Final Research Report
AVL-Equipped Vehicles as Traffic Probe Sensors
T9903-99 Transit Vehicles as Probes

AVL-Equipped Vehicles as Traffic Probe Sensors

By

Daniel J. Dailey
Associate Professor
University of Washington
Dept. of Electrical Engr.
Seattle, Washington 98195

Fredrick W. Cathey
Research Scientist
University of Washington
Dept. of Electrical Engr.
Seattle, Washington 98195

Washington State Transportation Center (TRAC)
Univeristy of Washington, Box 354802
University District Building, Suite 535
1107 N.E. 45th Street
Seattle, Washington 98105-4631

Washington State Department of Transportation
Technical Monitor
Pete Briglia
ITS Program Manager

Sponsored by

Washington State Transportation Commission
Department of Transportation
Olympia, Washington 98504-7370

Transportation Northwest (TransNow)
University of Washington
135 More Hall, Box 352700
Seattle, Washington 98195-2700

in cooperation with
U.S. Department of Transportation
Federal Highway Administration

January 2002

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
WA-RD 534.1		

4. TITLE AND SUBTITLE	5. REPORT DATE
TRAFFIC VEHICLES AS TRAFFIC PROBE SENSORS	January 2002
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT NO.
Daniel J. Dailey, Fredrick W. Cathey	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NO.
Washington State Transportation Center (TRAC) University of Washington, Box 354802 University District Building; 1107 NE 45th Street, Suite 535 Seattle, Washington 98105-4631	11. CONTRACT OR GRANT NO. Agreement T9903, Task 99
12. SPONSORING AGENCY NAME AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
Research Office Washington State Department of Transportation Transportation Building, MS 47370 Olympia, Washington 98504-7370 Martin Pietz, Project Manager, 360-705-7974	Draft Research Report 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This study was conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. ABSTRACT

In this report, we present new algorithms that use transit vehicles as probes to determine traffic speeds and travel times along freeways and other primary arterials. We describe a mass transit tracking system based on automatic vehicle location (AVL) data and a Kalman filter to estimate vehicle position and speed. We also describe a system of "virtual" probe sensors that measure transit vehicle speeds using the track data. Examples showing the correlation between probe data and inductance loop speed trap data are presented. We also present a method that uses probe sensor data to define vehicle speed along an arbitrary roadway as a function of space and time, a speed function. We present the use of this speed function to estimate travel time given an arbitrary starting time. Finally, we introduce a graphical application for viewing real-time speed measurements from a set of virtual sensors that can be located throughout King County on arterials and freeways.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Automatic vehicle location (AVL), Kalman filter, transit vehicles, speed sensors	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22616

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
None	None		

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. This document is disseminated through the Transportation Northwest (TransNow) Regional Center under the sponsorship of the U.S. Department of Transportation UTC Grant Program and through the Washington State Department of Transportation. The U.S. government assumes no liability for the contents or use thereof. Sponsorship for the local match portion of this research project was provided by the Washington State Department of Transportation. The contents do not necessarily reflect the official views or policies of the U.S. Department of Transportation or Washington State Department of Transportation. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

Disclaimer ..... **i**

1. Introduction **1**

2. Transit Database and AVL Data **4**

3. Tracker Component **6**

4. Kalman Filter Models **7**

4.1 Determination of Filter Parameters *12*

5. Probes **14**

5.1 GIS Index System *15*

5.2 Sensor Generator *17*

6. Results **19**

6.1 ProbeView *19*

6.2 Analysis *20*

7. Conclusions **28**

References **29**

LIST OF FIGURES

Figure 1.1: Data flow diagram for ProbeView. 2
Figure 2.1: Time series of reported distance into trip. 5
Figure 4.1: Time series of distance into trip. 10
Figure 4.2: Time series of estimated speed. 11
Figure 4.3: Residuals betweeen measurements and estimates. 11
Figure 4.4: Smoother velocity error estimates. 11
Figure 5.1: Chain of oriented arcs. 15
Figure 5.2: Sensor Generator snapshot. 18
Figure 6.1: ProbeView snapshot. 19
Figure 6.2: ProbeView data snapshot. 20
Figure 6.3: Probe data on Wednesday, June 13, 2001, (left) and on Thursday, June 14, 2001 (right), Aurora Avenue North. 21
Figure 6.4: Probe data on Friday, June 25, 2001, I-5 South. 21
Figure 6.5: Probe and speed trap data on Friday, June 15, 2001, I-5 South. 22
Figure 6.6: Corrected probe speeds for June 13, 14, and 15, 2001, I-5 South. 23
Figure 6.7: Probe and speed trap data on Wednesday, June 13, 2001, SR 520. 24
Figure 6.8: Probe and lane 1 speed trap data on Wednesday, June 13, 2001, SR 520. 24
Figure 6.9: Three consecutive days of probe data, I-5 North. 25
Figure 6.10: Speed as a function of time and distance. 25
Figure 6.11: Contour plot of speed, darker is slower. 26
Figure 6.12: Travel time as a function of departure time. 27

1. INTRODUCTION

Performance monitoring is an issue of growing concern both nationally and in Washington State. Travel times and speeds have always been of interest to traveler-information researchers, planners, and public agencies; and because travel times and speeds are key measures in performance monitoring, this interest is now greater than ever. However, deploying inductance loops, cameras, and other sensors on the roadway infrastructure to obtain this type of data is very expensive, and hence an alternative source of travel time and speed data is desirable. In this report, we present transit vehicles as traffic probe sensors and develop a framework in which to use vehicle position estimates as a speed sensor. An optimal filter method is presented that estimates acceleration, speed, and position as a function of space and time.

The goals of the "transit vehicles as probes" effort by the University of Washington Intelligent Transportation Systems (ITS) research group include the following:

1. Create a mass-transit vehicle tracking system that computes smooth estimates of speed and distance traveled for each vehicle in the fleet.
2. Create a network of virtual "speed sensors" on selected road segments using the "probe" vehicle speed estimates produced by the tracking system.
3. Make the speed measurements publicly available for traffic analysis purposes and for data fusion with camera and inductance loop measurements.
4. Create graphical applications for visualization of current and historical traffic conditions based on probe speed data.
5. Create tools for predicting point-to-point travel times based on smoothed historical speed data.

In this report, we describe achievements made toward accomplishing these goals. We use the data from a transit management automatic vehicle location (AVL) system, the King County Metro Transit AVL System [1], to construct "virtual sensors" that produce smoothed speed estimates. While the King County Metro AVL system uses a "dead reckoning" method to determine vehicle location, our probe sensor framework can also be used with AVL systems that employ the Global Positioning System (GPS).

Figure 1.1 is a data-flow diagram of the deployed system's architecture. The basic components are as follows: (1) a Tracker, (2) a Probe Speed Estimator, and (3) a display

application, ProbeView. The Tracker process receives the real-time stream of AVL position reports and outputs a corresponding stream of "Track" reports, including filtered vehicle position and speed estimates. The Probe Speed Estimator receives the track data and outputs probe speed reports when vehicles cross specified locations. The data flow between the components adheres to the self-describing data (SDD) protocol [2].

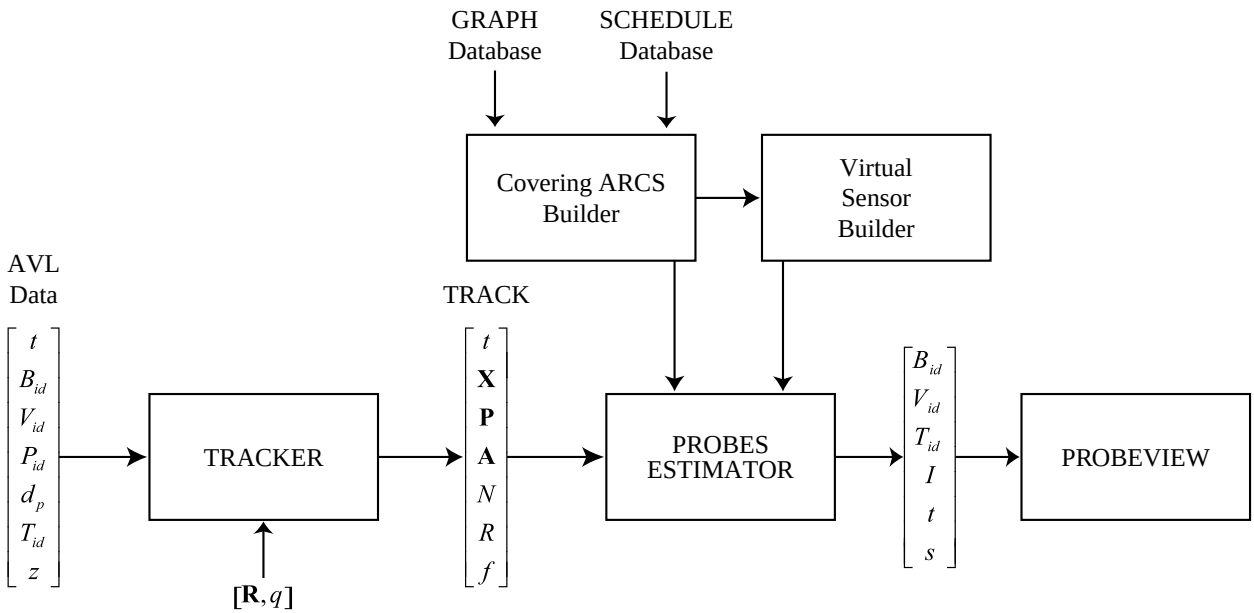


Figure 1.1: Data flow diagram for ProbeView.

To describe our algorithms, we first define terms and discuss essential concepts related to the transit scheduling system. This background information is necessary for understanding the AVL data as well as for presenting the problem of associating AVL data with actual road segments. We briefly describe the logic of the Tracker and present in more detail the framework for measurement and process models of the Kalman filter/smoother that estimates vehicle state (position, speed, and acceleration) from AVL position reports. The parameters for these models, variances of the measurement and process noise, were determined experimentally using a recently developed maximum marginal likelihood algorithm [3].

We discuss how the Probes Speed Estimator works to output vehicle speed estimates on specified roadways. To clarify the mapping of speed estimates onto roadway locations, we describe the geographical road segment database from which the bus schedule database is created. We introduce a GIS approach for organizing vehicle state estimates according to an

"oriented road segment" indexing system. We discuss the "Probe Sensor Generator," a map-based graphical tool that enables users to generate sensor locations, and we illustrate the ProbeView graphical map-based application for viewing real-time probe data and daily histories.

Finally, we present some examples and analysis. The first example shows the quality of the probe speed estimates at several locations. The second example compares inductance loop speed trap data with probe data at several locations and shows that probe sensor output is similar to loops data. In the third example, probe speed samples are collected in both time and space at locations along a chain of oriented road segments comprising a stretch of freeway. We construct a function from these data that gives speed as a bivariate function of time and distance, $dx/dt = f(x, t)$. Using integration techniques, we then estimate travel times between points at various times of day. These examples demonstrate that it is possible to create a speed and travel-time sensor using transit vehicles as probes.

2. TRANSIT DATABASE AND AVL DATA

Our basic assumptions in using a mass transit system as a speed sensor are as follows:

(1) There is a fleet of transit vehicles that travel along prescribed routes.

(2) There is a "transit database" that defines the schedule times and the geographical layout of every route and time point.

(3) There is an automatic vehicle location (AVL) system in which each vehicle in the fleet is equipped with a transmitter and periodically reports its progress back to a transit management center.

To clarify the terminology used here, a description of the AVL data as well as a conceptual description of the transit database is necessary. The database is described in terms defined by the ITS Transit Communications Interface Profile (TCIP). There are five relevant terms, which are as follows: (1) *time-point* (TP), (2) *time-point-interval* (TPI), (3) *pattern*, (4) *trip*, and (5) *block*. A *time-point* is a named location. The location is generally defined by two coordinates, either in Cartesian state-plane coordinates (as is the case in King County) or by geodetic latitude and longitude. Transit vehicles are scheduled to arrive or depart time-points at various times. A *time-point-interval* (TPI) is a named polygonal path directed from one time-point to another. The path is geographically defined by a list of "shape-points," where a shape-point is simply an unnamed location. Since one frequently needs to determine the distance of a vehicle along a path, each shape-point is augmented with its own distance into path. A *pattern* is a route specified by a sequence of TPIs, where the ending time-point of the i-th TPI is the starting time-point of the $(i+1)^{st}$ TPI. A *trip* is a pattern with an assignment of schedule times to each of the time-points on the pattern. A *block* is a sequence of trips. Each transit vehicle is assigned a block to follow over the course of the day. Some blocks are long and are covered by different vehicles at different times of day.

A typical AVL system will produce real-time reports of vehicle location based on technologies such as dead reckoning or satellite GPS position measurements. The King County Metro AVL system is based on a dead reckoning technique that uses odometer measurements with position corrections whenever the vehicle passes a "sign-post" that is a small radio beacon. However, the vehicle positioning technique is not critical, and GPS-based AVL systems can also

be used to generate the required information, as shown in Bell [3]. The AVL reports are created for each transit vehicle traveling the roadways and are transmitted at a rate of once per minute. These reports are the dynamic input data for the Tracker that is the first of the components used to produce the virtual sensor data presented here. Real-time reports from this system are freely available in the SDD format [4] from UW host "sdd" on port 8412. Archived data are available for post-processing at http://avllog.its.washington.edu:8080/.

A vehicle report produced by the King County Metro AVL system includes the following information: (1) time t, (2) block-identifier B_{id}, (3) vehicle-identifier V_{id}, (4) pattern-identifier P_{id}, (5) distance-into-pattern d_p, and (6) trip-identifier T_{id}. The block and vehicle identifiers are used by the tracker to correlate the report with the correct track. The pattern-identifier provides the link between vehicle distance-into-pattern and vehicle position on an identifiable road, permitting the geographical indexing of estimated speed. For tracking purposes, it is convenient to identify how far the vehicle has traveled into its block. In software upstream of the Tracker, we augment the AVL report with (7) distance-into-block z. Distance-into-block is simply the current distance-into-pattern plus the sum of the lengths of the preceding patterns on the block.

Figure 2.1 shows a time-series plot of reported distance-into-pattern for a vehicle traveling along its first trip of the day. Here, time is measured in minutes (min) after midnight and distance in feet. The circles represent scheduled time-points, and the continuous curve rising from left to right is the polygonal line joining these points. The short horizontal line segments are drawn to help visualize estimated schedule deviations of the vehicle.

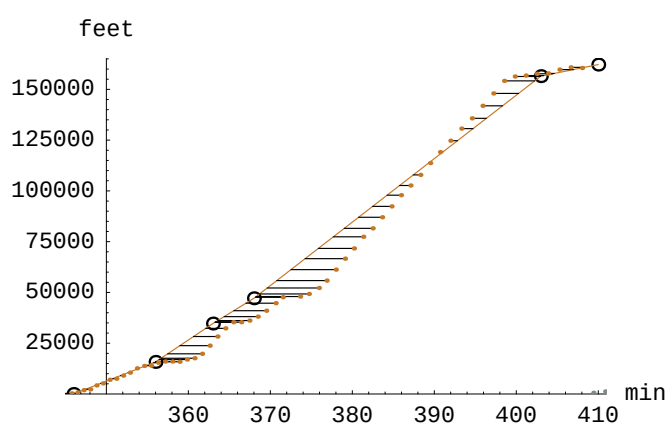


Figure 2.1: Time series of reported distance into trip.

3. TRACKER COMPONENT

The purpose of the Tracker component is to filter the stream of AVL reports and provide smoothed estimates of vehicle location and speed. Real-time track reports are available in SDD format from UW host "carpool.its.Washington.edu" on TCP port 9010. The tracker schema consist of two tables: the actual track data table and a static contents table that defines terms.

In order to perform its task, the Tracker maintains an internal "Track" data structure for each block identified in the transit schedule database. A Track consists of the following: (1) time t, (2) Kalman filter state $\mathbf{X}$ and covariance $\mathbf{P}$, (3) last AVL report $\mathbf{A}$, (4) number of updates N, (5) number of consecutive rejected reports R, and (6) speed validity flag f. For each AVL report received, the Tracker determines the Track with corresponding block-identifier and then takes one of three actions: Track *initialization*, Track *update*, or report *rejection*.

Track *initialization* consists of recording the report, initializing a Kalman filter, zeroing counters, and marking speed as not valid (it takes at least two measurements to determine speed). Tracks are initialized for a number of reasons. For example, a Track "ages out" if the time since last update is greater than a specified threshold and so must be initialized as soon as fresh data have been received. A Track will also be re-initialized if the reported vehicle id changes or there is a large change in reported distance due to a "sign post" correction. Finally, a Track is re-initialized if two reports in a row have been rejected for reasons discussed below.

The Track *update* consists of recording the report, updating the Kalman filter, zeroing the rejection count, and marking speed as valid. However, before an update is begun, some sanity checks are performed. To guard against false alarms and wild measurements, we perform a X^2 residual test, which determines whether the difference between reported and predicted distance is reasonable. If the residual is too large, the report is *rejected*. A report is also *rejected* if updating the filter would result in an unreasonable speed or a non-positive definite covariance matrix.

AVL reports for a given vehicle are received at an average rate of one per minute, and track data for the vehicle are periodically propagated and output at a nominal rate of once every 20 seconds. Because of low sample rate, the starting and stopping of a bus at bus stops is not generally observable. This "fine grain" behavior is compensated for in the noise terms of the Kalman filter [4].

4. KALMAN FILTER MODELS

Within the Tracker, we use a Kalman filter to transform a sequence of AVL measurements into smooth estimates of vehicle dynamical state, including vehicle speed. This filter requires a measurement and a process model. These models depend on several parameters, including the variances for measurement and process noise. We estimated representative values experimentally using the method of maximum marginal likelihood, as specified in Bell [3].

To implement a Kalman filter, the following must be specified: (1) a state-space, (2) a measurement model, (3) a state transition model, and (4) an initialization procedure. Once these items have been specified, one may employ any one of a number of implementations of the Kalman filter/smoother equations (see Bell [3], Jazwinski [6], Tung et al. [7], or Press et al. [8]) to transform a sequence of measurements into a sequence of vehicle state estimates.

To represent the instantaneous dynamical state of a vehicle, we selected a 3-dimensional state space. We denote a vehicle state vector by

$$X = \begin{pmatrix} x & v & a \end{pmatrix}^T \quad , \tag{4.1}$$

where x is distance into block, v is speed, and a is acceleration. (The superscript T denotes transpose.) We use the foot as unit of distance and minute as unit of time. A measurement, z, provided by the AVL system is simply a noisy estimate of the vehicle's distance into block, and our measurement model is given by

$$z = HX + \varepsilon = x + \varepsilon \quad . \tag{4.2}$$

Here H = (1 0 0) is the "measurement matrix," and ε denotes a random measurement error, assumed to have a Normal distribution with variance R. The variance is treated as a model parameter with nominal value of R=(500ft)^2.

We assume a simple dynamic model for the evolution of state defined by the first order system of linear stochastic differential equations,

$$\begin{aligned} dx &= v\ dt \\ dv &= a\ dt \quad . \\ dz &= dw \end{aligned} \tag{4.3}$$

Here dt is the differential of time, and dw is the differential of Brownian motion representing randomness in vehicle acceleration. By definition of Brownian motion (see Chapter 3, Section 5 of Jazwinski [6]), the expectation

$$E\left(dw^2\right) = q^2 dt \quad , \tag{4.4}$$

where q^2 is a model parameter. In the absence of a measurement correction, the variance of acceleration grows linearly with time. We selected a value for q^2 of (264 ft/min^2)^2/min = (3 mph/min)^2/min

The differential equations (4.3) are written in vector form as follows

$$dX = F\ X\ dt + G\ dw \quad , \tag{4.5}$$

where

$$F = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \qquad G = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad . \tag{4.6}$$

Integrating over a time interval (t, $t + \delta t$), we obtain the state transition model

$$X\left(t + \delta t\right) = \Phi\left(\delta t\right) X\left(t\right) + W\left(\delta t\right) \quad , \tag{4.7}$$

where $X(t)$ and $X(t + \delta t)$ denote vehicle state values at times t and $t + \delta t$, respectively, and $W(\delta t)$ is the accumulated error in X over δt.

$$\Phi\left(\delta t\right) = \exp\left(F \delta t\right) = \begin{pmatrix} 1 & \delta t & \delta t^2/2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{4.8}$$

is the “transition matrix.” The accumulated error $W(\delta t)$ has covariance

$$Q(\delta t) = \begin{pmatrix} \delta t^5/20 & \delta t^4/8 & \delta t^3/6 \\ \delta t^4/8 & \delta t^3/3 & \delta t^2/2 \\ \delta t^3/6 & \delta t^2/2 & \delta t \end{pmatrix} q^2 \quad . \tag{4.9}$$

See the text following Theorem 7.1 of Jazwinski [6] for a discussion of integration.

Finally, to run a Kalman filter, we need an initialization procedure, a method for computing an initial value for the state vector and its associated error covariance matrix. These initial values, $\hat{X}_0$ and P_0, are based on the initial measurement z_0 (at time t_0) and measurement variance R. We set $\hat{X}_0 = \begin{pmatrix} z_0 & 0 & 0 \end{pmatrix}$ and set P_0 equal to a diagonal matrix with entries R, (30 mph)^2 and (16 mph/min^2). The initial variances for speed and acceleration were based on engineering judgment and selected large enough so that subsequent measurements forced initial errors to decrease rapidly.

Now, given a sequence of measurements $z_1, z_2, \ldots z_N$ at times $t_1, t_2, \ldots t_N$, the Kalman filter transition equations are

$$\begin{aligned} \hat{X}_k^- &= \Phi_k \hat{X}_{k-1} \\ P_k^- &= \Phi_k P_{k-1} \Phi_k^T + Q_k \end{aligned} \quad , \tag{4.10}$$

where $\hat{X}_k^-$ is the prediction of the state vector at the kth step, $\Phi_k = \Phi(t_k - t_{k-1})$ is the transition matrix between the k-1 to the kth step, and $Q_k = Q(t_k - t_{k-1})$ is the corresponding transition covariance matrix. The data update equations are

$$\begin{aligned} \hat{X}_k &= \hat{X}_k^- + K_k\left(z_k - H\hat{X}_k^-\right) \\ P_k &= \left(I - K_k H\right)P_k^- \end{aligned} \quad , \tag{4.11}$$

where

$$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1} \quad . \tag{4.12}$$

A residual χ^2 test is performed before update to guard against bad measurements. Let $v = z_k - H\hat{X}_k^-$ denote the residual and $S = \left(HP_k^- H^T + R\right)^{-1}$ the residual covariance. Then $\chi = v \big/ \sqrt{(S)}$ should have a Gaussian distribution, and the update is rejected if χ^2 is too large (e.g., $\chi^2 > 3^2$).

The equations for Kalman "smoothed" estimates of state $\overline{X}_k$ for k=1, … N are given by the backward recurrence

$$\overline{X}_{k-1} = \hat{X}_{k-1} + P_{k-1}\Phi_k^T\left(P_k^-\right)^{-1}\left(\overline{X}_k - \hat{X}_k^-\right) \ , \tag{4.13}$$

where $\overline{X}_N = \hat{X}_N$. See Tung et al. [7] for a derivation of the Kalman filter/smoother formulas.

For our real-time application, we used the Kalman filter equations above. For data analysis and parameter estimation, we used an implementation of the Kalman smoother that is identified as Algorithm 4 of Bell [3]. This algorithm, in addition to computing the smoothed estimates of state, also computes the marginal likelihood of the measurement sequence.

Figures 4.1 and 4.2 show example time series plots of vehicle distance into trip and speed, in miles per hour (mph), as produced by the Kalman smoother. Figure 4.3 is a plot of residuals, the measurement minus the prediction. The straight horizontal lines indicate the 500-ft measurement uncertainty, and the slightly slanted horizontal lines show the 1-sigma distance error estimate computed by the smoother. Figure 4.4 shows the plot of the 1-sigma speed error estimated by the smoother. The AVL reports for this example span a time-point-interval containing a stretch of freeway.

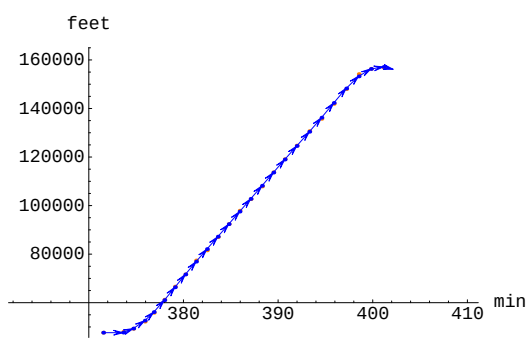


Figure 4.1: Time series of distance into trip.

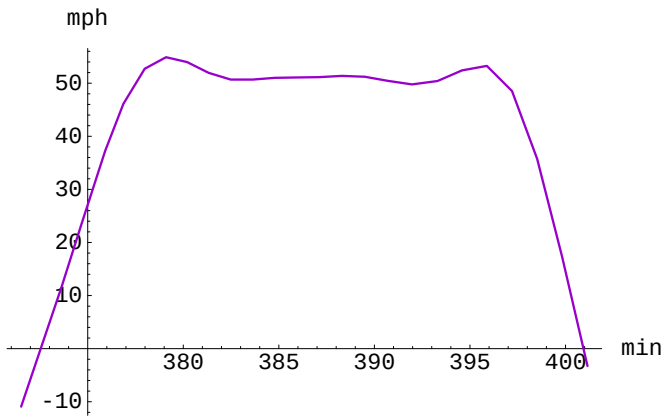


Figure 4.2: Time series of estimated speed.

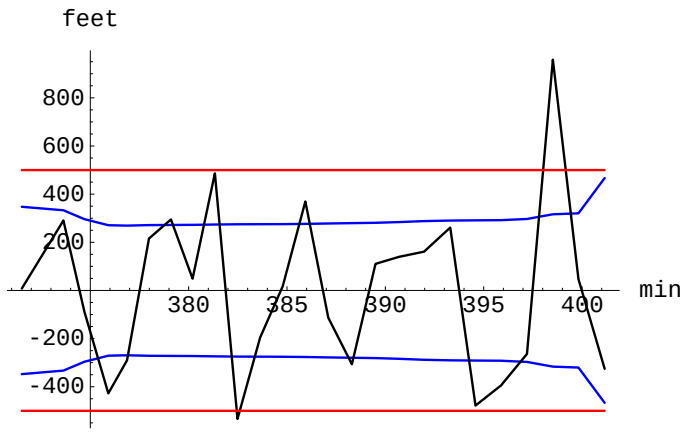


Figure 4.3: Residuals between measurements and estimates.

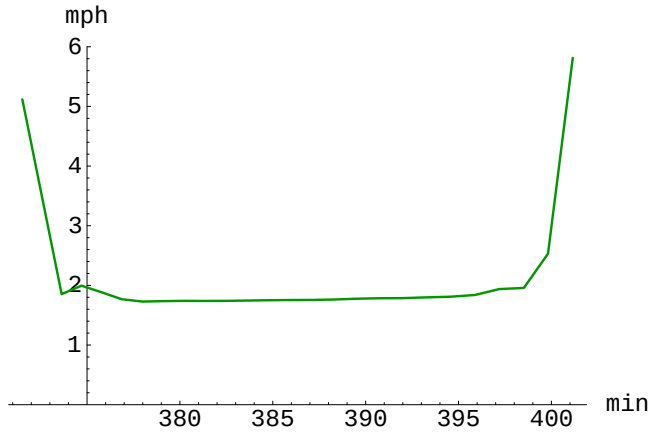


Figure 4.4: Smoother velocity error estimates.

The filter approach just presented relies on a set of parameters that reflect the variability in the measurement and the state. In many applications, these are input as "engineering judgment." In the next section, we describe a method to obtain a "best" set of these parameters.

4.1 DETERMINATION OF FILTER PARAMETERS

Our measurement and process models depend on two parameters: the measurement variance, R and the rate of change of the variance of process noise, q^2 . Using the method of "maximum marginal likelihood," we obtained "optimal" values for these parameters in a number of experiments with different measurement sequences. Our goal was not to perform an exhaustive statistical analysis but rather to find some representative values that would give reasonable filter performance. Note that the parameters determined in this manner don't necessarily have a physical interpretation, but they do provide an optimal fit of measurement data to a smooth trajectory.

Although the method of maximum marginal likelihood for parameter estimation is well known to statisticians, the fact that it can be used effectively for estimating parameters in the setting of Kalman-Bucy filters is not widely reported. This method was first proposed in Bell [3], where an effective procedure is provided for evaluating the marginal likelihood function. We briefly describe the theory.

Let **Z** and **X** denote vector-valued random variables representing a measurement sequence and a corresponding sequence of state vectors, and let ξ denote the parameter vector (R, q^2). A formula is given in Bell [3] (Eq. 4) for the joint probability density function $p_{\mathbf{Z},\mathbf{X}}(z,x;\xi)$ in terms of the measurement and process models like those described above. (The symbols z and x in this context denote real-valued vectors, and usage is not to be confused with that in the preceding section.) The cited reference provides an algorithm (Algorithm 4) which, when given a measurement vector z and parameter vector ξ, simultaneously computes the maximum likelihood estimate for the state vector sequence (the Kalman smoother estimate)

$$x^*_{z,\xi} = \arg\max_x \left\{ p_{\mathbf{X}|\mathbf{Z}}\left(x|z;\xi\right) \right\} , \tag{4.14}$$

and evaluates the marginal density function

$$p_{\mathrm{Z}}(z;\xi) = \int p_{\mathrm{Z,X}}(z,x;\xi)dx \quad . \tag{4.15}$$

As usual, the algorithm actually works in terms of the negative logs of the various probability densities.

In each of our experiments, we used the cited algorithm to define an objective function $f_z(\xi) = -\log\left(p_{\mathbf{z}}(z;\xi)\right)$, depending on the measurement sequence *z*, and then used Powell's conjugate direction minimization algorithm (Chapter 10, Section 5 of Press et al. [8]) to find the optimal parameter values for each measurement sequence *z*,

$$\xi_z^* = \arg\min_x \left\{ f_z(\xi) \right\} \quad . \tag{4.16}$$

We observed that the values obtained in each experiment were roughly the same and fluctuated around R = (500 ft)^2 and q^2 = (3mph/min)^2/min.

5. PROBES

In this section, we describe the procedure in use by the Probe Component to determine the times and speeds of transit vehicles as they cross specified “probe sensor” location. Real-time Probe reports are available from UW host “carpool.its.washington.edu” on TCP port 9011. The probes schema consists of four tables: the actual probe data table and three static tables providing definitions, sensor locations, and the road database.

Initially, Probes is supplied with a file of user-defined sensor locations. Rather than using state-plane coordinates, each location is specified by identifying a road-segment and a distance along the segment. This is accomplished using a “geographical information system” (GIS) road-segment database as described below. Using the specified location data, Probes determines blocks in the schedule database that will have vehicles crossing sensors and also determines the distance into block of each such crossing.

To determine when a vehicle actually crosses a sensor, and what its speed is, Probes maintains a simple data structure on each block consisting of the following data:

- block-identifier
- last track received
- next sensor crossing
- list of sensor crossings on block.

The basic idea of the Probes algorithm is that as successive track reports for a block are received, the reported distance-into-block is compared with the distance-into-block of the next sensor crossing. If the vehicle has crossed the sensor, then time and speed at the crossing are linearly interpolated from the current and preceding tracks. Of course, there are complicating issues that must be handled. Track re-initializations must be detected and the validity of the track speed must be monitored to ensure that time and speed are not improperly computed.

When vehicle data are output at a given sensor, it is important to also indicate which way the vehicle was moving along the road. We refer to the pair consisting of sensor location and direction of crossing as a GIS index.

Data transmitted on the Probe output stream include the following: (1) block-identifier B_{id}, (2) vehicle-identifier V_{id}, (3) trip-identifier T_{id}, (4) GIS index I, (5) time t, and (6) speed s.

5.1 GIS INDEX SYSTEM

In this section, we describe a geographical system for indexing traffic data. This system uses the "geographical information system" (GIS) road-segment database that underlies the construction of the bus schedule time-points and time-point-intervals described in Section 2. We note that the database for King County is based on U.S. Census Bureau TIGER files.

The GIS road segment database is a realization of a mathematical structure called a "directed graph." One set of "nodes" represents points on roads and one set of "arcs" represents segments of road between nodes. Like a time-point-interval, an arc has a start node and an end node and has a sequence of shape-points that define a polygonal path from start to end.

We specify a traffic "probe sensor location" in terms of the GIS database by identifying an arc and a distance along the arc. Note that traffic might move in two directions along the corresponding roadway or move opposite to the direction of the arc. We define a "GIS index" for traffic data to consist of a probe sensor location and a direction, or orientation. The orientation is +1 if the direction of traffic is along the direction of the arc and is -1 otherwise.

In order to assign GIS indices to transit vehicle data, we use the fact that scheduled vehicle paths are constructed from the GIS road segment database. The GIS database and the transit database are related as follows:

- item each time-point is a GIS node
- each time-point-interval (TPI) is the result of "welding together" a chain of oriented GIS arcs. See Figure 5.1.

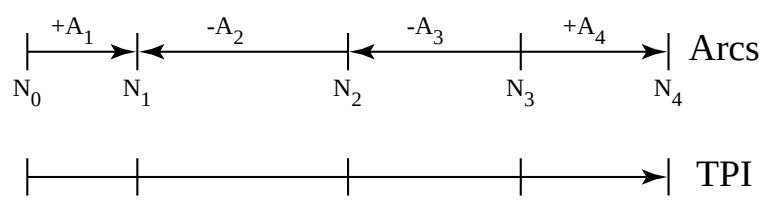


Figure 5.1: Chain of oriented arcs.

Now consider a probe sensor location specified by a distance d into some arc A. Suppose that $A = A_i$ occurs as the ith arc on some TPI and has orientation O with respect to this TPI. Then a formula for the distance of the sensor into the TPI, d_{TPI}, is given by

$$d_{\mathrm{TPI}} = \bar{d} + \sum_{j<i} \left| A_j \right| \quad , \tag{5.1}$$

where $|A_j|$ denotes the length of arc A_j and where

$$\bar{d} = \begin{cases} d & \text{if } O = 1 \\ \left| \mathrm{A_i} \right| - d & \text{otherwise} \end{cases} \tag{5.2}$$

Any transit vehicle traversing this TPI should produce a speed measurement with GIS index = $\{A, d, O\}$.

The transit database does not identify the chain of GIS arcs used to construct a TPI. The method we used to determine this information is described by the following algorithm.

- First, determine the unique GIS node, N_0, with the same position as the TPI start time-point. Also determine a GIS arc, A_1, which either starts or ends at N_0 and whose other node, N_1, has the same position as a TPI shape-point further down the list. The interior shape-points of the arc A_1 should coincide with shape-points on the TPI. If this arc is directed from N_0 to N_1, assign it the positive orientation, otherwise the negative orientation.

- Now assume inductively that $i > 0$ and we have determined the ith node, N_i, and oriented arc, $\pm A_i$, in the chain. Iterate the following step until N_i has the same position as the TPI end time-point.

- Find an arc, A_{i+1}, which either starts or ends at N_1 and whose other node, N_{i+1}, has the same position as a TPI shape-point further down the list. The interior shape-points on the arc should coincide with shape-points on the TPI. If this arc is directed from N_i to N_{i+1}, assign it the positive orientation, otherwise the negative. Set $i := i+1$.

We maintained the GIS nodes in a hash table keyed by position in order to efficiently look up a node given a position. In addition, each node had an associated list of incident arcs so that we could quickly determine which arcs started or ended at that node.

5.2 SENSOR GENERATOR

To make the selection of probe sensor locations a feasible task, we developed an interactive map-based graphical application. This program displays the GIS database in map form and allows the user to create and edit files of sensor locations. The program, written in Java 2, supports scrolling and zooming.

Sensors are created and deleted by “clicking the mouse” near a displayed road. If the mouse is near an existing sensor, then the sensor is deleted; otherwise, a new sensor is created. The algorithm described in Cathey and Dailey [10] is used to find the closest sensor location. This results in an arc and a distance into that arc, in state-plane coordinates, that corresponds to the mouse screen coordinates.

The application allows the user to view inductance loop cabinet locations so that probe sensors may be created nearby for comparative analysis. It is also possible to view the placement of TPIs so that no attempts are made to create sensors on roads with no transit vehicles.

Using this tool, we set up a network of probe sensors on the freeways and primary arterials of King County. To support analysis, and potential data fusion applications, a number of probe sensors are located next to inductance loop cabinets on freeways I-5, I-90, I-405, and SR 520. See Figure 5.2.

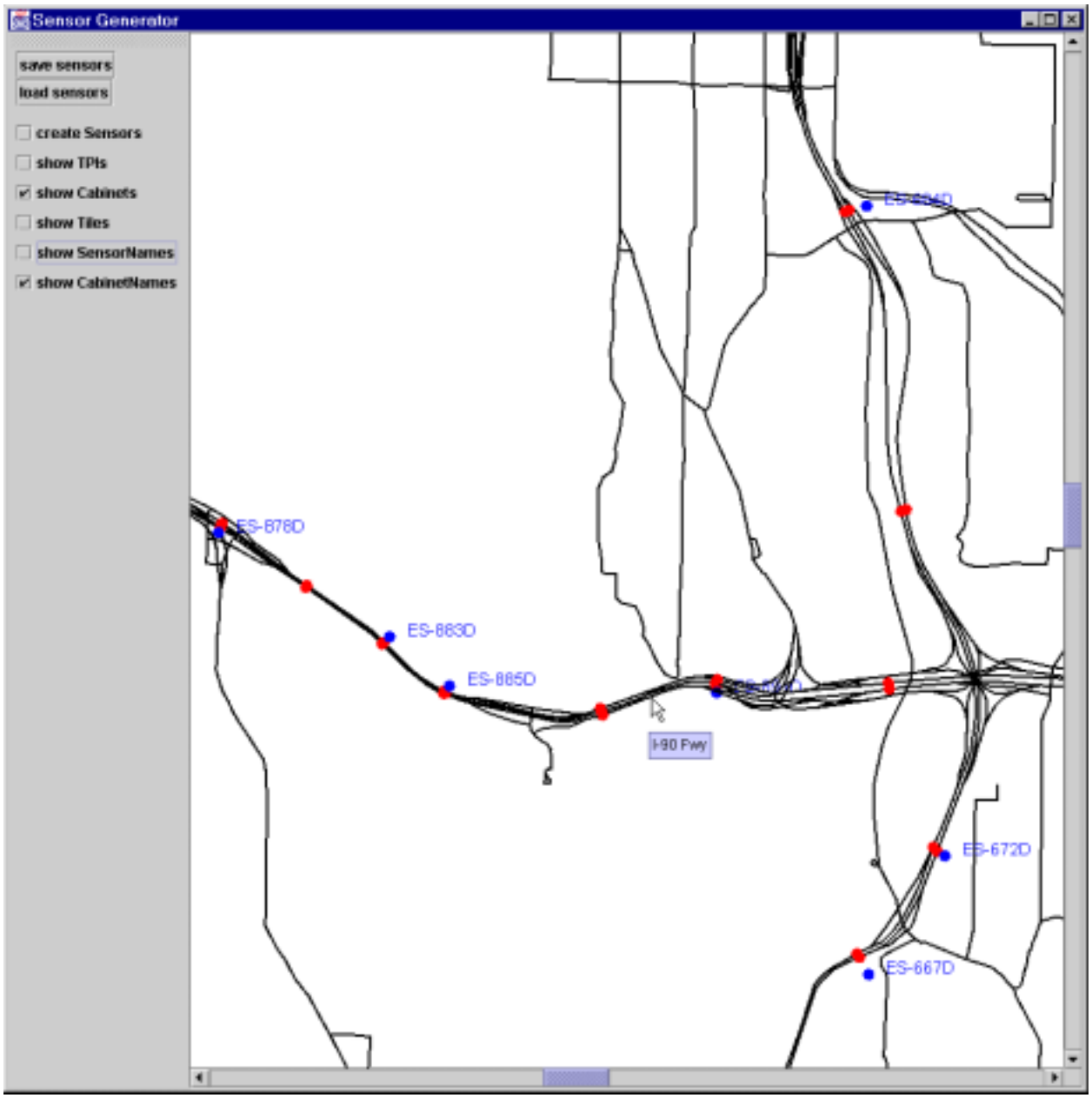


Figure 5.2: Sensor Generator snapshot.

6. RESULTS

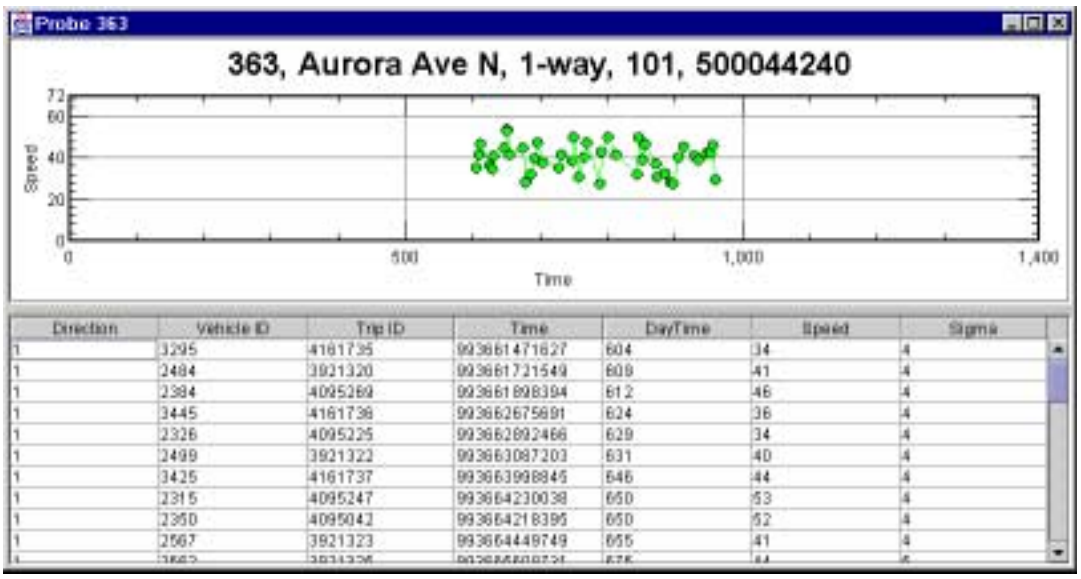
The character of the virtual sensor data produced by the framework just presented can be evaluated both quantitatively and qualitatively. To these ends, we first present a graphical presentation application created to allow real-time interaction with this data stream and then present a quantitative analysis of a representative set of data.

6.1 PROBEVIEW

A graphical map-based application was developed that connects to the Probes output stream and allows the user to view current transit vehicle speeds at all probe sensor locations in King County. For each sensor, speed of the most recent vehicle crossing is shown in a small bubble next to the sensor location, provided the time of the crossing is within 15 minutes of current time. Since vehicles in the U.S. generally travel on the right hand side of the road, a speed bubble is drawn on the right of a road segment if the vehicle is moving in the direction of the segment and is drawn on the left otherwise. However, for sensors on freeway express lanes, the speeds are centered directly on the sensor location. See Figure 6.1.

Figure 6.1: ProbeView snapshot.

The program has a programmed “tool tip” that detects mouse motion and displays the name of the closest road segment or information about the closest sensor. In Figure 6.1 the tool tip is displaying information about sensor 363 on Aurora Avenue North, which is on the one-way road segment whose ID is 500044240. When the left mouse button is pressed on this sensor, a window pops up showing a time series plot of speeds and a table of probe vehicle data. See Figure 6.2.



Direction	Vehicle ID	Trip ID	Time	DayTime	Speed	Sigma
1	3295	4161735	993661471627	604	34	4
1	2484	3921320	993661721549	609	41	4
1	2384	4095269	993661898394	612	46	4
1	3445	4161738	993662675691	624	36	4
1	2326	4095225	993662892466	629	34	4
1	2499	3921322	993663087203	631	40	4
1	3425	4161737	993663998845	646	44	4
1	2315	4095247	993664230038	650	53	4
1	2350	4095042	993664218395	650	52	4
1	2567	3921323	993664449749	655	41	4

Figure 6.2: ProbeView data snapshot.

6.2 ANALYSIS

For analysis purposes, probe data sets at all sensor locations were collected for three days, beginning Wednesday, June 13, and ending Friday, June 15, 2001.

Probe data have a large variability, and so we used a simple exponential smoothing filter on the data stream before comparison to other types of sensors. The transformation from raw speeds x_n to smoothed speeds y_n is defined recursively by the formula

$$y_n = \lambda y_{n-1} + (1-\lambda)x_n \text{ for } n > 0 \quad , \tag{6.1}$$

where $y_0 = x_0$. We used a smoothing factor of $\lambda = 0.7$. Note that since this filter is recursive, it is suitable for use in applications that process the real-time Probe data stream.

Figure 6.3 shows superpositions of smoothed and raw probe speed data collected on successive days at a point on Aurora Avenue North just south of the Aurora Bridge.

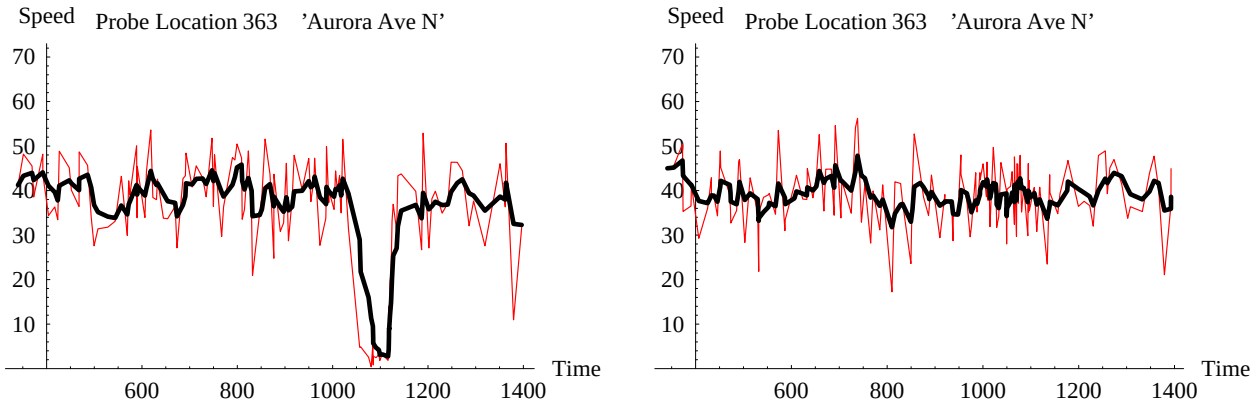


Figure 6.3: Probe data on Wednesday, June 13, 2001, (left) and on Thursday, June 14, 2001 (right), Aurora Avenue North.

Note that transit vehicle traffic appears to move at an average speed of about 40 mph all day long on both days, with the exception of an incident at time 1100 (6:20 p.m.) on Wednesday, June 13. On Wednesday, 134 samples were observed, and the mean error between the smoothed and raw speeds was -0.1 mph with a standard deviation of 5.3 mph. Similar performance was observed on Friday.

Figure 6.4 shows a superposition of smoothed and raw probe data on Friday, June 15, 2001, at probe location 417 on the I-5 South in Tukwila.

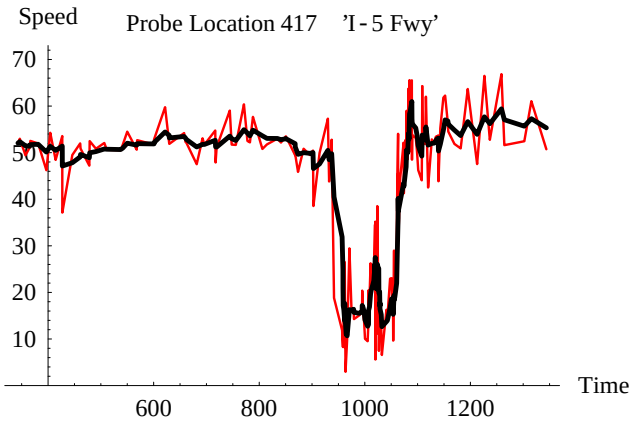


Figure 6.4: Probe data on Friday, June 25, 2001, I-5 South.

The location of probe sensor 417 was chosen next to a traffic management system (TMS) cabinet, which produces “speed-trap” traffic measurements using inductance loop sensors. Loop measurements from all TMS sensors are archived in the UW Traffic Data Acquisition and Distribution (TDAD) data-mine (see Dailey and Pond [11]). Using the TDAD Query Interface [11], we obtained speed-trap measurements on June 15, 2001, and compared them with the probe data. Figure 6.5 shows a time series of the smoothed probe data (dark line) superimposed over speed-trap data for four lanes of southbound traffic at cabinet ES-074. Linear interpolation functions were constructed for each lane of speed-trap data and then evaluated at probe report times.

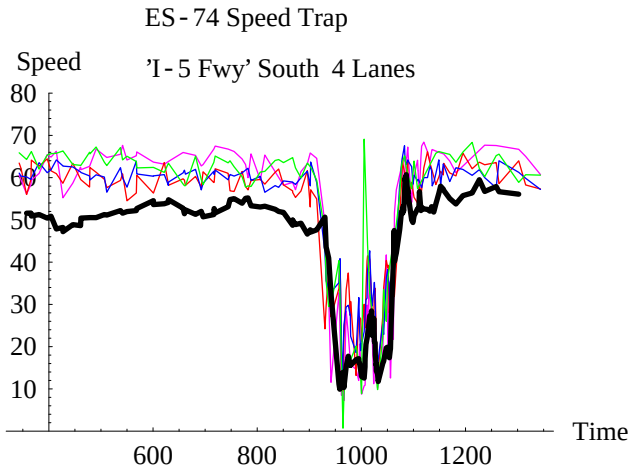
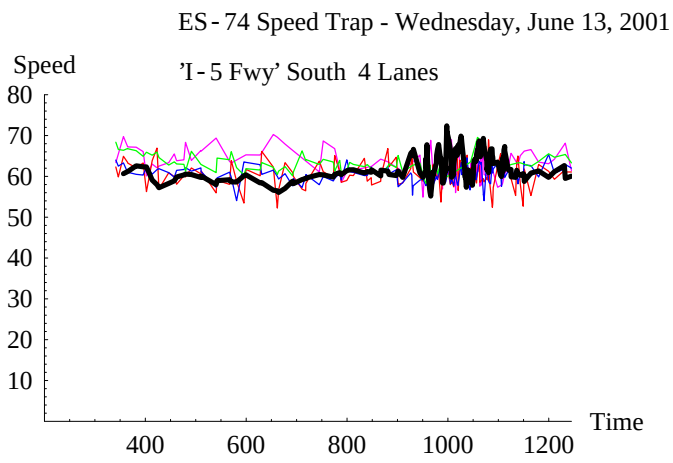
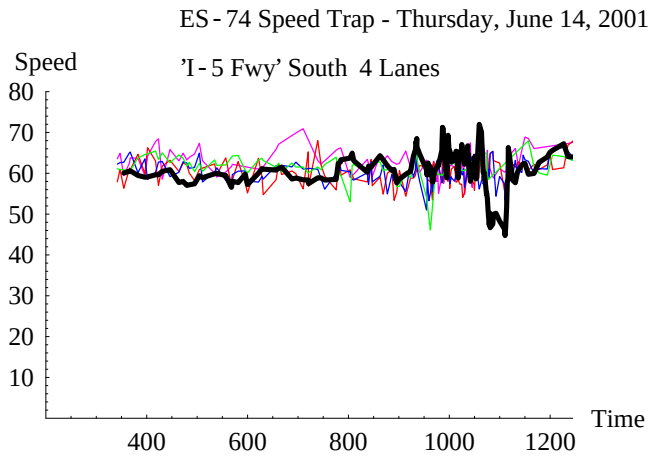


Figure 6.5: Probe and speed trap data on Friday, June 15, 2001, I-5 South.

The figure shows that probe speed measurements seem to be uniformly lower than speed-trap measurements on this day. The median difference between probe and speed-trap speeds for lane 1 was approximately 8 mph for each of the three days sampled. Figure 6.6 shows the effects of shifting the probe data by this median offset.

Figure 6.7 shows probe data superimposed over speed-trap data for three lanes of westbound traffic at cabinet ES-520 on State Road 520 just west of Lake Washington Boulevard. N.E. on Wednesday, June 13, 2001. Figure 6.8 shows that the probe data compare closely to speed-trap data for lane one. The median error between probe and trap-speeds was less than 1 mph on all three days.





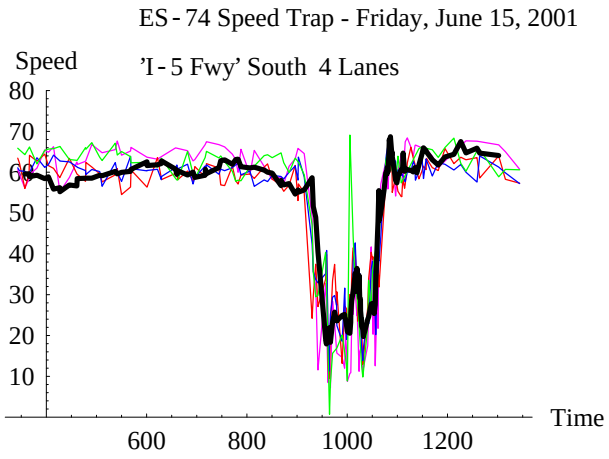


Figure 6.6: Corrected probe speeds for June 13, 14, and 15, 2001, I-5 South.

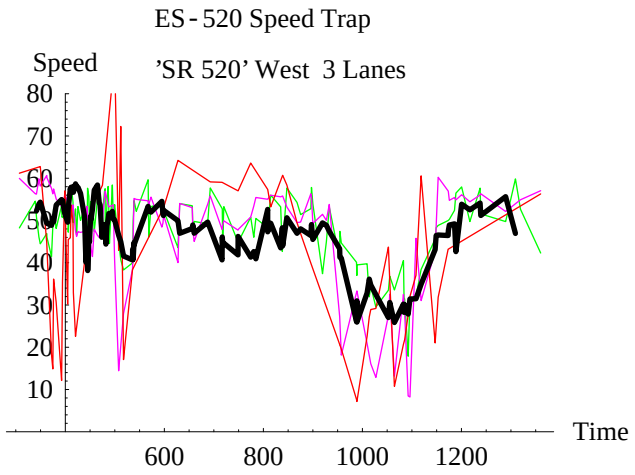


Figure 6.7: Probe and speed trap data on Wednesday, June 13, 2001, SR 520.

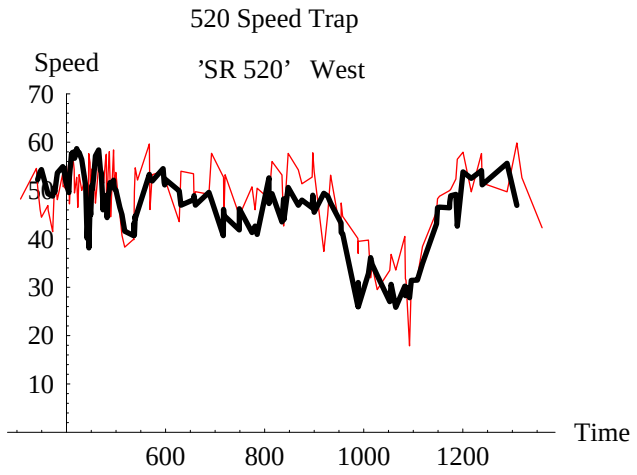


Figure 6.8: Probe and lane 1 speed trap data on Wednesday, June 13, 2001, SR 520.

Figure 6.9 shows a superposition of filtered probe speeds for the three days at probe location 6 on I-5 North between N 85th Street and NE Northgate Way. Unfortunately, Cabinet ES-154 at this location was malfunctioning and reporting speeds of 0 mph at all times. Nevertheless, the basic similarity in shape of the three series shows the promise of using historical data to develop a speed prediction function.

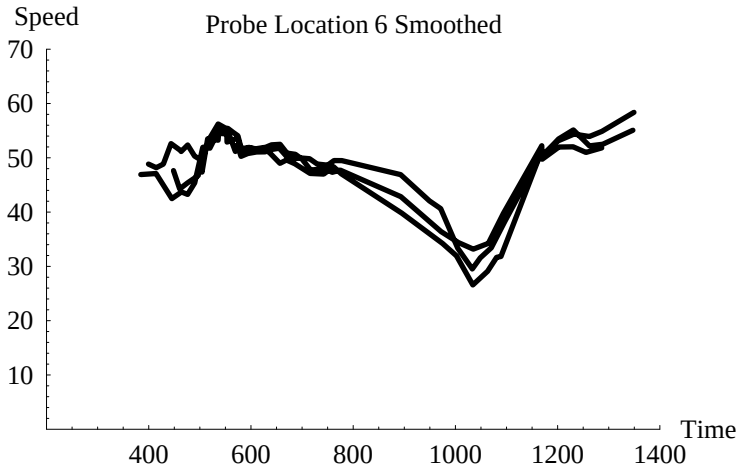


Figure 6.9: Three consecutive days of probe data, I-5 North.

Figure 6.10 shows a surface plot of speeds as a function of time and distance, $v = f(x,t)$, along a 2-mile stretch of I-5 South of S Spokane Street. The function was defined by 2-dimensional interpolation of speed values collected at GIS-indices on a chain of arcs defining the freeway.

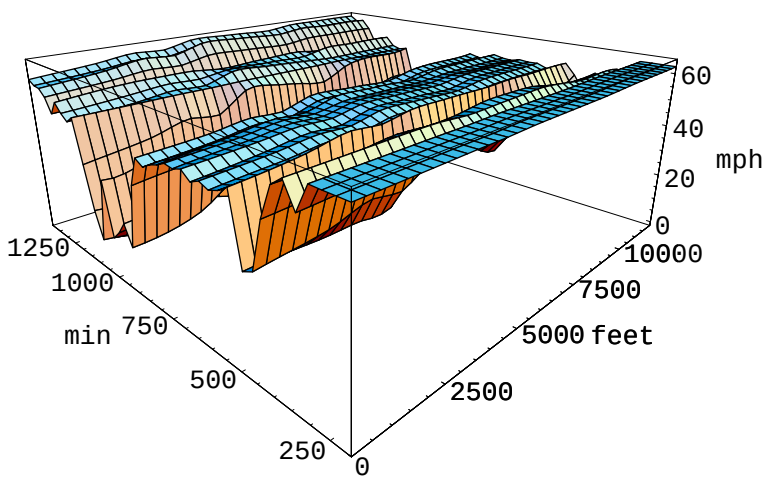


Figure 6.10: Speed as a function of time and distance.

Figure 6.11 shows a contour plot of the speed as a function of time and space, where the darker regions are slower speeds. To estimate travel times given this speed function, the ordinary differential equation,

$$\frac{dx}{dt} = f(x,t) \ , \tag{6.2}$$

is solved numerically using Euler's method. The two heavy white lines and the central heavy black line in Figure 6-11 are the trajectories of three solutions, and the shape of the trajectory depends heavily on the shape of the speed function. In particular, note the character of the bottom heavy white line that traverses a period and region that has slow speeds. This demonstrates that to accurately estimate travel time, speed must be an explicit function of space and time. For example, to estimate the travel time between two points as a function of time, we select a start time t_0 and solve the ODE for time t_1, subject to constraints $x(t_0) = 0$ feet and $x(t_1) =$ 11000 feet, to obtain travel time $t_1 - t_0$. The right of Figure 6.12 shows a plot of the travel times for this stretch of road as a function of departure time. The largest travel time peak, found at 1050 minutes, is associated with the bottom solution trajectory in the left of Figure 6.11.

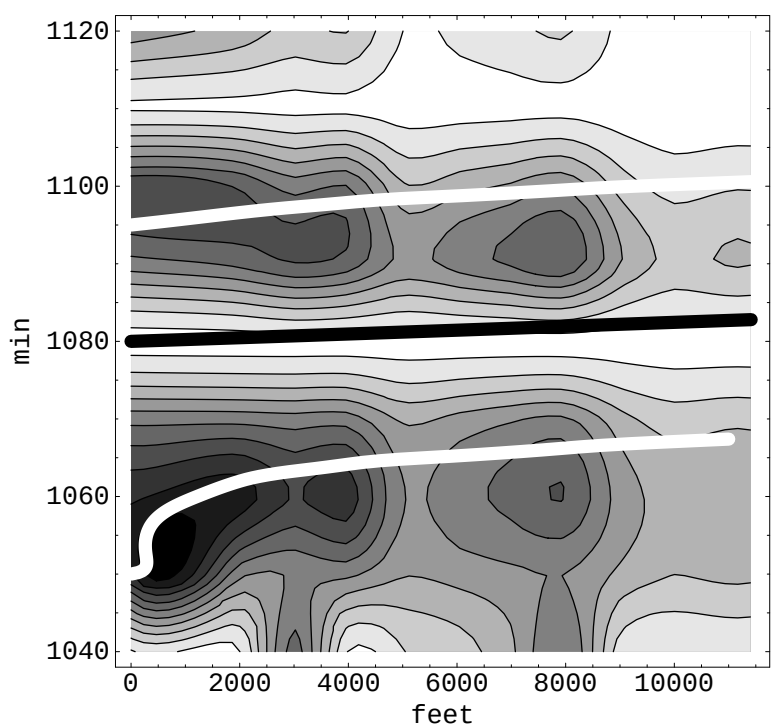


Figure 6.11: Contour plot of speed, darker is slower.

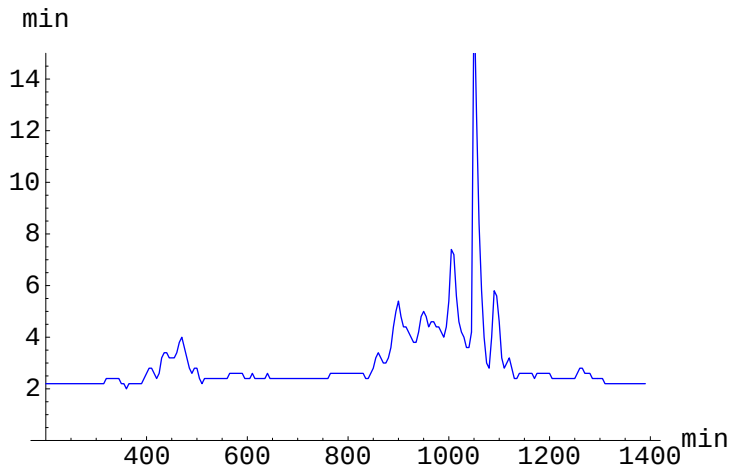


Figure 6.12: Travel time as a function of departure time.

7. CONCLUSIONS

In this report, we presented progress on the “transit vehicles as probes” effort at the University of Washington. We described a mass transit tracking system based on AVL data and a Kalman filter to estimate vehicle position and speed. We also described a system of “virtual” probe sensors that measure transit vehicle speeds using the track data. Graphical applications for viewing real-time speed measurements and for specifying probe sensor locations were described. Examples showing the correlation between Probe data and inductance loop speed trap data suggest that a layout of probe sensors on freeways and arterials can be specified to extend the limited use of speed traps. We also described a method that used probe sensor data to define vehicle speed along an arbitrary roadway as a function of space and time. We presented the use of this speed function to estimate travel time given an arbitrary starting time.

REFERENCES

1. Dailey, D.J., M.P. Haselkorn, K. Guiberson, and P.J. Lin. *Automatic Transit Location System*. Final Technical Report WA-RD 394.1. Washington State Department of Transportation, February 1996.

2. Dailey, D.J., D. Meyers, and N. Friedland. A Self Describing Data Transfer Methodology for ITS Applications. *Transportation Research Record 1660*, TRB, National Research Council, Washington, D.C., pp. 140-147, 1999.

3. Bell., B.M. The Marginal Likelihood for Parameters in a Discrete Gauss-Markov Process. *IEEE Transactions on Signal Processing*, Vol. 48, No. 3, pp. 870-873, March 2000.

4. Dailey, D.J., S.D. Maclean, F.W. Cathey, and Z. Wall. An Algorithm and a Large Scale Implementation. *Transportation Research Record*, TRB, National Research Council, Washington, D.C., to appear 2002.

5. Dailey, D.J. A Statistical Algorithm for Estimating Speed from Single Loop Volume and Occupancy Measurements. *Transportation Research B*, Vol. 33B, No. 5, pp. 313-322, June 1999.

6. Jazwinski, A.H. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

7. Tung, F., H.E. Rauch, and C.T. Striebel. Maximum Likelihood Estimates of Linear Dynamic Systems. *American Institute of Aeronautics and Astronautics Journal*, Vol. 3, pp. 1445-1450, August 1965.

8. Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, New York, 2nd edition, 1988.

9. Anderson, B.D.O. and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1979.

10. Cathey, F.W. and D.J. Dailey. A Prescription for Transit Arrival/Departure Prediction Using Automatic Vehicle Location Data. *Transportation Research C*, in press, 2001.

11. Dailey, D.J. and L. Pond. TDAD: An ITS Archived Data User Service (ADUS) Data Mine. *Transportation Research Board 79th Annual Meeting* (Preprint CD-ROM), 9-13 January 2000, Washington, D.C.