traffic simulation package CORSIM were generated. An incident was presented in the simulation to simulate traffic congestion.

The data collected from different sources will make it more comprehensive to evaluate

the proposed methods and existing methods under various traffic conditions.

CHAPTER IV

METHODOLOGY I: UNSCENTED KALMAN FILTER (UKF)

4.1   INTRODUCTION

This chapter will first identify the nonlinear problem of speed estimation. To overcome

the shortcomings of the EKF that was developed for the nonlinear problem, the UKF

method will be proposed. Both the EKF and the UKF are members of the KF family.

They are developed for nonlinear systems. The operations of the KF, the EKF and the

UKF will be described in details. In the following, data described in the previous chapter

will be applied to the proposed method as well as the EKF. Finally, estimation results

from both methods will be compared and evaluated.

4.2   NONLINEAR SYSTEM OF SPEED ESTIMATION

As shown in the study by Dailey (1999), the problem of speed estimation is a nonlinear

system. In this system, the MEVL ($\overline{L}_k$) and the speed variance (σ_k^2) are major variables

contributing to the nonlinearity. At a given location, $\overline{L}_k$ varies over time and the

variation is mainly determined by the involvement of trucks and other long vehicles.

This variable, however, is hard to estimate with single loop outputs. Thus, a common

value $\overline{L}$ is usually used during estimation. Figure 4.1 shows an example of average

vehicle lengths varying over time from 5 a.m. to midnight, using real world data

collected from the vehicle detection test bed in Austin, Texas. In this figure, the time interval is 30 seconds.
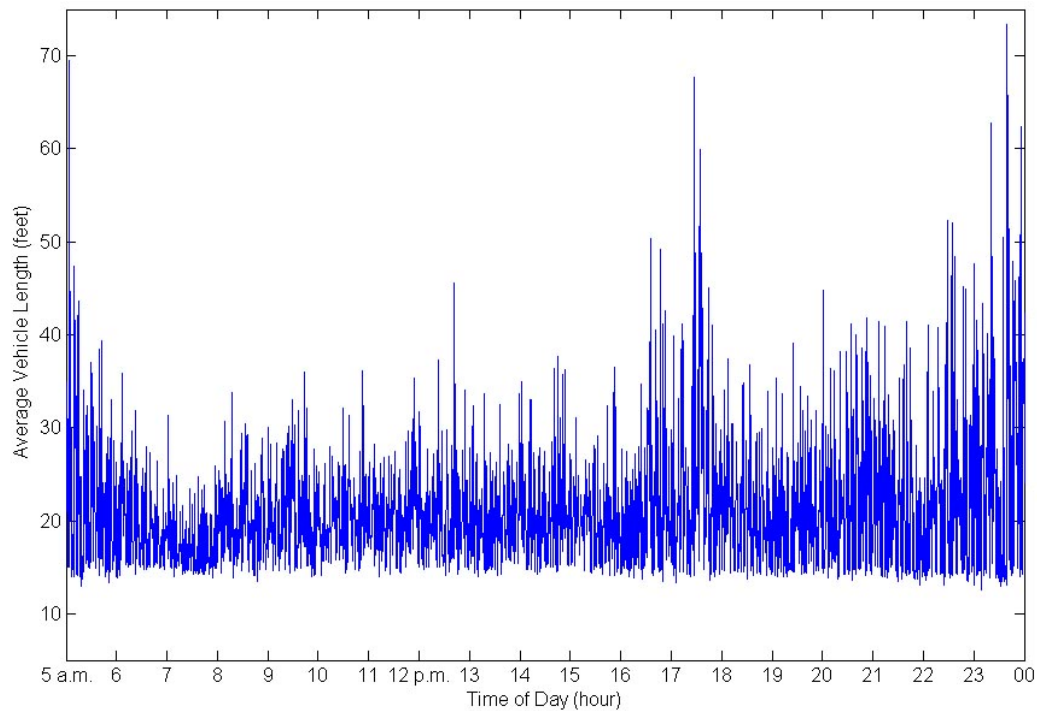
Fig. 4.1 Average vehicle lengths over time

It can be observed that average vehicle lengths during nighttime are generally larger than those during daytime, while the figure does not show any clear pattern of the average vehicle length with time. The average vehicle length sometimes is up to 70 feet during night time. It can also be very large during daytime in that some of the average vehicle lengths are larger than 50 feet between 16:00 and 18:00. Moreover, vehicle lengths during morning peak hours (6:00-9:00 hours) are lower than those during afternoon peak hours (16:00-19:00 hours). Finally, it is found that the lowest vehicle

lengths are close to a threshold value, such as 13.5 feet in this case. This is because passenger cars are generally longer than this value. In CORSIM, the shortest vehicle length is 14 feet, which shows consistency with ground observed values.

The effect of speed variance (σ_k^2) on the nonlinearity of the speed estimation system was not explored well in past studies. This is because outputs from either single loop or dual-loop detectors do not provide individual vehicle speed information. Peek ADR-6000 detectors make it possible to analyze the influence of speed variance. Real-world data from ADR-6000 detectors can be compiled to calculate the ratio of speed variance over squared speed ($\sigma_k^2 / \bar{s}_k^2$). An example of the ratio over time is shown in Figure 4.2, which also displays speed $\bar{s}_k$ over time (5 a.m. to midnight) in the upper plot.

From this figure, it can be observed that the ratio $\sigma_k^2 / \bar{s}_k^2$ is almost negligible under free-flow traffic conditions. However, under congested traffic conditions with low speeds ($\bar{s}_k$), the ratio $\sigma_k^2 / \bar{s}_k^2$ becomes large. High $\sigma_k^2 / \bar{s}_k^2$ values occur under congested traffic conditions. The maximum value of $\sigma_k^2 / \bar{s}_k^2$ is as high as 0.25. The example shows that speed variance sometimes are significant, and should have certain effects on speed estimation if ignored.
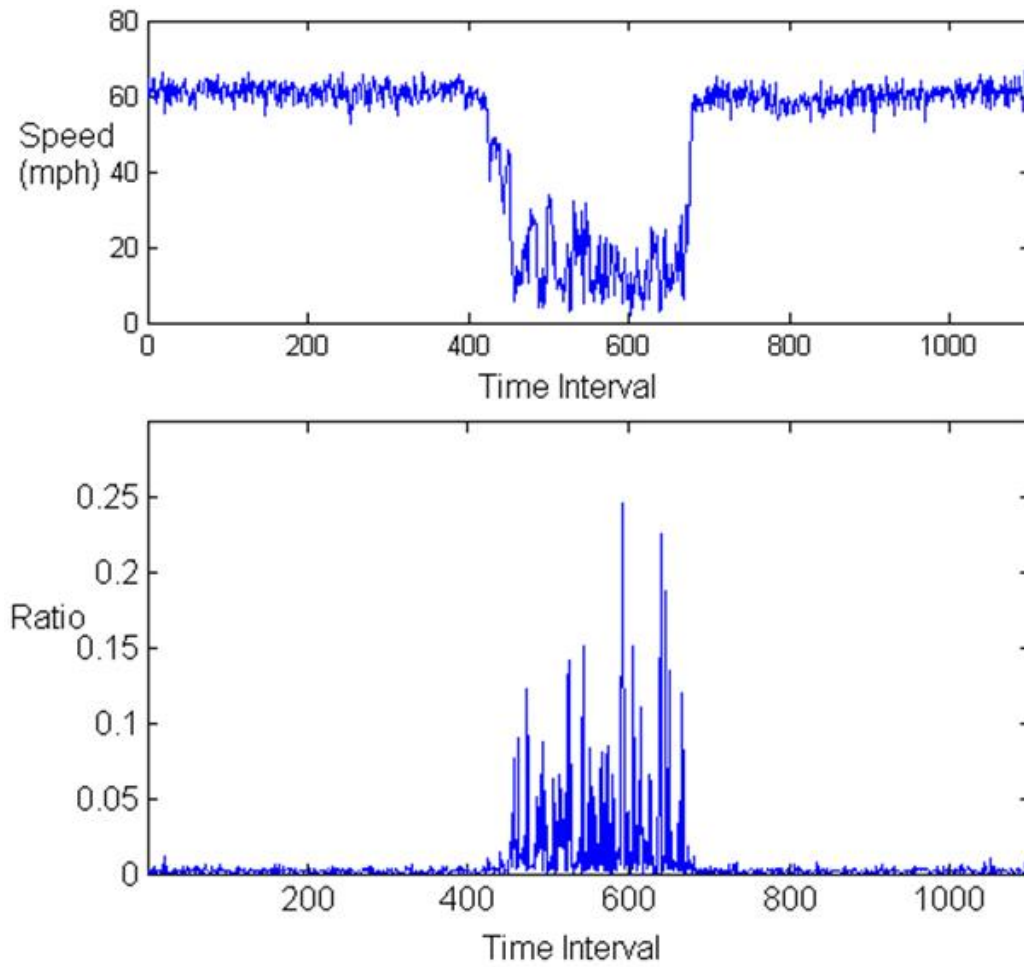
Fig. 4.2 Speed and the ratio of speed variance to squared speed over time

4.3   KALMAN FILTERS

To solve the nonlinear problem of speed estimation, Dailey (1999) presented an EKF method. As discussed in Chapter II, although the EKF is able to deal with nonlinear problems, there are several issues regarding this method. A better approach to handle

nonlinear systems is thus desirable for speed estimation. In this study, a new method, the UKF, is proposed for this problem.

Realizing the flaws existing in the EKF, Julier et al. (1996; 1997; 2000) presented a new estimator—the UKF. The UKF can be applied to nonlinear systems without the linearization steps required by the EKF. The UKF can achieve the second order or higher accuracy for nonlinear applications. Several studies have shown the superiority of the UKF for nonlinear systems (Merwe et al., 2004; Shin and Naser, 2004; Wan and Merwe, 2000). The UKF has been applied to many problems such as state estimation, parameter estimation and machine learning, yet it was rarely used in the field of transportation. In this section, the KF, the EKF, and the UKF are described as follows.

4.3.1 Kalman Filter (KF)

The KF, proposed by Kalman (1960), is one of the most advanced methods in modern control theory. The KF can be defined as an optimal recursive data processing algorithm. For a better understanding of the definition, the meanings for *optimal*, *recursive*, and *data processing algorithm* are described as follows (Maybeck, 1979):

- Optimal means that the KF incorporates all information that can be provided to it. The KF uses (a) knowledge of the system and measurement device dynamics, (b) the statistical description of the system noises, measurement errors, and uncertainty in the dynamics models, and (c) any available information about initial conditions of the variables of interest. Along with above information, it

processes all measurements to estimate the current value of the variables of interest.

- Recursive means that the KF does not require all previous data to be kept in storage and reprocessed every time a new measurement is taken. Only the estimated state from the previous time step and current measurement are needed to obtain the estimate of the current state. This is a very important feature for the practicality of filter implementation.

- In most practical applications, the filter is actually a data processing algorithm and is just a computer program in a central processor.

The KF is a dynamic system (a system varying with time) consisting of two parts, as is shown in Figure 4.3. In the first part, the new state is predicted through a process equation (equation 2.8). The equation uses the information of the previous state. After the new state is predicted, the measurement can be predicted via a measurement equation (equation 2.9). It can be seen that the dynamic system uses *prior* knowledge for prediction. As is mentioned in Chapter II, the operation of the KF includes two steps, and the prediction belongs to the first step. The second step is the correction, in which the predicted state is updated based on the difference (innovation) of the true and predicted measurements.
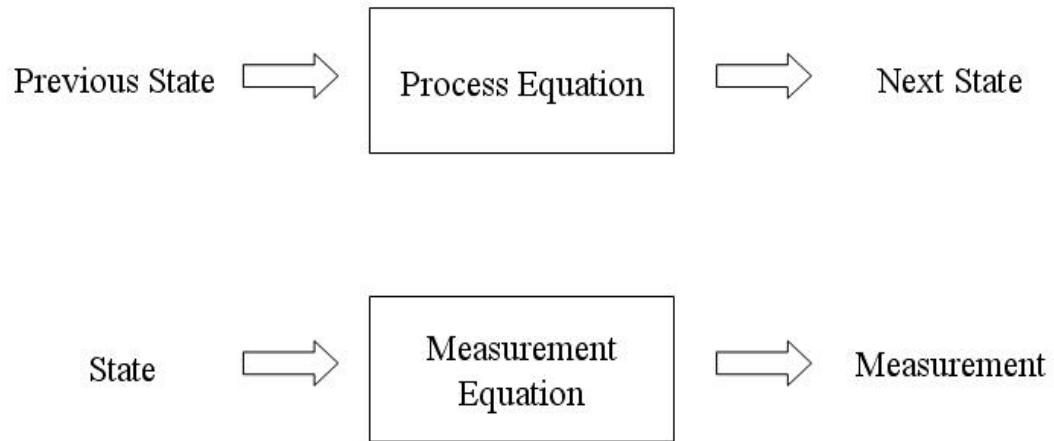
Fig. 4.3 Dynamic system of KF

Time Update (Prediction)

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Initial $\hat{x}_0$ and P_0

$k = k+1$

Measurement Update (Correction)

(1) Compute the Kalman gain

$$K_k = P_k^- H^T \left(HP_k^- H^T + R \right)^{-1}$$

(2) Update estimate with measurement y_k

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = \left(I - K_k H \right) P_k^-$$
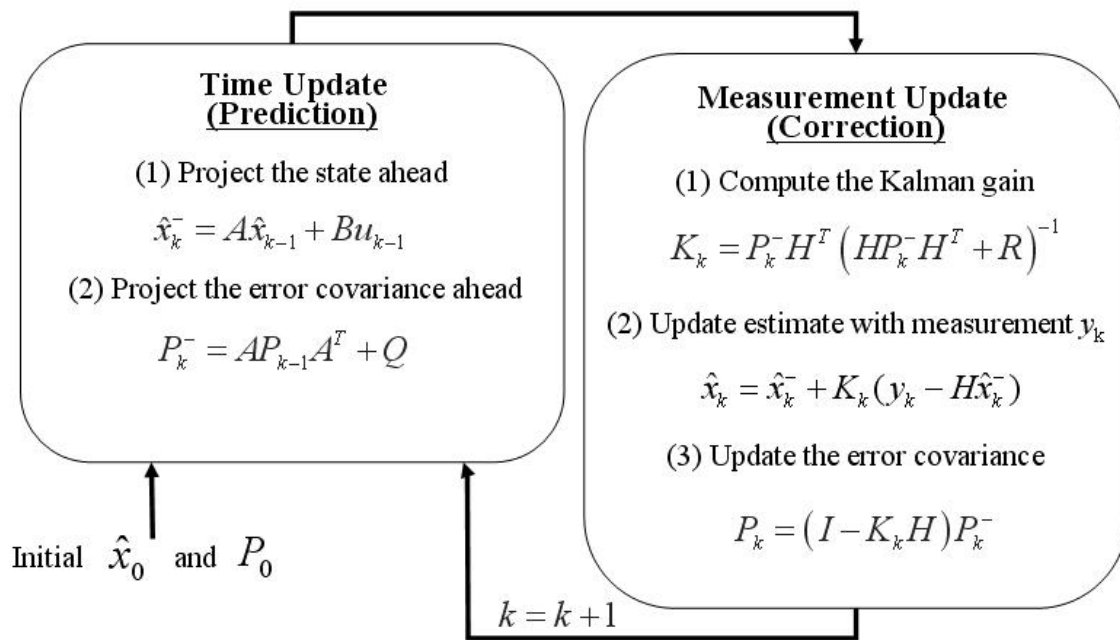
Fig. 4.4 Operation of the KF (Welch and Bishop, 2001)

The operation of the KF is shown in Figure 4.4. Q and R represent the process noise covariance and the measurement noise covariance, respectively. The process noise (v_k in equation 2.8) and the measurement noise (n_k in equation 2.9) are assumed to be white (zero-mean) and Gaussian: $p(v) \sim N(0,Q)$, $p(n) \sim N(0,R)$. With initial values of the previous state $\hat{x}_{k-1}$ and the covariance P_{k-1}, the KF projects the state and error covariance ahead in the time update step. The first task in the second step is the computation of the Kalman gain, which is the one that yields Minimum Mean Square Error (MMSE) estimates. The second task is to update the state by incorporating the measurement (y_k). The updated state ($\hat{x}_k$) is called *a posteriori* state. Correspondingly, the predicted state ($\hat{x}_k^-$) is *a priori* state. The final task in the measurement update is to compute *a posteriori* error covariance.

Note that the KF is a Minimum Mean Square Error (MMSE) estimator. If the error in the posterior stat estimation is $x_k - \hat{x}_k$, then the KF seeks to minimize $E[(x_k - \hat{x}_k)^2]$. This is equivalent to minimize the trace of the posterior error covariance. By minimizing the trace, we can determine the optimal Kalman gain.

In the formulation of the KF, three basic assumptions are used (Maybeck, 1979). First, the system is assumed to be linear, which means that the KF can only be applied to linear problems. Measurement is a linear function of state and the next state is a linear function of previous state. Second, both the process noise and the measurement are white. Whiteness implies that the noise value is not correlated in time. Thus, the knowledge of the current noise does no good for predicting the noise value at other time

intervals. Whiteness also means that the noise has equal power at all frequencies (a term used in power spectral density of signal). Third, the KF assumes that random variables (RV) such as state and noises have Gaussian distributions. The probability of a Gaussian RV has the shape of a normal curve.

The KF has been applied to many fields such as robotics (Wen and Durrant-Whyte, 1992), image processing (Durrant-Whyte et al., 1990), economics (LeRoy and Roger, 1977) and so on. The KF also has had many applications in transportation. For example, Okutani and Stephanedes (1984) used the KF for forecasting short-term freeway traffic flow. Kessaci et al. (1989) presented the KF to estimate traffic-turning movement ratios based on loop detector data. The KF was used to construct an autonomous driving system employed on public roads (Behringer et al., 1992) and improve the accuracy and reliability of an Omege-GPS (Global Positioning System) aircraft navigation system (Schlachta and Studenny, 1990).

4.3.2   Extended Kalman Filter (EKF)

As mentioned above, the KF can be only applied to linear systems to estimate the state of a discrete-time controlled process. However, in many cases, the system dynamics (state and measurement) are nonlinear. The KF is not applicable under such situations. Thus, the development of the EKF is to make the KF applicable to nonlinear systems through linearizing the current mean and covariance. Instead of linear equations 2.8 and 2.9, the process and measurement is now governed by nonlinear equations:

$$x_k = f(x_{k-1}, u_k, v_{k-1}) \qquad\qquad (4.1)$$

$$y_k = h(x_k, n_k) \tag{4.2}$$

where f and h are nonlinear functions. The EKF linearizes the equations using the partial derivatives. For details about partial derivatives, refer to Welch and Bishop (2001).

In the problem of speed estimation, the function f is linear while h is nonlinear. f denotes the relationship between current state and previous state. The following equation defines the state, measurement, and h (Dailey, 1999):

$$x_k = \begin{bmatrix} \overline{s}_k \\ \overline{s}_{k-1} \end{bmatrix}, \ y_k = \begin{bmatrix} O_k / N_k \\ O_{k-1} / N_{k-1} \end{bmatrix}, \ h(x_k) = \frac{L_k}{T} \begin{bmatrix} \dfrac{\sigma_s^2 + \overline{s}_k^2}{\overline{s}_k^3} \\ \dfrac{\sigma_s^2 + \overline{s}_{k-1}^2}{\overline{s}_{k-1}^3} \end{bmatrix} \tag{4.3}$$

It can be seen that the h represents a nonlinear relationship between state and measurement with the consideration of speed variance. The function h is derived from equation 2.5.

Similar to Taylor series in equation 2.11, the EKF linearizes the measurement equation 4.2 about a point x_k^p:

$$h(x_k) = h(x_k^p) + dh(x_k^p)(x_k - x_k^p) \tag{4.4}$$

In the implementation of the EKF, the point x_k^p can be represented by the previous state x_{k-1}. A new linearized measurement equation can be created:

$$\dot{y}_k = \dot{H}x_k + n_k \tag{4.5}$$

where

$$\dot{y}_k = y_k - h(x_k^P) + dh(x_k^P)x_k^P, \quad \dot{H} = dh(x_k^P),$$

and

$$dh(x_k) = \begin{pmatrix} -\dfrac{L_{k-1}}{T}[\dfrac{3\sigma_s^2 + \bar{s}_{k-1}^2}{\bar{s}_{k-1}^4}] & 0 \\ 0 & -\dfrac{L_{k-1}}{T}[\dfrac{3\sigma_s^2 + \bar{s}_{k-2}^2}{\bar{s}_{k-2}^4}] \end{pmatrix}.$$ After linearization, the measurement

equation is linear and can be used in the KF framework.

The basic operation of the EKF is the same as the general linear discrete KF and has

two steps: time update and measurement update. The operation of the EKF is shown in

Figure 4.5. Also, the program of the EKF coded by Matlab is presented in Appendix E.

**Time Update
(Prediction)**

(1) Project the state ahead

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + W_k Q_{k-1}W_k^T$$

Initial $\hat{x}_0$ and P_0

$$k = k+1$$

**Measurement Update
(Correction)**

(1) Compute the Kalman gain

$$K_k = P_k^- H^T (H P_k^- H^T + V_k R V_k^T)^{-1}$$

(2) Update estimate with measurement y_k

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-, 0))$$

(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$
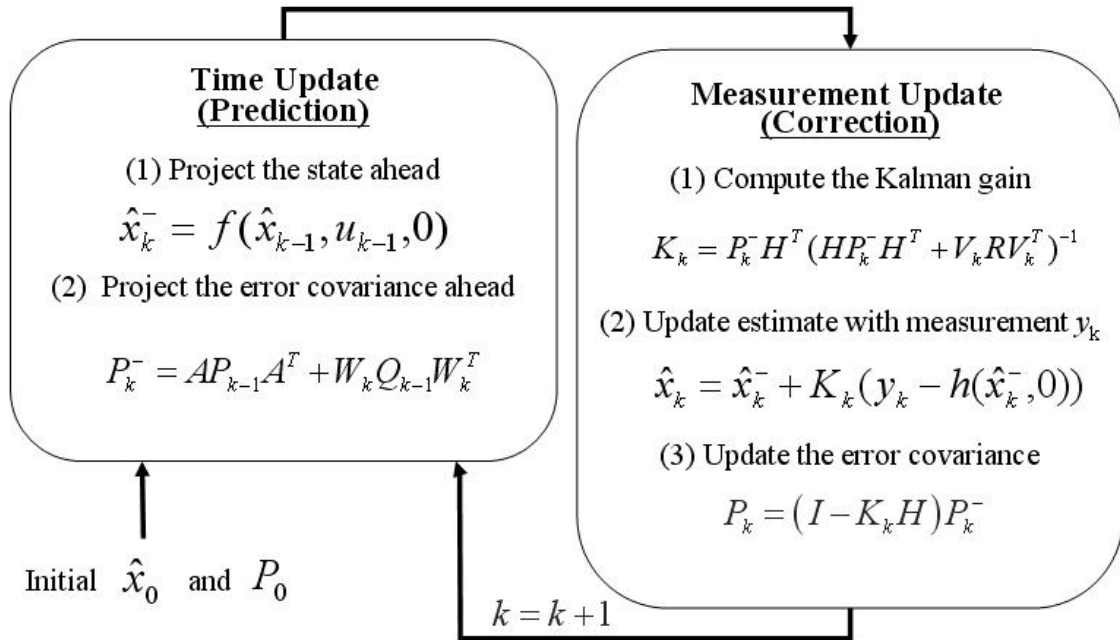
Fig. 4.5 Operation of the EKF (Welch and Bishop, 2001)

4.3.3 Unscented Kalman Filter (UKF)

As mentioned in section 2.4.4, the EKF has its shortcomings in dealing with nonlinear systems. To overcome its weaknesses, a new approach of the UKF is presented for speed estimation. The UKF still assumes that the state distribution is Gaussian; however, instead of linearizing the nonlinear system, it uses a minimal set of deterministically chosen sigma points that can completely capture the true mean and covariance of the state. When propagated through the true nonlinear system, the sigma points can capture the posterior mean and covariance accurately (Julier and Uhlmann, 1997).

The unscented transformation (UT) is the fundamental part of the UKF. It is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. The UT builds on the principle that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation (Julier and Uhlmann, 1997). Let x be a d-dimensional random variable with mean $\bar{x}$ and covariance P_x. x is propagated through a nonlinear function

$$y = g(x).$$
<div align="right">(4. 6)</div>

To calculate the statistics of y, a set of $2d + 1$ weighted points (or sigma points) are deterministically selected so that their sample mean and sample covariance are $\bar{x}$ and P_x. The sigma points are chosen by the following equation.

$$\chi_0 = \overline{x} \qquad\qquad w_0 = \frac{\kappa}{d+\kappa} \qquad\qquad i = 0$$

$$\chi_i = \overline{x} + (\sqrt{(d+\kappa)P_x})_i \quad w_i = \frac{1}{2(d+\kappa)} \qquad i = 1,...,d \qquad\qquad (4.7)$$

$$\chi_i = \overline{x} + (\sqrt{(d+\kappa)P_x})_i \quad w_i = \frac{1}{2(d+\kappa)} \qquad i = d+1,...,2d$$

where

κ = provides an extra degree of freedom to fine-tune the higher order moments of the approximation;

$\sqrt{(d+\kappa)}$ = a scaling factor that determines the spread of sigma points around $\overline{x}$;

$(\sqrt{(d+\kappa)P_x})_i$ = the ith column of the matrix square root of $(d+\kappa)P_x$; and

w_i = the weight which is associated with the ith point.

The Cholesky factorization method (Press et al., 1992) can be used to calculate the matrix square root. Once the sigma points are selected, they are propagated through the nonlinear function to yield the set of transformed sigma points

$$\mathsf{y}_i = g(\chi_i) \quad i = 0,1,...,2d . \qquad\qquad (4.8)$$

Then, the approximated mean, covariance and cross-variance of y can be calculated. The transformation procedure is as follows (Julier and Uhlmann, 1997):

1) The mean is calculated by the weighted average of the transformed sigma points,

$$\overline{y} \approx \sum_{i=0}^{2d} w_i^{(m)} \mathsf{y}_i \qquad\qquad (4.9)$$

2) The covariance and cross-covariance are given by the weighted outer product of the sigma points and/or transformed sigma points,

$$P_y \approx \sum_{i=0}^{2d}\sum_{j=0}^{2d} w_{ij}^{(c)}(\mathbf{y}_i - \overline{y})(\mathbf{y}_i - \overline{y})^T \tag{4.10}$$

$$P_{xy} \approx \sum_{i=0}^{2d}\sum_{j=0}^{2d} w_{ij}^{(c)}(\chi_i - \overline{x})(\mathbf{y}_i - \overline{y})^T \tag{4.11}$$

where $w_i^{(m)}$ and $w_{ij}^{(c)}$ are scalar weights of mean and covariance respectively. All weights should be equal or greater than zero. Figure 4.6 provides a schematic diagram of the unscented transformation, where $\gamma = \sqrt{(d+\kappa)}$.
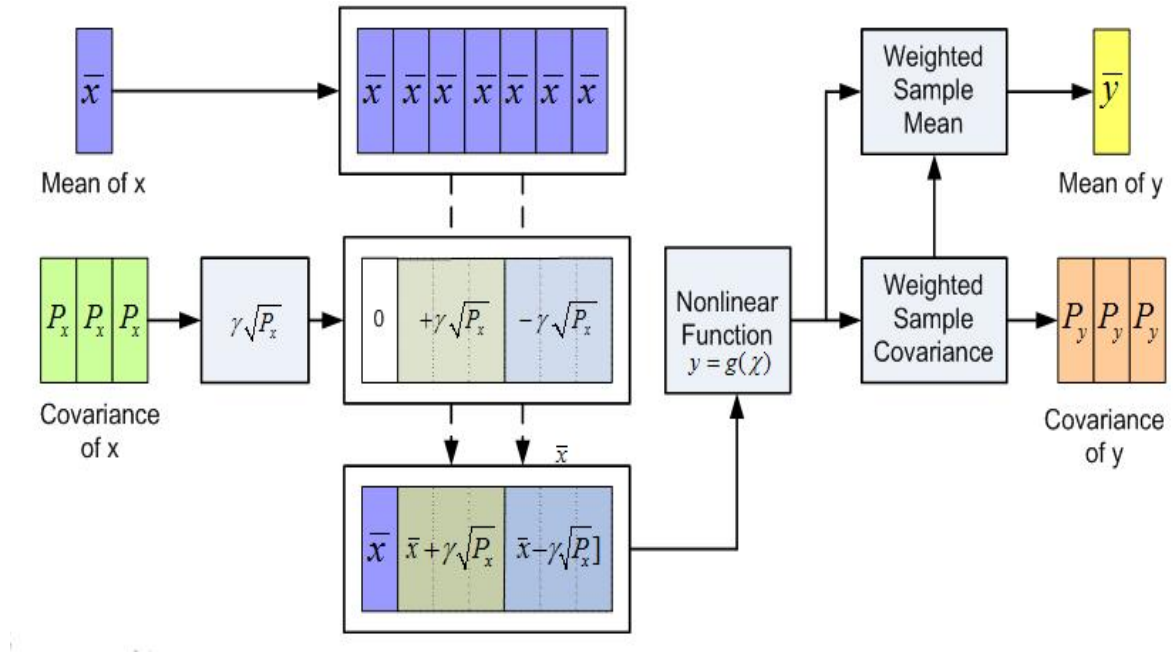
Fig. 4.6 Unscented transformation of the UKF

To demonstrate the difference between the sigma point approach and linearization, Merwe et al. (2004) drew 5000 samples from a known Gaussian prior and propagated the samples through a nonlinear function. The result of the posterior sample mean and covariance are shown in Figure 4.7. In this example, the dimension of the random variable x is 2. Thus, only 5 sigma points were used for capturing sample mean and covariance.
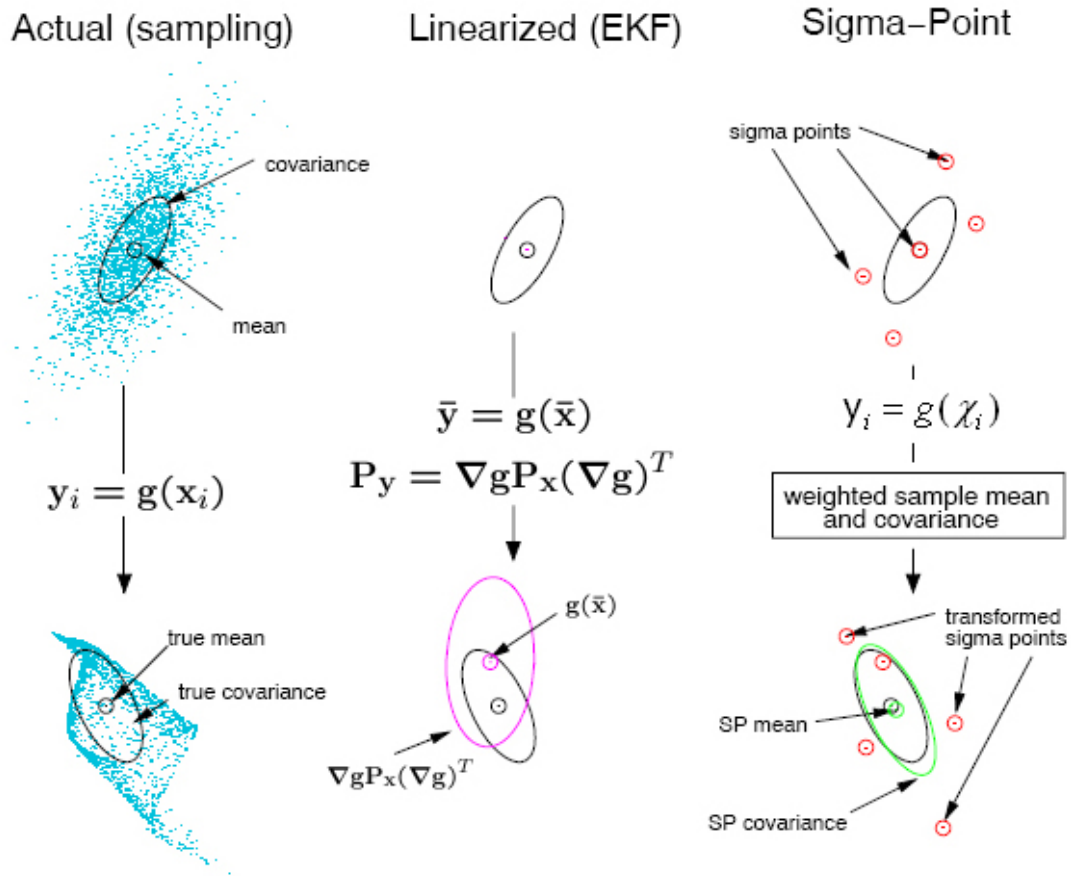
Fig. 4.7 Comparison of sigma point approach and linearization (Merwe et al, 2004)

The left plot shows the true statistics as calculated by a Monte Carlo method. The statistics of the posterior random variable by a linearization approach as used in the EKF is shown in the middle plot. The errors of the mean and covariance by this approach are visible. The right plot shows the results from the sigma-point approach. It can be seen that there is almost no bias error in both the sample mean and the covariance. The superiority of the sigma-point approach is clearly demonstrated.

One property of the selection of sigma points in the UT is that with the increase of the dimension of the state space, the radius of the sphere that bounds all the sigma points increases. Under such situations, the sigma points are possible to sample non-local effects, although they still capture the sample mean and covariance correctly (Merwe et al, 2004). In order to address the problem, the sigma points can be scaled away or from the mean of the prior distribution by a proper choice of κ:

$\kappa = 0$: The distance of the ith sigma point from $\overline{x}$ and $\left| \chi_i - x \right|$ is proportional to $\sqrt{d}$;

$\kappa > 0$: The sigma points are scaled away $\overline{x}$;

$\kappa < 0$: The sigma points are scaled towards $\overline{x}$.

A Scaled Unscented Transformation (SUT) was developed to solve this problem (Julier, 2002). The SUT replaces the original set of sigma points with a transformed set by

$$\chi_i^{'} = \chi_0 + \alpha(\chi_i - \chi_0) \quad i = 0,...,2d , \tag{4.12}$$

where a is a positive scaling parameter to minimize possible higher order effects. The weights of sigma points can be correspondingly transformed by

$$w_i^{'} = \{ \begin{array}{ll} w_0/\alpha^2 + (1-1/\alpha^2) & i = 0 \\ w_i/\alpha^2 & i = 1,...,2d \end{array} \tag{4.13}$$

By setting

$$\lambda = \alpha^2(d+\kappa) - d , \tag{4.14}$$

the sigma point selection and scaling can be combined into one step:

$$\begin{array}{lll} \chi_0 = \overline{x} & w_0^{(m)} = \dfrac{\lambda}{d+\kappa} & i = 0 \\[2mm] \chi_i = \overline{x} + (\sqrt{(d+\kappa)P_x})_i \ \ i = 1,...,d & w_0^{(c)} = \dfrac{\lambda}{d+\kappa} + (1-\alpha^2+\beta) & i = 0 \\[2mm] \chi_i = \overline{x} + (\sqrt{(d+\kappa)P_x})_i \ \ i = d+1,...,2d & w_i^{(m)} = w_i^{(c)} = \dfrac{1}{2(d+\kappa)} & i = 1,...,2d \end{array} \tag{4.15}$$

where β is a parameter to incorporate prior knowledge of the distribution of x, $w_i^{(m)}$ represents the weight of mean, and $w_i^{(c)}$ denotes the weight of covariance of the ith sigma point.

The operation of the UKF is shown in Figure 4.8. In addition to the selection of sigma points, the UKF is similar to the KF and has the time update and measurement update steps. Again, the time update projects the state and the error covariance ahead; the operations in the measurement update state include computing the Kalman gain, updating the estimate of state with the consideration of current measurement, and updating the error covariance to obtain the posterior estimate.
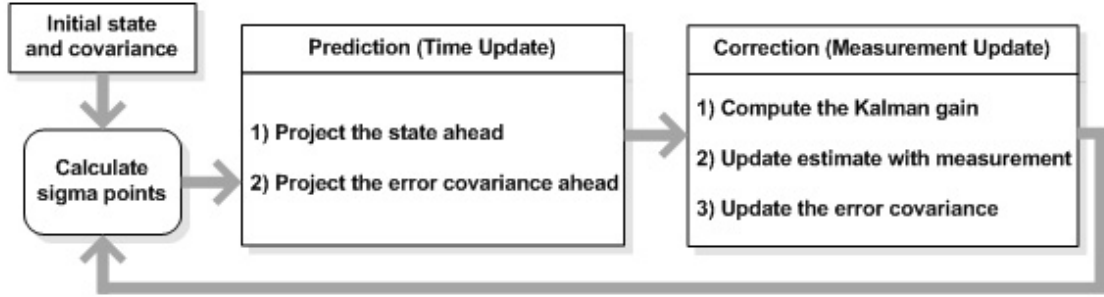
Fig. 4. 8 Operation of the UKF

4.4 IMPLEMENTATION OF THE UKF

From Figure 4.8, it can be seen that the UKF directly applies the UT (or SUT) to the recursive KF framework. In the implementation of the UKF, the state random variance is redefined as

$$x_k^\alpha = \begin{bmatrix} x_k^x \\ x_k^v \\ x_k^n \end{bmatrix} = \begin{bmatrix} x_k \\ v_k \\ n_k \end{bmatrix},$$
(4.16)

where x_k is the original state, v_k is the process noise, n_k is the measurement noise. The sigma-point selection scheme in equation 4.15 is applied to x_k^α to calculate the corresponding sigma points $\{\chi_{k,i}^\alpha;\ i=0,...,2d\}$, where $\chi_{k,i}^\alpha \in \Re^{d_x+d_v+d_n}$, d_x is the original state dimension, d_v is the process noise dimension, and d_n is the measurement noise dimension. Similarly, the state covariance is established by the individual covariance of x, v, and n:

$$P^a = \begin{pmatrix} P_x & 0 & 0 \\ 0 & Q_v & 0 \\ 0 & 0 & R_n \end{pmatrix},$$

(4.17)

where Q_v is the process noise covariance and R_n is the measurement noise covariance. Thus, by incorporating the state space with the noise random variables, the effects of the noises on the system dynamics and observations can be captured with the same level of accuracy as the state.

The complete algorithm of the UKF is shown as follows (Julier and Uhlmann, 1997):

- Initialization.

$$\overline{x}_0 = E[x_0], \quad P_{x_0} = E[(x_0 - \overline{x}_0)(x_0 - \overline{x}_0)^T]$$

$$\overline{x}_0^\alpha = E[x_0^\alpha] = E[\overline{x}_0 \quad 0 \quad 0]^T$$

$$P_0^\alpha = E[(x_0^\alpha - \overline{x}_0^\alpha)(x_0^\alpha - \overline{x}_0^\alpha)^T] = \begin{pmatrix} P_{x_0} & 0 & 0 \\ 0 & Q_v & 0 \\ 0 & 0 & R_n \end{pmatrix}$$

where $x_0^\alpha = [x_0 \quad v_0 \quad n_0]^T$, v denotes the process noise variable, and u is the measurement noise variable.

- For time intervals $k = 1, ..., \infty$

  1) Calculation of sigma points.

  $$\chi_{k-1}^\alpha = [\overline{x}_{k-1}^\alpha \quad \overline{x}_{k-1}^\alpha + \gamma\sqrt{P_{k-1}^\alpha} \quad \overline{x}_{k-1}^\alpha - \gamma\sqrt{P_{k-1}^\alpha}]$$

  (4.18)

  2) Time update.

  $$\chi_{k|k-1}^x = f(\chi_{k-1}^x, \chi_{k-1}^v)$$

  (4.19)

$$\bar{x}_k^- = \sum_{i=0}^{2d} w_i^{(m)} \chi_{i,k|k-1}^x \tag{4.20}$$

$$P_{x_k}^- = \sum_{i=0}^{2d} w_i^{(c)} (\chi_{i,k|k-1}^x - \bar{x}_k^-)(\chi_{i,k|k-1}^x - \bar{x}_k^-)^T \tag{4.21}$$

where $\chi_k^\alpha = [(\chi^x)^T \ (\chi^v)^T \ (\chi^n)^T]^T$, $w_i^{(m)}$ is again the weight of mean and $w_i^{(c)}$ denotes

the weight of covariance for the ith sigma point.

    3)   Measurement update.

$$Y_{k|k-1} = h(\chi_{k|k-1}^x, \chi_{k-1}^n) \tag{4. 22}$$

$$\bar{y}_k^- = \sum_{i=0}^{2d} w_i^{(m)} Y_{i,k|k-1} \tag{4. 23}$$

$$P_{\bar{y}_k} = \sum_{i=0}^{2d} w_i^{(c)} (Y_{i,k|k-1} - \bar{y}_k^-)(Y_{i,k|k-1} - \bar{y}_k^-)^T \tag{4. 24}$$

$$P_{x_k y_k} = \sum_{i=0}^{2d} w_i^{(c)} (\chi_{i,k|k-1}^x - \bar{x}_k^-)(Y_{i,k|k-1} - \bar{y}_k^-)^T \tag{4. 25}$$

$$K_k = P_{x_k y_k} P_{\bar{y}_k}^{-1} \tag{4. 26}$$

$$\bar{x}_k = \bar{x}_k^- + K_k(y_k - \bar{y}_k^-) \tag{4. 27}$$

$$P_{x_k} = P_{x_k}^- - K_k P_{\bar{y}_k} K_k^T \tag{4. 28}$$

where y_k is the measurement, h is the function described in Equation 2.2 denoting the

relationship between observations and states, and K_k is the Kalman gain.

The algorithm of the UKF is coded in Matlab. The realization of this algorithm can

be seen in Appendix E.

The implementation of the UKF requires similar initial information as the EKF.

Firstly, a fixed MEVL needs to be preset. This value is also required for most speed

estimation methods. In a practical application, the MEVL can be obtained from

historical vehicle classification. For example, vehicle length data from TMS was used for estimating the MEVL (Dailey, 1999). Secondly, both the UKF and the EKF require the input of process noise and measurement noise. In the EKF, both values are experimentally determined and fixed throughout the whole estimation process (Dailey, 1999). In the UKF, speed variance is used for modeling process noise. Since speed variance data are not available from single loop outputs, a constant value of speed variance is experimentally determined and used. The measurement noise in the UKF is represented by the variance of observations ($\sigma_{O/N}^2$). It can be recursively determined by the variance of measurements based on last noise value ($\sigma_{O_{k-1}/N_{k-1}}^2$) and the current measurement (O_k/N_k). Thus, it is an easy and efficient way to account for the measurement noise. Finally, as mentioned in Section 2.4.4, a state-transition model is needed for predicting the new state based on the previous state. In the EKF, the current speed ($\overline{s}_k$) is predicted by the two previous weighted speeds ($\overline{s}_{k-1}$ and $\overline{s}_{k-2}$) (equation 2.12). The weights of these two previous states are derived by using the AR method. However, it is very simple to model the state-transition in the UKF because equal weights of $\overline{s}_{k-1}$ and $\overline{s}_{k-2}$ are used. Therefore, the implementation of the UKF is actually easier than the EKF.

4.5   ESTIMATION RESULTS AND DISCUSSION

In this section, ADR-6000 detector data and dual-loop data described in Chapter III were used for speed estimation. Both the EKF and the UKF were implemented to the datasets

using the developed Matlab programs. To evaluate the accuracy of estimates, Measures of Effectiveness (MOEs) are selected. The MOEs include Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The MAE and RMSE are calculated by

$$MAE = \frac{1}{M} \sum_{k=1}^{M} \left| \overline{s}_k - s_k \right| \tag{4.29}$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{k=1}^{M} [\overline{s}_k - s_k]^2} \tag{4.30}$$

where M is the total number of time intervals, $\overline{s}_k$ is the estimated speed of the kth time interval, and s_k is the observed speed of the kth time interval.

From the equations, it can be seen that the MAE is used to measure the average magnitude of absolute errors. The MAE is a linear score that puts equal weights to all the individual differences ($\left| \overline{s}_k - s_k \right|$).The RMSE is the square root of Mean Square Error (MSE), which can capture both the variance of errors and the bias of estimates. It gives relatively large weights to large errors because the errors are squared before averaged. Both MOEs are negatively-oriented scores, that is, lower values are better.

Peek ADR-6000 data from the SH6 vehicle detection test bed were first used for speed estimation. As mentioned in Chapter III, traffic flow at this location is low to medium. Traffic congestion usually did not exist except for special situations. One-day of data (January 27th, 2004) were collected and complied into time intervals of 30 seconds. The results of speed estimation from the UKF are shown in Figures 4.9 and 4.10 for lanes 1 and 2, respectively.
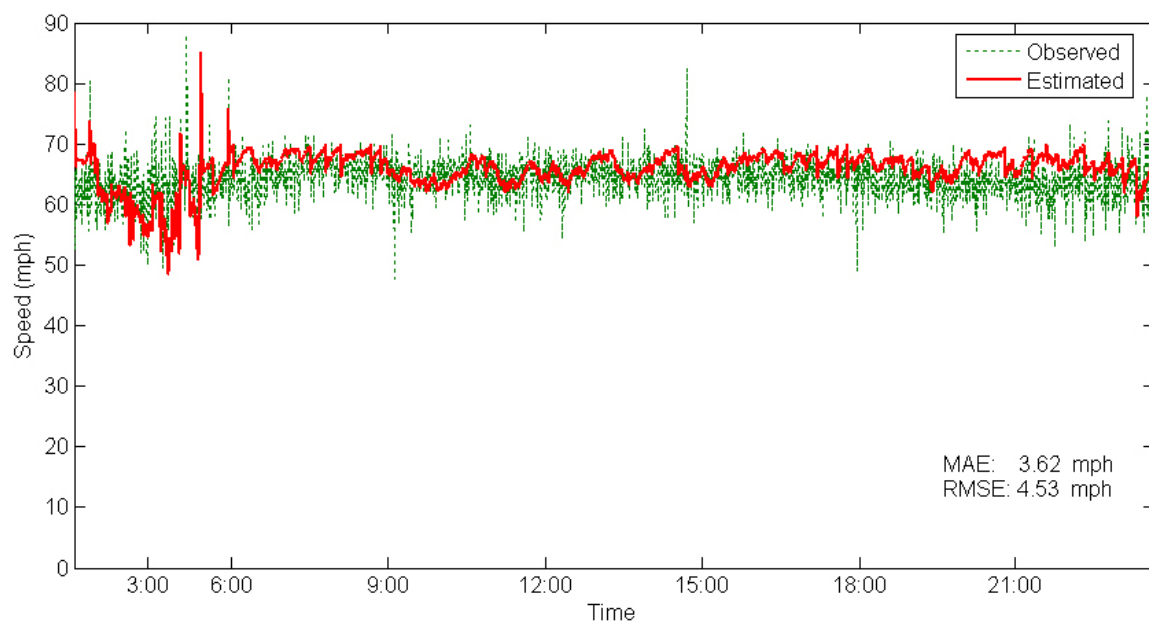
Fig. 4.9 Speed estimation results from the UKF at SH6 on Jan. 27, 2004 (lane 1)
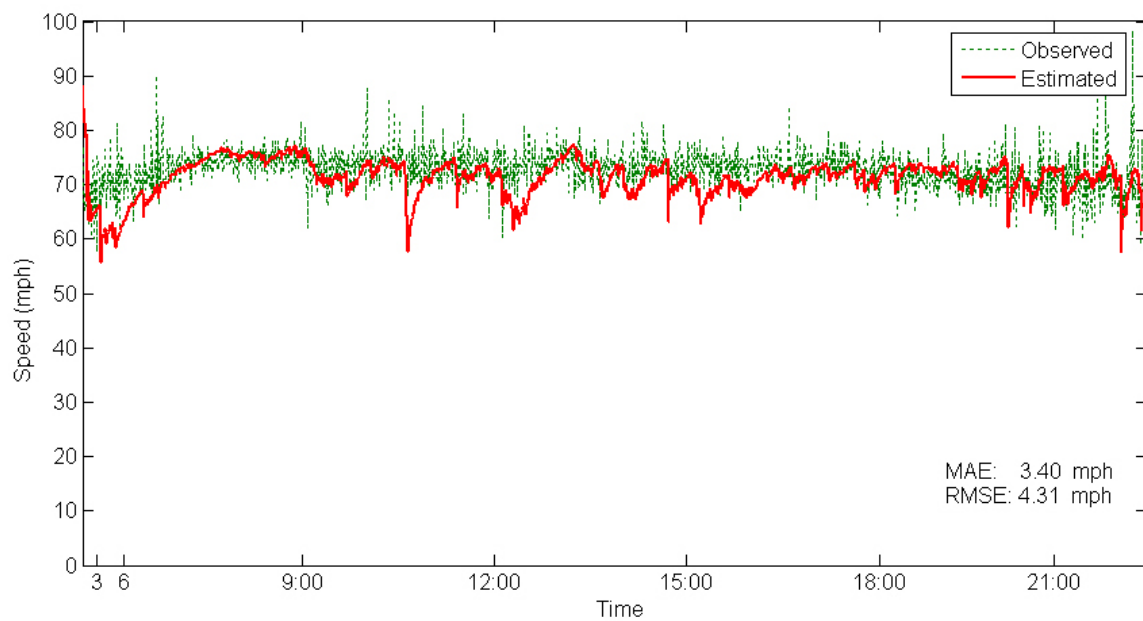
Fig. 4.10 Speed estimation results from the UKF at SH6 on Jan. 27, 2004 (lane 2)

From the comparison of the estimated and observed speeds, it can be seen that the UKF had good performance for speed estimation under low-medium traffic flow conditions. In speed estimation, those time intervals that did not have any vehicle passing by were discarded. Thus, the number of time intervals in an hour varies with time. By comparing lane 1 and lane 2, we can see that the number of polling intervals of lane 1 is larger than that of lane 2. This is because lane 1 (shoulder lane) has higher traffic volumes and for some intervals traffic existed only in lane 1. Along with higher truck percentages distributed in lane 1, estimation results of this lane are less accurate than those of lane 2. The MAE and RMSE values of lane 2 are 3.40 mph and 4.31 mph while the values for lane 1 are 3.62 and 4.53, respectively.

The EKF and the *g*-estimator method were also implemented to the lane 1 dataset and estimation results are shown in Figures 4.11 and 4.12. From the figures, it is evident that the UKF had better performance than the EKF and the g-estimator method; the g-estimator had the worst performance. The EKF had better estimates during daytime than those during night time. The g-estimator, however, generated large variation of speeds as shown by a wide band of speed estimates in Figure 4.11. As expected, this method did not have good performance for speed estimation even under normal traffic conditions. For this reason, the g-estimator will not be used for further comparison.
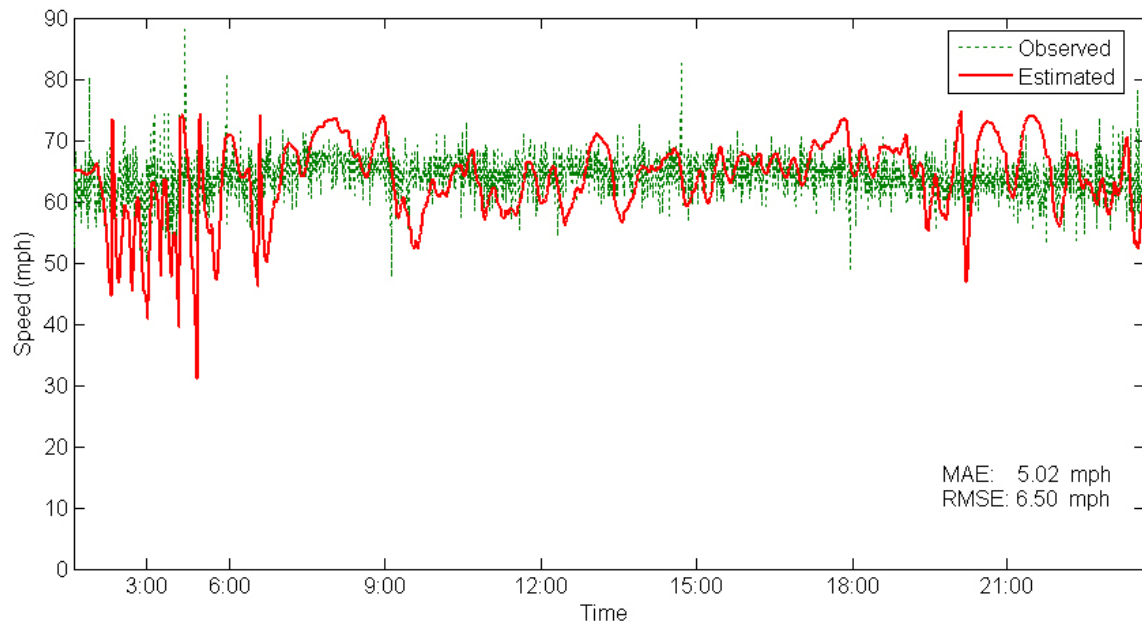
Fig. 4.11 Speed estimation results from the EKF at SH6 on Jan. 27, 2004 (lane 1)
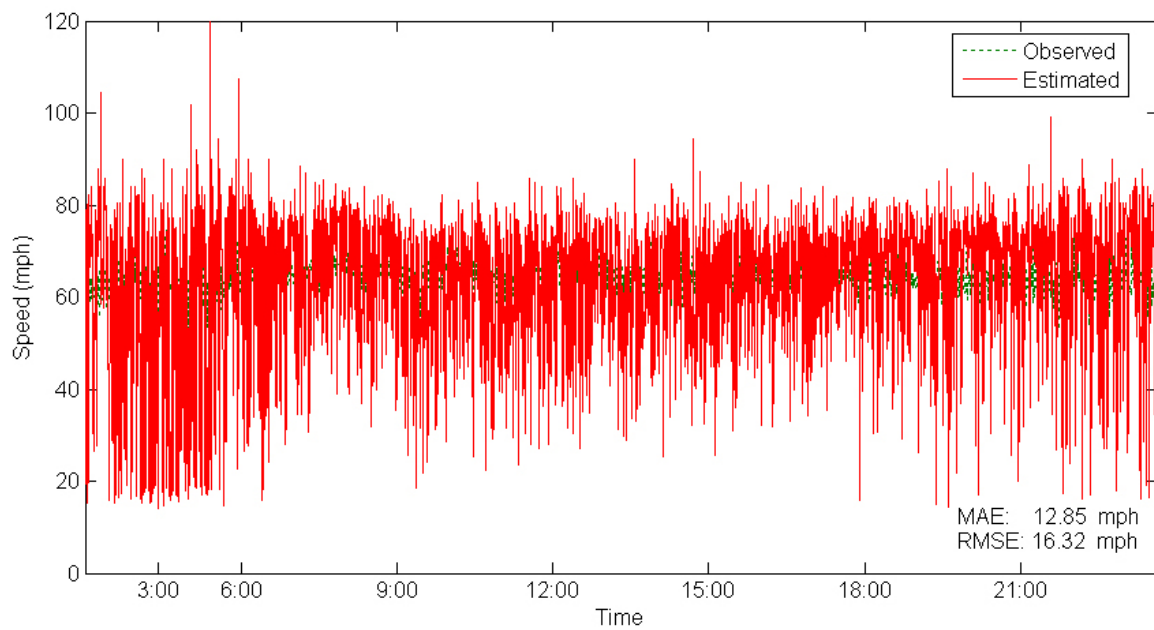
Fig. 4.12 Speed estimation results from the *g* method at SH6 on Jan. 27, 2004 (lane 2)

In the following, ADR-6000 data collected from the IH-35 test bed were used for speed estimation. Two weekday datasets were collected on October 27 and November 9, 2004 from the 4 southbound through lanes. Estimation results are shown in Figures 4.13-20. It can be seen that the UKF generated accurate and stable estimates for all datasets. From the MOEs, it is found that the estimates became more accurate from lane 1 to lane 4. For example, the MAE and RMSE are 3.74 mph and 5.13 mph of lane 1 on Oct. 27 and they decrease to only 2.70 mph and 3.62 mph of lane 4. This again can be explained by truck percentages and traffic volumes. Lanes 1 and 2 had serious traffic congestion during AM/PM peak hours so that vehicle speeds dropped down and sometimes were lower than 10 mph. Even under low speed conditions, the estimated speeds still captured real world speed variations very well. During the first few hours, estimated speeds had relatively large variations even though traffic flow was low. This was caused by high percentage of trucks. The MEVL had large variations during this time period, which can be as short as a passenger car's length and as long as a multi-trailer truck's length. The influence of MEVL during night time can be also identified from the literature. In some studies, speed estimation during night time was not even considered, which is reasonable since traffic flow analysis during daytime is more important for traffic control and management.
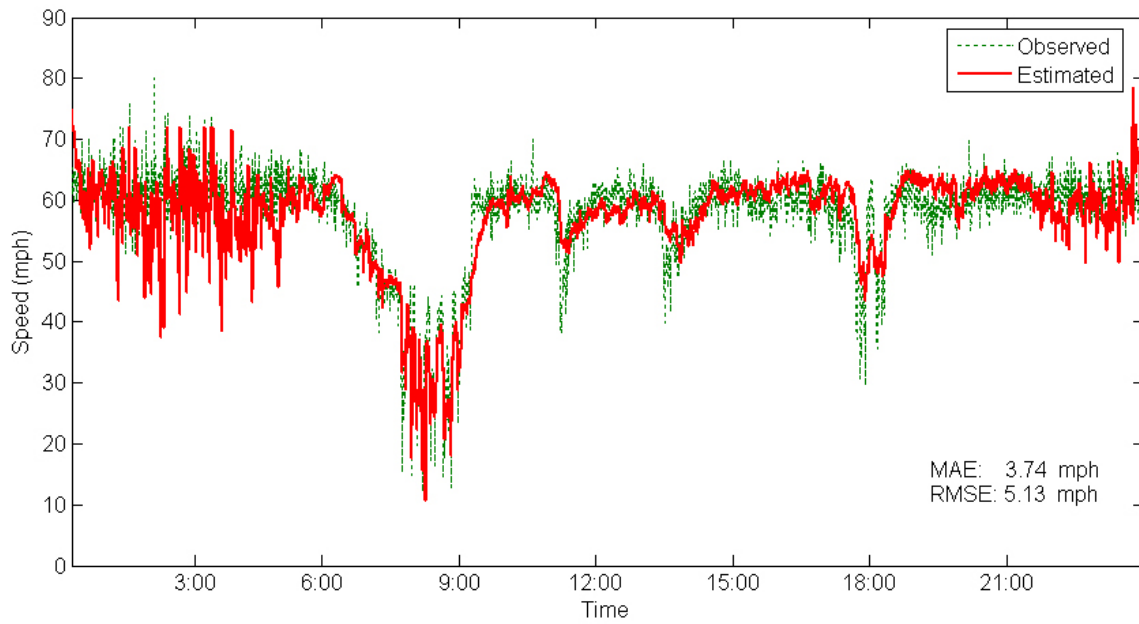
Fig. 4.13 Speed estimation results from the UKF at IH-35, Austin, on Oct. 27, 2004 (lane 1)
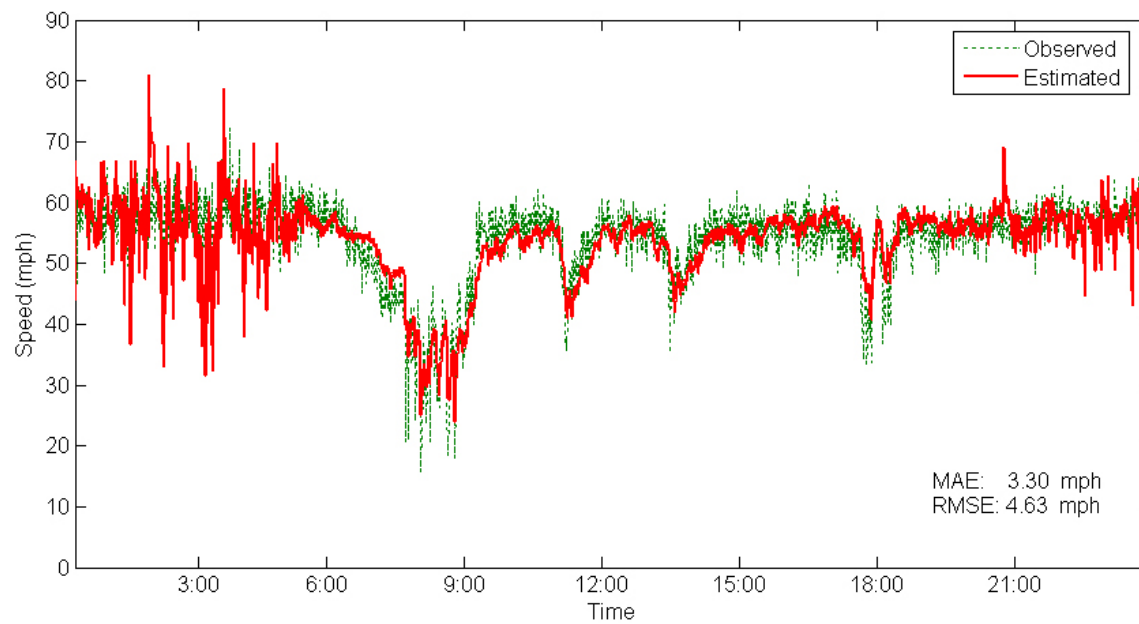
Fig. 4.14 Speed estimation results from the UKF at IH-35, Austin, on Oct. 27, 2004 (lane 2)
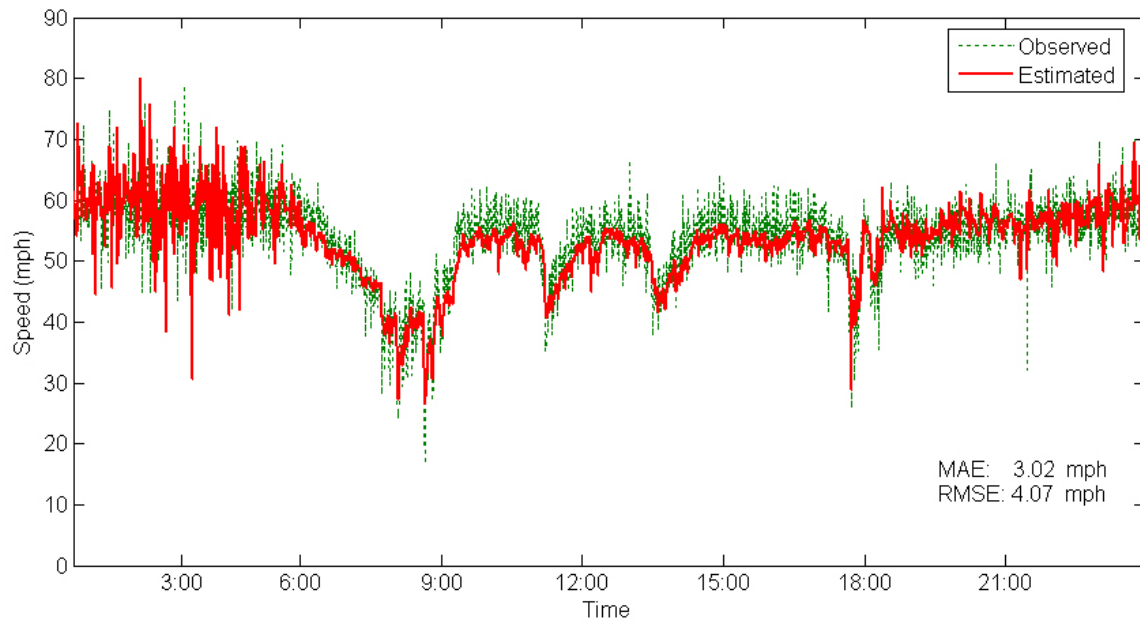
Fig. 4.15 Speed estimation results from the UKF at IH-35, Austin, on Oct. 27, 2004
(lane 3)
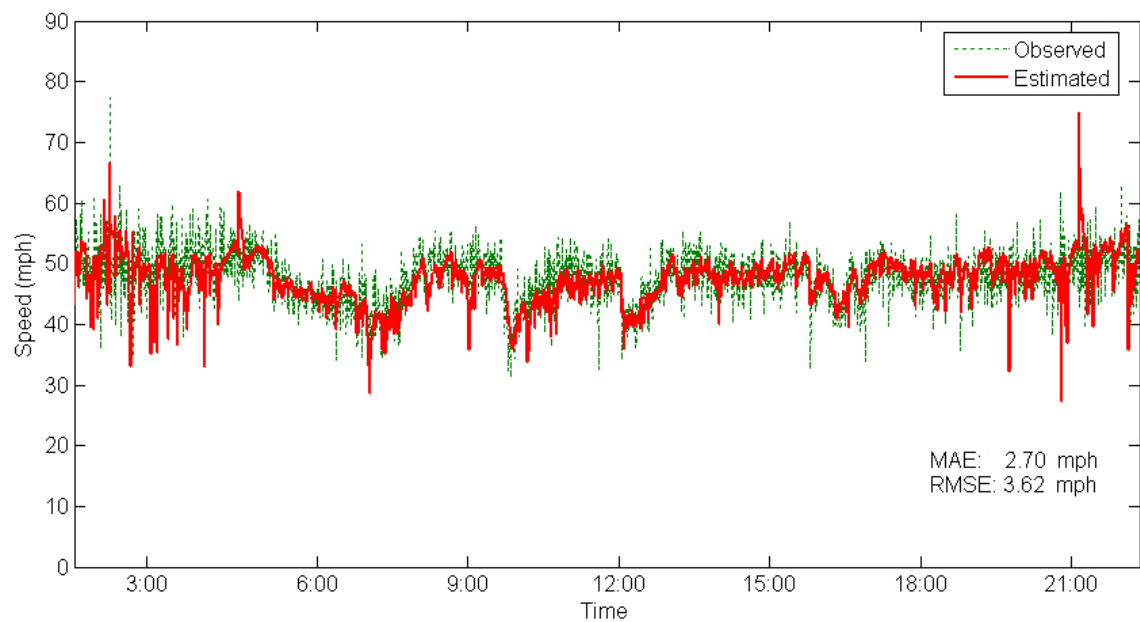
Fig. 4.16 Speed estimation results from the UKF at IH-35, Austin, on Oct. 27, 2004
(lane 4)
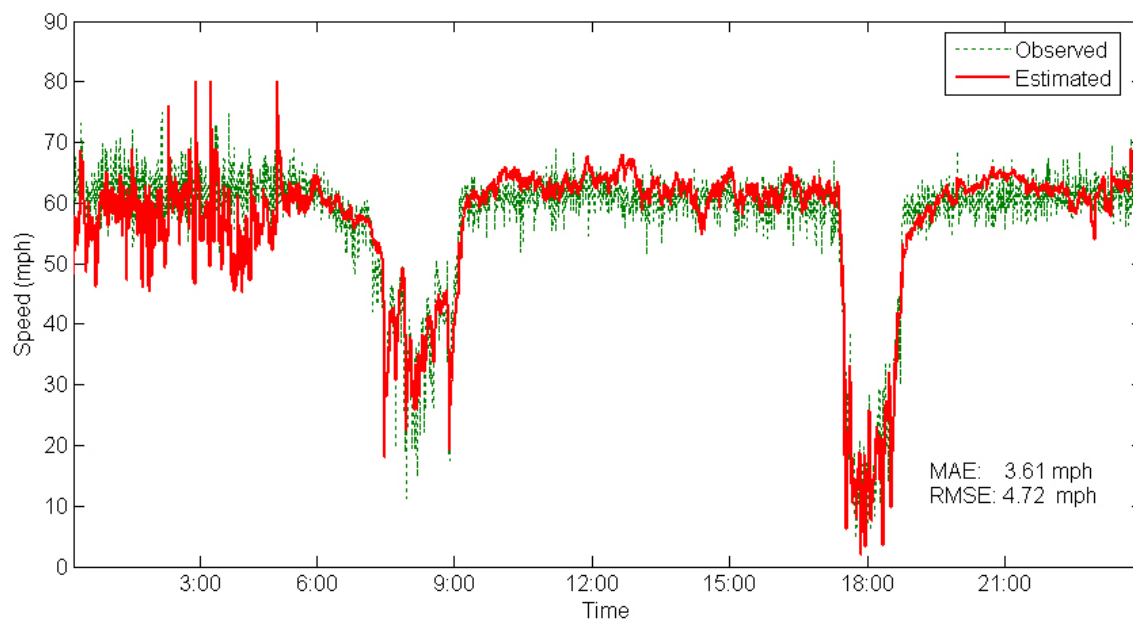
Fig. 4.17 Speed estimation results from the UKF at IH-35, Austin, on Nov. 9, 2004 (lane 1)
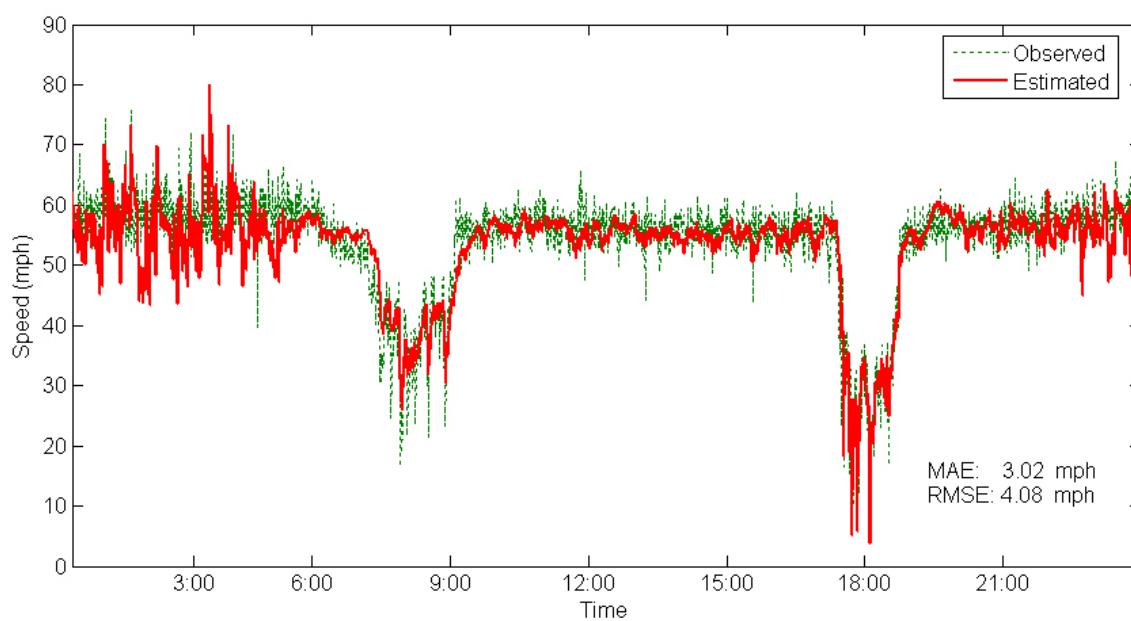
Fig. 4.18 Speed estimation results from the UKF at IH-35, Austin, on Nov. 9, 2004 (lane 2)
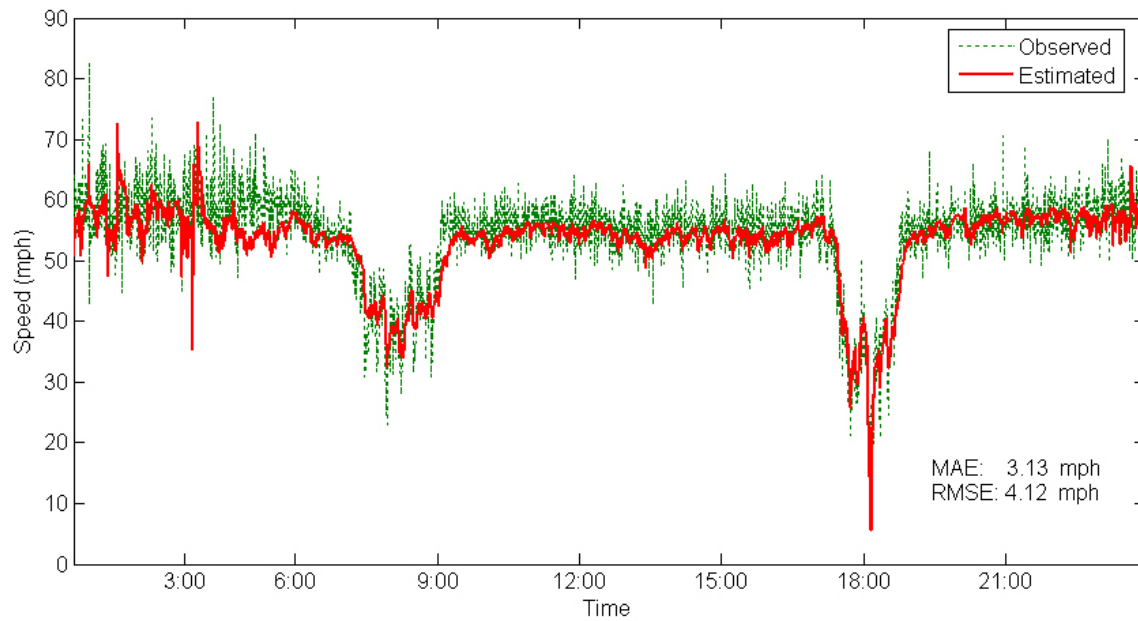
Fig. 4.19 Speed estimation results from the UKF at IH-35, Austin, on Nov. 9, 2004 (lane 3)
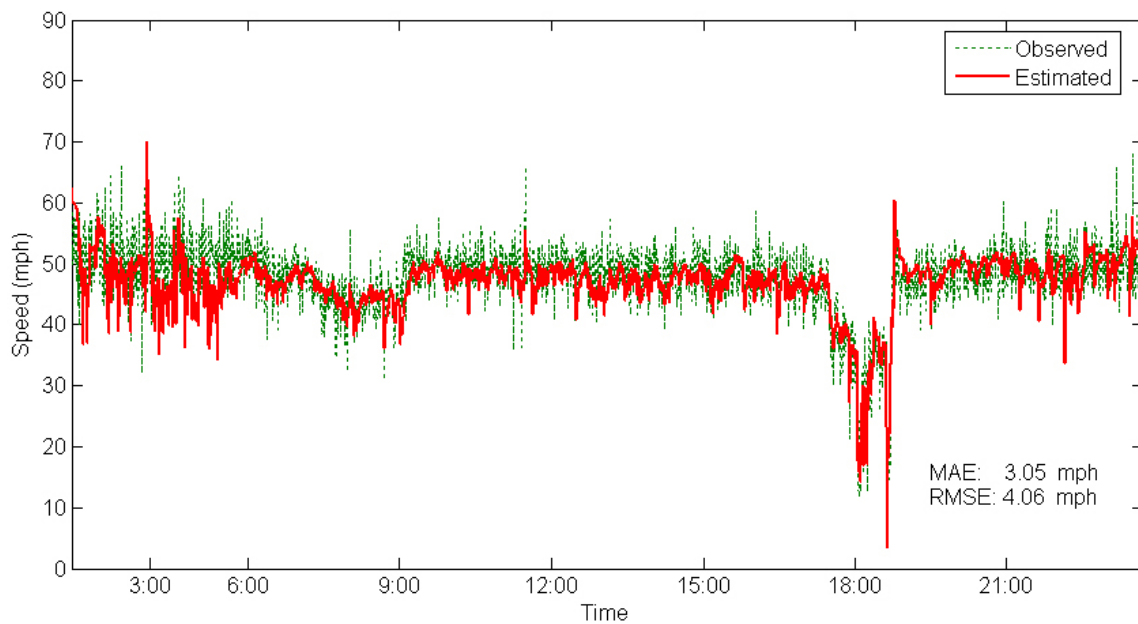
Fig. 4.20 Speed estimation results from the UKF at IH-35, Austin, on Nov. 9, 2004 (lane 4)

Peek ADR-6000 data enable the research to test speed estimation algorithms under different polling times. The lane 4 dataset collected on Oct.27 was also compiled into 20s and 1min time intervals. The UKF method was implemented and the results are shown in Figure 4.21. It is found that with larger duration of time intervals, the MAE and RMSE were lower, but the UKF could still maintain good estimations when the time interval is as small as 20 seconds. Obviously, the UKF method can be applied to single loop detectors that output data with different polling intervals.

Finally, one week of dual-loop detector data collected from IH-35 in San Antonio were used for speed estimation. The estimation results from the UKF are shown in Figure 4.22. This dataset enables us to observe the performance of the UKF throughout multiple days. The results show that the UKF generated accurate estimates of speed. The estimated speeds followed the measured speeds very well. A closer look into the dual-loop dataset found that measured speeds sometimes were very high during nighttime. The highest measured average speed during a time interval is 150 mph with two vehicles detected. Such high speed data might not have been accurately measured. However, such erroneous data did not evidently affect UKF speed estimation results.
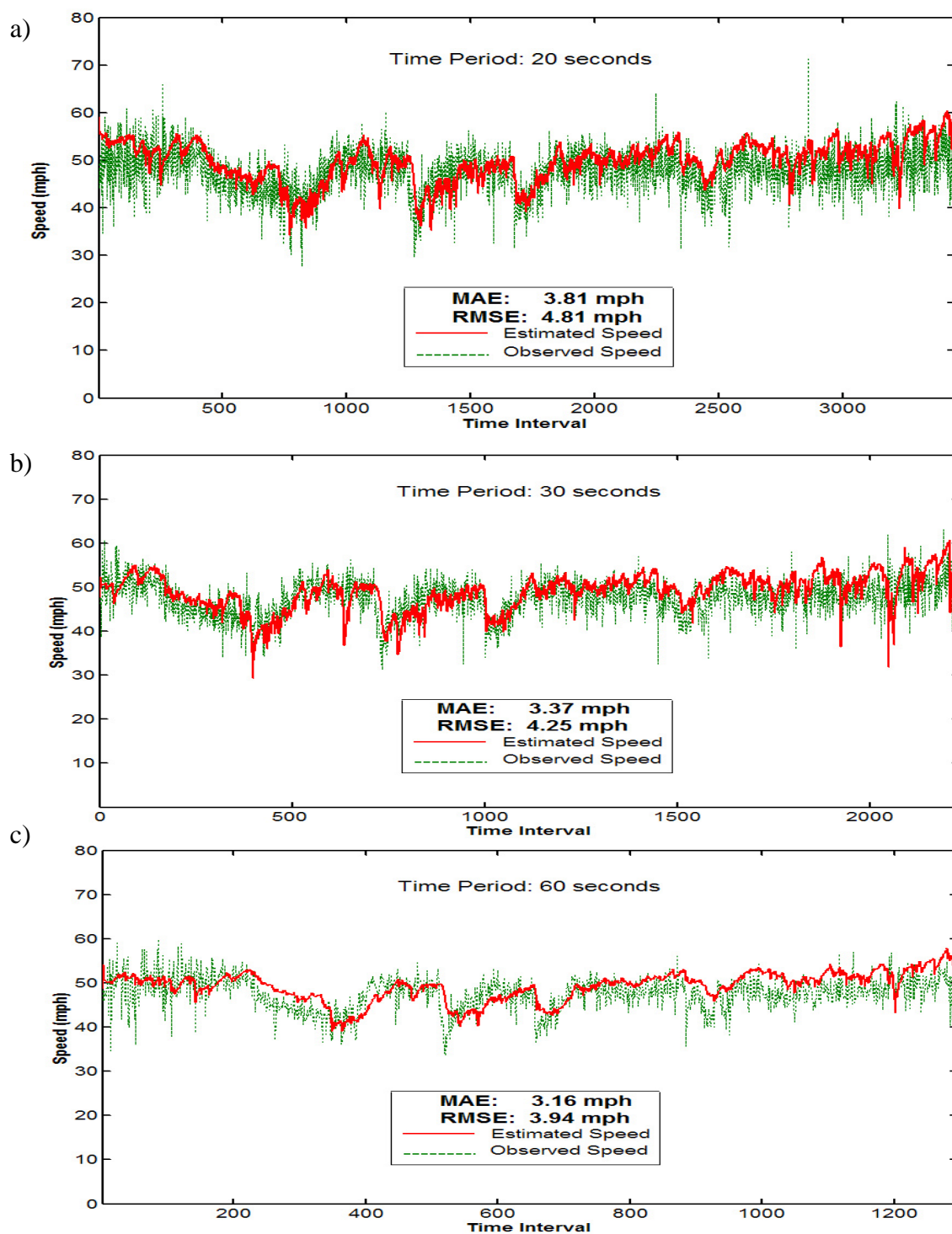
Fig. 4.21 Estimated and observed speeds (lane 4, IH-35, Austin, Oct.27, 2004)
a) 20s time interval b) 30s time interval c) 60s time interval
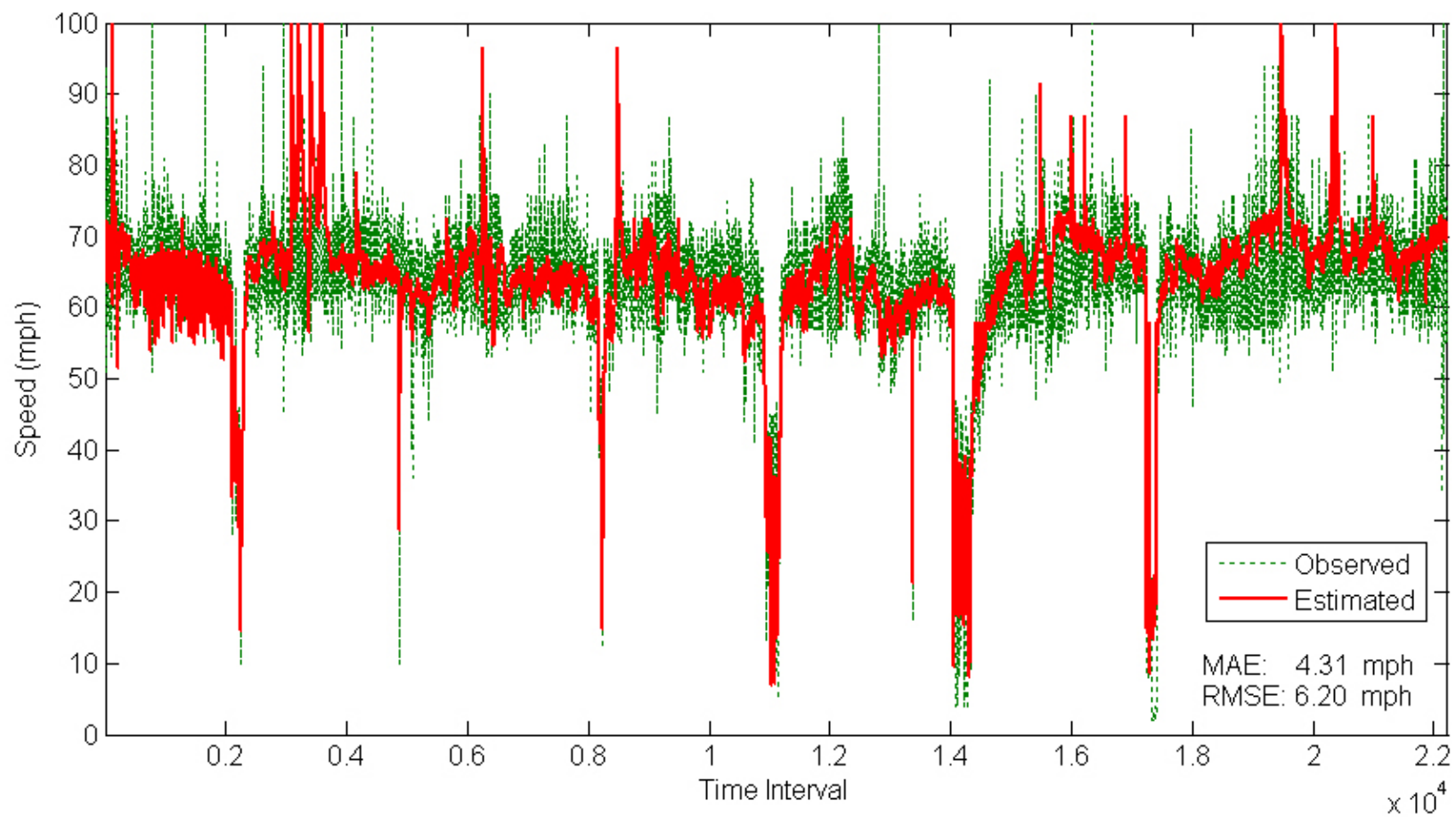
Fig. 4.22 Estimated speeds from the UKF on IH-35 in San Antonio (lane 1, Feb.10-16, 2003)

The EKF was also implemented to the above datasets. The comparison of results from the UKF and the EKF are shown in Table 4.1. Results show that the UKF had better estimates than the EKF for all datasets with lower MAEs and RMSEs. The superiority of the UKF is thus demonstrated. From the Austin dataset, it can be seen that the performance of both the EKF and the UKF became better from lane 1 to lane 4.

Table 4.1 Comparison of Speed Estimation Results

Location	Date	Lane No.	EKF		UKF	
			MAE (mph)	RMSE (mph)	MAE (mph)	RMSE (mph)
S.H.6 in College Station						
	Jan.27, 2004	Lane 1	5.02	6.50	3.62	4.53
		Lane 2	4.72	6.02	3.40	4.31
IH-35 in Austin						
	Oct.27, 2004	Lane 1	7.36	10.17	3.74	5.13
		Lane 2	7.03	10.05	3.30	4.63
		Lane 3	5.18	7.00	3.02	4.07
		Lane 4	4.18	5.6	2.70	3.62
	Nov.09, 2004	Lane 1	5.65	7.24	3.61	4.72
		Lane 2	5.27	7.16	3.12	4.08
		Lane 3	4.59	5.91	3.13	4.1
		Lane 4	4.91	6.74	3.05	4.06
IH-35 in San Antonio						
	Feb.10-16, 2003	Lane 1	7.78	10.08	4.31	6.20

In order to examine whether or not the MAEs from the EKF and UKF significantly differ from each other, paired t-tests were conducted. For each lane, assume that the absolute errors of the EKF and the UKF are e_{Ek} and e_{Uk}, where k is the index of time interval. Then, the difference between e_{Ek} and e_{Uk} is $d_k = e_{Ek} - e_{Uk}$. The test statistic can be calculated by

$$t = \frac{\bar{d}}{\sqrt{s^2 / M}} , \tag{4.31}$$

where $\bar{d}$ is the mean of $d_k, k = 1..., M$, s^2 is the sample variance, M is the sample size (number of pairs), and $M - 1$ equals to the number of degrees of freedom. It should be noted that before the paired t-tests, a normal test for a dataset of the differences was examined. It was found that the differences were normally distributed. Thus, it is assumed that the differences of other datasets are normally distributed so that the paired t-tests can be used.

The results of paired t-tests for MAEs are shown in Table 4.2. It can be seen that the 2-tailed p-values for all datasets are 0.000, which means that the difference between the estimation errors from the UKF and the EKF are significant at 95% confidence level. The powers of the tests are almost 100% (=1- p-value). Therefore, the superiority of the UKF over the EKF can be identified. For example, the difference of mean for lane 1 dataset in College Station is 1.40 mph with a sample size of 2471, and the calculated t value is 16.89, then the two-sided p-value can be obtained by using the t value and the number of degree of freedom (2470). Moreover, the 95% confidence interval is [1.23

Table 4.2 Paired Samples t-test for MAEs

Location	Date	Lane	Paired Differences					t	df	Sig. (2-tailed)
			Mean	Std. Dev.	Std. Err. Mean	95% Confidence Interval of the Difference				
						Lower	Upper			
SH6 (College Station)	Jan.27, 2004	1	1.40	4.11	0.08	1.23	1.56	16.89	2470	0.000
		2	1.32	4.71	0.11	1.02	1.44	11.52	1949	0.000
IH-35 (Austin)	Oct.27, 2004	1	3.62	6.45	0.12	3.40	3.88	30.09	2819	0.000
		2	3.73	6.27	0.12	3.45	3.92	31.14	2816	0.000
		3	2.16	4.46	0.09	1.99	2.33	25.37	2733	0.000
		4	1.48	3.17	0.06	1.35	1.60	23.48	2541	0.000
	Nov.29, 2004	1	2.04	4.75	0.09	1.86	2.22	23.13	2796	0.000
		2	2.15	4.76	0.09	1.91	2.27	23.50	2783	0.000
		3	1.46	3.24	0.06	1.34	1.58	23.56	2741	0.000
		4	1.86	3.80	0.07	1.71	2.01	24.08	2536	0.000
IH-35 (San Antonio)	Feb.10-16, 2003	1	3.47	6.20	0.04	3.39	3.55	83.49	22225	0.000

mph, 1.56 mph], which means that we are 95% confident that the true mean (difference) lies between 1.23 mph and 1.56 mph. The 95 confidence interval is an important measure of the reliability of the test. In all tests, the values of lower bounds are greater than zero.

4.6   SENSITIVITY ANALYSIS

As mentioned above, speed variance information cannot be obtained from single loop outputs. For this reason, it was set as a fixed value in the implementation of the UKF algorithm. Therefore, it is necessary to examine the influence of the speed variance on speed estimation. The lane 1 dataset collected from Austin on Oct.27 was used for this purpose. Different values of square root of speed variance (σ), ranging from 0.05 mph to 10 mph, were assigned for experiments. Estimation results are shown in Figure 4.23.

It is found that speed variance did not affect estimation results significantly. The lowest MOEs were obtained when the square root of speed variance (σ_s) was between 2 mph and 3 mph. With the decrease or increase of σ_s outside the range, estimation errors became larger. However, the errors increased relatively slowly. This is very important for speed estimation because it will be "safe" to set the value for σ_s within a range. Further examination found that smaller values of σ_s were more favorable for uncongested flow conditions than larger values, and vise versa. This is reasonable because speed variance becomes larger under traffic congestion conditions. Therefore, it is recommended that σ_s be larger than 2 and less than 8 in practice.

From the figure, it can be also found that the RMSE curve is similar to the MAE curve in shape. As was mentioned, the RMSE is able to measure the variance of errors and the bias of estimations, and the MAE measures estimation errors. It can be inferred that the increase of σ_s did not increase the variance of errors evidently.
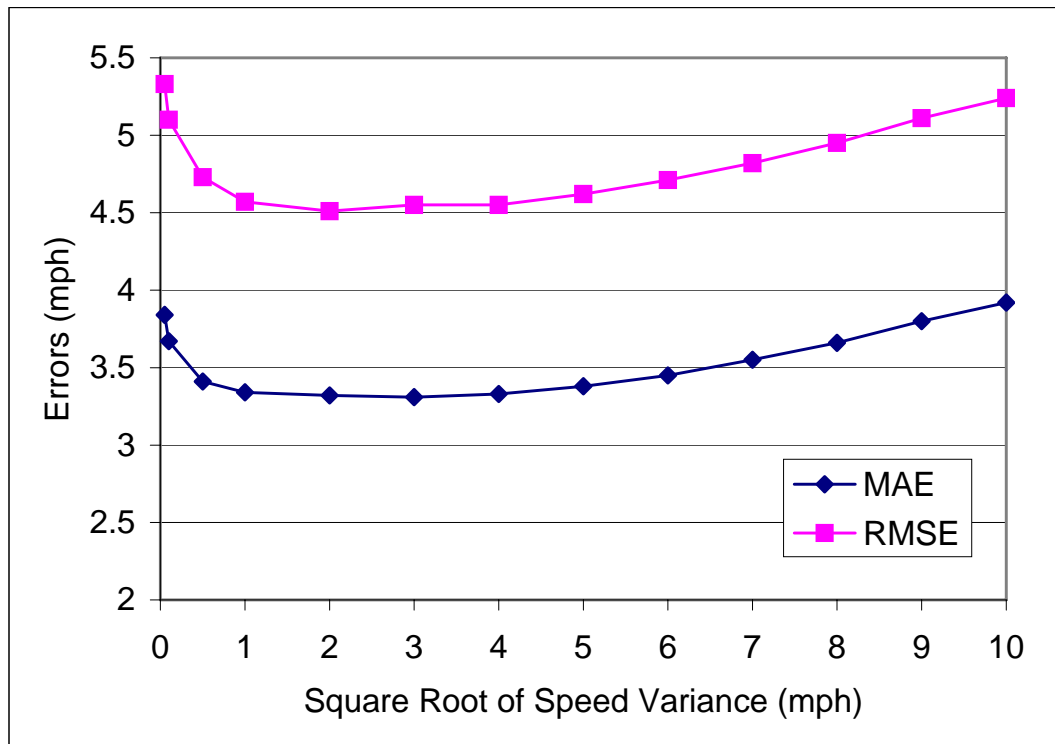
Fig. 4.23 Sensitivity analysis of speed variance with 30s time interval
(lane 1, IH-35, Austin, Oct.27, 2004)

4.7   SUMMARY

In this chapter, the problem of speed estimation was first examined. It was showed that the problem is nonlinear, especially under congested traffic conditions. Thus, a UKF that performs well for nonlinear systems was proposed for speed estimation.

Peek-ADR 6000 detector data and dual-loop data collected from different locations were applied to the proposed method as well the EKF. Estimation results from both methods were compared and evaluated. It was found that the UKF generated more accurate estimates than the EKF. Finally, the effect of speed variance on speed estimation was analyzed. It was found that speed variance did not have great effects on speed estimation.

CHAPTER V

METHODOLOGY II: UNSCENTED PARTICLE FILTER (UPF)

5.1   INTRODUCTION

In the previous chapter, an UKF method was developed for the nonlinear speed estimation problem. Despite that the UKF was demonstrated to have better performance than the EKF, this method uses the Gaussian assumption that may affect the accuracy of speed estimation. Therefore, this chapter will discuss the assumption behind the use of the UKF. A new method will then be proposed to avoid the limitation of the UKF while taking advantage of its strength.

5.2   LIMITATION OF THE UKF

The UKF, like the EKF, assumes a Gaussian parametric form of the posterior (Merwe, 2000). The assumption means that the distribution of the state ($\overline{s}_k$), the process noise (v_k), and the observation noise (n_k) are Gaussian distributed. Thus, the Gaussian posterior can fail in non-Gaussian problems with multi-modal and/or heavy tailed posterior distributions.

In the speed estimation problem, although the distributions of noises are difficult to analyze, the distribution of the state can be examined. We can check the distribution of speed ($\overline{s}_k$). Vehicle speeds are usually assumed to be normally distributed under free (or

nearly free) flow conditions. Thus, the distribution of speed under such conditions is first examined. An example of speed distribution is shown in Figure 5.1 using speed data gathered from S.H.6 with time interval of 30 seconds. It can be seen that most speed values are between 55 mph and 70 mph. Also, part "b)" shows that the values are nearly symmetric around 65 mph, with slightly left skewness. From the Q-Q plot, it seems that the speed data fit a normal distribution, except some values in the tails. However, hypotheses test for goodness-of-fit to a normal distribution at the 95% significance level showed that this dataset did not fit a normal distribution with a p-value less than 0.001. This could have been caused by the values distributed in the tails.
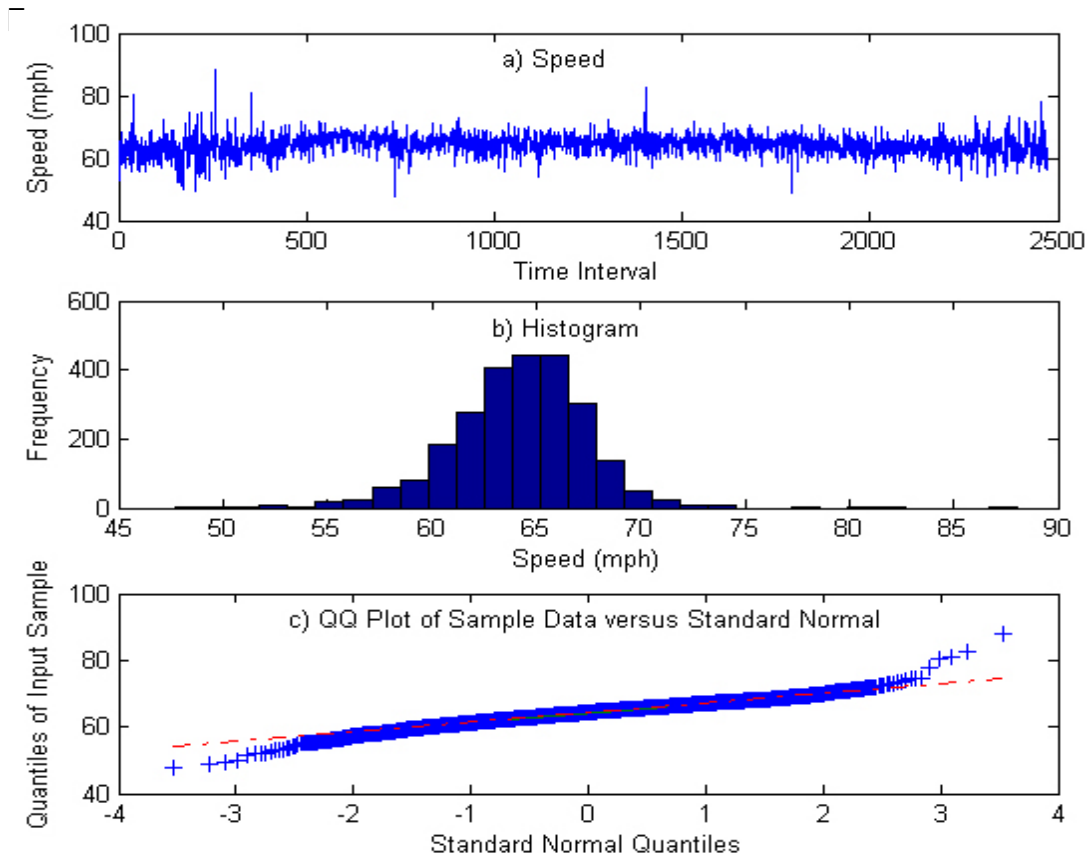
Fig. 5.1 Speed distribution under normal traffic conditions

Next, the normality of speed under traffic congestion situations is examined. Figure 5.2 shows the distribution of speed under such conditions using data from IH-35 with 30-sec time intervals.
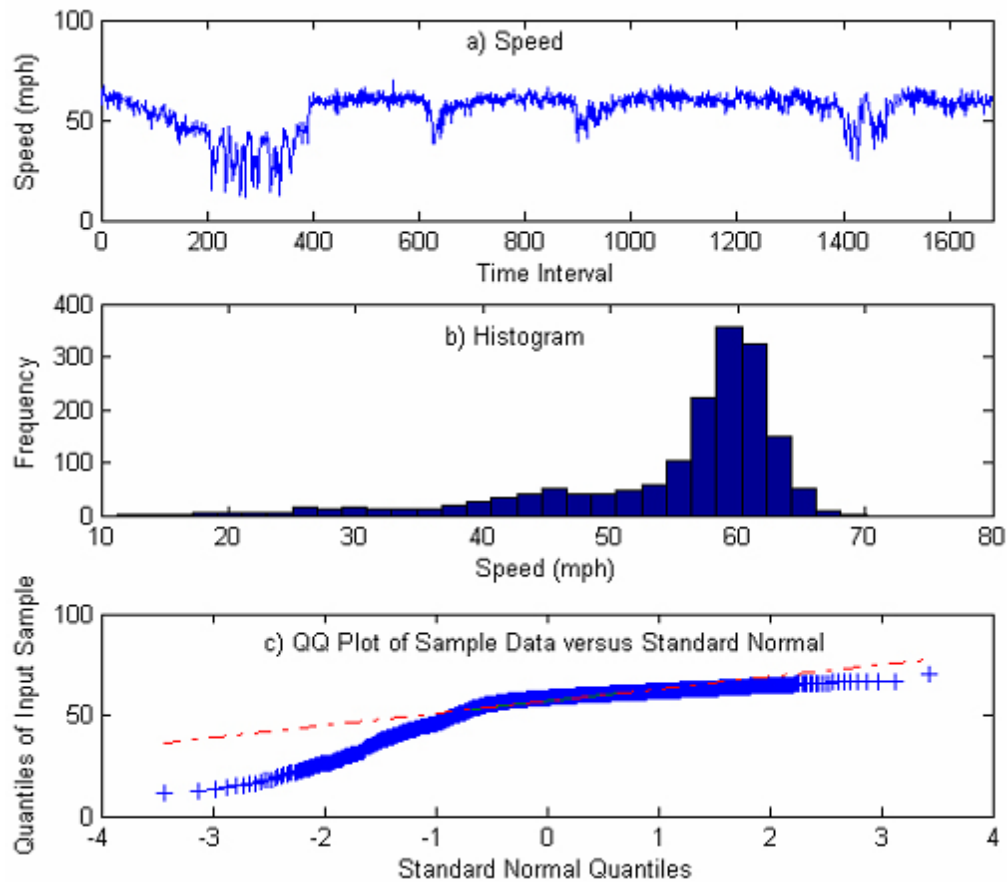
Fig. 5.2 Speed distribution under congested traffic conditions

Part "a)" of this figure shows measured speed data over time with time interval of 30 seconds. Traffic congestion existed during the peak periods. From the histogram (part "b)"), it can be seen that the distribution of speed has a heavy tail in the left side, which

is caused by low speed values. Part "c)" is the corresponding normal Q-Q plot of speed data. The Q-Q plot displays a highly left (negative) skewed distribution of speed data. Hypotheses test of a normal distribution results in the p-value of 0. It is obviously that the speed distribution is not normally distributed.

Some other distributions have been proposed in the past to model speed data. For example, log-normal distribution was found to be appropriate when the speed distribution is unimodal and right skewed (a tail is on the right side) (Haight and Mosher, 1962; Gerlough and Huber, 1975). The composite distribution was used when the traffic stream includes two classes of vehicles and has a bimodal distribution (May, 1990; Dey et al., 2006). The data from our study and many previous studies have showed that speed did not typically follow a normal distribution. Therefore, the Gaussian assumption tends to be invalid in practice. The invalid assumption in the UKF can have effects on speed estimation. A remedy needs to be developed to solve the weakness of the UKF.

5.3   METHODOLOGY

To overcome the unrealistic Gaussian assumption, one can use nonparametric techniques, such as the Particle Filter (PF), which do not depend on the Gaussian assumption. Nonparametric techniques are developed based on the PF, which is also called the sequential Monte Carlo method. The PF uses a set of random particles to approximate the posteriors instead of using a functional form. The PF was first introduced into the statistics and physics in the fifties (Hammersley and Morton, 1954;

Rosenbluth and Rosenbluth, 1955). Many PF algorithms have been proposed in the following decades (Akashi and Kumamoto, 1977; Handschin, 1970; Gordon et al., 1993). However, most of them use the state transition prior $p(x_k|x_{k-1})$ as the *proposal distribution* to draw particles. As a result, the particles may have low likelihood as the state transition does not take into account the most recent observation y_k.

Based on the advantages and limitations of both parametric and nonparametric techniques, a hybrid filter of the Unscented Particle Filter (UPF) that combines the nonparametric PF and the parametric UKF is suggested (Merwe et al. 2000). In the UPF, the PF provides the general probabilistic framework for nonlinear non-Gaussian systems, while the UKF generates proposal distributions for the PF, taking the most recent observation into account.

5.3.1   Particle Filter (PF)

Using the nonparametric method, a set of particles can be drawn to approximate the posterior distribution $p(x_{0:k}|y_{1:k})$:

$$\hat{p}(x_{0:k}|y_{1:k}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_{0:k}^{(i)}}(dx_{0:k}) \tag{5.1}$$

where $\delta(d)$ is the Dirac Delta Function and the samples $\{x_{0:k}^{(i)}; i=1,...,N\}$ are drawn from the posterior distribution. The approximation converges if N is large enough (Doucet, 1998). However, this approximation is only of theoretical significance as it is often impossible to sample directly from the posterior distribution. To solve this

difficulty, we can sample from a known proposal distribution $q(x_{0:k}|y_{1:k})$. Thus, the posterior distribution can be approximated by *properly weighted* particles drawn from the proposal distribution (Liu and Chen, 1998):

$$\hat{p}(x_{0:k}|y_{1:k}) = \sum_{i=1}^{N} w_k(x_{0:k}^{(i)})\delta_{x_{0:k}^{(i)}}(dx_{0:k}) \tag{5.2}$$

The unnormalized importance weights are given by:

$$w_k^{(i)} = p(y_{1:k}|x_{0:k}^{(i)})p(x_{0:k}^{(i)})/q(x_{0:k}^{(i)}|y_{1:k}) \tag{5.3}$$

The importance weights are further normalized through:

$$\widetilde{w}_k^{(i)} = w_k^{(i)} / \sum_{j=1}^{N} w_k^{(j)} \tag{5.4}$$

To achieve a sequential estimate of the posterior distribution, it is important to develop a recursive calculation of weights. Assumptions are made that the current state is independent on future observations, the states follow a Markov process, and observations are conditionally independent given the states (Isard and Blake, 1996; Merwe et al., 2000). With those assumptions, a recursive estimate for the importance weights is given by:

$$\widetilde{w}_k^{(i)} = \widetilde{w}_{k-1}^{(i)} \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} \propto \frac{p(x_{0:k}|y_{1:k})}{q(x_{0:k}|y_{1:k})} \tag{5.5}$$

where $p(y_k|x_k^{(i)})$ is the likelihood, $p(x_k^{(i)}|x_{k-1}^{(i)})$ is again the transition prior,

$q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})$ is the proposal distribution and $\dfrac{p(x_{0:k}|y_{1:k})}{q(x_{0:k}|y_{1:k})}$ is called the importance

ratio.

So far, the first step in the PF is called the Sequential Importance Sampling (SIS). In this step, the proposal distribution is used twice. First, particles are drawn from the proposal distribution; and second, each particle's importance weight is calculated based on the proposal distribution. To choose an appropriate proposal distribution, the proposal distribution that minimizes the variance of the importance weights is advocated (Doucet et al., 1999).

It is found that the variance of the importance ratio ($\dfrac{p(x_{0:k}|y_{1:k})}{q(x_{0:k}|y_{1:k})}$) increases over time

(Kong et al., 1994; Doucet et al., 1999), which means that one of the importance weights tends to one while others become zero after a few iterations. To avoid the degeneration of the SIS, a residual re-sampling step is used to eliminate samples with low importance weights and multiply samples with high importance weights. The re-sampling procedure first calculates $\tilde{N}_i = [N\tilde{w}_k^{(i)}]$ ([ ] rounds a number towards zero), and then computes the

remaining $\overline{N}_k = N - \displaystyle\sum_{i=1}^{N} \tilde{N}_k^{(i)}$ with new weights $w_k'^{(i)} = (\tilde{w}_k^{(i)}N - \tilde{N}_i)/\overline{N}_k$. Finally, the

results ($\overline{N}_k$ and $w_k'^{(i)}$) are used to update $\tilde{N}_i$. See (Liu and Chen, 1998) for more details on the re-sampling procedure.

The last step of the PF is the output step. The output is a set of samples that can be used to approximate the mean and covariance of the posterior x_t. The approximated mean is the estimated state at the kth time interval. In summary, the algorithm of the PF can be illustrated by Figure 5.3. In this figure, assuming $N = 10$ particles are drawn at the kth time step.

Draw particles from proposal distribution
$$q(x_t^{(i)} \mid x_{0:t-1}^{(i)}, y_{1:t}), i = 1, \dots, N$$

Calculate importance weights using likelihood function with new observation
$$w_t^{(i)} \propto p(y_t \mid x_t^{(i)})$$

Re-sample particles
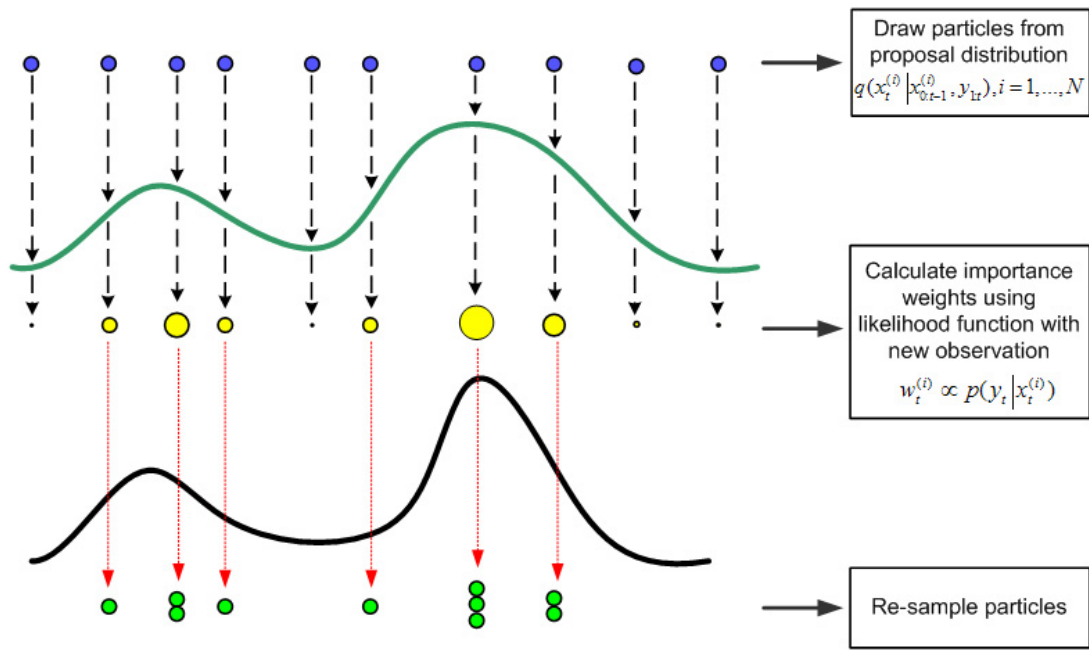
Fig. 5.3 Schematic diagram of the PF (Merwe et al., 2000)

5.3.2 Unscented Particle Filter

As has been mentioned, the parametric (UKF) and nonparametric (PF) techniques have their strengths and weaknesses. To utilize their good features and avoid their limitations, the hybrid UPF combining the PF and the UKF is proposed for the nonlinear non-Gaussian speed estimation problem. Estimated speeds from the UKF are used as the

proposal distribution for the PF to draw particles. The operation of the UPF is described in the following steps.

Step 1.   Sequential Importance Sampling

    a. Draw the particles $x_t^{(i)}, i = 1,..., N$ with the UKF to obtain the proposal distribution $q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})$.

        i. Calculation of sigma points (UKF)

        ii. Time update (UKF)

        iii. Measurement update (UKF)

    b. Sample particles $x_t^{(i)}, i = 1,..., N$ from the proposal distribution.

    c. Evaluate the importance weights (equation 5.3).

    d. Normalize the importance weights (equation 5.4).

Step 2.   Re-sampling

Multiply particles with high importance weights and suppress particles with low importance weights.

Step 3.   Output

Approximate the posterior distribution $p(x_{0:t} | y_{1:t})$ using a set of samples (equation 5.1).

5.4   IMPLEMENTATION OF THE UPF

5.4.1   Model Establishment

Before applying the UPF to speed estimation, three models need to be established: the process model (in the UKF), the measurement model (in the UKF), and the likelihood model (in the PF). These three models are developed and described as follows.

5.4.1.1 Process Model

The general state-transition model representing the relationship of the predicted state and the previous state(s) is shown in equation 4.1. In the EKF (Dailey, 1999), the predicted state is estimated by previous two states with assigned weights. Those two weights are determined using least squares estimates of the AR with 2 orders based on experimentally measured speed data. While in the UKF, we simply assign those two weights equally (=0.5). Thus, at the kth time step, the process model is given by

$$x_k^{-(j)} = (\hat{x}_{k-1}^{(j)} + \hat{x}_{k-1}^{(j)})/2 + n_k^{(j)}, \quad j = 0,...,2n \tag{5.6}$$

where j is an index and equals to the number of sigma points, n is the dimension of the state space, and $n_k^{(j)}$ is the process noise. In the speed estimation problem, the process noise can be determined by speed variance. In reality, either measurement or calculation of speed variance (σ_s^2) is impossible and thus it is commonly set as a constant value experientially.

5.4.1.2 Measurement Model

Equation 4.2 represents the general measurement model. The function h denotes the relationship between state and measurement. In speed estimation, h is the nonlinear model presented in equation 2.5. Like in the EKF, the measurement error is determined experientially in the UKF. A simple and efficient way to determine the measurement error v_k is using the variance of measurements $\sigma^2_{O/N}$. Thus, v_k at the kth time interval can be recursively calculated by the last error (v_{k-1}) and current measurement O_k / N_k.

5.4.1.3 Likelihood Model

There is no simple expression for the likelihood model. However, the likelihood model can be established using the measurement innovation ($\Delta_k^{(i)}, i = 1,..., N$), which is the difference between the observation and the predicted observation. It is suggested that higher weights will be assigned to those particles with lower residuals. In this way, the relationship of the likelihood, the measurement noise n_k, and the measurement innovation is established by

$$p(y_k | x_k^{(i)}) \propto (n_k)^{-1} \exp(-\frac{(\Delta_k^{(i)})^2}{2}) \qquad (5.7)$$

5.4.2 Implementation

The complete algorithm of the UPF is shown as follows (Merwe et al., 2000).

- Initialization: $k = 0$.

For $i = 0, ..., N$, draw the particles x_0^i from the prior and initialize the following variables:

$$\bar{x}_0^{(i)} = E[x_0^{(i)}]$$

$$P_0^{(i)} = E[(x_0^{(i)} - \bar{x}_0^{(i)})(x_0^{(i)} - \bar{x}_0^{(i)})^T]$$

$$\bar{x}_0^{(i)\alpha} = E[x_0^{(i)\alpha}] = E[\bar{x}_0^{(i)} \ 0 \ 0]^T$$

$$P_0^{(i)\alpha} = E[(x_0^{(i)\alpha} - \bar{x}_0^{(i)\alpha})(x_0^{(i)\alpha} - \bar{x}_0^{(i)\alpha})^T] = \begin{pmatrix} P_0^{(i)} & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{pmatrix}$$

In this study, N equals to 100.

● For time intervals $k = 1, ..., \infty$

1) Sequential Importance sampling

❖ For $i = 0, ..., N$:

◆ Calculate Sigma Points

$$\chi_{k-1}^{(i)\alpha} = [\bar{x}_{k-1}^{(i)\alpha} \ \ \bar{x}_{k-1}^{(i)\alpha} + \gamma\sqrt{P_{k-1}^{(i)\alpha}} \ \ \bar{x}_{k-1}^{\alpha} - \gamma\sqrt{P_{k-1}^{(i)\alpha}}]$$

where $\gamma = \sqrt{d + \kappa}$.

◆ Propagate the particle into future (time update):

$$\chi_{k|k-1}^{(i)x} = f(\chi_{k-1}^{(i)x}, \chi_{k-1}^{(i)v})$$

$$\bar{x}_{k|k-1}^{(i)} = \sum_{j=0}^{2d} w_j^{(m)} \chi_{j,k|k-1}^{(i)x}$$

$$P_{k|k-1}^{(i)} = \sum_{j=0}^{2d} w_j^{(c)} (\chi_{j,k|k-1}^{(i)x} - \bar{x}_{k|k-1}^{(i)})(\chi_{j,k|k-1}^{(i)x} - \bar{x}_{k|k-1}^{(i)})^T$$

$$\mathbf{y}^{(i)}_{k|k-1} = \mathrm{h}(\chi^{(i)x}_{k|k-1}, \chi^{(i)n}_{k-1})$$

$$\overline{y}^{(i)}_{k|k-1} = \sum_{j=0}^{2d} w^{(m)}_{j} \mathbf{y}^{(i)}_{i,k|k-1}$$

• Incorporate new observation (measurement update):

$$P_{\overline{y}_k \overline{y}_k} = \sum_{j=0}^{2d} w^{(c)}_{j} (\mathbf{y}^{(i)}_{j,k|k-1} - \overline{y}^{(i)}_{k|k-1})(\mathbf{y}^{(i)}_{j,k|k-1} - \overline{y}^{(i)}_{k|k-1})^{T}$$

$$P_{x_k y_k} = \sum_{j=0}^{2d} w^{(c)}_{j} (\chi^{(i)x}_{j,k|k-1} - \overline{x}^{(i)}_{k|k-1})(\mathbf{y}^{(i)}_{i,k|k-1} - \overline{y}^{(i)}_{k|k-1})^{T}$$

$$K_k = P_{x_k y_k} P^{-1}_{\overline{y}_k \overline{y}_k}$$

$$\overline{x}^{(i)}_{k} = \overline{x}^{(i)}_{k|k-1} + K_k (y_k - \overline{y}^{(i)}_{k|k-1})$$

$$\hat{P}^{(i)}_{k} = P^{(i)}_{k|k-1} - K_k P_{\overline{y}_k \overline{y}_k} K^{T}_{k}$$

Sample $\hat{x}^{(i)}_{k} \sim q(x^{(i)}_{k} | x^{(i)}_{0:k-1}, y_{1:k})$.

❖ For $i = 0,...,N$, evaluate the importance weights up to a normalizing constant.

$$w^{(i)}_{k} \propto \frac{p(y_k | \hat{x}^{(i)}_{k}) p(\hat{x}^{(i)}_{k} | x^{(i)}_{k-1})}{q(\hat{x}^{(i)}_{k} | x^{(i)}_{0:k-1}, y_{1:k})}$$

❖ For $i = 0,...,N$, normalize the importance weights.

2) Suppress particles $(\hat{x}^{(i)}_{0:k}, \hat{P}^{(i)}_{0:k})$ with high or low importance weights ($\tilde{w}^{(i)}_{k}$) to obtain N random particles $(\tilde{x}^{(i)}_{0:k}, \tilde{P}^{(i)}_{0:k})$.

3) The output is a set of samples that can be used to approximate the posterior distribution:

$$p(x_{0:k} \mid y_{1:k}) \approx \hat{p}(x_{0:k} \mid y_{1:k}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_{0:k}^{(i)}}(dx_{0:k})$$

The algorithm of the UKF is coded in Matlab. The realization of this algorithm can be seen in Appendix E. The implementation of the UPF requires similar initial information as both the EKF and the UKF, as is described in Section 4.4.

5.5 ESTIMATION RESULTS AND DISCUSSION

The UPF algorithm was first implemented to some datasets used in the previous chapter. Parameter settings (i.e., MEVL and noises) in the UPF are same as those of the UKF so that they can be compared and evaluated. The MOEs for result evaluation, again, include the MAE and the RMSE shown by equations 4.28 and 4.29.

The estimated results from the UPF for two datasets (lane 1, S.H.6, Jan.26, 2004; lane1, IH-35, Nov.09, 2004) are shown in Figures 5.4 and 5.5, demonstrating speed estimation under both normal and congested traffic conditions. These two figures are corresponding to Figures 4.9 and 4.17, in which the results from the UKF are shown. It can be seen that the UPF had very accurate speed estimations for both cases. For example, the MAE and RMSE for the IH-35 dataset are only 3.20 mph and 4.23 mph. Moreover, the UPF had better estimates than the UKF with lower MOE values.
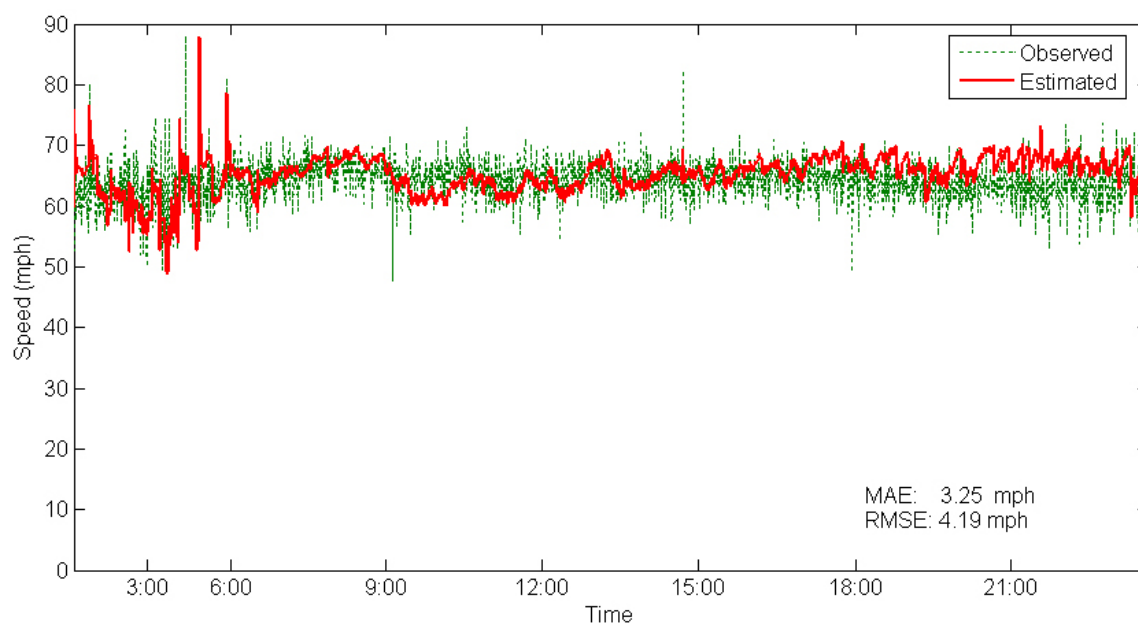
Fig. 5. 4 Estimation results from the UPF at SH6 on Jan. 26, 2004 (lane 1)
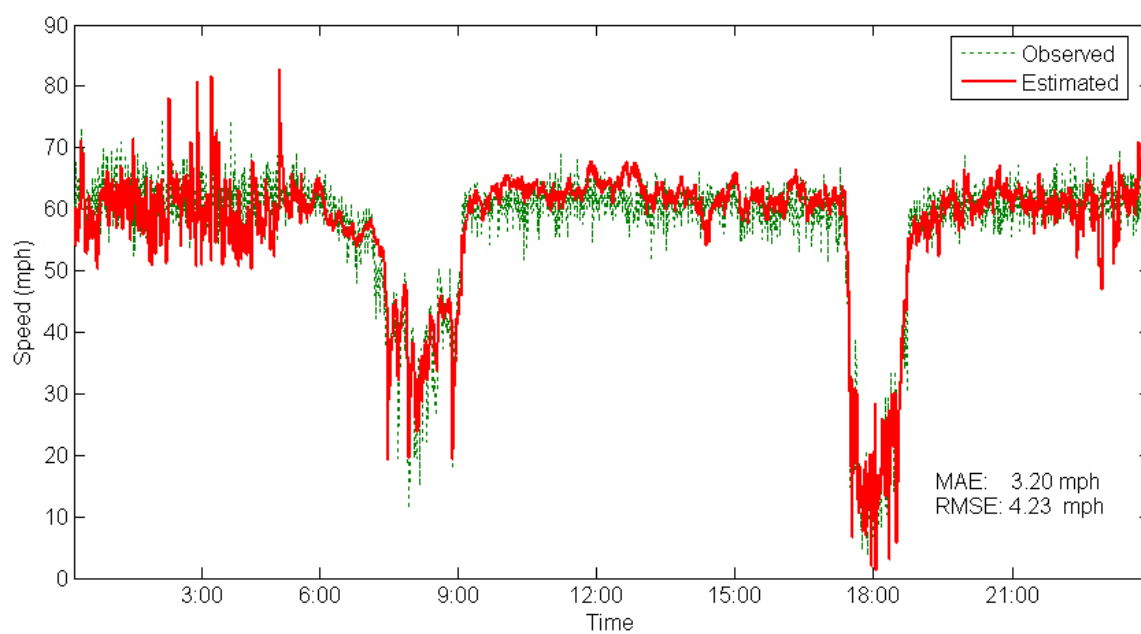
Fig. 5.5 Estimation results from the UPF at IH-35, Austin, on Nov. 09th, 2004 (lane 1)

To examine the performance of different methods under congestion conditions, a 2-hour period (17:00 – 19:00) from Figure 5.5 was extracted and is shown in Figure 5.6. It also includes estimation results from the UKF and the EKF. The MAE values for the UPF, UKF, and EKF are 3.26 mph, 4.59 mph, and 5.97 mph, respectively; correspondingly, the RMSE values are 4.24 mph, 5.74 mph, and 7.43 mph. From the figure, it can be observed that the UPF captured the variation of speed very well. The UKF, although not better than the UPF, still had good performance. The EKF, however, seems to have latency in speed estimation, which means that this method had a time delay in response to speed variations. Thus, it usually detected the variation of speed after around 2 time intervals. To test the latency, the study used the estimations that were 2 time intervals ahead as the current estimations and calculated the errors between estimated and observed values. It was found that the MAE and RMSE were 4.19 mph and 5.28 mph, which are even better than those results of the UKF. Of course, this is infeasible to do in practice since we have no knowledge of future estimates.

Figure 5.7 shows estimation results from the UPF for the double loop detector dataset, which has been applied to the UKF and the EKF in Section 4.5. The MAE and RMSE are 3.95 mph and 5.28 mph, respectively. The UPF had more accurate estimates than both the UKF and the EKF, which had the MAEs of 4.31 mph and 7.78 mph, and RMSEs of 6.20 mph and 10.08 mph.
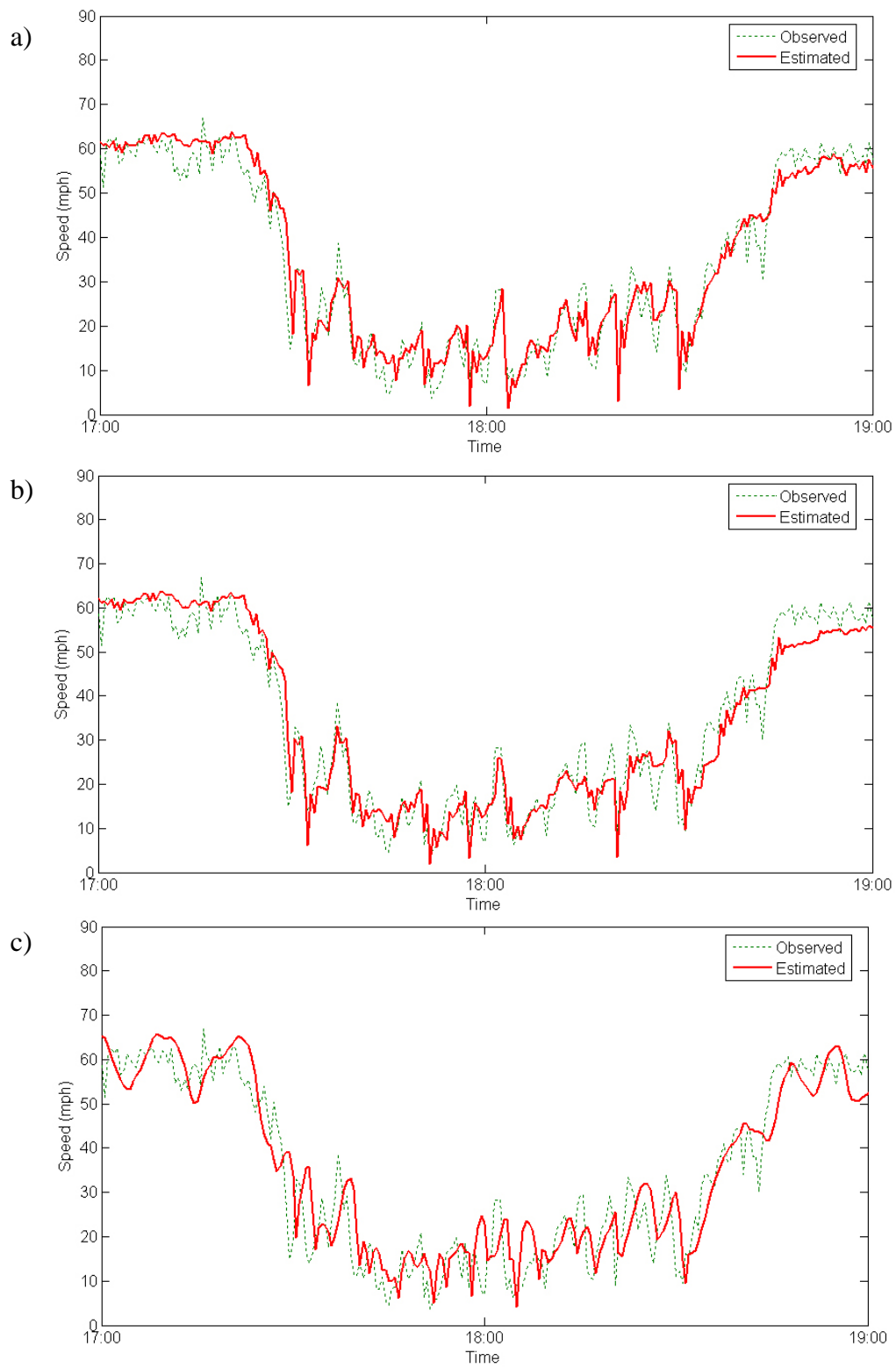
Fig. 5. 6 Comparison of results under congested conditions. a) UPF. b) UKF. c) EKF.
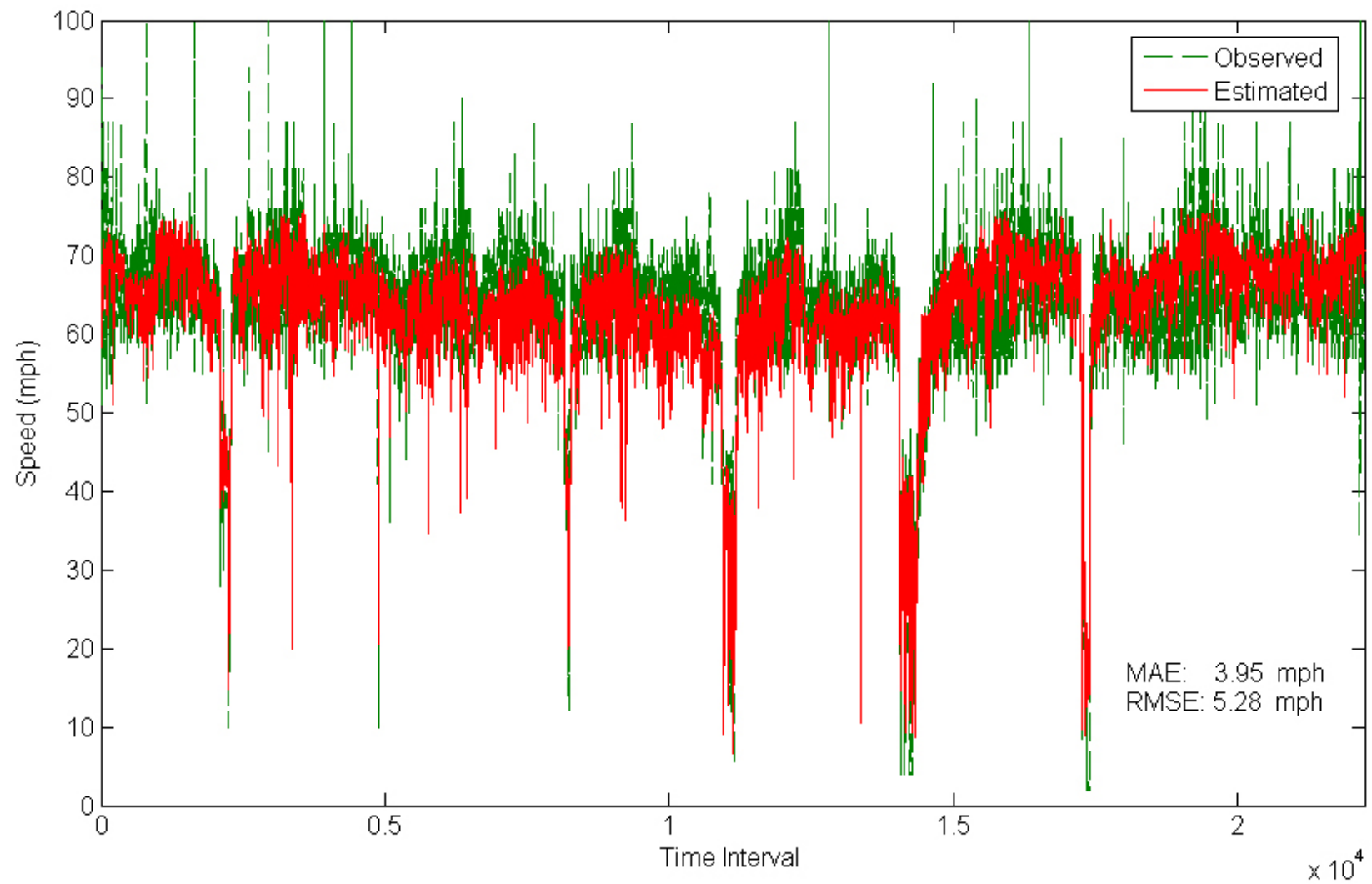
Fig. 5. 7 Estimation results from the UPF at IH-35, San Antonio, from Feb.10 – 16, 2003 (lane 1)

For a better comparison, a 2.5-hour period (from 16:00 to 18:30) of speed estimation during afternoon peak hours on Friday (Feb. 14, 2003) was selected, and results from those three methods are displayed in Figure 5.8. Urban freeways usually have the heaviest traffic loads during this period. As shown in the figure, traffic speeds first went down quickly as caused by congestion, then varied between 5 mph to 40 mph for 2 hours, and finally recovered gradually to 50 mph between 18:00-18:30.

The UPF, the UKF, and the EKF have MAEs of 3.55 mph, 4.67 mph, and 7.71 mph, and RMSEs of 4.78 mph, 5.86 mph, and 10.44 mph, respectively. It is obvious that the UPF had the best estimation accuracy among them. From part "a)" of this figure, it can be seen that the estimation curve of the UPF followed the observation curve very well. The UKF had good estimations but was still worse than the UPF. The EKF again had the worst performance as caused by its weaknesses (linearization, latency, etc).

In addition to Peek ADR-6000 and double loop detector data, simulated data were also used for speed estimation. Occupancy and count data from surveillance detectors were used to estimate speeds, and observed speed data were for performance evaluation. The three filtering methods were applied to the 2-hour simulated dataset as described in Chapter III. The MAEs of the UPF, the UKF, and the EKF are 2.08 mph, 2.66 mph, and 3.47 mph, and correspondingly, the RMSEs are 2.73 mph, 3.44 mph, and 5.23 mph. It is obvious that the UPF had the most accurate estimations.
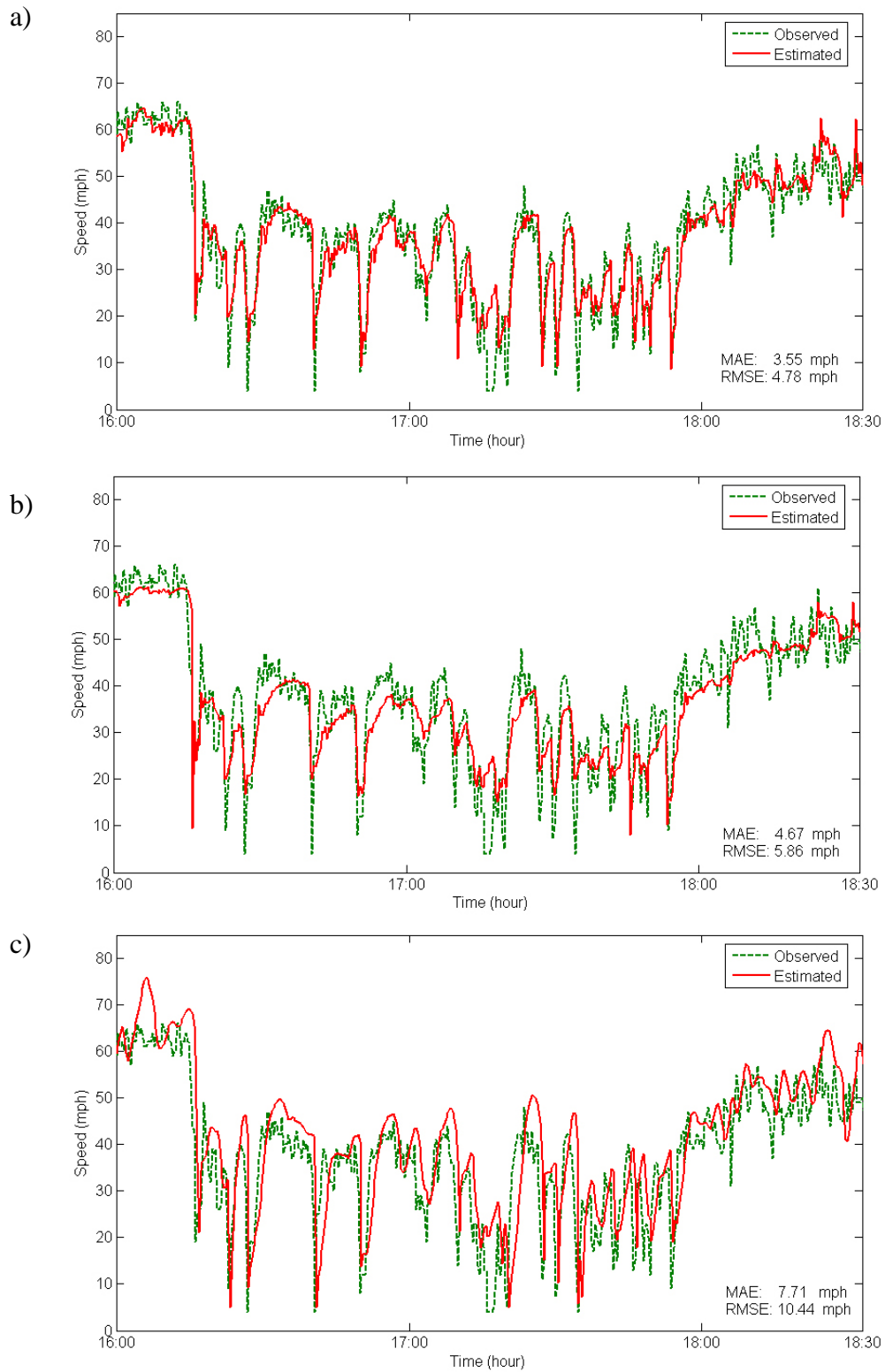
Fig. 5. 8 Comparison of results. a) UPF. b) UKF. c) EKF.

Figure 5.9 shows observed speed and estimation errors from those three methods. The vertical dotted lines in part "a)" represent the onset and end of the incident. It can be seen that the traffic flow underwent drastic speed reductions with the presence of the incident, then remained low between 10 mph and 20 mph during the one and a half hours of traffic congestion, and recovered quickly to the normal speed. From part "b)" of this figure, it is found that the UPF and the UKF had comparable estimation results under normal traffic conditions. While under congested situations, the UPF performed better, especially during the presence of the incident. The UKF had several relatively large errors after the onset of the incident, while the UPF had accurate estimations. The EKF did not have good estimates during the incident, as can be seen from those three peaks of overestimation errors. Moreover, the EKF had large errors during the recovery of the congested traffic flow; the UPF only had one relatively large error during this time period.
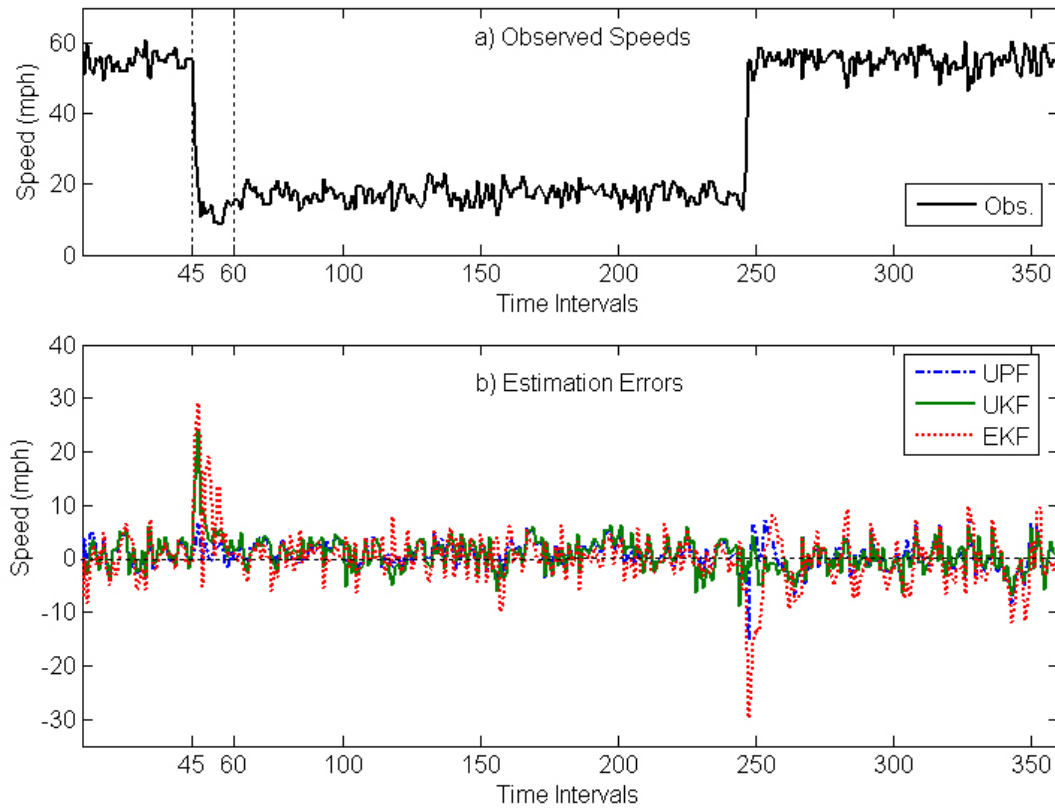
Fig. 5.9 Estimation errors of simulated data

The estimation results for five different datasets are summarized in Figures 5.10 and 5.11, which show MAEs and RMSEs respectively. The five columns represent the following five datasets from different locations and/or dates.

- S.H.6: Lane 1 on Jan. 26, 2004

- IH-35 in Austin (1): Lane 1 on Oct. 27, 2004

- IH-35 in Austin (2): Lane 1 on Nov. 09 , 2004

- IH-35 in San Antonio: Lane 1 from Feb 10 to 16^t, 204

- CORSIM: 2-hour simulation with the involvement of incident

From both figures, it can be easily observed that the UPF outperforms the UKF, although its improvement is not as much as that of the UKF over the EKF.
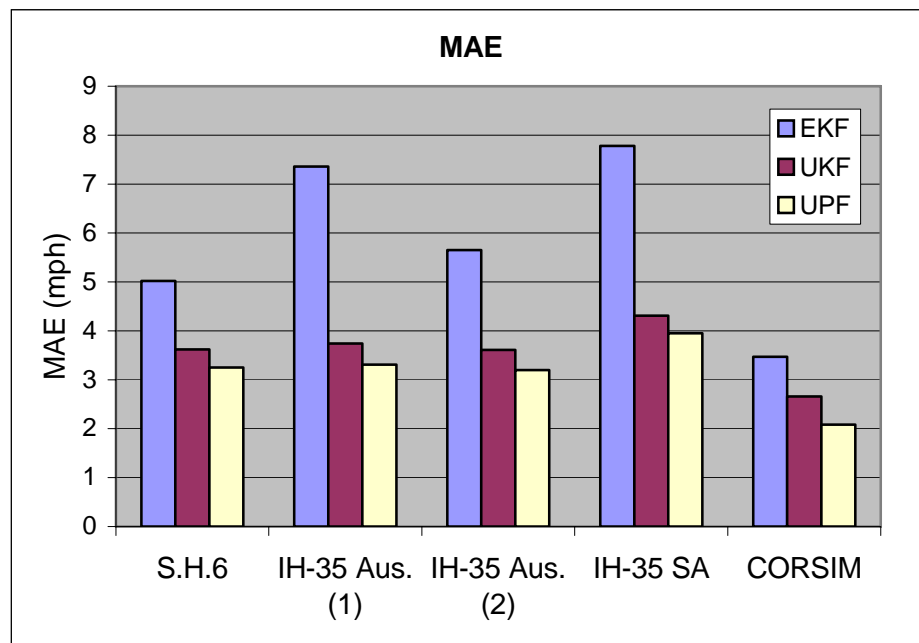
Fig. 5.10 Comparison of MAEs
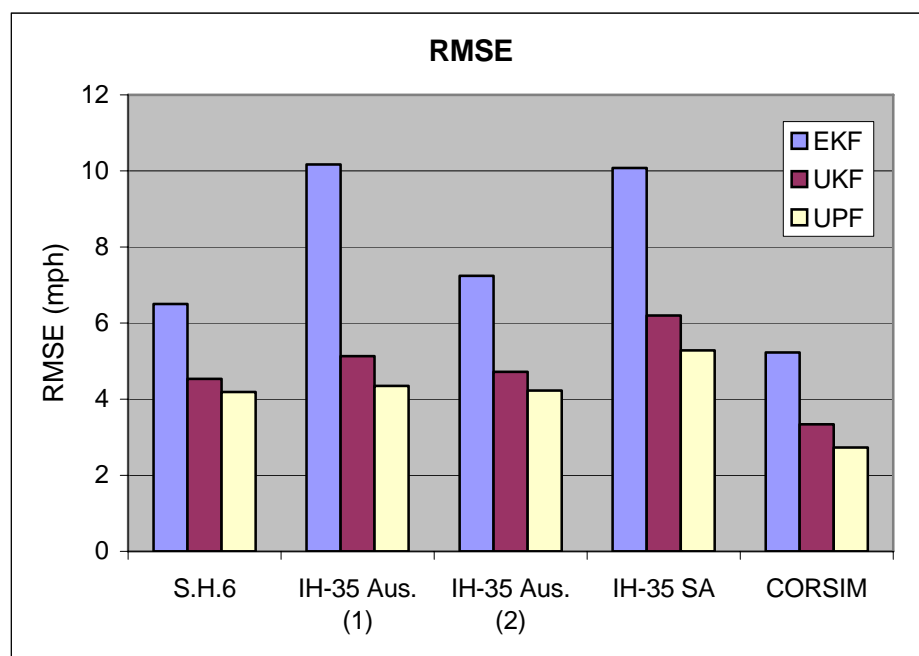
Fig. 5.11 Comparison of RMSEs

In order to show whether or not the UPF significantly improves the accuracy of speed estimation over the UKF, paired t-tests were carried out for the MAEs at a 5% significance level. It is again assumed that the differences are normally distributed. Testing results are presented in Table 5.1. It can be seen that the 2-tailed p-values for all datasets are 0.000, which means that the difference between the estimation errors from the UPF and the UKF are significant at the 95% confidence level. Moreover, the lower bounds of the 95% confidence intervals are positive. Therefore, testing results confirm that the UPF is superior to the UKF.

Table 5.1 Paired t-tests for MAEs of the UPF and the UKF

Location	Date	Lane	Paired Differences					t	df	Sig. (2-tailed)
			Mean	Std. Dev.	Std. Err. Mean	95% Confidence Interval of the Difference				
						Lower	Upper			
SH6	Jan.27 2004	1	0.37	2.15	0.04	0.29	0.46	8.56	2470	0.000
IH-35 (Austin)	Oct.27 2004	1	0.51	3.72	0.09	0.33	0.68	5.58	1679	0.000
	Nov.29 2004	1	0.36	1.79	0.03	0.30	0.43	10.69	2796	0.000
IH-35 (San Antonio)	Feb.10 - -16 2003	1	0.36	3.66	0.02	0.30	0.41	13.95	22225	0.000
CORSIM	N/A	1	0.58	2.12	0.11	0.36	0.80	5.17	359	0.000

5.6   SUMMARY

In this chapter, the speed estimation problem was modeled as a nonlinear non-Gaussian system. Real world data were used to show that the distribution of speed was not normally distributed, and this contradicted the Gaussian assumption used behind the KF family. Thus, the non-parametric PF technique was introduced for solving the nonlinear non-Gaussian problem. The PF, however, had its weakness in sampling from the posterior distribution. As a result, the UPF method was proposed for speed estimation. This method combines the UKF and the PF to avoid their limitations as well as absorb their strengths.

The EKF, the UKF, and the UPF were implemented to data collected both from fields and simulations. Estimation results as well as hypothesis tests confirmed that the UPF had more accurate estimates than the UKF, although the improvement was not as much as that of the UKF over the EKF.

The number of particles (N) used for sampling from the proposal distribution was set as 100. The selection of N was based on preliminary experiments using different number of particles that took the values of 50, 100, and 200. It was found that the results using 100 and 200 particles had nearly no difference, while the results using 100 particles had some improvement over those using 50 particles. Hence, $N=100$ was used for the implementation of the UPF algorithm. The results were not shown in the dissertation since they are not the focus of the research and $N=100$ is a reasonable value. However, it should be noted that the value of N is related to the computational cost of the UPF. The larger the N value, the higher computational cost the UPF requires. Thus, although the

UPF had better performance than the UKF, it has taken longer times to execute the UPF algorithm because the UKF algorithm is embedded in it. If the UPF is used to process a large amount of data, the computation time will be significant, but it becomes negligible for on-line speed estimation as only one measurement is taken every time interval.

CHAPTER VI

EXTENSIONS

6.1   INTRODUCTION

In the previous two chapters, the UKF and the UPF methods were proposed for speed estimation, and the accuracy of speed estimation was significantly improved through the implementations of the developed algorithms. With the achievement, the dissertation work will be beneficial to traffic operations by providing operating improvements on freeway networks. The benefits can be foreseen by investigating several applications described as follows.

6.2   TRAVEL TIME ESTIMATION

The estimation of travel time is very important for the purpose of both traffic management and traveler information provision. Because of the wide implementation of loop detectors, travel time estimation using ILD data has been the focus of numerous studies. Many speed-based travel time estimation methods have been developed in the past. Among those methods, the extrapolation methods are the simplest and most widely accepted techniques for travel time estimation using loop detector outputs.

This extrapolation method was first presented in the Travel Time Data Collection Handbook (1998). The development of this method is based on the assumption that speed does not vary between two detection points. Thus, the travel time between the two

points can be calculated as the distance divided by the speed (Ferrier, 1999; Lindveld and Thijs 1999; Quiroga, 2000; Lindveld et al., 2000; Cortes et al. 2002; Van Lint and van der Zijpp, 2003; Li et al., 2006). The schematic diagram of this method is shown in Figure 6.1.
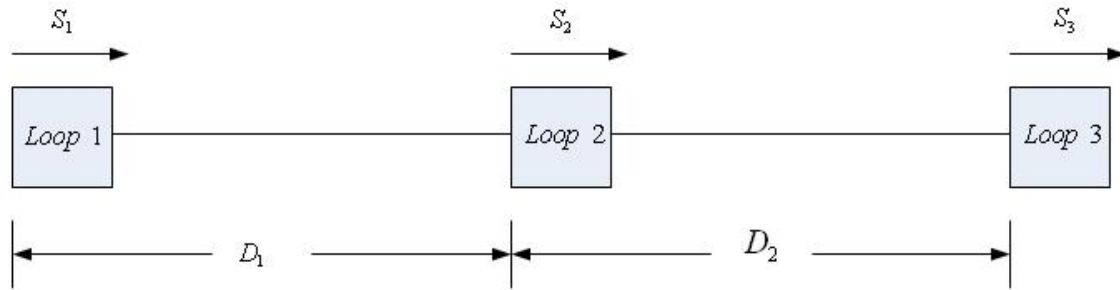
Fig.6.1 Schematic diagram of extrapolating travel time

To calculate the travel time between loops 1 and 2, there are three different ways of extrapolation. The first approach uses the average speed of S_1 and S_2 for the calculation. Thus, the travel time between loop 1 and loop 2 is

$$tt_{1-2} = \frac{2 * D_1}{S_1 + S_2}, \tag{6.1}$$

where S_1 and S_2 are the average speeds at stations 1 and 2, D_1 is the distance between stations 1 and 2, and tt_{1-2} is the estimated travel time.

The second way of travel time estimation uses the minimum speed of S_1 and S_2 for calculation. Thus, the travel time is calculated by

$$tt_{1-2} = \frac{D_1}{Min(S_1, S_2)},$$
(6.2)

The third way can be calculated by

$$tt_{1-2} = \frac{1}{2}(\frac{D_1}{S_1} + \frac{D_1}{S_2}).$$
(6.3)

From the above equations, it can be shown that the second approach will result in the largest travel time, and the first approach has the smallest value.

No matter which method is used for the estimation of travel time, speed is the only independent variable, since the distance between two measurement points is known. Thus, the accuracy of travel time is directly governed by the accuracy of speeds S_1 and S_2. The improvement of speed estimation using single loop detector outputs will directly improve the estimation of travel time and provide more accurate traveler information. Moreover, the travel time estimation using a series of single loops is cost-effective. Finally, other applications (i.e., delay analysis) that might use travel time information as an input will also benefit from the improvement of speed estimation.

6.3   INCIDENT DETECTION

As reported in a TTI's Urban Mobility Report (Schrank and Lomax, 2002), incidents are responsible for around 53-58 percent of the total delay experienced by motorists in urban areas. Thus, improving the performance of incident detection to reduce response time is very important. Over the past three decades, numerous studies have been conducted to develop incident detection algorithms including comparative algorithms (Payne et al.,

1976; Tignor and Payne, 1977; Collins et al., 1979; Masters et al., 1991; Persaud and Hall, 1989; Antoniades and Stephanedes, 1996), statistical algorithms (Dudek et al., 1974; Levin and Krause, 1978), modeling algorithms (Persuad et al., 1990; Willsky et al., 1980), and time-series algorithms (Cook et al., 1974; Ahmed and Cook, 1982; Stephanedes and Chassiakos, 1993). The most commonly used measures of performance for the evaluation of incident detection algorithms are 1) detection rate, 2) detection time, and 3) false alarm rate (Carvell, 1997).

The comparative algorithms are simple methods by "comparing speed, volume, and/or occupancy from a single loop station or between two detectors stations against thresholds that define when incident conditions are likely" (Bridya et al., 2005). For example, the PATREG algorithm detected incident by checking current speed against preset thresholds (Collines et al., 1979); the Catastrophe theory used speed, count, and occupancy as variables for incident detection, and the alarm sounded when speed dropped dramatically without a corresponding increase in occupancy and count (Persaud and Hall, 1989); the McMaster algorithm used the speed-occupancy and flow-occupancy charts for incident detection based on data from a single loop station (Antoniades and Stephanedes, 1996).

With more accurate speed information provided, it is possible to decrease the detection time without sacrificing reliability. As mentioned in Chapter V, the EKF method has the weakness of latency in speed estimation. This will result in longer detection times, while it can be avoided by using the proposed methods. Thus, the

dissertation work will help improve the performance of the comparative algorithms for automated incident detection.

6.4   LARGE TRUCK VOLUME ESTIMATION

In addition to speed, volume, and occupancy, real-time vehicle classification information is also an important input for traffic control and management. In the study by Wang and Nihan (2003), a question was posed: "Can single loop detectors do the work of dual-loop detectors?" With accurate estimation of speed from single loop outputs, we are able to say 'yes' to the question. As pointed out by Wang and Nihan (2003), accurate speed estimation is the key to produce reasonable vehicle classification information in that vehicle length can be straightforwardly calculated once speed is known. Therefore, the issue of the estimation of large truck volume was initiated and addressed.

To estimate large truck volume, vehicles are classified into two categories: Small Vehicles (SVs) and Large Trucks (LTs). A LT or a long vehicle is defined as a truck with a length greater than or equal to 12.19 m (40 feet) as in the studies of Wang and Nihan (2004) and Kown et al. (2002). The classification was based on the analysis of vehicle length distribution on freeways. Figure 6.2 shows and example of the distribution. From this figure, a bi-modal distribution is identified. The distributions of the SV class and the LT class were further explored and shown in Figure 6.2. It was found that both distributions of SVs and LTs were approximately normally distributed.

With the simple classification, the normal assumption of vehicle length, and accurate speed estimated from single loop data provided, it is capable of estimating the number of
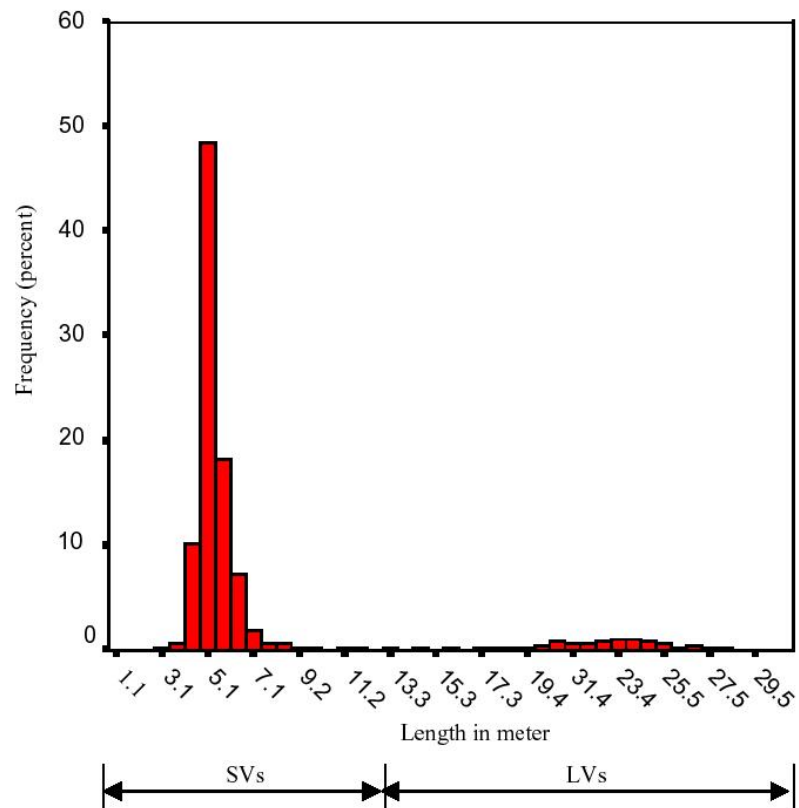
Fig.6.2 Length distribution of vehicles (Wang and Nihan, 2004)

(a) SV Class

(b) LV Class
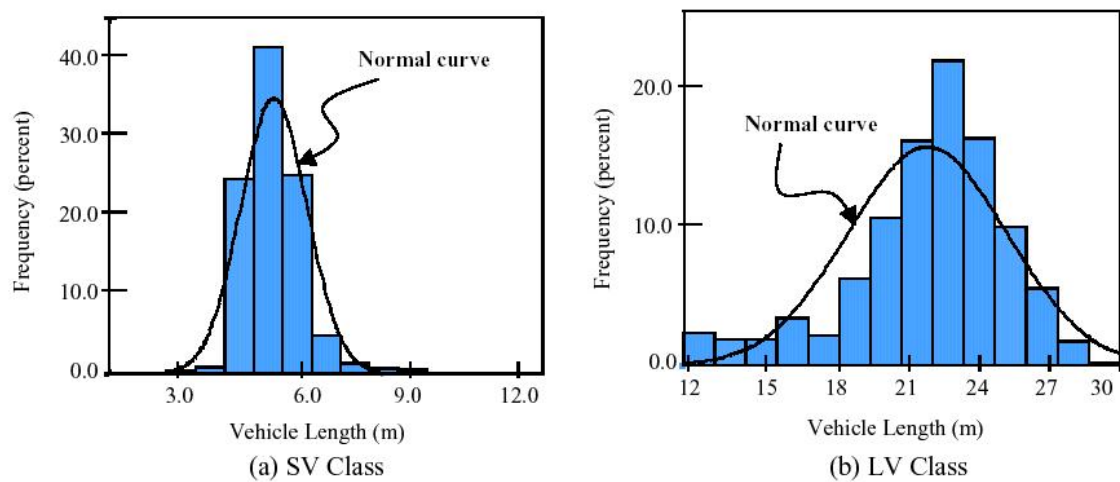
Fig.6.3 Vehicle lengths distributions with normal distribution curves (Wang and Nihan, 2004)

LTs with a reasonable accuracy. This will certainly provide valuable information for transportation planning, design, control, and operation.

6.5   SUMMARY

With the completion of this dissertation, the research can be applied to many aspects of traffic operations to improve operating performance. In this chapter, three examples of applications, speed-based travel time estimation using a series of single loops in the network, incident detection using comparative algorithms, and large truck volume estimation were used to illustrate the potential benefits of the dissertation work. The contributions, however, are not limited to the examples and can be extended to applications that require speed information from single loop outputs. In addition, the research enables cost-effective analyses of the applications without expensive detection systems.

CHAPTER VII

SUMMARY AND CONCLUSIONS

The problem statement of the dissertation identified the need to estimate speed accurately using single loop outputs. A summary of how the problem was addressed in this dissertation, the conclusions reached, and recommendations for future work are provided below.

7.1   SUMMARY

Count and occupancy collected from single loop detectors can be used to estimate speed, which can provide important information for traffic operations. Moreover, single loop detectors have been the most widely employed detectors on the U.S. highways with low costs. For these reasons, many studies have been conducted in the past to develop methodologies for speed estimation. Among them, the EKF has achieved some success and is able to generate relatively good estimates. The EKF was developed to apply to the nonlinear system of speed estimation (Dailey, 1999). However, as discussed in Chapter II, the EKF still has several issues in speed estimation.

To overcome the weaknesses of the EKF in dealing with nonlinear systems, the dissertation proposed the UKF method that has better performance for nonlinear systems. The dissertation further pointed out the common problem (Gaussian assumption) existing in the KF family that could affect the performance of the UKF, and showed that the assumption might not meet real-world conditions. To solve this

problem, the UPF method that is applicable to nonlinear non-Gaussian systems was developed to avoid the weaknesses of the UKF and PF, and assimilate their strengths as well. The algorithms of above methods were programmed in MATLAB and can be used for real-time speed estimation.

Field data collected from different locations and different days were used for the study. The datasets were collected from two types of detectors, Peek-ADR 6000 detectors and double loop detectors. Also, simulated data from the microscopic simulation program CORSIM were generated under freeway incident conditions. The EKF, the UKF, and the UPF were implemented to the datasets. Finally, the estimation results from the three methods were compared and evaluated. It was showed that the proposed methods had better performance under various traffic flow conditions.

7.2   CONCLUSIONS

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

- The problem of speed estimation was established as a nonlinear system in the past (Dailey, 1999). The dissertation further identified the problem as a non-Gaussian system. Overall, speed estimation can be treated as a nonlinear non-Gaussian problem.

- The UKF method was introduced to improve speed estimation. This method is able to overcome some limitations in the EKF method and has better performance in dealing with nonlinear systems.

- To find a solution for the nonlinear non-Gaussian problem of speed estimation, the hybrid method UPF was proposed. The method was established by combining the UKF and the PF. As a result, it is able to incorporate the strengths and avoid the limitations of the UKF and the PF.

- The implementations of the three methods (EKF, UKF, and UPF) are not difficult. They require similar external information such as the MEVL and initial inputs in their implementations.

- It was found that the proposed methods had good performance in speed estimation under various traffic flow conditions. The comparison and evaluation of the estimation results showed that both proposed methods had better estimation results than the EKF. Thus, the proposed methods are found to be promising methods for speed estimation using singe loop detector outputs.

- The results proved that the UPF had better performance than the UKF. However, the computational cost of the UPF is higher since it incorporates the UKF into the operation. It should be noted that the computation time will not be an issue for on-line applications.

- The dissertation work can be beneficial to real-time traffic operations. The improvement of speed estimation will improve the performance of applications such as travel time estimation using a series of single loops in the network, incident detection, and large truck volume estimation. Therefore, the work enables traffic analysts to use single loop outputs in a more cost-effective way.

7.3   FUTURE RESEARCH

- Single loop detectors may output erroneous data or even have system failures in practice. Thus, it is needed to carry out error checking and/or quality control on single loop outputs. Future research should be conducted to estimate speed under such situations to improve robustness. Since several algorithms regarding error checking and quality control of single loop data have been developed in the past, they can be combined into the research.

- The dissertation work has focused on speed estimation on freeway sections. Future research can be carried out to expand speed estimation to other facilities such as on-ramps that have single loop installed.

REFERENCES

Ahmed, S.A.., and Cook, A.R. (1982). "Application of time-series analysis techniques to freeway incident detection." *Transportation Research Record 841,* Transportation Research Board, National Research Council, Washington, D.C.

Antoniades, C.N., and Stephanedes, Y. J. (1996). "Single-station incident detection algorithm (SSID) for sparsely instrumented freeway sites." *Proceedings of 4th International Conference on Advanced Technologies in Transportation Engineering*, Capri, Italy, 218-221.

Akashi, H., and Kumamoto, H. (1977). "Random sampling approach to state estimation in switching environments." *Automatica* 13, 429–434.

Athol, P. (1965). "Interdependence of certain operational characteristics within a moving traffic stream." *Highway Research Record 72*, Highway Research Board, National Research Council, Washington D.C., 58-87.

Behringer, R., Holt, V., and Dickmanns, D. (1990). "Road and relative ego-state recognition." *Proceedings of the Intelligent Vehicles '92 Symposium*, Detroit, MI., 385-390.

Bozic, S.M. (1994). *Digital and Kalman filtering: an introduction to discrete-time filtering and optimum linear*. Halsted Press, NY.

Bridya, R.E., Johnson, J.D., and Balke K.N. (2005). "An investigation into the evaluation and optimization of the automatic incident detection algorithm used in TxDOT traffic management systems." *Report No. FHWA/TX-06/0-4770*. Texas Transportation Institute, Texas A&M University, College Station, TX.

Carvell, J.D., Balke, K., Ullman, J., Fitzpatrick, K., Nowlin, L., and Brehmer, C. (1997). "Freeway management handbook." *Report No. FHWA-SA-97-064*. Federal Highway Administration, U.S. Department of Transportation, Washington, DC.

Coifman, B. (2001). "Improved velocity estimation using single loop detectors." *Transportation Research Part A*, 35, 863-880.

Coifman, B., and Cassidy, M.J. (2002). "Vehicle reidentification and travel time measurement on congested freeways." Transportation Research A, 36, 899-917.

Coifman, B., Dhoorjaty, S., and Lee, Z.H. (2003). "Estimating median velocity instead of mean velocity at single loop detectors." *Transportation Research Part C*, 11, 211-222.

Collins, J.F., Hopkins, C.M., and Martin, J.A. (1979). "Automatic incident detection – TRRL algorithms HIOCC and PATREG." *TRRL Supplementary Report 526*, Transport and Road Research Laboratory, Crowthorne, Berkshire.

Cook, A.R., and Cleveland, D.E. (1974). "Detection of freeway capacity-reducing incidents by traffic-stream measurements." *Transportation Research Record 495,* Transportation Research Board, National Research Council, Washington, D.C.

CORSIM user's guide (Software Help Menu) (2001), Federal Highway Administration, U.S. Department of Transportation, Washington, D.C.

Cortes, C. E., Lavanya, R., Oh, J. S., and Jayakrishnan, R. (2002). "A general purpose methodology for link travel time estimation using multiple point detection of traffic." *Presented at the 81st Annual Meeting of the Transportation Research Board*, CD-ROM, National Research Council, Washington D.C.

Dailey, D.J. (1999). "A statistical algorithm for estimating speed from single loop volume and occupancy measurements." *Transportation Research Part B*, 33B(5), 313-322.

Dey, P.P., Chandra, S., Gangopadhya, S. (2006). "Speed distribution curves under mixed traffic conditions." *Journal of Transportation Engineering 132 (6)*, 475-481.

Doucet, A. (1998). "On sequential simulation-based methods for Bayesian filtering." *Technical Report CUED/F-INFENG/TR310*, Department of Engineering, University of Cambridge.

Doucet, A., Gordon, N.J., Krishnamurthy, V. (1999). "Particle filters for state estimation of jump markov linear systems." *Technical Report CUED/F-INFENG/TR 359*, Cambridge University Engineering Department, Cambridge, UK.

Dudek, C.L., Messer, C.J., and Nuckles, N.B. (1974). "Incident detection on urban freeways." *Transportation Research Record 495*, Transportation Research Board, National Research Council, Washington, D.C.

Durrant-Whyte, H.F., Rao, B.Y.S., and Hu, H. (1990). "Toward a fully decentralized architecture for multi-sensor data fusion." *Proceedings 1990 IEEE International Conference on Robotics and Automation*, 2, Los Alamitos, CA., 1331-1336.

Ferrier, P. J. (1999). "Comparison of vehicle travel times and measurement techniques along the I-35 corridor in San Antonio, Texas." Master's thesis, Department of Civil Engineering, Texas A&M University, College Station, TX.

Gerlough, D.G., and Huber, M.J. (1975). "Traffic flow theory-A monograph." *Transportation Research Board, Special Report 165*, Washington D.C., 37 and 205-206.

Gordon, N., Salmond, D., Smith, A. (1993). "Novel approach to nonlinear/ non-Gaussian Bayesian state estimation." *IEEE Trans. Radar, Signal Processing* 140, 107-113.

Gordon, R.L., Reiss, R.A., Haenel, H., Case, E.R., French, R.L., Mohaddes, A., and Wolcott, R. (1996). "Traffic control systems handbook." *Report No. FHWA-SA-95-032*, Federal Highway Administration, U.S. Department of Transportation, Washington, D.C.

Haight, F.A., and Mosher, W.W. (1962). "A practical method for improving the accuracy of vehicular speed distribution measurements." *Highway Research Board Bulletin 341*, HRB, Washington D.C., 92-116.

Hall, F.L. (1987). "An interpretation of speed-flow-concentration relationships using catastrophe theory." *Transportation Research Part A*, 21A (3), 191-201.

Hall, F. and Persuad, B. (1989). "Evaluation of speed estimates made with single detector data from freeway management systems." *Transportation Research Record*: *Journal of the Transportation Research Record 1232*, National Research Council, Washington, D.C., 9-16.

Hammersley, J.M., Morton, K.W. (1954). "Poor man's Monte Carlo." *Journal of the Royal Statistical Society B* 16, 23–38.

Handschin, J.E. (1970). "Monte Carlo techniques for prediction and filtering of non-linear stochastic processes." *Automatica* 6, 555–563.

Hellinga, B. (2002). "Improving freeway speed estimates from single-loop detectors." *Journal of Transportation Engineering*, 128 (1), 58-67.

Isard, M., Blake, A. (1996). "Visual tracking by stochastic propagation of conditional density." *Proc. 4th European Conf. Computer Vision*, 343-356.

Ishimaru, J.M. and Hallenbeck, M.E. (1999). "*Flow evaluation design technical report.*" Washington State Transportation Center, Seattle, WA.

Julier, S. J. (2002). "The scaled unscented transformation." *Proceedings of the American Control Conference*, 6, Anchorage, AK, 4555 – 4559.

Julier, S., and Uhlmann, J. (1996). *A general method for approximating nonlinear transformations of probability distributions*. Oxford University: Oxford.

Julier, S., and Uhlmann, J. (1997). "A new extension of the Kalman filter to nonlinear systems." *International Symposium of Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, FL, 182-193.

Julier, S., Uhlmann, J., and Durrant-Whyte, H.F. (2000). "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators." *IEEE Transactions on Automatic Control*, 45 (3), 477–482.

Kalman, R.E. (1960). "A new approach to linear filtering and prediction problems." *Transaction of ASME, Journal of Basic Engineering*, 82, 35 – 45.

Kell, J.H., Fullerton, I.J., and Mills, M.K. (1990). "*Traffic detector handbook*," 2nd Edition. Federal Highway Administration, U.S. Department of Transportation, Washington D.C.

Klein, L.A. (2003). *Traffic detector handbook*. Federal Highway Administration, U.S. Department of Transportation, Washington, DC.

Kong, A., Liu, J.S., and Wong, W.H. (1994). "Sequential imputations and Bayesian missing data problems." *Journal of the American Statistical Association 89*, 278–288.

Kwon, J., Varaiya, P., and Skabardonis, A. (2002). "Estimation of truck traffic volume from single loop detector using lane-to-lane speed correlation." *Presented at the 81st Annual Meeting of the Transportation Research Board*, CD-ROM, National Research Council, Washington, D.C.

LeRoy, S.F, and Eaud, R.N. (1977). "Applications of the Kalman filter in short-run monetary control." *International Economic Review*, 18(1), 195-207.

Levin, M.L., and Krause, G.M. (1978). "Incident detection: A Bayesian approach." *Transportation Research Record 682*, Transportation Research Board, National Research Council, Washington, D.C.

Li, R., Rose, G., and Sarvi, M. (2006). "Evaluation of speed-based travel time estimation models." *Journal of Transportation Engineering*, 132(7), 540-547.

Lin, W.H., Dahlgren, J., and Huo, H. (2004). "Enhancement of vehicle speed estimation with single loop detectors." In *Transportation Research Record: Journal of the Transportation Research Record 1870*, National Research Council, Washington, D.C., 147-152.

Lindveld, C. D., and Thijs, R. (1999). "On-line travel time estimation using inductive loop data: The effect of instrumentation peculiarities." *Proceedings of the 6th Annual World Conference on Intelligent Transportation Systems,* (CD-ROM), Toronto.

Lindveld, C., Thijis, R., Bovy, P. H., and Van der Zijpp, N. J. (2000). "Evaluation of online travel time estimators and predictors." *Transportation Research Record 1719*, Transportation Research Board, Washington, D.C., 45-53.

Liu, J., Chen, R. (1998). "Sequential Monte Carlo methods for dynamic systems." *Journal of American Statistical Association* 93, 1031-1041.

Masters, P.H., Lam, J.K., and Wong, J. (1991). "Incident detection algorithms of COMPASS – An advanced traffic management system." *Vehicle Navigation and Information System Conference Proceedings*. Society of Automotive Engineers, Inc. Warrendale, PA, 295-310.

MathWorks, Inc. (2002), *MATLAB documentation*, Version 6.5.0.180913a, Release 13, Natick, MA.

May, A.D. (1990). *Traffic flow fundamentals*. Prentice Hall, Inc., NJ.

Maybeck, P.S. (1979). *Stochastic models, estimation, and control*. Academic Press, Inc., NY.

McDermott, J.M. (1980). "Freeway surveillance and control in the Chicago area." *Transportation Engineering Journal*, 106 (3), American Society of Civil Engineers, 333-348.

McShane, W.R., Roess, R.P., and Prassas, E.S. (1998). *Traffic engineering*, 2nd Edition. Prentice Hall, Upper Saddle River, NJ.

Merwe, R., Wan, E., and Julier, S. (2004). "Sigma-Point Kalman filters for nonlinear estimation and sensor-fusion - applications to integrated navigation." *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, R I.

Merwe, R., de Freitas, N., Doucet, A., and Wan, E. (2000). "The Unscented Particle Filter", *Advances in Neural Information Processing Systems* (NIPS13), MIT Press, Eds. T. K. Leen, T. G. Dietterich and V. Tresp, December 2000.

Michalopoulos, P., and Hourdakis, J. (2001). "Review of non-intrusive advanced sensor devices for advanced traffic management systems and recent advances in video detection." *Proceedings of the Institution of Mechanical Engineers, Part I, Journal of systems and control engineering*, 215(4), University of Minnesota, 345-355.

Middleton, D., and Parker, R. (2000). "*Vehicle detection workshop, participant notebook*." Texas Transportation Institute, College Station, TX.

Middleton, D., and Parker, R. (2002). "Vehicle detection evaluation." *Report No. FHWA/TX-03/2119-1*, Texas Transportation Institute, College Station, TX.

Mimbela, L.E., and Klein, L.A. (2000). "*Summary of vehicle detection and surveillance technologies used in intelligent transportation systems*." The Vehicle Detector Clearinghouse, Southwest Technology Development Institute, U.S. Department of Transportation, Federal Highway Administration.

Nihan, N. L., Zhang, X., and Wang, Y. (2002). "Evaluation of dual-loop data accuracy using video ground truth data." *Research Report TNW2002-02*, University of Washington, Seattle, WA.

Qgut, K.S. (2004). "An alternative regression model of speed-occupancy relation at the contested flow level." *ARI*, the Bulletin of the Istanbul Technical University, 54 (2), 70-76.

Oh, S., S.G. Ritchie, and C. Oh. (2002). "Real-time traffic measurement from single loop inductive signatures." *Transportation Research Record 1804*, Transportation Research Board, National Research Council, Washington, D.C., 98-106.

Okutani, I., and Stephanedes, Y.J. (1984). "Dynamic prediction of traffic volume through Kalman filtering theory." *Transportation Research Part B*, 18(1), 1-11.

Owen, L.E., Zhang, Y., Rao, L., and McHale, G. (2000)."Traffic flow simulation using CORSIM." *Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, 1143-1147.

PATH (Partners for Advanced Transit and Highways) (1997). "Workshop on *research, development, and testing of traffic surveillance technologies*." Richmond Field Station, CA.

Payne, H.J., Helfenbein, E.D., and Knobel, H.C. (1976). "Development and testing of incident detection algorithms, Volume 2: Research Methodology and Detailed Results." *Report No. FHWA-RD-76-20*. U.S. Department of Transportation, FWHA, Washington, DC.

Peek Traffic, (2004). "*ADR-6000 automatic data recorder.*" <http://www.ustraffic.net/products/data/ADR-6000-05.pdf>, (Sept. 16, 2005).

Persaud, B. N., and Hall, F.L. (1989). "Catastrophe theory and patterns in 30-second freeway traffic data: Implications for incident detection." *Transportation Research Part A*, 2, 103-113.

Persuad, B.N., Hall, F.L., and Hall, L.M. (1990). "Congestion identification aspects of the McMaster incident detection algorithm." *Transportation Research Record 1287,* Transportation Research Board, National Research Council, Washington, D.C.

Potter, T., (2005). "*The evolution of inductive loop detector technology.*" Reno A&E, <http://www.renoae.com/Documentation/MISC/Advances%20in%20Loop%20Dete ctor%20Technology.pdf>, (Aug. 23, 2006).

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (1992). *Numerical recipes in C: the art of scientific computing*, 2 ed. Cambridge University Press, NY

Pushkar, A., Hall, F., and Acha-Daza, J. (1994). "Estimation of speeds from single-loop freeway flow and occupancy data using Cusp Catastrophe theory model." *Transportation Research Record 1457*, Transportation Research Board, Washington D.C., 149-157.

Quiroga, C. (2000). "Assessment of dynamic message travel time information accuracy." *Proceeding of the North American Travel Monitoring Conference and Exposition*, Middleton, WI, 1-13.

Raj, J., and Rathi, A. (1994). "Inductive loop tester – ILT II." *Summary Report No. FHWA-SA-94-077*, U.S. Department of Transportation, Washington D. C.

Rosenbluth, M.N., Rosenbluth, A.W. (1955). "Monte Carlo calculation of the average extension of molecular chains." *Journal of Chemical Physics* 23, 356–359.

Schlachta, H.B. and Studenny, J. (1990). "Interoperability versus integration of Omega and GPS." *Journal of Navigation*, May 1990, 229-237.

Schrank, D., and Lomax, T. (2002). *The 2002 urban mobility report*. Texas Transportation Institute, Texas A&M University, College Station, TX.

Shin, E.H., and Naser, E.S. (2004). "An Unscented Kalman filter for in-motion alignment of low-cost IMUs." *IEEE Frames Conference Proceedings*, Monterey, CA, 273–279.

Stephanedes, Y.J. and Chassiakos, A.P. (1993). "Applications of filtering techniques for incident detection." *Journal of Transportation Engineering*, 119(1), American Society of Civil Engineers, NY.

Sun, C., and Ritchie, S.G. (1999). "Individual vehicle speed estimation using single loop inductive waveforms." *Journal of Transportation Engineering*, 125 (6), 531-538.

Tignor, S.C., and Payne, H.J. (1977). Improved freeway incident detection algorithms. *Public roads*. U.S. Department of Transportation, FWHA, Washington, DC.

Travel Time Data Collection Handbook (1998), *Report No. FHWA-PL-98-035*, Texas Transportation Institute, College Station, TX.

Van Lint, J. W. C., Hoogendoorn, S. P., and van Zuylen, H. J. (2002), "Freeway travel time prediction with state space neural networks." Presented at the 81st TRB Annual Meeting of the Transportation Research Board, (CD-ROM), National Research Council, Washington D. C.

Wan, E., and Merwe, R. (2000). "The Unscented Kalman filter for nonlinear estimation." *Adaptive Systems for Signal Processing, Communications,and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158.

Wang, Y., and Nihan, N.L. (2000). "Freeway traffic speed estimation with single loop outputs." *Transportation Research Record*: *Journal of the Transportation Research Record 1727*, 120-126.

Wang Y., and Nihan, N.L. (2003). "Can single-loop detectors do the work of dual-loop detectors?" *Journal of Transportation Engineering*, 117(2), 178-188.

Wang, Y., and Nihan, N. L. (2004). "Dynamic estimation of freeway large truck volume based on single-loop measurements." *Intelligent Transportation Systems* 8, 133-141.

Welch, G., and Bishop, G. (2001). "*An introduction to the Kalman filter.*" SIGGRAPH 2001, Course 8, University of North Carolina at Chapel Hill.

Wen, W., and Durrant-Whyte, H.F. (1992). "Model-based multi-sensor data fusion." *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, 2, Nice, France, IEEE: Los Alamitos, CA, 1720-1726.

Willsky, A.S., Chow, E.Y., Gershwin, S.B., Greene, C.S., Houpt, P.K., and Kurkjian, A.L. (1980). "Dynamic model-based techniques for the detection of incidents on freeways." *IEEE Tranactions on Automatic Control*, AC-25(3). Institute of Electrical and Electronic Engineers, Piscataway, N.J.

Wilshire, R., Black, R., Grochoske R., and Higinbotham, J. (1985). "Traffic Control Systems Handbook." *Report No. FHWA-IP-85-12*. Institute of Transportation Engineers. Washington, D.C.

Woods, D.L., Cronin, B.P., and Hamm, R.A. (1994). "Speed Measurement with Inductance Loop Speed Traps." Texas Transportation Institute. *Research Report FHWA/TX-95/1392-8*. Texas A&M, College Station.

Yao, D., Gong, X., and Zhang, Y. (2004). "A hybrid model for speed estimation based on single-loop data." *Proceedings of the 7th International IEEE ITS Conference*, MoD3.3, Washington, D.C., 205-209.

Zeeman, E. C. (1977). *Catastrophe theory*. Addison-Wesley, Reading, MA.

APPENDIX A

NOTATIONS

k - Time step index

N - Traffic count

l_d - Detection zone length

l_v - Vehicle length

L - Effective vehicle length

$\bar{L}$ - Mean effective vehicle length

g - Estimator that equals to 1 over mean effective vehicle length

O - Occupancy

$\bar{s}$ - Average speed

s - Vehicle speed

t - Presence time

t_{on} - Instant of time the detector detects a vehicle

t_{off} - Instant of time the vehicle exits the detector

T - Duration of time intervals

σ_s^2 - Speed variance

tt - Travel time

D - Distance

APPENDIX B

ACRONYMS

Table B.1 List of Acronyms

Acronym	Title
AR	Auto Regression
ATIS	Advance Traveler Information Systems
ATMS	Advanced Traffic Management Systems
CORSIM	Corridor Simulation
DOT	Department of Transportation
EKF	Extended Kalman Filter
FHWA	Federal Highway Administration
FRESIM	Freeway Simulation
GPS	Global Positioning System
ILDs	Inductive Loop Detectors
ITS	Intelligent Transportation Systems
KF	Kalman Filter
LCD	Liquid Crystal Display
LOS	Level of Service
NEMA	National Electrical Manufactures Association
MAE	Mean Absolute Error
MEVL	Mean Effective Vehicle Length
MMSE	Minimum Mean Square Error

Acronym	Title
MOE	Measurement Of Effectiveness
MOS-LSI	Metal Oxide Semiconductor—Large Scale Integration
MSE	Mean Square Error
NETSIM	Network Simulation
PF	Particle Filter
RMSE	Root Mean Square Error
RTMS	Remote Traffic Microwave Sensor
SIS	Sequential Importance Sampling
TMS	Traffic Management System
TRAFED	Traffic Network Editor
TRAFVU	Traffic Visualization Utility
TSC	Traffic System Center
TSIS	Traffic Software Integrated System
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
UT	Unscented Transformation
SUT	Scaled Unscented Transformation
VID	Video Image Detector
VIP	Video Image Processor
WIM	Weigh-in Motion

APPENDIX C

MICROSCOPIC TRAFFIC SIMULATION

CORSIM INPUT FILE (.TRF FILE)

```
12345678 1 2345678 2 2345678 3 2345678 4 2345678 5 2345678 6 2345678 7 234567
                                      1  302007                          0   1
      1   0   0   3      7981 0000  0                3   0        7781    7581   2
36003600                                                                        3
                    30                                                          4
   0         0   0   0   0   0   0   0   0   0   0                               5
   1    2 603           2      01           8002              20  18  60  0     11
8001   1               2      01           2                 20  18      0     11
   1    2    100                                                               21
8001   1    100                                                                21
   1    8001                                                                   35
   2       1                                                                   35
   1 1                                                                         36
   2 1                                                                         36
   1    2    1    5000  1111   100  0                                          42
8001   14000  10   0  100                                                      50
8001   14000   04000  104000  204000  30                                 1    53
   1    2 600   1    2                                                         55
   1    2 900 300    2                                                         55
   2   35          120                    0  20   0   0                120    58
   6   53          120                    0  36   0   0                120    58
   7   53          120                    0  24   0   0                120    58
   8   64          120                    0   9   0   0                120    58
   4   40          120                    0  11 100   0               2500    58
        20                                                                     64
   0                                                                          170
```

```
8002    1065     101                          195
8001       0     101                          195
   1     352      99                          195
   2     955     101                          195
   0   3                                      210
8001   14000   10    0   100                   50
   0                                          170
   1    0    0                                210
```

APPENDIX D

CORSIM SIMULATION OUTPUTS

Table D.1 CORSIM Outputs (Half-Hour)

Time	Speed (mph)	Vehicle Count (veh/20s)	on-time (s)	Occupancy (%)
0:00:20	56.8	11	4.9	24.5
0:00:40	53.1	10	4.5	22.5
0:01:00	57.6	11	4.3	21.5
0:01:20	55.1	11	3.9	19.5
0:01:40	58	12	3.6	18
0:02:00	51.1	11	5.1	25.5
0:02:20	59.7	12	3.6	18
0:02:40	52.7	11	3.8	19
0:03:00	51.1	10	4	20
0:03:20	56	10	3.2	16
0:03:40	54.4	11	5	25
0:04:00	56.3	11	3.5	17.5
0:04:20	49.5	11	4.9	24.5
0:04:40	55.7	11	3.9	19.5
0:05:00	56.8	11	4.3	21.5
0:05:20	52.1	11	4.3	21.5
0:05:40	54.1	12	3.9	19.5
0:06:00	55	11	3.5	17.5
0:06:20	52	12	4.1	20.5
0:06:40	52	11	4.1	20.5
0:07:00	52	11	3.8	19
0:07:20	55.6	10	4.6	23
0:07:40	53.6	11	4.2	21
0:08:00	54.2	11	3.5	17.5
0:08:20	58.6	12	4.2	21
0:08:40	57.1	11	4.2	21
0:09:00	56	11	3.6	18

Time	Speed (mph)	Vehicle Count (veh/20s)	on-time (s)	Occupancy (%)
0:09:20	60.8	11	3.7	18.5
0:09:40	50.5	12	4.5	22.5
0:10:00	50	11	4.3	21.5
0:10:20	57.5	11	3.9	19.5
0:10:40	57.8	12	4	20
0:11:00	54.9	11	3.5	17.5
0:11:20	55.5	12	4.6	23
0:11:40	54.4	11	3.7	18.5
0:12:00	56.6	11	3.7	18.5
0:12:20	56.9	12	4.5	22.5
0:12:40	56.8	11	4	20
0:13:00	58.6	11	3.8	19
0:13:20	52.8	10	3.8	19
0:13:40	53.2	12	4.2	21
0:14:00	52.9	11	4.2	21
0:14:20	55.3	11	3.5	17.5
0:14:40	55.4	12	5.1	25.5
0:15:00	55.4	11	3.6	18
0:15:20	31.3	13	9.5	47.5
0:15:40	19.1	15	15.6	78
0:16:00	10.8	9	14.5	72.5
0:16:20	14.9	11	14.3	71.5
0:16:40	11.6	10	16.9	84.5
0:17:00	12.5	7	14.3	71.5
0:17:20	14.1	9	15.4	77
0:17:40	9.1	7	15.5	77.5
0:18:00	9	7	18.8	94
0:18:20	8.8	8	18.3	91.5
0:18:40	8.5	8	18.9	94.5
0:19:00	13.4	10	14.7	73.5
0:19:20	15.3	9	14.2	71
0:19:40	13.8	11	13.3	66.5
0:20:00	15.4	9	14.3	71.5
0:20:20	15.8	9	12.4	62

Time	Speed (mph)	Vehicle Count (veh/20s)	on-time (s)	Occupancy (%)
0:20:40	14.9	8	13.5	67.5
0:21:00	13.1	11	15.3	76.5
0:21:20	18.9	14	13.3	66.5
0:21:40	21.6	16	14.2	71
0:22:00	18.8	15	15.2	76
0:22:20	18.2	12	13.1	65.5
0:22:40	19.1	13	14.1	70.5
0:23:00	16.4	8	10.9	54.5
0:23:20	17.8	13	16.7	83.5
0:23:40	14.4	7	12.3	61.5
0:24:00	14.3	8	13.2	66
0:24:20	14.3	7	13	65
0:24:40	18.6	13	14.6	73
0:25:00	18.5	15	14.6	73
0:25:20	21	14	13	65
0:25:40	17	13	14.3	71.5
0:26:00	16.5	12	14.2	71
0:26:20	20.6	14	14.2	71
0:26:40	19.3	13	13.8	69
0:27:00	15.5	8	13.4	67
0:27:20	18	14	15.6	78
0:27:40	18.2	12	13.9	69.5
0:28:00	12.7	6	12.3	61.5
0:28:20	13.8	11	14.2	71
0:28:40	15.9	10	13.9	69.5
0:29:00	16.7	13	15	75
0:29:20	14.7	12	13.8	69
0:29:40	17.2	13	14.3	71.5
0:30:00	17	12	14.4	72

APPENDIX E

MATLAB PROGRAMS

OCCUPANCY PROGRAM

20 Second of Polling Interval

```
%This program compiles PVR data collected from Peek ADR-6000 detectors into a
%polling interval of 20 seconds. The raw data were imported into Excel beforehand.
%Outputs include occupancy, volume, speed, and vehicle length.

clc;
clear all;

raw_data = xlsread('austin10274.xls');
[n m] = size(raw_data);
j=1;
TI(j) = 1;
total = 0;

date = datevec(a(1,1));
hour(1) = date(1,4);
minute(1) = date(1,5);
second(1) = date(1,6);

if second(1) < 21
  key_ = 1;
elseif second(1) < 41
  key_ = 2;
else
  key_ = 3;
end

for k =2:n
  date = datevec(raw_data(k,1));
  hour(k) = date(1,4);
  minute(k) = date(1,5);
  second(k) = date(1,6);

  if second(k) < 21
    key = 1;
  elseif second(k) < 41
```

```
     key = 2;
   else
     key = 3;
   end

   if (hour(k)==hour(k-1)) & (minute(k) == minute(k-1))
     if (key_ == key)
       else
         key_ = key;
         j = j+1;
         TI(j) = k;
     end
   else
     key_ = key;
     j = j+1;
     TI(j) = k;
   end

end

j = j+1;
k = k+1;
TI(j) = k;

for kk = 2:j
  nd(kk-1) = TI(kk) - TI(kk-1);
  total = total + nd(kk-1);
 %calculate the average vehicle length during time interval kk-1; convert unit from m to
%feet)
  length(kk-1) = 3.28084*sum(a(TI(kk-1):(TI(kk)-1),2))/nd(kk-1);
 %calculate the average vehicle speed during time interval kk-1; convert unit from
%m/sec to mph
  speed(kk-1) = 2.237*sum(a(TI(kk-1):(TI(kk)-1),3))/nd(kk-1);
%occupancy during time interval kk-1
  occu(kk-1) =  sum(a(TI(kk-1):(TI(kk)-1),4))/(0.2);
end

output = [nd' occu' speed' length'];

%export data into a .DAT file
diary on
diary autin10274.dat
output
diary off
```

30 Second of Polling Interval

```
%This program compiles PVR data collected from Peek ADR-6000 detectors into a
%polling interval of 30 seconds. The raw data were imported into Excel beforehand.

clc;
clear all;

raw_data = xlsread('austin10274.xls');
[n m] = size(raw_data);

j=1;
TI(j) = 1;
total = 0;

date = datevec(a(1,1));
hour(1) = date(1,4);
minute(1) = date(1,5);
second(1) = date(1,6);

if second(1) < 31
  key_ = 1;
else
  key_ = 2;
end

for k =2:n
  date = datevec(raw_data(k,1));
  hour(k) = date(1,4);
  minute(k) = date(1,5);
  second(k) = date(1,6);

  if second(k) < 31
   key = 1;
  else
   key = 2;
  end

  if (hour(k)==hour(k-1)) & (minute(k) == minute(k-1))
   if (key_ == key)
     else
       key_ = key;
       j = j+1;
```

```
        TI(j) = k;
      end
  else
    key_ = key;
    j = j+1;
    TI(j) = k;
  end

end

j = j+1;
k = k+1;
TI(j) = k;

for kk = 2:j
  nd(kk-1) = TI(kk) - TI(kk-1);
  total = total + nd(kk-1);
%calculate the average vehicle length during time interval kk-1; convert unit from m to
%feet)
  length(kk-1) = 3.28084*sum(a(TI(kk-1):(TI(kk)-1),2))/nd(kk-1);
 %calculate the average vehicle speed during time interval kk-1; convert unit from
%m/sec to mph
  speed(kk-1) = 2.237*sum(a(TI(kk-1):(TI(kk)-1),3))/nd(kk-1);
%occupancy during time interval kk-1
  occu(kk-1) =  sum(a(TI(kk-1):(TI(kk)-1),4))/(0.2);
end

output = [nd' occu' speed' length'];

%export data into a .DAT file
diary on
diary autin102742.dat
output
diary off
```

EXTENDED KALMAN FILTER

```
%This program is used for speed estimation using the EKF algorithm.

clc;
clear all;

load data.dat; %load the .DAT file
Data_loop = data;

Si=size(Data_loop);
Ddim1=Si(1,1);
Ddim2=Si(1,2);

Ndim=2;  % the dimension of state
Length_bar(1) = 30/5280;  % average vehicle length

T = 30/3600; % duration of polling interval

x(1)=65;; % initial speed
x_hat(1)= x(1);
x(2)=65;
G=[1.91,-.91;1,0];
K=[0,0;0,0];

Varq(1)=5;
Q=[Varq(1),0;0,Varq(1)];  % process noise

Varr(1)=0.05;
R=[Varr(1),0;0,Varr(1)];  % observation noise

P_pri=[0,0;0,0];
P_post=P_pri(1);

z(1)=0;
z_a(1)=0;

count=Data_loop(1,1);
occu=Data_loop(1,2)/100;
 z(2)=occu/count;
Length_bar(2)=Length_bar(1);

Varq(2)=Varq(1);
```

```
Varr(2)=Varr(1);
Q=[Varq(2),0;0,Varq(2)];
R=[Varr(2),0;0,Varr(2)];

H_m=[-(Length_bar(2)/T)*(x_hat(1).^2+3*Varq(2).^2)/x_hat(1).^4,0;0,0];

P_pri=G*P_post*G'+Q;
K= P_pri*H_m'*inv(H_m*P_pri*H_m'+R);
P_post=P_pri-K*H_m*P_pri;

z_a(2)=z(2)-(Length_bar(1)/T)*(x_hat(1).^2+Varq(1).^2)/x_hat(1).^3-
((Length_bar(1)/T)*(x_hat(1).^2+3*Varq(1).^2)/x_hat(1).^4)*x_hat(1);

x(2)= x_hat(1)+K(1,1)*(z_a(2)-H_m(1,1)*G(1,1)*x_hat(1));
Z_m=[z_a(2),z_a(1)]';

x_hat(2)= x(2);

l(1)=1;
l(2)=2;
i=2;
for j=2:Ddim1
    count=Data_loop(j-1,1);
    occu=Data_loop(j-1,2)/100;
    Length_bar(j) = 30/5280;
    if count == 0
    else
    i= i +1;
    l(i)=i;
    z(i)=occu/count;

    X_ss=[x_hat(i-1),x_hat(i-2)]';
    x_hat(i)=0;

   z_a(i)=z(i)-(Length_bar(i-1)/T)*(x_hat(i-1).^2+Varq(i-1).^2)/x_hat(i-1).^3-
((Length_bar(i-1)/T)*(x_hat(i-1).^2+3*Varq(i-1).^2)/x_hat(i-1).^4)*x_hat(i-1);

    X_s=[x_hat(i),x_hat(i-1)]';
    Z_m=[z_a(i),z_a(i-1)]';
    Z_mm=[z(i),z(i-1)]';

    Varq(i)=Varq(i-1);
    Varr(i)=Varr(i-1);
    Q=[Varq(i),0;0,Varq(i)];
```

```matlab
    R=[Varr(i),0;0,Varr(i)];

    H_p=[(Length_bar(i-1)/T)*(x_hat(i-1).^2+Varq(i-1).^2)/x_hat(i-1).^3,(Length_bar(i-
2)/T)*(x_hat(i-2).^2+Varq(i-2).^2)/x_hat(i-2).^3]';

    H_m=[-(Length_bar(i-1)/T)*(x_hat(i-1).^2+3*Varq(i-1).^2)/x_hat(i-1).^4,0;0,-
(Length_bar(i-2)/T)*(x_hat(i-2).^2+3*Varq(i-2).^2)/x_hat(i-2).^4];

    %time update
    X_s=G*X_ss;
    P_pri=G*P_post*G'+Q;

    %Measurement update
    K= P_pri*H_m'*inv(H_m*P_pri*H_m'+R);
    P_post=P_pri-K*H_m*P_pri;

    z_m=Z_mm - H_p + H_m*X_ss;

    X_a=K*(Z_m - H_m*G*X_s);
    X_aa=[X_a(1,1),0]';
    X_s=X_s+X_aa;

    x_hat(i)=X_s(1,1);
    z_a(i)=Z_m(1,1);
    end
end

xh_fil = (x_hat(1,2:(j+1)))';
xh_actu = Data_loop(:,3);
plot(1:j,xh_fil,1:j,xh_actu)
axis([1 j 0 90 ]) ;
xlabel('Time Interval')
ylabel('Speed (mph)')

for i=1:j
 error(i) = xh_fil(i)-xh_actu(i);
end
MAE = mean(abs(error))

RMSE(1) = sqrt(mean((xh_fil(1:j,1)-xh_actu(1:j,1)).^2));
fprintf('%d:%d  Root-mean-square-error (RMSE) of estimate : %4.3f\n', 1, 1, rmse(1));

var_RMSE = var((xh_fil(1:j,1)-xh_actu(1:j,1)));
fprintf('%d:%d  Variance of estimate errors: %4.3f\n', 1, 1, var_RMSE);
```

UNSCENTED KALMAN FILTER

```
%This program is used for speed estimation using the UKF algorithm.

clc;
clear all;

load data.dat; %load the .DAT file
Data_loop = data;

Si=size(Data_loop);
Ddim1=Si(1,1);
Ddim2=Si(1,2);

T = 30/3600;
var_sp = 5;

j=0;

for i=1:Ddim1
    count_r = Data_loop(i,1);
    occu_r = Data_loop(i,2);
    if (count_r~=0) & (occu_r~=0)
        j = j + 1;
        count(j) = count_r;
        occu(j) = occu_r/100;
        veh_l(j) = 30/5280;
        Y(j) = occu(j)/count(j);
    end
end

Xdim = 1;    %state dimension
Odim = 1;    %observation dimension
U1dim = 0;   %state input dimension
U2dim = 0;   %observation input dimension
Vdim = 1;    %state noise dimension
Ndim = 1;    %observation noise dimension

mean_RMSE = zeros(1,1);  % buffer for MC results for each algorithm
var_RMSE  = zeros(1,1);

N = j;       % number of observed data
```

```matlab
alpha = 1;   % scale factor (UKF parameter)
beta  = 2;   % optimal setting for Gaussian priors (UKF parameter)
kappa = 0;   % optimal for state dimension=2 (UKF parameter)


xh = zeros(1,N);    % state estimation buffer
xh(1,1) = 60;        % initial estimate of state E[X(0)]
Px = 0.5;            % initial state covariance

xh_  = zeros(1,N);
yh_  = zeros(1,N);
inov = zeros(1,N);

L = Xdim + Vdim + Ndim;       % augmented state dimension
nsp = 2*L+1;                  % number of sigma-points
kappa = alpha^2*(L+kappa)-L;  % compound scaling parameter

W = [kappa 0.5 0]/(L+kappa);   % sigma-point weights
W(3) = W(1) + (1-alpha^2) + beta;

Sqrt_L_plus_kappa = sqrt(L+kappa);

Zeros_Xdim_X_Vdim     = zeros(1,1);
Zeros_Vdim_X_Xdim     = zeros(1,1);
Zeros_XdimVdim_X_Ndim = zeros(2,1);
Zeros_Ndim_X_XdimVdim = zeros(1,2);

for i=2:(N+1)
   if (U1dim==0), UU1=zeros(0,nsp); end
   if (U2dim==0), UU2=zeros(0,nsp); end

   % TIME UPDATE
   if i==2
     Z   = cvecrep([xh(1,i-1); 0; 0], nsp);
   else
     Z   = cvecrep([(xh(1,i-1)+xh(1,i-2))/2; 0; 0], nsp);
   end

   if Px==0
      Sx = 0;
   else
     Sx = chol(Px); % sqrt of state error covariance
   end
```

```matlab
    if var_sp==0
       Sv = 0;
    else
     Sv = chol(var_sp);   %sqrt of state noise covariance
    end

    if var_Y(i-1)==0
       Sn = 0;
    else
     Sn = chol(var_Y(i-1));  %sqrt of observation noise covariance
    end

    SzT  = [Sx Zeros_Xdim_X_Vdim; Zeros_Vdim_X_Xdim Sv];
    Sz   = [SzT Zeros_XdimVdim_X_Ndim; Zeros_Ndim_X_XdimVdim Sn];
    sSz  = Sqrt_L_plus_kappa * Sz;
    sSzM = [sSz -sSz];
    Z(:,2:nsp) = Z(:,2:nsp) + sSzM;  % build the sigma-point set

    %-- Calculate predicted state mean
    X_ = Z(1,:) +Z(2,:);  %get predicted state
    X_bps = X_;
    xh_(:,i) = W(1)*X_(:,1) + W(2)*sum(X_(:,2:nsp),2);

    noise(i-1) = var_sp;
    temp1 = X_ - cvecrep(xh_(:,i),nsp);

     Px_ = W(3)*temp1(:,1)*temp1(:,1)' + W(2)*temp1(:,2:nsp)*temp1(:,2:nsp)'; %priori
state error covariance

     for k=1:nsp
       Y_(1,k) = (var_sp.^2+X_bps(1,k)^2).*veh_l(i-1)/(T*X_bps(1,k)^3)+Z(3,k);   %
propagate through observation model
     End

    %-- Calculate predicted observation mean
    yh_(:,i) = W(1)*Y_(:,1) + W(2)*sum(Y_(:,2:nsp),2);
    temp2 = Y_ - cvecrep(yh_(:,i),nsp);

    Py  = W(3)*temp2(:,1)*temp2(:,1)' + W(2)*temp2(:,2:nsp)*temp2(:,2:nsp)';

    % MEASUREMENT UPDATE
    Pxy = W(3)*temp1(:,1)*temp2(:,1)' + W(2)*temp1(:,2:nsp)*temp2(:,2:nsp)';
    KG(i) = Pxy / (Py);
    inov(:,i) = Y(:,i-1) - yh_(:,i);
```

```
   xh(:,i) = xh_(:,i) + KG(i)*inov(:,i);
   Px = Px_ - KG(i)*Py*KG(i)';
end

xh_fil = xh(1,2:(j+1))';
xh_actu = Data_loop(:,3);
plot(1:j,xh_actu,1:j,xh_fil)
axis([1 j 0 90 ]) ;
xlabel('Time Interval')
ylabel('Speed (mph)')

for i=1:j
   error(i) = xh_fil(i)-xh_actu(i);
end
MAE = mean(abs(error))

RMSE(1) = sqrt(mean((xh_fil(1:j,1)-xh_actu(1:j,1)).^2));
fprintf('%d:%d  Root-mean-square-error (RMSE) of estimate : %4.3f\n', 1, 1, rmse(1));

var_RMSE = var((xh_fil(1:j,1)-xh_actu(1:j,1)));
fprintf('%d:%d  Variance of estimate errors: %4.3f\n', 1, 1, var_RMSE);
```

UNSCENTED PARTICLE FILTER

```matlab
%This program is used for speed estimation using the UPF algorithm.

clc;
clear all;

load data.dat;%
Data_loop = data;

Si=size(Data_loop);
Ddim1=Si(1,1);
Ddim2=Si(1,2);

T = 30/3600;
var_sp = 5;
j=0;

for i=1:Ddim1
    count_r = Data_loop(i,1);
    occu_r = Data_loop(i,2);
    if (count_r~=0) & (occu_r~=0)
        j = j + 1;
        count(j) = count_r;
        occu(j) = occu_r/100;
        veh_l(j) = 30/5280;
        Y(j) = occu(j)/count(j);
    end
end

Xdim = 1;
Odim = 1;
U1dim = 0;
U2dim = 0;
Vdim = 1;
Ndim = 1;

mean_RMSE = zeros(1,1);
var_RMSE  = zeros(1,1);

NOV = j;

N=100;                          %number of particles
```

```matlab
alpha = 1;          % scale factor (UKF parameter)
beta  = 2;           % optimal setting for Gaussian priors (UKF parameter)
kappa = 0;            % optimal for state dimension=2 (UKF parameter)


xh(1,1) =60 ;
x_temp2 = ones(1,N)*60;      %"x  = ParticleFilterDS.particles; "
Px = .5;                      %initial state noise covariance
Sx = ones(1,N)*Px;           %"Sx = ParticleFilterDS.particlesCov; "

weights = cvecrep(1/N,N);      % Initial particle weights = 1/N
normWeights = cvecrep(1/N,N);
estimate   = zeros(Xdim,NOV);
SxPred  = zeros(Xdim,Xdim,N);
xNew    = zeros(Xdim,N);
xPred   = zeros(Xdim,N);

ones_numP = ones(N,1);
ones_Xdim = ones(1,Xdim);

proposal = zeros(1,N);

normfact = (2*pi)^(Xdim/2);              %sqrt of 2*pi

for i=2:NOV+1,
  OBStemp = Y(i-1);               % inline cvecrep, the first obs value
  OBS = OBStemp(:,ones_numP);

  randBuf = randn(Xdim,N)/5;

  for k=1:N,
      %Start of UKF
      %Obtain the proposal distribution (xh(:,k)) from the UKF.
      %End of UKF
      xNew(:,k) = xh(:,k);
      SxPred(:,:,k) = Px;
      xPred(:,k) = xNew(:,k) + SxPred(:,:,k)*randBuf(:,k);
  end


  %Start of prior
   if i==2
     x_temp   = xh(1,i-1);
```

```matlab
    else
       x_temp    = (xh(1,i-1)+xh(1,i-2))/2;
    end
    x_temp2 = ones(1,N)*x_temp;

    x_noise = xPred - x_temp2;

    prior = x_noise.^(alpha-1).*exp(x_noise*(-1/beta)) + 1e-99;
    %End of prior

    %Start of likelihood
    for ii=1:N,
      Y_temp(1,ii) = (var_sp.^2+xPred(1,ii).^2).*veh_l(i-1)/(T*xPred(1,ii).^3);
    End

    Y_noise = OBS - Y_temp;
    likelihood = zeros(1, N);      % preallocate likelihood matrix
    foo = Sn.^2 ./ Y_noise;
    likelihood = exp(-0.5./sum(foo.*foo, 1))./(normfact*abs(prod(diag(Sn.^2))));
    %End of likelihood

    difX = xPred - xNew;
    for k=1:N,
       cholFact = SxPred(:,:,k);
       foo2 = cholFact \ difX(:,k);
       proposal(k) = exp(-0.5*foo2'*foo2) / abs(normfact*prod(diag(cholFact))) + 1e-99;
       weights(k) = weights(k) * likelihood(k) * prior(k) / proposal(k);
    end

    if sum(weights)<1e-10
       weights = cvecrep(1/N,N);
    else
       weights = weights / sum(weights);
    end

    %calculate estimate
    muFoo = sum(weights(ones_Xdim,:).*xPred,2);
    estimate(:,i) = muFoo;        % expected mean

    %Resample
    S = 1/sum(weights.^2);     % calculate effective particle set size
    if (S < N)                 % resample if S is below threshold
       outIndex  = residualresample(1:N,weights);
       x_temp2 = xPred(:,outIndex);
```

```
    for k=1:N,
        Sx(:,:,k) = SxPred(:,:,outIndex(k));
    end
    weights = normWeights;
  else
    x_temp2  = xPred;
    Sx = SxPred;
  end

  xh(:,i) = estimate(:,i);
  Px = Sx(1);

end

xh_fil = estimate(1,2:(j+1))';
xh_actu = Data_loop(:,3);
plot(1:j,xh_actu,1:j,xh_fil)
axis([1 j 0 90 ]) ;
xlabel('Time Interval')
ylabel('Speed (mph)')

for i=1:j
  error(i) = xh_fil(i)-xh_actu(i);
end
MAE = mean(abs(error))

RMSE(1) = sqrt(mean((xh_fil(1:j,1)-xh_actu(1:j,1)).^2));
fprintf('%d:%d  Root-mean-square-error (RMSE) of estimate : %4.3f\n', 1, 1, rmse(1));

var_RMSE = var((xh_fil(1:j,1)-xh_actu(1:j,1)));
fprintf('%d:%d  Variance of estimate errors: %4.3f\n', 1, 1, var_RMSE);
```

VITA

Name: Zhirui Ye

Permanent Address   Nanjing, Jiangsu Province, China 210011

Email Address: zhirui@gmail.com

Education   Ph.D., Civil Engineering, Texas A&M University,
                             December 2007
                             M.S., Civil Engineering, Southeast University,
                             Nanjing, China, May 2003
                             B.S., Civil Engineering, Southeast University,
                             Nanjing, China, July 2000