

# Dynamic Model-Based Filtering for Mobile Terminal Location Estimation

Michael McGuire, *Member, IEEE*, and Konstantinos N. Plataniotis, *Member, IEEE*

**Abstract**—Mobile terminal location is an important area of research because of its applications in location-sensitive browsing and resource allocation. This paper presents a method for reducing the error in mobile terminal location estimation. A preprocessor using nonparametric estimation is used to reduce the effects of non-line-of-sight and multipath propagation on the location procedure. A model-based dynamic filter is presented that uses an accurate model of mobile terminal motion to combine information from location measurements made at different time instances together to create an improved location estimate. The model of mobile terminal motion has a kinematic state space model describing the physical rules governing terminal motion and a control model that describes the human control input into the motion process. Location dependency in the control input model is used to derive a new dynamic filter. This filter provides greatly improved accuracy over previously known location techniques and is much more robust to variations in the mobile terminal motion and nonlinear effects in the propagation environment.

**Index Terms**—Cellular land mobile radio, filters, position location.

## I. INTRODUCTION

MUCH research has been performed on mobile terminal location in wireless cellular networks. Even without sophisticated location methods, a wireless cellular network has some knowledge of the location of a communicating mobile terminal. The handoff algorithm determines which base station serves the mobile terminal at any given time, which gives statistical knowledge about the mobile terminal's location. This location information is returned in Phase I of the Federal Communication Commission's emergency 911 wireless location requirement [1]. For more precise location estimates, the relationship between radio signals' characteristics and the relative positions of the mobile terminal and base stations is exploited to generate location estimation procedures. Measurements that have been proposed for the location of mobile terminals include the angle of arrival (AoA) of the radio signals from the mobile terminals to the base stations, the received signal strength (RSS) of the radio signal of the mobile terminal at the base station, the time of arrival (ToA) of the radio signal from the base station at the mobile terminal, and the time difference of arrival (TDoA) between signals from multiple base stations at the mobile terminal [2]–[5]. No matter which measurements are used to locate the

mobile terminal, there are errors in the position estimates resulting from noise in the measurements.

It has been shown that time filtering of the location estimates can reduce the location errors [6]–[10]. A filtering algorithm improves the estimation of mobile terminal location by combining the location information from measurements made at several sampling time periods together into an improved location estimate. The filtering algorithms presented in the literature were selected in a heuristic manner, with the main criterion of selection being ease of implementation. A filter's error reduction performance is highly dependent on the relationship between the filter's structure and the properties of the random processes that compose the motion and measurement generation processes for the mobile terminals. The filter parameters in the mobile terminal selection literature were selected to be optimal for the data sets in each paper. Methods for parameter selection for different scenarios were not presented. Thus, the robustness of the filters to data sets generated by mobile terminal motions other than those presented in the papers is uncertain.

This paper presents a location system where a time-based filter based on models of the mobile terminal motions is used to reduce the error in mobile terminal location. The mobile terminal motion model consists of a kinematic model, which describes the physical rules controlling the motion of mobile terminal motion, and a user control input decision model, which describes the user decisions concerning the motions of the mobile. The parameters of these models are obtained from real-world measurements of pedestrian and vehicular motion. The control input decision model's parameters are based on known rules for the motion of vehicles or pedestrians. The dependency of the control input selection on the terminal position is exploited to improve the estimation method. The advantage of this method is that the parameters of the models are easily mapped from field measurements, which makes the application of the filter to different scenarios simple.

This method is a network-based location solution since only the cellular network can have access to the necessary information about the local propagation and physical environment around the mobile terminal that is required by the location estimate filters. A terminal-based location solution, where all location estimation calculations are performed in the mobile terminals, could not use this method without prohibitively large downloads of information to the terminals from the base stations containing the local model parameters. The filtering algorithm described in this paper can be applied to other measurement types, such as GPS measurements.

The method will be evaluated for mobile terminals located within road vehicles moving through dense urban areas. Dense

Manuscript received February 19, 2002; revised November 27, 2002 and March 20, 2003. This work was supported by the Nortel Institute for Telecommunications under a Grant.

The authors are with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: mmcguire@dsp.toronto.edu).

Digital Object Identifier 10.1109/TVT.2003.814219

urban areas are the regions of greatest interest to cellular network operators since these regions have the highest densities of users. These are also likely to be the areas where third-generation cellular networks are first introduced. Mobile terminals in road vehicles can have high velocities and uncertainty in their motion. This makes estimating their locations of greater difficulty than for relatively low-mobility pedestrian-carried mobile terminals. Mobile terminals in road vehicles derive the greatest benefit from model-based estimation location estimation.

The location estimation filtering algorithm proposed in this paper is independent of the type of measurements used to locate the mobile terminal. ToA measurements are used in the simulation within this paper to evaluate the effectiveness of the method. The modulation and multiple-access schemes proposed for most next-generation cellular networks allow for high-resolution time measurements, which makes ToA location highly likely to be the location technology used in these networks [11].

Section II describes the measurement model. The radio propagation environment used to evaluate the location procedure is discussed, and the preprocessing of the radio measurements before the filtering procedure is outlined. Section III describes the motion model for the mobile terminals. The kinematic and user control input models that dictate mobile terminal motion behavior are described. The model-based filter and control input estimator are described in Section IV. The setup of the simulations used to evaluate the location estimation algorithm is outlined in Section V. The results of the simulations are presented within Section VI. Our conclusions and possible topics for future research are summarized in Section VII.

## II. PROPAGATION MEASUREMENT MODEL

The measurements taken to locate the mobile terminal are ToA measurements. The propagation time from the base station to mobile terminal is measured. These propagation time measurements are converted to propagation distance measurements by multiplication of the time measurements by the speed of light. There are nonremovable errors in the time measurements resulting from noise in the measurement systems and propagation environment.

The measurement error for propagation distance has been shown in most cases to be near Gaussian in [12]. The variance of the measurement error when there is only a single propagation path is mostly a function of the signal power, interference power, and noise power at the receiver. The propagation effect mainly responsible for increasing the error in propagation distance measurements is multipath propagation, during which the radio signals travel from the transmitter to receiver via multiple paths, each with its own attenuation and transmission delay. Multipath propagation results in the error distance measurements' having higher variances and positive nonzero means [13]. The positive bias resulting from the nonzero error mean is created by the probability that the time measurement device will incorrectly detect one of the extra longer propagation paths as the shortest propagation path instead of the true shortest distance path. The increased variance is a result of the transmission energy's being spread over multiple paths, which makes the estimation of the

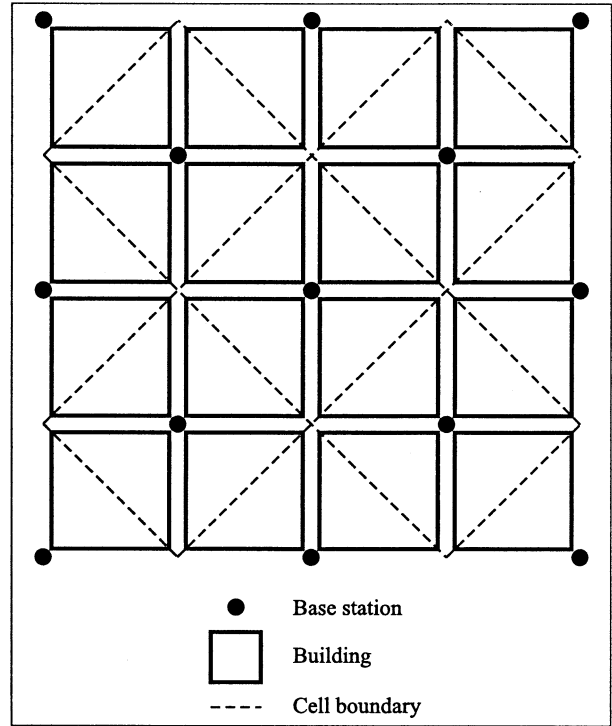


Fig. 1. Simulation environment layout.

propagation time more difficult than if all the transmission energy was in one path.

The environment used to evaluate the location estimation procedure is a simple Manhattan model that has been used in the mobile terminal location literature to evaluate radio location performance [2]. The layout of the environment is shown in Fig. 1. The positive  $y$ -direction will be said to be north, making the positive  $x$ -direction east. The city blocks are 300 m long, and the streets are 20 m wide. Base stations are located in the intersection at every second block. This environment and base-station layout is typical of dense urban areas. The handoff algorithm for the cellular network is a perfect distance-based algorithm; a mobile terminal has knowledge of which base station it is closest to and communicates with the network via that base station. This makes the cell of each base station, the region where all the mobile terminals would communicate with the base station, a diamond-shaped region, as shown in Fig. 1.

The measurement noise assumed in the model is Gaussian. The distance measurement vector is given by

$$\mathbf{z}(k) = \mathbf{d}(k) + \varepsilon(k) \quad (1)$$

where  $\mathbf{z}(k)$  is the measurement vector at sample interval  $k$ ,  $\mathbf{d}(k)$  is the vector of propagation distances from the base stations to the mobile terminal at sample interval  $k$ , and  $\varepsilon(k)$  is a random vector representing measurement noise. The length of  $\mathbf{z}(k)$  is equal to the number of base-station measurements used to locate the mobile in one sample interval. The covariance vector of the measurement noise vector  $E[\varepsilon(k)\varepsilon(k)^T]$  is equal to  $\sigma_d^2$  multiplied by an identity matrix of the appropriate size.

During line-of-sight (LOS) propagation, when the shortest distance straight line propagation path from base station  $j$  to

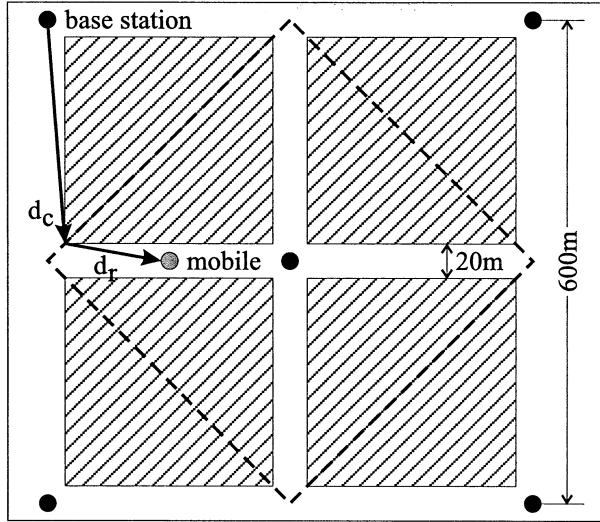


Fig. 2. Manhattan propagation environment.

mobile terminal is not blocked  $d_j(k)$ , the  $j$ th entry of the distance vector  $\mathbf{d}(k)$  is the distance from the base station to the mobile terminal. This is not true during non-line-of-sight (NLOS) propagation when the shortest distance straight line propagation path is blocked by a building or other obstruction. For the Manhattan model used in the simulations, it is assumed that during NLOS propagation, the signal diffracts around corners and the shortest propagation path length is the distance from the base station to the corner plus the distance from the corner to mobile terminal. An example of an NLOS propagation distance is given by ( $d_j = d_c + d_r$ ) in Fig. 2.

The measurement noise is Gaussian with a mean of 16.0 m and a standard deviation  $\sigma_d$  of 16 m. The parameters of the noise density are taken from [2, Fig. 5], which simulated multipath propagation based on the COST 207 urban power delay profile in microcells and analyzed the performance of a ToA distance measurement system.

This paper discusses a network-based algorithm. Propagation distance measurements for mobile terminal location are made by the base stations. It is assumed that only the five closest base stations to the mobile terminal can measure the propagation delay. This constraint results from signal loss and channel reuse considerations. It is assumed that base stations other than the closest five either would not receive a strong enough signal from the mobile terminal to be able to measure the propagation delay or would have reassigned the mobile terminal's channel for use by another mobile terminal of greater proximity to them [14]. Of these five base stations, only the three base stations with the lowest propagation delay measurements are used to locate the mobile terminal. The distance measurement vector  $\mathbf{z}(k)$  has a length of three and consists of the distance measurements calculated from the three lowest propagation delay measurements. Field measurements have shown that three or more base stations can make measurements for the mobile terminal more than 90% of the time in urban areas [15].

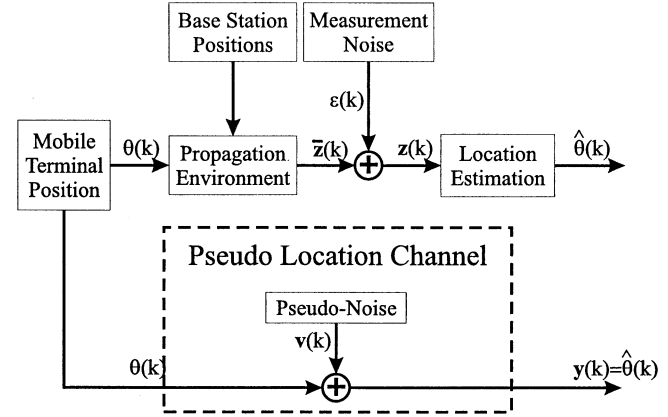


Fig. 3. Zero memory estimator for ToA location estimation.

### A. Zero Memory Estimation

The true location of the mobile terminal at sample interval  $k$  will be designated  $\theta(k)$ . The time measurement vector for sampling interval  $k$  is designated  $\mathbf{z}(k)$ , as was described above. The estimated location of the mobile terminal at sampling interval  $k$  is designated  $\hat{\theta}(k)$ .

The so-called zero memory estimator is a preprocessor for the location estimation system. It allows the model-based location estimation filter to operate without knowledge of the nonlinearities in the propagation environment. This is illustrated in Fig. 3. The model-based filter receives  $\mathbf{y}(k) = \hat{\theta}(k)$ , the estimated position of the mobile terminal at sample time  $k$  based only on the measurements made during interval  $k$ . This synthetic measurement  $\mathbf{y}(k)$  is a linear function of the true system state at time  $k$  as opposed to  $\mathbf{z}(k)$ , which is a nonlinear function of the true system state.

Most location estimation techniques assume LOS propagation where the shortest distance straight line path between the mobile terminal and base station is not blocked. During NLOS propagation, when the LOS propagation path is blocked by a building or other obstruction, location estimates made assuming LOS propagation suffer from a bias [2]. The use of recursive filters to estimate and remove this bias has been proposed [9]. These filters require a model not only for the motion of the mobile terminals but also for the time evolution of the effect of NLOS propagation on the measurements. The time evolution of NLOS effects is a complicated function of the motion of the mobile terminal, the location of radio propagation obstacles, and the location of the base stations. This means that the NLOS bias estimation algorithm is extremely complicated and must be adjusted if the motion model parameters change.

The approach taken to solve the NLOS problem in this paper is to use a survey of propagation measurements to characterize the propagation environment. Nonparametric estimation, based on survey measurements made of the radio propagation environment, is then used to convert measurements made by mobile terminals at unknown locations to location estimates. These location estimates do not assume an LOS propagation model, so they do not have a bias when NLOS propagation occurs.

The estimation problem is stated as estimating the location of a mobile terminal at an unknown location within region  $\mathcal{S}$  from measurement vector  $\mathbf{z}(k)$  given the survey set. The survey data set consists of measurements  $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$  made at respective true locations  $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_n\}$ . Since the survey set is the same for all mobile terminals within a given cell for all sampling intervals, the survey measurements and locations are not given a time index. The probability density function of the survey point locations is the same as the probability density function of the mobile terminal location. For our estimation problem, this means the survey points for the cell the mobile terminal is known to be residing in are used in the estimation procedure.

The survey measurement set can come from many sources. During network configuration, propagation measurements are made to ensure that the base stations are providing good coverage of the network area [16]. These propagation measurements can be converted for use with the zero memory estimator. Computer models could also be used when field measurements are considered too expensive. Survey data obtained from computer models will have higher measurement error than survey data obtained from field measurements since the computer model cannot perfectly match the true propagation environment. The results presented in this paper are based on survey data taken from the simulated field measurements. A mixture of the two methods is also possible. Field measurements would be made in high-occupancy areas, where the network provider earns more revenue and wishes to ensure good service, and computer models would be used in areas with lower occupancy densities, where location accuracy is not as large a concern.

The zero memory estimation formula is

$$\hat{\boldsymbol{\theta}}(k) = \frac{\sum_{j=1}^n \boldsymbol{\theta}_j(h_z)^{-m} K_z\left(\frac{\mathbf{z}(k) - \mathbf{z}_j}{h_z}\right)}{\sum_{j=1}^n (h_z)^{-m} K_z\left(\frac{\mathbf{z}(k) - \mathbf{z}_j}{h_z}\right)} \quad (2)$$

where  $K_z(\cdot)$  is a kernel function that decreases monotonically from the origin,  $h_z$  is a smoothing constant, and  $m$  is the length of the measurement vector  $\mathbf{z}(k)$  [17], [18].

This estimation equation can be rewritten to show that the estimated position is a weighted averaged of the survey point positions with the weights' being determined by the measurement data. Equation (2) is rewritten as [18]

$$\mathbf{y}(k) = \hat{\boldsymbol{\theta}}(k) = \sum_{j=1}^n \boldsymbol{\theta}_j w_j(\mathbf{z}(k)) \quad (3)$$

where

$$w_j(\mathbf{z}(k)) = \frac{K_z\left(\frac{\mathbf{z}(k) - \mathbf{z}_j}{h_z}\right)}{\sum_{i=1}^n K_z\left(\frac{\mathbf{z}(k) - \mathbf{z}_i}{h_z}\right)}. \quad (4)$$

An examination of (3) shows that the cost of this zero memory location estimation in terms of the number of operations is linear with respect to the number of survey points used. The zero memory location estimate requires memory to hold all the survey data. This should not be a considerable expense since the amount of memory required is not large and the cost of this memory in the base stations is not expensive (on the order of memory cost for a personal computer). In addition, the survey data are identical for all mobile terminals connected

to a base station. Other zero memory location estimation methods require iterative solving techniques and calculation of the derivatives of the conditional density functions [2]. Our technique requires the calculation of the sum of the kernel functions for each survey point, which is less costly.

Since we are locating only road vehicles, all the survey points are located on the streets. As can be seen in Figs. 1 and 2, each cell has two streets that run perpendicular to each other. One line of survey points is sampled down the center of the street parallel to the  $x$ -axis, and another line of survey points is sampled along the center of the street parallel to the  $y$ -axis. Since there are  $n$  survey points in each cell, the survey points on each line are separated by  $600 \cdot 2/n$  m.

The kernel function used in this paper is called the Parzen Gaussian kernel [19]. It is given by

$$K(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi}}\right)^m |\mathbf{C}|^{-\frac{1}{2}} \exp\left(-\frac{\mathbf{x}^T \mathbf{C}^{-1} \mathbf{x}}{2}\right). \quad (5)$$

For the ToA estimation problem, the matrix  $\mathbf{C}$  is set to be an identity matrix of size  $m$ .

An examination of how  $h_z$  is substituted into the kernel function shows that  $h_z$  has some relation to the variance of the measurement noise [19]. The delay spread can be measured during network configuration. This allows for the variance of the propagation distance measurement noise to be calculated so this value could be known to network operators. Experience has shown that with nonparametric estimation, using a value for the smoothing parameters in the same order of magnitude as the optimal value gives estimates with errors nearly as good as if the optimal value for the smoothing parameter was used. An analysis of parameter selection for location estimation has been performed in other work [17], [18]. Experimentation with a model similar to the one considered here suggests that parameter values such as  $h_z = 2\sigma_d$  and  $n = 100$  provide excellent results.

An advantage of the nonparametric estimator is that in addition to estimating the location of the mobile terminal, an estimate of the covariance of the measurement noise is readily available. This is obtained from

$$\begin{aligned} \mathbf{R}(k) &= \text{Cov}(\hat{\boldsymbol{\theta}}(k) | \mathbf{z}(k)) \\ &= \text{Cov}(\boldsymbol{\theta}(k) | \mathbf{z}(k)) \\ &= \text{E}[\boldsymbol{\theta}(k) \boldsymbol{\theta}(k)^T | \mathbf{z}(k)] \\ &\quad - \text{E}[\boldsymbol{\theta}(k) | \mathbf{z}(k)] \text{E}[\boldsymbol{\theta}(k) | \mathbf{z}(k)]^T. \end{aligned} \quad (6)$$

The calculation of  $\text{E}[\boldsymbol{\theta}(k) | \mathbf{z}(k)] \approx \hat{\boldsymbol{\theta}}(k)$  is accomplished using the estimator described above. An estimator for the value of  $\text{E}[\boldsymbol{\theta}(k) \boldsymbol{\theta}(k)^T | \mathbf{z}(k)]$  can be derived to be [20]

$$\text{E}[\boldsymbol{\theta}(k) \boldsymbol{\theta}(k)^T | \mathbf{z}(k)] \approx \sum_{j=1}^n \boldsymbol{\theta}_j \boldsymbol{\theta}_j^T w_j(\mathbf{z}(k)) \quad (7)$$

where  $w_j(\mathbf{z}(k))$  is given by (4). This covariance can be used to evaluate the accuracy of the estimation, with a low covariance estimate implying high accuracy, and is an additional input to a dynamic estimator. The covariance estimate allows a dynamic estimator to give more weight to the zero memory location estimates with higher estimated accuracy than the location estimates with low estimated accuracy. One such scenario

is when the mobile terminal is located at a position relative to the measuring base stations where zero memory location estimators cannot provide high-accuracy location estimates. In this situations, there are relatively large regions that create measurement vectors  $\mathbf{z}(k)$  that are almost the same. This is called the geometric dilution of precision (GDOP) problem [21]. In this situation, the covariance estimate from (6) will be high, indicating a low-accuracy zero memory estimate.

### III. MOTION MODEL

This section describes the motion of road vehicles in urban environments. This model is split into two parts. The first part is a kinematic model that describes the physical rules governing the motion of a mobile terminal in response to the forces acting on it. The second part of the model is a control decision model. The control decision model describes the human decisions that determine the control inputs into the kinematic model for mobile terminal motion.

From the description of vehicle motion, two mathematical models are derived. The first model is an estimation model, described in Section IV, which is the time evolution model for system state consisting of terminal location and velocity used by the location estimation algorithm. The second model is the simulation model, described in Section V, which we use to generate location tracks for mobile terminals to evaluate our location estimate algorithm. Both models share the same kinematic model of vehicle motion. The models differ on the control decision level. The estimation control decision model is a general decision model designed to handle many different kinds of mobile motions subject to the constraints an urban environment places on vehicular motion. The simulation control decision model is designed to create a reasonable reproduction of actual human control decisions for a vehicle moving through an urban environment. This simulation control decision model is used only to evaluate the filter's performance; it was not used to generate the filter.

#### A. Kinematics of Vehicle Motion

This section describes a state space model for describing the motion of a mobile terminal located in a road vehicle. A state space model describes the time evolution of the system state of the mobile terminal in terms of differential equations in continuous time, or time difference equations in discrete time.

Several state space models have been developed to model the kinematics of road vehicle motion [22], [23]. These state space models are of greater complexity than required for the mobile tracking application. These state space models are deterministic, requiring precise knowledge of the control input, road conditions, and vehicle parameters. This knowledge is not known to the location estimation system, and estimating it would be computationally expensive. This can be avoided by inserting random process noise into the state space model of the mobile terminal motion. The random process noise models the uncertainty caused by lack of knowledge of road conditions and vehicle parameters.

There are several state space models proposed in the literature for mobile terminal motion using random process noise,

such as mobile position modeled as Brownian motion [8], velocity modeled as Brownian motion [7], and position modeled as fractional Brownian motion [24]. None of the models was based on observed characteristics of a vehicle's motion; instead, they were primarily selected for computational simplicity. This resulted in some of the models' having characteristics disparate from actual vehicular motion. For example, the velocity modeled as Brownian motion model has the characteristic that the variance of velocity tends to infinity with increasing time.

We propose a state model based on observed vehicle motion characteristics incorporating random process noise. A vehicle is subject to several friction and drag forces. Two major forces resist the motion of the vehicle: rolling resistance and air resistance [25]. Rolling resistance is generated by slip between the vehicles' wheels and the driving surface and friction inside of the vehicle. Air resistance is generated by the force of the air around the vehicle against its motion. Both of these increase with the vehicle's speed. The result of these forces is that if the vehicle is subject to constant driving force, the acceleration of the vehicle will decrease as the velocity increases [26].

A random motion model that matches this observation is based on a modified form of the Langevin equation [27]. A linear drag term  $\alpha$  is present in the kinematic model to model the effects of air and rolling resistance. Mobile terminal motion in one dimension is given by

$$\ddot{x}(t) = -\alpha\dot{x}(t) + w(t) + u(t) \quad (8)$$

where  $x(t)$  is mobile position,  $\dot{x}(t)$  is mobile velocity,  $\ddot{x}(t)$  is mobile acceleration,  $u(t)$  is a deterministic function representing driver control, and  $w(t)$  is random process noise. The process noise models random effects such as noise in the control system of the vehicle, variations between drivers, and random road conditions.

The control input  $u(t)$  is the driver's input into the system, which controls the direction in which the vehicle is moving, in which direction it will accelerate, and so on. The value of  $u(t)$  directly influences the mean speed of the vehicle in control input direction.

If  $u(t) = 0$ , then the mobile position will wander around  $x(t) = 0$  with  $E[x(t)] = E[\dot{x}(t)] = E[w(t)] = 0$ . A positive value of  $u(t)$  will cause the mobile terminal to move in the positive direction  $\lim_{t \rightarrow \infty} E[\dot{x}(t)] = u(t)/\alpha$  if  $u(t)$  is constant. A negative value of  $u(t)$  has the opposite effect. The driver of the vehicle will select  $u(t)$  based on the vehicle's present location, the speed limit and the final desired destination. Observations of real vehicle speeds by vehicular traffic engineers have shown that the distribution of vehicle speed at a fixed locations can be modeled as Gaussian [26]. To match this observation, the process noise  $w(t)$  is selected as a white Gaussian process  $E[w(t)] = 0$  and  $\text{Var}[w(t)] = \sigma^2$ . The variance of the velocity given the control input is determined by the variance of the process noise  $\lim_{t \rightarrow \infty} \text{Var}(\dot{x}(t)|u(t)) = \text{Var}(w(t))/(2\alpha)$ . The maximum mean acceleration of the vehicle is  $u(t) \text{ m/s}^2$ , which is attained when  $\dot{x}(t) = 0 \text{ m/s}$ .

The location estimator works with discrete time samples of the location state for which a discrete time motion model is required. Fortunately, as long as the time between changes in the

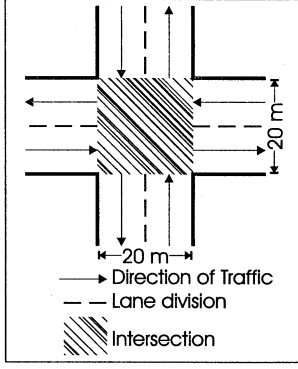


Fig. 4. Lane layout and intersection configuration.

control input  $u(t)$  is longer than the time constant of the continuous time state space model  $1/\alpha$ , it is possible to easily convert the continuous time model to a discrete state space time model. The discrete time motion model is given by

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) + \mathbf{w}(k) \quad (9)$$

where  $\mathbf{x}(k)$  is the location position and velocity state at time interval  $k$ .  $\Phi$ ,  $\Gamma$ , and the covariance matrix of the process noise are computed as shown in Appendix I based on the continuous time evolution equation (8).

To match the operation characteristics in typical urban settings in North America, a mean velocity of 54 km/h or 15 m/s should be introduced. The maximum mean acceleration during standard driving for a standard passenger automobile is 2.5 m/s<sup>2</sup> [25]. To match these performance values, we set  $\alpha = 1/6$  and the standard control input  $u(t) = 2.5$ . We used a value of  $\sigma^2 = 1/3$  to give a standard deviation of 1 m/s (3.6 km/h) in the velocity.

### B. Control Input Model

The control input is the human control on the motion of the mobile terminal. In terms of the kinematic model above, it determines the value of the input functions  $u(t)$  or  $\mathbf{u}(k)$  in discrete time. A driver's decision on what action to take at each point in a journey is determined by the location of the destination as well as other factors such as traffic conditions.

Several types of decision models have been proposed in the mobility modeling literature. The simplest models only change the control inputs on the boundaries of cells and do not consider street layout [28], [29]. More complex motion models have been proposed that allow the control inputs to change at any time [30]–[32]. The control inputs are kept constant over time periods with random lengths sampled from an exponential distribution. The model in [30] models realistic direction-changing behavior for vehicles. Before a vehicle makes a major direction change, it must slow down or come to a stop. Again, street layout is not considered in these models. Simple models that consider street layout have been described in the literature as well [33]. These models assume that vehicles can change their velocity and directions instantaneously. Realistic vehicle braking and turning behavior are not considered. The models allow for simple simulation and analysis of the behavior of groups of mobile terminals

TABLE I  
SETTING FOR MOVING NORTH ON STREET

Control Input	Value
$u_x(k)$	$\beta(p_x(k) - 5)$
$u_y(k)$	$C$

but are not designed to provide realistic behavior for a single mobile terminal.

The control decision model we propose is designed for accurate modeling of motion for a single mobile terminal. The street layout is considered and the mobile terminal will only make turns at intersections. Lane use is considered, and the mobile terminal will stay in the proper lane for its chosen direction. The mobile terminal will also brake before intersections so that it is moving slowly while making major direction changes.

As a driver approaches an intersection, his selected route will determine what lane he will use, the probability that he will brake, and which direction he will move away from the intersection. If he is going to turn, he will have to brake in order to make the corner safely. If he has decided that he is going to go straight through an intersection, he will only stop if a traffic control signal forces him to, or if there is some form of traffic blockage.

Vehicles remain in the center of their respective lanes with only small variations unless passing or turning. A standard North American intersection layout is shown in Fig. 4. Turning usually only takes place in the central area of the intersection. Turning takes place outside of intersections when the vehicle has reached its final destination and the vehicle is turned into a parking area.

1) *Simple Motion Without Turns*: This decision model describes the control behavior when the mobile is located away from intersections or when the mobile terminal is traveling down a highway. The allowed direction of all motion is in one of two directions: up or down the street. Motion is restricted by the lane in which the mobile terminal.

The control inputs are categorized as  $u_{\parallel}(k)$ , which is the control input in the direction parallel to the street direction, and  $u_{\perp}(k)$ , which is perpendicular to the street direction. For example, if the street is parallel to the  $Y$ -axis, then  $u_y(k) = u_{\parallel}(k)$  and  $u_x(k) = u_{\perp}(k)$ . The control input  $u_{\parallel}(k)$  is used to control the speed along the street. The control input  $u_{\perp}(k)$  is applied toward the center of the lane to keep the mobile terminal within the proper lane.

A deterministic constant input of  $u_{\parallel}(k) = C$  is applied in the direction of motion along the street. This will result, as described in Section III-A, in a mean velocity of  $C/\alpha$  meters/second in that direction after a period of initial acceleration. The other control input will be set to  $u_{\perp}(k) = \beta(p_{\perp}(k) - c)$ , where  $\beta$  is a control constant,  $p_{\perp}(k)$  is the position of the mobile on the coordinate axis perpendicular to the street direction, and  $c$  is the location of the center of the lane.

For example, if the mobile terminal is heading in the positive  $y$ -direction in a lane whose middle is located at  $x = 5$  m, the control inputs will be set as shown in Table I.

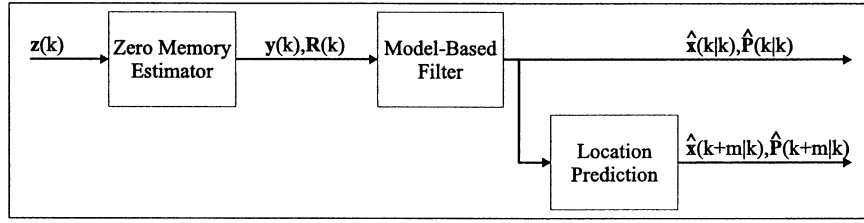


Fig. 5. Location estimation system.

2) *Motion With Turns*: Turning maneuvers make the mobility model much more complicated. The long-term behavior models required to model when a mobile terminal makes a turn will not be considered here. Only the short-term behavior involved with making turns and handling intersections will be shown.

Street layout information is required to accurately model turning behavior. A mobile terminal in a vehicle can only make turns at street intersections. To safely make a turn, a mobile terminal must be traveling at a low speed. Therefore, when a mobile is traveling at a higher speed down a street, it must slow down before an intersection in order to make a turn. If a mobile terminal is not going to turn at an intersection, it will only slow down if there is some form of traffic signal or blockage that forces it to do so.

#### IV. ESTIMATION AND FILTERING ALGORITHM

The zero memory estimator only uses the measurements taken at sample interval  $k$  to estimate the mobile location during period  $k$ . However, there will be dependence between the mobile's location at period  $k$  and its location during the previous periods. The model-based filter takes the estimate of the location from the zero memory estimator and combines it with estimates of location from previous sampling intervals to obtain an improved estimate of the current location. To accomplish this, it uses information about how the system state evolves over time to extract information from previous time intervals about the present system state to improve the location estimate. This also allows the location estimate system to estimate velocities for the mobile terminal that cannot be calculated from location measurements made at a single time instant. This facilitates the estimation of future mobile terminal system states.

The system state of the mobile terminal at sample interval  $k$  is designated  $\mathbf{x}(k)$ , and the measurement vector at sample interval  $k$  is designated  $\mathbf{y}(k)$ . The estimate of  $\mathbf{x}(k)$ , given measurements  $\{\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(k)\}$ , is designated  $\hat{\mathbf{x}}(k|k)$ . The covariance of  $\hat{\mathbf{x}}(k|k)$  is designated  $\mathbf{P}(k|k)$ . We select the estimator that minimizes the mean squared error

$$\text{MSE}(\hat{\mathbf{x}}(k|k)) = \mathbb{E} \left\{ [\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)]^T [\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)] \right\}. \quad (10)$$

The system state estimation system is shown in Fig. 5. To estimate the system state, two relations need to be known: 1) that between the system state and the measurements and 2) the time evolution behavior of the system state. The relationship between measurements and system state was described in Section II. The time evolution of the system state assumed by the filter, called

the state space model, is described in Section IV-A. The actual filtering algorithm will be described in Section IV-B.

Raw measurements are not used as inputs to the filtering algorithm because of the complex relationship between propagation distance measurements and mobile terminal location caused by NLOS and multipath propagation. The zero memory estimator uses nonparametric estimation to create a synthetic measurement  $\mathbf{y}(k)$ , which is a linear function of the system state

$$\begin{aligned} \mathbf{y}(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \quad (11)$$

where  $\mathbf{v}(k)$  is a random vector representing the error from the nonparametric zero memory location estimation procedure. As a by-product of nonparametric estimation, the zero memory estimator also computes  $\mathbf{R}(k)$ , an estimate of the covariance of  $\mathbf{v}(k)$ . The zero memory estimator preprocessor, as well as giving the dynamic estimator an input with a linear relationship to the system state, also provides statistical information about the measurements to the dynamic estimator.

An advantage of the separate zero memory estimator and model-based filter algorithms is that the algorithms can be changed independently if the measurements or mobility mode of the terminal change. For example, ToA measurements may be used to locate the mobile terminal in heavy urban areas, while AoA measurements may be used in suburban and rural areas. It is certainly possible that a mobile terminal could move from one area to another during a communication session. With the algorithmic split, the time evolution algorithm could remain the same while the measurement algorithm is changed as the mobile terminal moves from one region to another.

##### A. Estimation Model

The estimation model assumed by the location estimation filter is a simplified version of the motion model described in Section III. The estimation motion model must concentrate on short-term modeling of mobile terminal motion. It must also be general enough to be able to handle most of the possible mobile terminal motions without any knowledge of parameters that would not be readily available to field implementations of the algorithm.

The assumed kinematic model for mobile terminal motion is

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) + \mathbf{w}(k). \quad (12)$$

The matrices  $\Phi, \Gamma$  and the covariance matrix  $\mathbf{Q}_{\text{model}} = \mathbf{Q}$  are assumed known and are based on the matrices reviewed in Section III. We assume that the parameters of the motion such as drag coefficient and process noise covariance are known to the

TABLE II  
INPUTS FOR MOBILE TERMINAL DIRECTIONS

Direction	Control Input
North	$\mathbf{u}(k) = \mathbf{u}_1 = \begin{bmatrix} 0 \\ C \end{bmatrix}$
South	$\mathbf{u}(k) = \mathbf{u}_2 = \begin{bmatrix} 0 \\ -C \end{bmatrix}$
East	$\mathbf{u}(k) = \mathbf{u}_3 = \begin{bmatrix} C \\ 0.0 \end{bmatrix}$
West	$\mathbf{u}(k) = \mathbf{u}_4 = \begin{bmatrix} -C \\ 0.0 \end{bmatrix}$
None	$\mathbf{u}(k) = \mathbf{u}_5 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$

estimation system. The values for these model parameters can be calculated based on the measurements made in the network area by vehicular traffic engineers.

The only part of the kinematic estimation model that cannot be estimated ahead of time or controlled is the control input of the user. To approximate the behavior of the user, the estimation model assumes the control input vector is one of five possible input vectors that are matched to the mobile terminal being stopped or moving in one of the cardinal compass directions matched to directions of the streets, as shown in Table II.

If there are other street directions, then other control input vectors would be added to the above set to match them. The constant  $C$  in the control input set is matched to the maximum observed acceleration of the mobile terminals. The relationship of this constant with the velocities and drag coefficient of the mobile terminal motions is explained in Section III.

Vehicles in urban environments usually only change directions in intersections. When the mobile terminal is not in intersections, the control input usually remains constant. That is,  $\Pr[\mathbf{u}(k) = \mathbf{u}(k-1)] \approx 1$  if the mobile terminal is not in an intersection. If the mobile is located in an intersection, there is a substantial probability that the control input will change. The state transition probabilities in each region are described in Table III in the form of matrices. The optimal value of the constant  $P(\text{TOSELF})$  is a function of the probability that the mobile will turn at the intersection. If the probability of turning is high, then the optimal  $P(\text{TOSELF})$  will be small. If the probability of turning is low, then the optimal  $P(\text{TOSELF})$  will be close to one. Good values of  $P(\text{TOSELF})$  range from 3/4 to 1 for a turning probability greater than 0.1.

TABLE III  
ESTIMATION MODEL: STATE TRANSITION

Terminal Location	Transition Probabilities for Terminal Location
$\mathbf{x}(k)$	$\Theta_{ij}(\mathbf{x}(k)) = \Pr[\mathbf{u}(k+1) = \mathbf{u}_i   \mathbf{u}(k) = \mathbf{u}_j, \mathbf{x}(k)]$
Not in Intersection	$\begin{bmatrix} c_1 & c_2 & c_2 & c_2 & c_2 \\ c_2 & c_1 & c_2 & c_2 & c_2 \\ c_2 & c_2 & c_1 & c_2 & c_2 \\ c_2 & c_2 & c_2 & c_1 & c_2 \\ c_2 & c_2 & c_2 & c_2 & c_1 \end{bmatrix}$
In Intersection	$\begin{bmatrix} c_3 & c_4 & c_4 & c_4 & c_4 \\ c_4 & c_3 & c_4 & c_4 & c_4 \\ c_4 & c_4 & c_3 & c_4 & c_4 \\ c_4 & c_4 & c_4 & c_3 & c_4 \\ c_4 & c_4 & c_4 & c_4 & c_3 \end{bmatrix}$

$$c_1 = 0.999, c_2 = (1 - 0.999/4), c_3 = P(\text{TOSELF}), \text{ and } c_4 = (1 - P(\text{TOSELF}))/4$$

### B. Model-Based Estimation

The probability density function that describes the evolution of the location state is given by

$$\begin{aligned} f(\mathbf{x}(k+1), \mathbf{u}(k+1) | \mathbf{x}^k, \mathbf{u}^k) \\ = f(\mathbf{x}(k+1), \mathbf{u}(k+1) | \mathbf{x}(k), \mathbf{u}(k)) \end{aligned} \quad (13)$$

where  $\mathbf{x}^k = \{\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(k)\}$  and  $\mathbf{u}^k = \{\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(k)\}$ . Equation (13) shows that the system state and control input  $(\mathbf{x}(k), \mathbf{u}(k))$  are jointly a Markov-one process [27]. The conditional density of the system state and control input vectors at interval  $k+1$  given their values at interval  $k$  are independent of the vectors' values at any other preceding time interval. The control input is a function of the mobile's present location as well as the final destination of the trip and factors such as present traffic conditions.

If a discrete time evolution process is a Gaussian Markov process and the measurement equation is linear with Gaussian noise, then the optimal filtering algorithm is the Kalman filter [34]. The type of density of  $\mathbf{v}(k)$ , the measurement noise, is not known. It has been shown that the Kalman filter can still provide good performance in many cases when the measurement noise is not Gaussian, so it is still applied to this problem. The time evolution process for mobile terminal location is a Gaussian Markov process if  $\mathbf{u}(k)$  is known, as shown in (12). Thus, it is possible to optimally estimate  $\mathbf{x}(k)$  if  $\mathbf{u}(k)$  is known for all  $k$ , if  $\mathbf{v}(k)$  is assumed to be Gaussian.



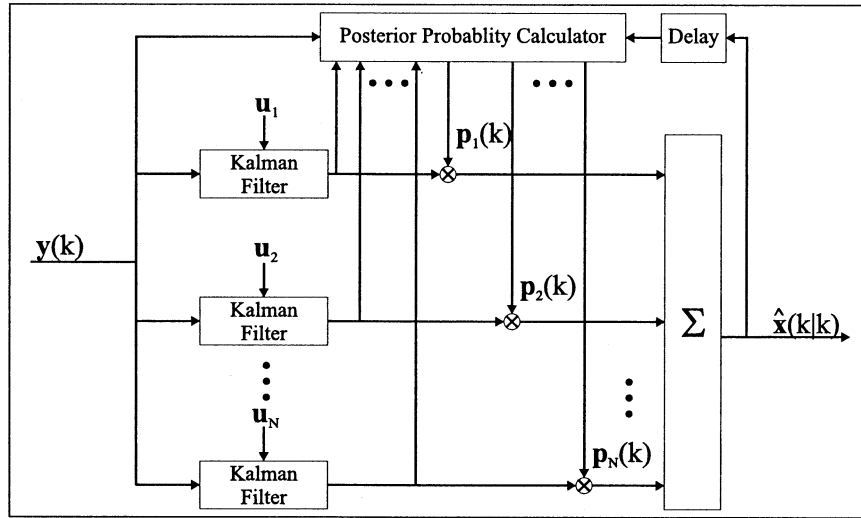


Fig. 6. Interactive multiple model filter structure.

The Kalman filter is briefly described in Appendix II. The use of Kalman filters allows for a location estimate to be calculated along with an estimate of the variance of the location estimate error. The required inputs to the Kalman filter are the measurement vector  $y(k)$  and the estimated variance of the measurement vector  $R(k)$ . From  $R(k)$ , the Kalman filter calculates different weights on the measurements from different sampling periods to calculate the current location of the mobile terminal. If the measurement for a sampling period  $k$  has high estimated error values within its  $R(k)$  matrix, then it will have less weight in the final location estimate. This reduces the impact of GDOP, as isolated measurements with high GDOP will not be given much weight in location estimate calculations. If there are several sampling periods with large  $R(k)$ , perhaps caused by GDOP, the accuracy of the final location estimates will be reduced but the calculated error covariance by the Kalman filter will be high, so that the user will be aware of the inaccuracy. The final accuracy will still be higher than for zero memory estimator since information from measurements taken in different sampling intervals will be combined.

The location process is a Gaussian Markov process, allowing optimal estimation by the Kalman filter, only if the user input vector  $u(k)$  is known. Unfortunately, the value of  $u(k)$  must be estimated by the location estimation process. A commonly applied solution would be to augment the state vector with the control input and then use a Kalman-like filter to estimate the control input as well as the system state. Unfortunately,  $x(k)$  and  $u(k)$  are not jointly Gaussian, and the control input process function is discontinuous. This makes the augmented motion model of  $x(k)$  and  $u(k)$  nonlinear. Many of the standard nonlinear filtering algorithms, such as the extended Kalman filter, will have a high probability of not converging because of the discontinuities in the control input process [35].

As was shown in Section III, during normal vehicle motion,  $u(k)$  takes on the value of one member from a small discrete set of possible control input vectors. That is,  $u(k) \in \{u_1, u_2, \dots, u_N\}$ . Section IV-A gave a model where the control input vector process is described as a Markov

process where the control input state transition probabilities are a function of the current location of the mobile terminal. Knowledge of the relationship between  $x(k)$  and  $u(k)$  and their time evolutionary behavior are used to derive a new filtering algorithm.

Given the problem with state augmentation techniques, an alternative approach to the problem of estimating the system state is to decompose the model of the system state into several possible models for the location model. Each different model corresponds to a different possible value of the control input vector. An estimate of the system state for the mobile terminal is calculated for each of the possible models. The estimation problem is then how to calculate the probability weight for the location estimate calculated with the different assumed models and how to use these estimates and probability weights to calculate a combined location estimate [36].

The estimation motion model assumes that the input vector selection process is a Markov chain given the system state

$$\begin{aligned} \Pr[u(k) = u_j | x(k-1), u(k-1), u(k-2), \dots, u(0)] \\ = \Pr[u(k) = u_j | x(k-1), u(k-1)]. \end{aligned} \quad (14)$$

The input vector selection process is specified by the initial selection and control input vector state transition probabilities. As a consequence of the switching behavior of the control input process, the estimation procedures for each of the different models need to interact to properly handle the switching probabilities.

The organization of the new multimodel filter is shown in Fig. 6. Several Kalman filters are run in parallel with different hypothetical values of the control input  $u(k)$ . Since each Kalman filter gets identical values for the process noise covariance  $Q$  and measurement noise covariance  $R(k)$ , each Kalman filter will have the same Kalman gain  $K(k)$  and covariance  $P(k|k-1)$  and  $P(k|k)$ . Most of the calculations are shared between the Kalman filters. Only the different location estimates need to be calculated using the hypothetical control input values. The output of the Kalman filters is provided to a

posterior calculation algorithm, which determines the posterior probability of each process model generating the observed input. The posterior probabilities are then used to generate multiplicative weights for each of the Kalman filter outputs so a combined state estimate can be calculated.

The final state estimate is given by

$$\begin{aligned}\hat{\mathbf{x}}(k|k) &= E[\mathbf{x}(k)|\mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)] \\ &= E\{E[\mathbf{x}(k)|\mathbf{u}(k-1)]|\mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)\} \\ &= \sum_{j=1}^N \mathbf{p}_j(k) \mathbf{x}_j(k|k)\end{aligned}\quad (15)$$

where  $\mathbf{x}_j(k|k)$  is the output of the  $j$ th Kalman filter and  $\mathbf{p}(k)$  is the weight vector with  $\mathbf{p}_j(k)$  being its  $j$ th entry, defined as the probability of model  $j$ 's being the true model given the observed measurements

$$\mathbf{p}_j(k) = \Pr[\mathbf{u}(k-1) = \mathbf{u}_j | \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)]. \quad (16)$$

Equation (15) is derived using the smoothing property of expectation  $E\{E[Y|X]\} = E[Y]$  [27]. In the multimodel filter, the density of  $\hat{\mathbf{x}}(k|k)$ , a sum of Gaussian densities with each model contributing one term, is approximated by a single Gaussian density. Without this approximation, the number of terms in the density calculation would grow exponentially with each sampling period. This approximation is justified since usually only one of the terms has a weight  $\mathbf{p}_j(k)$  near unity, while all other terms have weights near zero.

Bayes' theorem can be applied to (16) to obtain (17), as shown at the bottom of the page [37]. The denominator term is a normalizing constant independent of the input, so its value need not be calculated. The first term in (17) is expanded using the theorem of total probability as

$$\begin{aligned}\Pr[\mathbf{u}(k) = \mathbf{u}_i | \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)] \\ = \sum_{j=1}^N \Pr[\mathbf{u}(k) = \mathbf{u}_i | \mathbf{u}(k-1) = \mathbf{u}_j, \\ \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)] \mathbf{p}_j(k).\end{aligned}\quad (18)$$

The estimated probability weights of the models for the first sampling period  $\mathbf{p}(0)$  ideally would be the true initial conditional probabilities for the different control inputs. In comparison to the algorithm presented here, the so-called interactive multiple model (IMM) filter assumes that  $\mathbf{u}(k)$  is independent of the measurements  $\mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)$ . As shown in (14), the location and control input are dependent, which makes the measurements and control input dependent. Therefore, we cannot make the same independence assumption as the IMM algorithm.

The information about the system state contained in all measurements up to time interval  $k-1$  is summarized in the system state estimate  $(\hat{\mathbf{x}}(k-1|k-1), \mathbf{P}(k-1|k-1))$ . It can be easily

seen from (12) that given  $\mathbf{u}(k-1)$ ,  $\mathbf{x}(k)$  is independent of  $\mathbf{u}(k)$ , and therefore  $\mathbf{y}(k)$  is independent of  $\mathbf{u}(k)$  given  $\mathbf{u}(k-1)$ .

This allows us to use the approximation

$$\begin{aligned}\hat{\Pr}[\mathbf{u}(k) = \mathbf{u}_i | \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)] \\ = \sum_{j=1}^N \Pr[\mathbf{u}(k) = \mathbf{u}_i | \mathbf{u}(k-1) = \mathbf{u}_j, \\ \mathbf{x}(k-1) = \hat{\mathbf{x}}(k-1|k-1)] \mathbf{p}_j(k).\end{aligned}\quad (19)$$

The value of  $\hat{\mathbf{x}}(k-1|k-1)$  is the location estimate for the last sampling period calculated using (15). An alternative solution would be to use the location estimate from the filter with the highest probability weight to determine the state transition probabilities. But, as this gives a location estimate with higher variance, it also results in estimated transition probabilities with higher errors than using the weighted average.

To use this dependency in the state estimator, the estimated state determines the state transition matrix used in the model detection algorithm. The location region is split into  $N_{\text{region}}$  regions, each of which has a different control state transition matrix. The estimated control state transition matrix is given by

$$\hat{\Theta}(k) = \begin{cases} \Theta_1, & \text{if } \hat{\mathbf{x}}(k-1|k-1) \in \mathcal{S}_1 \\ \Theta_2, & \text{if } \hat{\mathbf{x}}(k-1|k-1) \in \mathcal{S}_2 \\ \vdots & \\ \Theta_{N_{\text{region}}}, & \text{if } \hat{\mathbf{x}}(k-1|k-1) \in \mathcal{S}_{N_{\text{region}}} \end{cases} \quad (20)$$

where  $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{N_{\text{region}}}\}$  are disjoint regions in the location space. For this application, the regions and transition probabilities are described in Table III. The weight update calculation becomes

$$\begin{aligned}\mathbf{p}_i(k+1) &= \frac{f(\mathbf{y}(k+1) | \mathbf{u}(k) = \mathbf{u}_i, \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1))}{f(\mathbf{y}(k+1) | \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1))} \\ &\times \sum_{j=1}^N \hat{\Theta}_{ij}(k) \mathbf{p}_j(k)\end{aligned}\quad (21)$$

where  $\hat{\Theta}(k)$  is determined by (20). The conditional density of  $\mathbf{y}(k+1)$  in (21) gives the evidence that the last measurement provides regarding the control input vector state. It is assumed that the measurement vector given the past measurements and control inputs is jointly Gaussian. The conditional density function is then given by

$$\begin{aligned}f(\mathbf{y}(k+1) | \mathbf{u}(k) = \mathbf{u}_i, \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)) \\ = \frac{1}{2\pi} |\mathbf{C}(k)|^{-\frac{1}{2}} \\ \times \exp \left\{ -\frac{1}{2} [\mathbf{y}(k+1) - \mathbf{m}_i(k)]^T \right. \\ \left. \times [\mathbf{C}(k)]^{-1} [\mathbf{y}(k+1) - \mathbf{m}_i(k)] \right\}\end{aligned}\quad (22)$$

where

$$\mathbf{m}_i(k) = \mathbf{H}[\Phi \hat{\mathbf{x}}(k|k) + \Gamma \mathbf{u}_i]$$

---


$$\begin{aligned}\mathbf{p}_i(k+1) &= \Pr[\mathbf{u}(k) = \mathbf{u}_i | \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1)] \\ &\times \frac{f(\mathbf{y}(k+1) | \mathbf{u}(k) = \mathbf{u}_i, \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1))}{f(\mathbf{y}(k+1) | \mathbf{y}(k), \mathbf{y}(k-1), \dots, \mathbf{y}(1))}\end{aligned}\quad (17)$$

the mean value of the estimated location for time interval  $k+1$  given the measurements up to time interval  $k$  assuming  $\mathbf{u}(k) = \mathbf{u}_i$  and

$$\mathbf{C}(k) = \mathbf{H}\mathbf{P}(k+1|k)\mathbf{H}^T + \mathbf{R}(k+1)$$

the covariance of the estimated measured location. The filter uses the difference between the expected predicted measurement and the true measurement, the “innovation sequence,” for each filter and the transition probabilities for the control input state to update the estimated probability weights for the possible control inputs [38].

When the input vector to the motion model is constant for several steps, the estimate for the Kalman filter with the input vector matched to the true input vector will have the lowest mean squared error. The other Kalman filter location estimates will have estimates of the system state with larger mean squared errors since they have assumed control inputs that do not match the true control input. The multimodel filter will give the Kalman filter matched to the true input vector the highest weight in the final system state estimate.

If the input vector changes after the control input has remained unchanged for several sample intervals, it takes the Kalman filter that is matched to the new input several sampling intervals to converge to the system state of mobile terminal. The newly matched Kalman filter must first remove the errors from its location estimate. Meanwhile, the Kalman filter that matches the previous input vector will generate state estimates with asymptotically increasing errors as the effects of the new mismatch build up. The effect of these transition effects is that the control input estimation algorithm requires several sampling periods to properly identify the new input vector.

One method to increase the rate at which the Kalman filters adjust to input vector changes is to elevate the assumed covariance of the process noise over the value given in the mobility model [37]. This increased covariance reflects uncertainty in the knowledge of the input vectors. The process noise covariance, a design parameter in the setting of Kalman filters, is set to

$$\mathbf{Q} = \mathbf{Q}_{\text{model}} + Q_u \mathbf{\Gamma}\mathbf{\Gamma}^T \quad (23)$$

where  $\mathbf{Q}_{\text{model}}$  is the process noise in the kinematic model of the mobility model and  $Q_u$  is a positive scalar constant.

Higher values of  $Q_u$  increase the magnitude of the values in  $\mathbf{Q}$  and thus  $P(k|k-1)$ , the covariance of the estimated system state. This higher covariance of the system state results in the Kalman filters' giving more weight to the measurements in estimating the system state than on the state predictions made using the kinematic model and assumed inputs. The Kalman filter that has an assumed input matching the true control input will have worse performance with higher  $Q_u$  since its state estimates will be placing less weight on accurate state predictions made by a good model than with lower  $Q_u$ . Conversely, the Kalman filters with assumed inputs not matching the true control input will give better performance with higher  $Q_u$  since their state estimates will be placing less weight on predictions made by partially mismatched models than for lower  $Q_u$ . The value of  $Q_u$  affects the speed at which the multimodel filter detects the Kalman filter with the matching control input after the control input changes.

TABLE IV  
POSTPROCESSING ADJUSTMENTS FOR LANE USE

Condition	Adjustment
$\hat{x}_2(k k) < -3$ meters/second	$\hat{x}_3(k k)_{\text{adjusted}} = \hat{x}_3(k k) + 5$
$\hat{x}_2(k k) > +3$ meters/second	$\hat{x}_3(k k)_{\text{adjusted}} = \hat{x}_3(k k) - 5$
$\hat{x}_4(k k) < -3$ meters/second	$\hat{x}_1(k k)_{\text{adjusted}} = \hat{x}_1(k k) - 5$
$\hat{x}_4(k k) > +3$ meters/second	$\hat{x}_1(k k)_{\text{adjusted}} = \hat{x}_1(k k) + 5$

The optimal value of  $Q_u$  is a tradeoff between the performance of the Kalman filter with the correct input vector and the performance of the filters with mismatched input vectors. Low values of  $Q_u$  will give asymptotically better performance if the input vector remains constant for a long period of time at the cost of longer convergence times after the input vector changes. The range of values for  $Q_u$  that give reasonable estimator performance is from zero to the variance of the control input in one of the coordinate axis directions. In this application, the control input for either the  $x$ - or  $y$ -directions can go from  $-C$  to  $C$ , where  $C$  is the maximum acceleration of the vehicle from Section IV-A. Since the mean of the control inputs is zero, it can be easily shown that the variance of the control input along one axis in this application is less than  $C^2$ ; therefore  $0 \leq Q_u < C^2$ .

1) *Postprocessing*: Another advantage of the model-based filtering approach is that dependency of vehicle location and velocity can be exploited to improve location accuracy. In this application, the zero memory estimator does not have enough resolution to identify in what lane the mobile terminal is traveling. If the survey points are taken in the center of the street, as described in Section II-A, the zero memory estimator will usually return estimated location in the center of the street when the mobile terminal is not located in intersections. The velocity estimate produced by the multimodel filter is used to improve the location estimate by adjusting the location estimate for the proper lane indicated by the estimated terminal velocity. The lane and direction of travel relationship are shown in Fig. 4. The postprocessing is given by Table IV and is performed when the zero memory location estimate is not within an intersection. No adjustments are performed if the mobile is moving at low velocity, as it could be performing maneuvers that are not restricted by lane use.

2) *Filter Initial Conditions*: The multimodel filter is recursive and causal. The filter uses its estimate of state at time interval  $k-1$  to estimate the state at time interval  $k$ . The recursive nature of the filter means the implementation memory requirements of the filter are finite, which allows it to be used in field implementations. In order to effectively estimate the state of the system, the initial estimate  $\hat{\mathbf{x}}(0|0)$  and its covariance  $\mathbf{P}(0|0)$  have to be accurate estimates of the true mean initial system state and covariance. The covariance matrix indicates the accuracy of the initial state estimate; high values in  $\mathbf{P}(0|0)$  indicate a large uncertainty in the initial state estimate.

A good selection of initial conditions for the filters is essential for the estimation algorithm to operate well. To provide unbiased state estimates, the state vector and covariance matrices used to initialize a Kalman filter must match the mean of the ini-

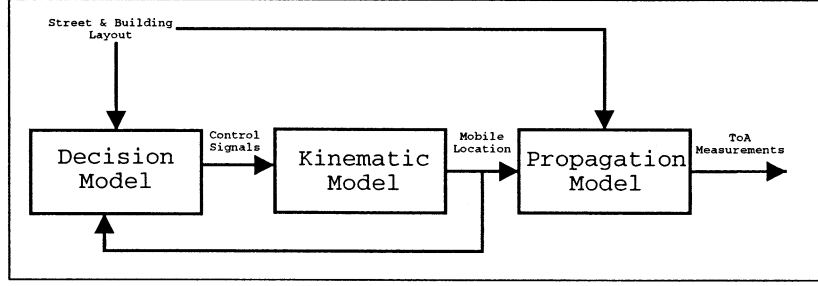


Fig. 7. Simulation models interaction.

tial state and covariance of the initial state for the process to be estimated. The initial conditions selected for the filters must also be reasonable, in that they should be values that would be available to a field implementation. In this application, we use the zero memory estimator to obtain initial estimates of the system state.

The initial conditions for all the Kalman filters in the multi-model estimation filter are identical, as we assume that the initial system state is independent of the control input values. In the absence of other information, all input vectors are equally likely at the first sampling interval  $\mathbf{p}_j(0) = (1/5)\forall j \in \{1, 2, 3, 4, 5\}$ . The zero memory estimator is applied to the first time delay measurements  $\mathbf{z}(0)$  to obtain an initial mobile terminal position estimate  $\mathbf{y}(0)$  and position covariance estimate  $\mathbf{R}(0)$ . The initial system state estimate is based on the first location estimate from the zero memory filter

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} \mathbf{y}_1(0) \\ 0 \\ \mathbf{y}_2(0) \\ 0 \end{bmatrix}.$$

The covariance of the first system state estimate is then

$$\mathbf{P}(0|0) = \begin{bmatrix} \mathbf{R}_{11}(0) & 0 & \mathbf{R}_{12}(0) & 0 \\ 0 & \frac{15^2}{3} & 0 & 0 \\ \mathbf{R}_{12}(0) & 0 & \mathbf{R}_{22}(0) & 0 \\ 0 & 0 & 0 & \frac{15^2}{3} \end{bmatrix}. \quad (24)$$

The variance of the velocity estimates  $\mathbf{P}_{22}(0|0)$  and  $\mathbf{P}_{44}(0|0)$  is based on the initial velocities in the  $x$ - and  $y$ -coordinates being uniformly distributed between  $-15$  and  $15$  m/s, which is taken from the estimator motion model. Initial velocity is assumed to be independent of initial position. It is assumed that the zero memory estimator is unbiased so the initial estimated system state obtained from it is an unbiased estimate of the true initial location. The initial covariance is only an estimate of the true covariance, which will result in errors in the state estimates. Fortunately, the system state process has limited memory, which means the dependency of the system state at interval  $k$  and the system state at interval  $k + m$  decreases with the magnitude of  $i$ . The effect of estimation error for time interval 0 will decrease with each additional time interval. The influence of measurements made in later sample periods will eventually remove the effect of initial location estimate errors.

3) *Location Estimation When Signal Unavailable:* It may occur that during a given sampling interval  $k$ , there are fewer than three base stations that receive a strong enough signal from

the mobile terminal to make a valid zero memory estimate. Field measurements have shown that this case is rare in urban environments [15]. The use of dynamic estimation allows a location estimate to still be calculated for these intervals. Other approaches from zero memory estimation do not provide a systematic method for location estimation during periods of signal unavailability since there is no formulation for  $\mathbf{x}(k)$  based on  $\mathbf{x}(k-1)$  [2], [21]. The estimated mobile terminal location for such a sampling interval is the predicted mobile terminal location from the last sampling interval with a measurement vector  $\hat{\mathbf{x}}(k|k-l)$ , where  $l$  is the number of sampling periods where  $y(k)$  has not been available. The predicted location can be easily calculated using the motion model

$$\begin{aligned} \hat{\mathbf{x}}(k|k-l) &= \mathbb{E}[\mathbf{x}(k)|\mathbf{y}(k-l), \dots, \mathbf{y}(1)] \\ &= \sum_{j=1}^N \mathbf{p}_{j(k)} \mathbf{x}_j(k|k-l) \end{aligned} \quad (25)$$

where

$$\begin{aligned} \mathbf{p}(k) &= \hat{\boldsymbol{\Theta}}(k-1) \mathbf{p}(k-1) \\ \mathbf{x}_j(k|k-l) &= \boldsymbol{\Phi}^l \mathbf{x}_j(k-l|k-l) + \sum_{a=0}^{l-1} \boldsymbol{\Phi}^a \boldsymbol{\Gamma} \mathbf{u}_j. \end{aligned}$$

$\hat{\boldsymbol{\Theta}}(k-1)$  is calculated from (20). The covariance of the location estimate is calculated by applying the recursion  $\mathbf{P}(k|k-1) = \boldsymbol{\Phi} \mathbf{P}(k-1|k-1) \boldsymbol{\Phi}^T + \mathbf{Q}$ ,  $l$  times. When the signal is again sufficient to allow for calculation of  $y(k)$ , the estimation procedure continues as described in the preceding sections, replacing estimated vectors and matrices predicted from the  $k-1$  interval with values estimated from the  $k-l$  interval as needed.

## V. SIMULATIONS

The tracking algorithms' performance is evaluated using simulations of vehicle motion in a dense urban area. The simulation model consists of three parts: the kinematic model, the decision model, and the propagation model. The kinematic model will determine the mobile terminal's acceleration, velocity, and position in response to control inputs. The decision model will mimic the driver's decisions as to lane selection and whether to turn or brake at an intersection. The propagation model generates the simulated measurements from the mobile terminal's location. The relationships between the different simulation models are summarized in Fig. 7.

The propagation model and kinematic model are based on the actual physical phenomena that govern radio signal propagation

TABLE V  
MOTION SIMULATION PARAMETERS

Parameter	Symbol	Value
Drag	$\alpha$	$\frac{1}{6}^*$
Lane Control	$\beta$	$-\frac{1}{4}$
Braking Distance	$B$	40 m
Turning Probability	$P(\text{turn})$	$\frac{2}{3}$
Maximum Mean Acceleration	$C$	2.5 meters/second <sup>2</sup>
Block Length	$D$	300 m
Street Width	$W$	20 m

\*  $\alpha$  value when not braking.

and vehicular motion and are described in Sections II and III, respectively. The decision model used to simulate mobile terminal motion is described in the section below. The constants used in the simulations are listed in Table V.

#### A. Decision Model

The decision model generates the control inputs into the kinematic model that will determine future mobile terminal position based on the mobile terminal's current location and the simulated street layout.

Modeling human driver decision patterns accurately is difficult for computer simulations. The solution proposed in this paper is to use a model for driver behavior that has higher entropy than actual driver behavior. This means that the mobile terminal position state at a sample interval  $k$  gives less information about the mobile terminal position state at interval  $k + N$  with  $N > 0$  in the simulation motion than for true vehicular motion. The simulated motion is harder to track than the motion that would be generated by human drivers since past measurements of mobile terminal position give less information about the present mobile terminal location than for motions generated by human decisions.

The simulated driver in the model described below makes decisions at every intersection independently of the previous intersection decision. Thus the tracking algorithm cannot use any form of long-term behavior model to improve performance. The tracking algorithm's performance for mobiles with motion controlled by human drivers is likely to be superior to that for mobiles with the control logic described below.

When the mobile terminal is not located near intersections, movement is restricted to the lane specified for the mobile's direction of motion. This behavior is as described in Section III-B1.

We use a simplified decision logic system to simulate driver decision behavior. The simulated driver's decision at each intersection is independent of the decision made at any other intersection during the mobile's journey.

A simple finite-state machine is used to control mobile decisions. The simulated driver is in one of four states: normal, braking, turning, and transit. The state transition diagram

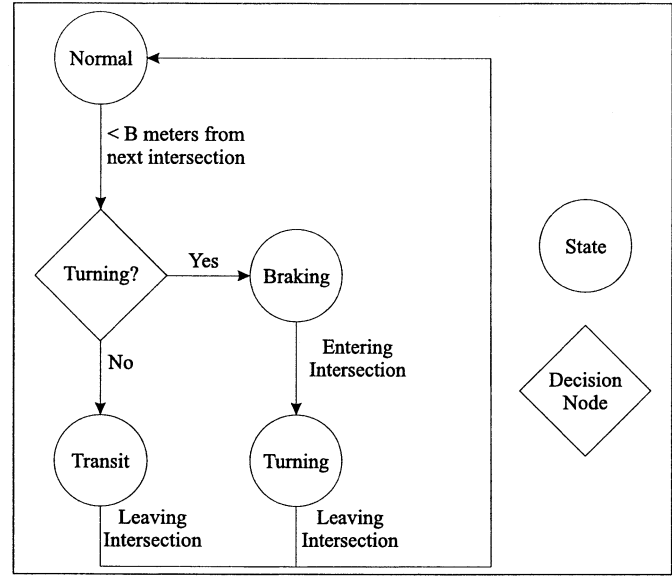


Fig. 8. State transition diagram for motion simulator.

is shown in Fig. 8. In a true urban environment, there is a random chance that a vehicle would be forced to brake at each intersection encountered because of traffic control signals or road congestion. In this simplified model, a vehicle only brakes at an intersection when making a turn, but it must always come to a complete stop before turning.

1) *Normal State*: The mobile starts in the normal state. The mobile terminal in this state is not located in an intersection. The mobile will, after an initial period of acceleration, move at the mean velocity of 15.0 m/s down the street while staying in the proper lane. The control inputs for the main direction of travel will be set as shown in Table II.

The position of the center of the next intersection in the decided direction of travel is calculated. If the mobile is within  $B$  m of the next intersection, the mobile decides if it will turn at the next intersection. The probability of turning is given by the constant  $P_{\text{turn}}$ . If the mobile is within  $B$  m of the next intersection and turning, it will transition to the braking state in the next sampling interval. If the mobile is within  $B$  m of the next intersection and not turning, it will change to the transit state in the next sampling interval. If the mobile terminal is farther than  $B$  m from the intersection, then the mobile remains in the normal state.

The distance  $B$  is set to 40 m. This distance was selected based on data on vehicle braking distances in typical urban environments [25]. In other environments, this parameter would be set based on the mean speed in the environment and road conditions. This information is also used by those designing road systems. The network operator needing this information could obtain it from the traffic control authorities for the area of interest.

2) *Braking State*: In the braking state, the mobile's motion will be reduced so that it will stop just inside of the intersection region. The control input in the main direction of travel will be set to zero. The control input in the direction perpendicular to the main direction of travel will still be set to hold the proper lane, just as in the normal state.

The drag needed to bring the mobile to a stop just after entering the intersection is calculated

$$\alpha_{\text{new}} = \frac{v}{d} \quad (26)$$

where  $v$  is the velocity in the selected direction of travel and  $d$  is the distance to the intersection entry point. The drag coefficient  $\alpha$  is set to the value in the interval  $[1/6, 5.0]$  nearest to  $\alpha_{\text{new}}$ . This interval represents the drag coefficients that the vehicle's brakes can generate. When the mobile terminal enters the intersection, the control state transitions to the turn state in the next sampling interval.

3) *Turning State*: In the turn state, the mobile will move the mobile terminal into the proper lane for its new direction of travel. Upon first entering the turn state, the mobile resets the drag coefficient  $\alpha$  back to  $1/6$  and sets the direction of travel to the new direction. The new direction for the mobile is 50% likely to be either of the perpendicular cardinal directions to the mobile's current direction of travel. When the mobile terminal leaves the intersection, it will change to the normal state in the next sampling period. The lane-holding logic is set for the new direction of travel to move the mobile terminal into the new lane.

4) *Transit State*: When the mobile terminal is in the transit state, control inputs will be set as in the normal state. When the mobile terminal is in the transit state, it will set control inputs as in the normal state according to Table II. When the mobile leaves the intersection, it will transit to the normal state in the next sampling period.

The value of  $P_{\text{turn}}$  was set to  $2/3$ . The result of this choice is that when a mobile approaches an intersection, it has an equal probability of going straight, turning left, or turning right. This is the maximum entropy case when the mobile is restricted from going back the direction it came.

### B. Initial Conditions

For the simulation studies, we choose initial conditions in a manner that replicates the random motion state of a mobile terminal that has just been switched on. To simplify the simulation, we always assume that the mobile terminal starts in the central cell with the base station located at coordinates  $(0,0)$ . This is not an unrealistic assumption, as when a mobile terminal initiates a call, it quickly identifies the base station to which it is closest. The location and velocity state parameters are uniformly distributed within the space of possible values. This is the distribution of maximum entropy when no other information is known about the mobile terminal's state.

First the direction of the mobile terminal motion is selected from the possible set of {North, South, East, West}. A position value  $P$  is sampled from a uniform distribution with a range of  $(-D, D)$  where  $D$  is the block length (300 m in Fig. 2). An initial speed  $S$  is sampled from a uniform distribution with a range of  $[0, 15.0]$ . A lane position value  $L$  is sampled from a uniform distribution with a range of  $[0, 10]$ .

From these random values, the initial state of the mobile terminal position is generated as shown in Table VI based on North American lane use.

TABLE VI  
INITIAL STATE OF MOBILE TERMINAL

Initial direction	$p_x(0)$	$v_x(0)$	$p_y(0)$	$v_y(0)$	$u_x(0)$	$u_y(0)$
North	$L$	0	$P$	$S$	0	2.5
South	$-L$	0	$P$	$-S$	0	-2.5
East	$P$	$S$	$-L$	0	2.5	0
West	$P$	$-S$	$L$	0	-2.5	0

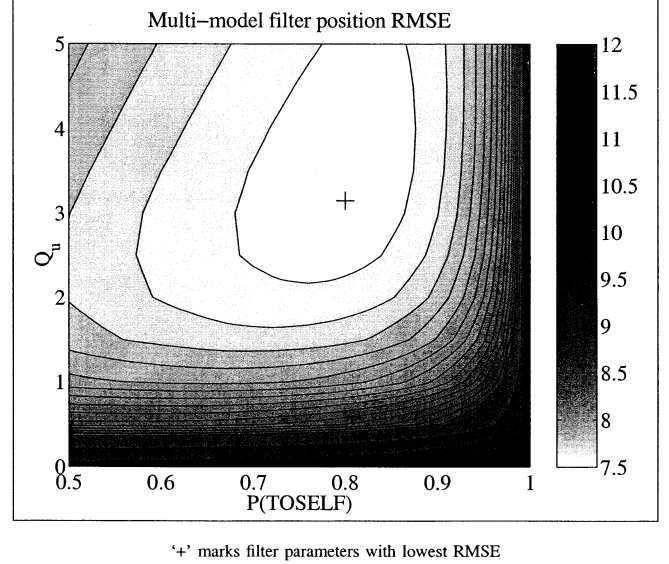


Fig. 9. Location RMSE for  $P_{\text{turn}} = 2/3$  with varying filtering parameters ( $\alpha = 1/6$ ).

## VI. RESULTS

The figures of merit used to analyze the performance of the estimation algorithm are the root mean square error (RMSE) of position and velocity. This figure of merit is widely used in filter evaluation and the location estimation literature [6], [21]. The RMSE is defined as

$$\text{RMSE} = \sqrt{E[\tilde{X}^2 + \tilde{Y}^2]} \quad (27)$$

where  $\tilde{X}$  is the error in the estimated  $x$ -coordinate and  $\tilde{Y}$  is the error in the estimated  $y$ -coordinate. For position RMSE, the  $x$ - and  $y$ -coordinates are the errors in the position estimate for the mobile terminal. For velocity RMSE, the  $x$ - and  $y$ -coordinates specify the velocity error in the estimated mobile terminal system state.

As was described in Section IV, the performance of the estimation algorithm is dependent on the user-selected values of  $Q_u$  and  $P(\text{TOSELF})$ . The optimal values of these parameters for a user-selected figure of merit is dependent on the probability that a mobile terminal will make a turn at an intersection  $P_{\text{turn}}$ . For  $P_{\text{turn}} = 2/3$ , Fig. 9 shows a contour plot for the position RMSE averaged over the initial 100-s interval after filtering is applied (the first 200 samples since the sampling period  $T$  is half a second) for a range of different values of  $Q_u$  and  $P(\text{TOSELF})$ . The minimum position RMSE was obtained for  $Q_u = 3.15$  and  $P(\text{TOSELF}) = 0.80$ . High RMSE is only obtained when

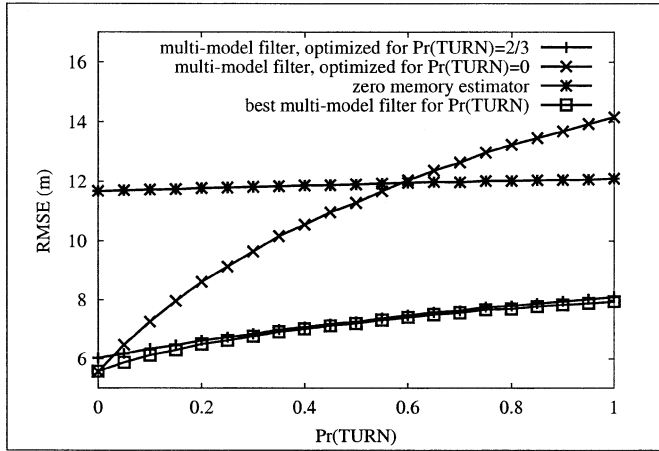


Fig. 10. Robustness with turning probability.

$Q_u \approx 0$  or  $P(\text{TOSELF}) \approx 1$ . Other than these boundary conditions, Fig. 9 shows that the values of  $Q_u$  and  $P(\text{TOSELF})$  can vary significantly from the optimal values without a large increase in the resulting RMSE.

We have shown that for a fixed  $P_{\text{turn}}$ , the multimodel filter is robust to variations of its parameters. It is impossible for the network designer to know the exact turning probability for the mobile terminals at the intersections. This value is also changing with road conditions and time of day. The robustness of a multimodel filter that has been optimized for a given turning probability when the true turning probability varies from the design value is essential if the filter is going to be successful in field implementations. Fig. 10 shows the position RMSE averaged over the first 200 samples after filtering is applied for filters optimized for different assumed values of  $P_{\text{turn}}$  when the true turning probability is varied. For comparison, the averaged RMSEs of the multimodel filters with parameters optimized for each  $P_{\text{turn}}$  are also plotted. The results for the multimodel filter optimized for  $P_{\text{turn}} = 2/3$  show good robustness over a wide range of values for turning probability. The filter optimized for  $P_{\text{turn}} = 0$  gave better performance for low turning probability, but its error increases dramatically with rising turning probability. Fig. 10 shows that the multimodel filter optimized for  $P_{\text{turn}} = 2/3$  gives averaged position RMSE almost as good as the optimal multimodel filter for a wide range of turning probabilities. For this reason, this set of multimodel filter parameters is recommended for field implementations. This graph shows what results can be expected if an accurate long-term behavior model for the mobile terminal motion is available. In this case, the turning probability of the mobile terminal would have less entropy. If the long-term behavior model is perfect, then the RMSE for turning probability of zero from Fig. 10 reflects what the multimodel filter could return. With less accurate long-term mobility models, the results for  $0 < P_{\text{turn}} \leq 2/3$  would reflect the possible performance.

The next set of simulations demonstrates the convergence of the filter algorithm. The position and velocity RMSE are calculated for each sample period  $k$  after filtering was initiated. The turning probability at each intersection was  $2/3$ . The multimodel filter used the optimal values of  $Q_u = 3.15$  and

TABLE VII  
SIMPLE KALMAN FILTER MODEL

Matrix	Value
$\Phi$	$\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$Q$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 \end{bmatrix}$

$P(\text{TOSELF}) = 0.80$  found during the last set of simulations. For comparison, the RMSE performance of a simple Kalman filter is also plotted. This Kalman filter uses the model specified by the matrices in Table VII with no control input. This is the dynamic model used for the Kalman filter in [8]. The assumed covariance of the process noise given by the entries of 1.5 in  $Q$  for the simple Kalman filter model were optimized for the best position RMSE performance. The position RMSE for the simple Kalman filter when using the parameter selection method presented in [8] is also shown. The zero memory estimator's position RMSE is plotted to show the improvement that filtering provides.

In [8], it is assumed that mobile terminal acceleration process is a zero-mean Gaussian process with variance  $\sigma_v^2$  in both the  $x$ - and  $y$ -coordinate directions. If this assumption is true, the magnitude of the acceleration vector for any given time is a Rayleigh distributed random variable with a mean value of  $\sigma_v \sqrt{\pi/2}$ . The average value of the acceleration vector's magnitude is calculated from acceleration measurements made during a sample run and the value of  $\sigma_v$  calculated. The change in the terminal velocity from one sampling instance to the next along one coordinate axis is then a Gaussian random variable with mean zero and variance of  $(\sigma_v \cdot T)^2$ , where  $T$  is the sampling period. We calculated the mean acceleration magnitude to be  $1.677 \text{ m/s}^2$  when the turning probability was  $2/3$ . We therefore replace the value of 1.5 in the  $Q$  matrix in Table VII with  $(0.5)^2 \cdot 1.677 = 0.41925$  for the simulations.

Fig. 11 shows the tracking performance for a random initial condition. The performance for the multiple model filter shows that the position RMSE converges to approximately 7.5 m within 10 s of the motion's starting. The simple Kalman filter converges to a position RMSE of 9 m with about the same convergence time. The zero memory estimator has a position RMSE of about 12 m. The line labeled "Simple Kalman Filter (Hellebrandt)" uses the parameter matrices calculated from [8]. The assumed process noise variance, for this case, is too low

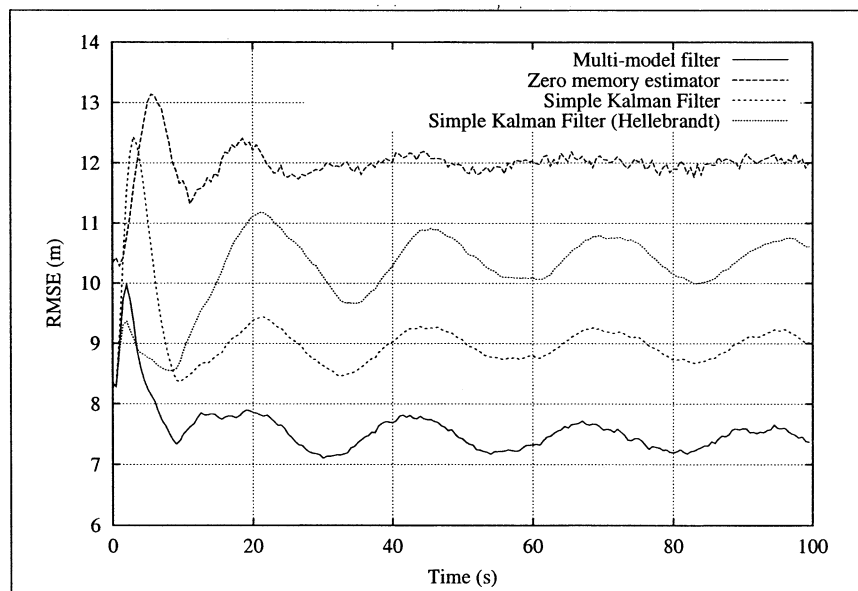


Fig. 11. Filtering location error performance.

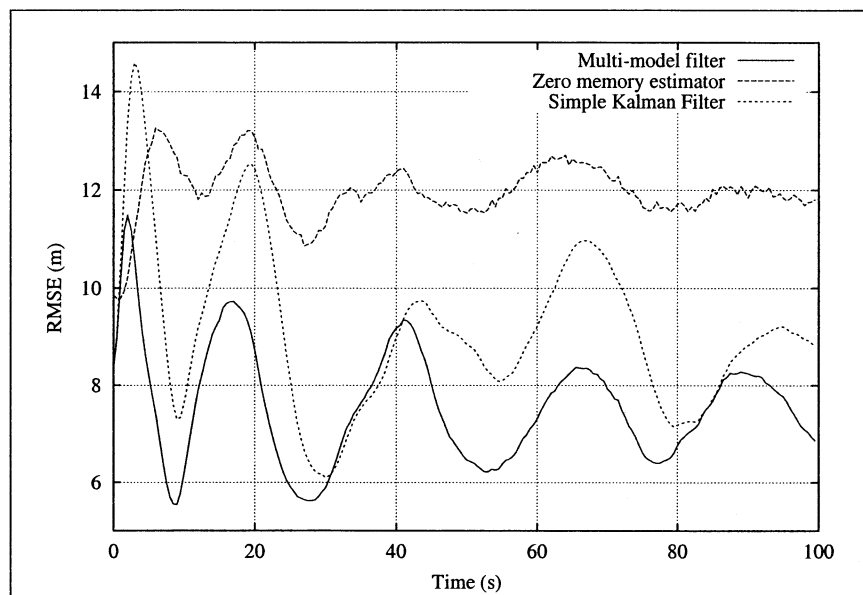


Fig. 12. Filtering location error performance (deterministic starting position).

to handle the turns in the intersections, so the simple Kalman filter diverges. From these results, it would appear that the multimodel filter only reduces the position RMSE by 1.5 m from the best simple Kalman filter, but this result does not show the differing RMSE in intersections.

To illuminate the effect of intersections on tracking performance, simulations were performed with the mobile terminal always starting 150 m away from the next intersection it would enter. The turning probability was set to 2/3. For the simulations with this deterministic starting condition, the mobile terminals enter intersections, on average, at 10 s, 30 s, 50 s, and every 20 s after that since the mean mobile terminal speed is 15 m/s and intersections are 300 m apart. Fig. 12 shows the position RMSE curves for simulations with the deterministic

starting condition. The simple Kalman filter's performance is greatly degraded immediately after the mobile terminal leaves an intersection and the control input changes. The damping that can be seen in the position RMSE error curves for the deterministic starting position is caused by increasing variance in the relative positions of the different simulated mobile terminals with respect to each other caused by the process noise in the simulation motion model. Both the multimodel and simple Kalman filters show a location dependency for position RMSE, giving lower RMSE performance between intersections. At intersections, the multiple model filter manages the higher uncertainty in mobile motion better than the simple Kalman filter, which has spikes of position RMSE error almost as high as the unfiltered position RMSE provided by the zero memory estimator.



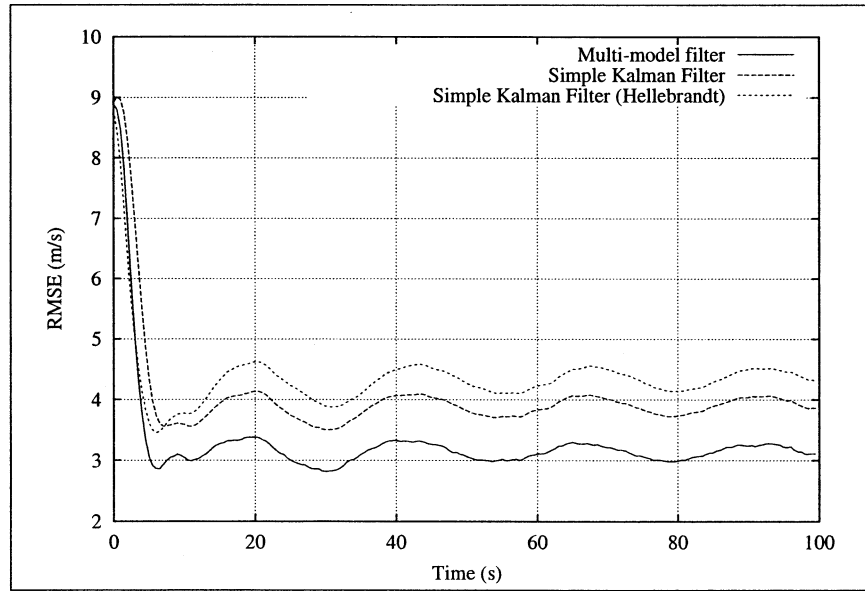


Fig. 13. Filtering velocity error performance.

Fig. 13 shows the velocity RMSE for the multimodel and simple Kalman filter. The multimodel filter provides velocity estimates with lower RMSE than the simple Kalman filter.

The multimodel filter has three advantages over the simple Kalman filter. First, it gives system state estimates with lower error. Second, it handles the uncertainty of mobile terminal motions in an intersection much more robustly than the simple Kalman filter, which has high spikes in estimation error because of intersections. Third, the multimodel filter is robust to varying turning probabilities for the mobile terminal motions. The simple Kalman filter's parameters need to be optimized for the specific turning probability encountered, and robustness for a different turning probability is not guaranteed. The simple Kalman filter does have lower computational cost but the multimodel filter is designed so the multiple Kalman filters contained within it all have the same Kalman gain. So the Kalman gain calculation, the operation requiring the most operations in the Kalman filter, is only performed once and the calculated value used for all the filters. The additional cost of the multiple model filter, the calculation of the posterior probability values from (21) and applying the weights in (15), is minimal compared to the Kalman gain calculation. The multimodel filter requires only a small amount of additional memory compared to the simple Kalman filter. The location estimate for each Kalman filter needs to be stored along with its respective posterior probability. Because each filter has an identical Kalman gain, each also has an identical posterior covariance estimate.

The last set of simulations shows the effect of uncertainty in the knowledge of the kinematic model of mobile terminal motion on the estimation performance of the multimodel filter. The value of  $\alpha$  used for the motion simulation and the estimation model in the filter algorithm are varied in the range [0.1, 0.2], which covers the drag coefficients for most passenger vehicles likely to be used in urban environments. The other parameters of the motion model from Section III were adjusted so the mean and variance of the mobile terminal velocities would remain the

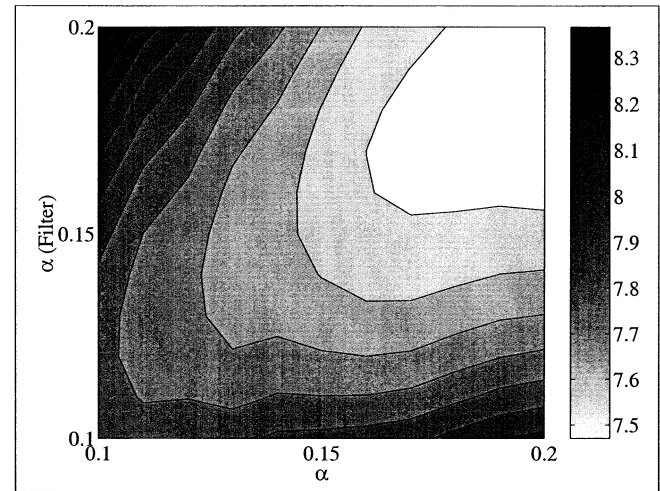


Fig. 14. Location RMSE with varying drag coefficients ( $P_{\text{turn}} = 2/3$ ).

same for all values of  $\alpha$  used. The mean velocities of the mobile terminals were kept constant so the maximum acceleration of the mobile terminals was set to  $C' = 15 \cdot \alpha$  and the mean velocity would remain at 15 m/s. The variance of the process noise was set to  $\sigma^2 = 2\alpha$  so the standard deviation of the velocities would remain at 1 m/s. A contour plot of RMSE location errors for ranges of values for assumed and true values of  $\alpha$  is shown in Fig. 14. The filter parameter values are set to  $Q_u = 3.15$  and  $P(\text{TOSELF}) = 0.8$ . Good performance is always obtained when the assumed  $\alpha$  in the filter  $\alpha(\text{Filter})$  matches the true  $\alpha$  in the motion simulator. It can be seen that the lowered performance for a mismatch between the true  $\alpha$  that generated the measurement data and the assumed  $\alpha$  in the filter algorithm is fairly small. It is possible to estimate  $\alpha$  during filter operation using an extended Kalman filter or the expectation maximization algorithm, but these results show that this estimation would give little extra benefit and at possible large computational cost [20].

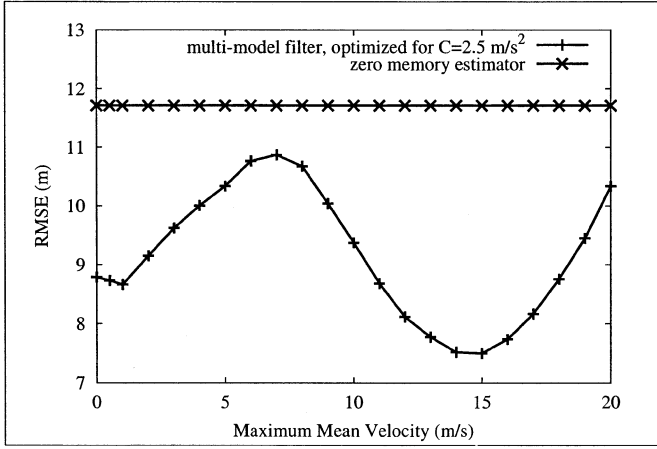


Fig. 15. Location RMSE with varying maximum mean velocity ( $P_{\text{turn}} = 2/3$ ,  $\alpha = 1/6$ ).

In general, larger values of true  $\alpha$  allow for better filter performance. This is a result of the accelerations of the vehicles' being higher, resulting in larger separations between the models in the measurement domain.

Fig. 15 shows the multimodel filter's performance when the assumed value of maximum acceleration  $C$  in the filter does not match the true value for the vehicle. As was shown in Section III-A, from  $C$  the maximum mean velocity is calculated as  $C/\alpha$ . The filter is optimized for  $C = 2.5 \text{ m/s}^2$  for a maximum mean velocity of 15 m/s. Fig. 15 shows that the filter works best when the true maximum mean velocity is near 15 m/s with the filter working nearly as well when the maximum mean velocity is near zero. The filter works well for low velocity because, in this case, the motion of the mobile terminal is well matched by the filter with the assumed value for  $\mathbf{u}(k)$  being the zero vector. This result shows that this filter gives good performance for both vehicular users and mobile terminals carried by low-speed pedestrians.

## VII. CONCLUSION

Estimation of mobile terminal positions based on measurements made at a single time instant will contain errors because of noise in the signal measurements. This paper introduces dynamic filtering to reduce the location estimation error by combining the information from signal measurements made at several time instances to calculate an improved location estimate. Mobile terminal velocity can be estimated jointly with mobile terminal position to facilitate prediction of future mobile terminal locations.

In order for dynamic estimation to be performed successfully, an accurate model for the mobile terminal motion is required. A dynamic model was described that separates the mobile terminal motion into a simple kinematic model for the physical laws governing terminal motion and a user decision model for the human decisions that affect mobile terminal motion.

A multiple model dynamic estimation algorithm was presented that estimates the control input for simple Kalman filter estimation algorithms. This dynamic estimation algorithm uses knowledge of the dependency between control input changes and the location of the mobile terminal to improve the accuracy of its location estimates.

The performance of the estimation algorithm was evaluated using simulations. The robustness of the algorithm to variations in the parameters of mobile terminal motion was demonstrated.

Future work in this area includes investigation of mobile terminal location prediction for resource allocation and handoff algorithms. The incorporation of long-term behavior models for mobile terminal motion is also under investigation. The evaluation of location estimation methods using dynamic filtering needs investigation. Many of the techniques used to evaluate zero memory estimation techniques, such as the GDOP [21], are less useful when dynamic filtering is applied. New performance measures need to be derived based on the effect that time-based filtering can have upon measurement accuracy.

## APPENDIX I

### STATE SPACE MODEL DERIVATION

The state vector is defined as

$$\mathbf{x}(t) = \begin{bmatrix} p_x(t) \\ v_x(t) \\ p_y(t) \\ v_y(t) \end{bmatrix} \quad (28)$$

where  $(p_x(t), p_y(t))$  is the location vector of the mobile terminal and  $(v_x(t), v_y(t))$  is the velocity vector of the mobile terminal. In continuous time, the state space model of vehicular motion can be given by

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\{\mathbf{w}(t) + \mathbf{u}(t)\} \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\alpha \end{bmatrix} \mathbf{x}(t) \\ &\quad + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \left\{ \begin{bmatrix} w_x(t) \\ w_y(t) \end{bmatrix} + \begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix} \right\}. \end{aligned} \quad (29)$$

The terms  $w_x(t)$  and  $w_y(t)$  represent zero-mean white Gaussian noise processes with variances of  $E[w_x(0)^2] = E[w_y(0)^2] = \sigma^2$ , which are the process noise terms for the continuous time dynamic model. The deterministic inputs, representing driver control input in the  $x$ - and  $y$ -directions, are given by  $u_x(t)$  and  $u_y(t)$ . These inputs determine the direction that the mobile terminal will move. If for all values of  $t \geq T_f$ ,  $\mathbf{u}_x(t) = \mathbf{u}_x(T_f)$  and  $\mathbf{u}_y(t) = \mathbf{u}_y(T_f)$ , then

$$\lim_{t \rightarrow \infty} E \left\{ \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} \right\} = \frac{1}{\alpha} \begin{bmatrix} u_x(\infty) \\ u_y(\infty) \end{bmatrix}.$$

Thus,  $\mathbf{u}_x(t)$  and  $\mathbf{u}_y(t)$  determine the final direction of motion. If the control inputs change, the mobile terminal motion will smoothly change to the new direction of motion as the drag term forces the velocity functions to remain continuous.

The asymptotic covariance of the velocities can be easily found to be

$$\lim_{t \rightarrow \infty} \text{Cov} \left\{ \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} \right\} = \begin{bmatrix} \frac{\sigma^2}{2\alpha} & 0 \\ 0 & \frac{\sigma^2}{2\alpha} \end{bmatrix}.$$

In practice, we can only sample measurements of the state of the system at discrete times. We will assume that the state is

sampled with a sampling period of  $T$  s. The discrete state vector is given by

$$\mathbf{x}(k) = \begin{bmatrix} p_x(kT) \\ v_x(kT) \\ p_y(kT) \\ v_y(kT) \end{bmatrix}. \quad (30)$$

A discrete version of the dynamic model can be obtained from the continuous time model [39]. We make the simplifying assumption that the input vector  $\mathbf{u}(t)$  changes only at the sample times. Obviously, the inputs can change at any time instant, not just at the sampling instants. The error introduced by this mismatch between the modeling assumptions and real model will be negligible provided the sampling period is small compared to the time constant of the continuous system  $\alpha^{-1}$ . The sampling period is set at  $T = 0.5$  s, which is less than the time constant of the system of  $1/\alpha = 6.0$  s—which justifies the assumption made to discretize the continuous state space model. The resulting discrete time dynamic model is given by

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma \mathbf{u} + \mathbf{w}(k) \quad (31)$$

where

$$\begin{aligned} \Phi &= \begin{bmatrix} 1 & \frac{(1-\exp(-\alpha T))}{\alpha} & 0 & 0 \\ 0 & \exp(-\alpha T) & 0 & 0 \\ 0 & 0 & 1 & \frac{(1-\exp(-\alpha T))}{\alpha} \\ 0 & 0 & 0 & \exp(-\alpha T) \end{bmatrix} \\ \Gamma &= \begin{bmatrix} \frac{\exp(-\alpha T)-1+\alpha T}{\alpha^2} & 0 \\ \frac{1-\exp(-\alpha T)}{\alpha} & 0 \\ 0 & \frac{\exp(-\alpha T)-1+\alpha T}{\alpha^2} \\ 0 & \frac{1-\exp(-\alpha T)}{\alpha} \end{bmatrix} \\ \mathbf{Q} &= E \left[ \int_0^T \exp(\mathbf{A}t_1) \mathbf{B} \mathbf{W}(t_1) dt_1 \right. \\ &\quad \times \left. \left( \int_0^T \exp(\mathbf{A}t_2) \mathbf{B} \mathbf{W}(t_2) dt_2 \right)^T \right] \\ &= E [\mathbf{w}(k) \mathbf{w}(k)^T] \\ &= \begin{bmatrix} r_{11} & r_{12} & 0 & 0 \\ r_{12} & r_{22} & 0 & 0 \\ 0 & 0 & r_{11} & r_{12} \\ 0 & 0 & r_{12} & r_{22} \end{bmatrix}. \end{aligned}$$

The components of the process noise covariance  $\mathbf{Q}$  are given by

$$\begin{aligned} r_{11} &= \frac{\sigma^2 (2\alpha T - 3 + 4\exp(-\alpha T) - \exp(-2\alpha T))}{2\alpha^3} \\ r_{12} &= \frac{\sigma^2 (1 - \exp(-\alpha T))^2}{2\alpha^2} \\ r_{22} &= \frac{\sigma^2 (1 - \exp(-2\alpha T))}{2\alpha}. \end{aligned}$$

For handoff measurements, mobile terminals measure the signal for the base stations they are using for primary communications but also the signal from other base stations. It is likely to be these measurements that will be extended for mobile terminal location purposes. Therefore, the sampling period was set to the approximate time between measurements in support of the handoff algorithm in GSM. Other networks standards, e.g., IS-95, have different sampling intervals for

handoff, but the handoff sampling periods are of the same order of magnitude, so the results are still valid.

## APPENDIX II KALMAN FILTER

The Kalman filter assumes that  $\hat{\mathbf{x}}(k-1|k-1)$  is a jointly Gaussian random vector with the mean  $\mathbf{x}(k-1)$ , the true system state at sample time  $k-1$ , and covariance matrix  $\mathbf{P}(k-1|k-1)$ . It also assumes the measurement vector for sample time  $k$  is given by

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (32)$$

where  $\mathbf{v}(k)$  is a jointly Gaussian random vector with zero mean and covariance matrix  $\mathbf{R}(k)$  that is independent of the process noise  $\mathbf{w}(k)$ .

The optimal filtering algorithm consists of two stages: the prediction stage and the correction stage. The prediction stage recursively predicts the system state at interval  $k$  from the estimate of the system state at interval  $k-1$ . The correction stage then uses the measurement taken at interval  $k$  and combines it with the prediction to get a corrected estimate of the state at interval  $k$ .

During the prediction stage, the value of  $\mathbf{x}(k)$  is predicted from the estimate of  $\mathbf{x}(k-1)$  using the measurements taken up until sampling interval  $k-1$ . The prediction is given by

$$\hat{\mathbf{x}}(k|k-1) = \Phi \mathbf{x}(k-1|k-1) + \Gamma \mathbf{u}(k-1) \quad (33)$$

with the covariance of the predicted system state given by

$$\begin{aligned} \mathbf{P}(k|k-1) &= E [(\hat{\mathbf{x}}(k|k-1) - \mathbf{x}(k))(\hat{\mathbf{x}}(k|k-1) - \mathbf{x}(k))^T] \\ &= \Phi \mathbf{P}(k-1|k-1) \Phi^T + \mathbf{Q}. \end{aligned} \quad (34)$$

The correction stage of the filter incorporates the new measurement for sampling interval  $k$ ,  $\mathbf{y}(k)$ , to improve the estimate of the location state. The first step to the correction is to calculate the innovation at sample time  $k$ ,  $\tilde{\mathbf{y}}(k)$ , the difference between the predicted measurement and actual measurement at time  $k$

$$\tilde{\mathbf{y}}(k) = \mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1). \quad (35)$$

The corrected estimate of the system state at time  $k$  is calculated using

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)\tilde{\mathbf{y}}(k) \quad (36)$$

where  $\mathbf{K}(k)$  is called the Kalman gain. The Kalman gain is given by

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T [\mathbf{R}(k) + \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T]^{-1}. \quad (37)$$

The updated covariance of the estimate is given by

$$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}]\mathbf{P}(k|k-1) \quad (38)$$

where  $\mathbf{I}$  is the appropriately sized identity matrix. The Kalman filter has been proven to be unbiased, i.e.,  $E[\hat{\mathbf{x}}(k|k)] = \mathbf{x}(k)$ , and optimal if the process noise and measurement noise densities are Gaussian and the time evolution and measurement equations are linear [34].

## REFERENCES

- [1] FCC, "Report and Order and Further notice of proposed rulemaking in the matter of revision of the commission's rules to ensure compatibility with enhanced 911 emergency calling systems," FCC docket 94-102, June 1996.

- [2] J. J. Caffery Jr and G. L. Stüber, "Subscriber location in CDMA cellular networks," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 406–416, May 1998.
- [3] Z. Salcic, "GSM mobile station location using reference stations and artificial neural networks," *Wireless Personal Commun.*, vol. 19, no. 3, pp. 205–226, Dec. 2001.
- [4] I. Jami, M. Ali, and R. F. Ormondroyd, "Comparison of methods of locating and tracking cellular mobiles," *Inst. Elect. Eng. Colloq. Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications*, pp. 1/1–1/6, 1999.
- [5] R. R. Collman, "Evaluation of methods for determining the mobile traffic distribution in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 1629–1635, Nov. 2001.
- [6] P.-C. Chen, "A cellular based mobile location tracking system," in *IEEE Vehicular Technology Conf.*, May 1999, pp. 1979–1983.
- [7] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling, location tracking, and trajectory prediction in wireless atm networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 922–936, Aug. 1998.
- [8] M. Hellebrandt and R. Mathar, "Location tracking of mobiles in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 1558–1562, Sept. 1999.
- [9] W. Kim, G.-I. Jee, and J. G. Lee, "Wireless location with NLOS error mitigation in Korean CDMA system," *3G Mobile Commun. Technol.*, pp. 134–138, Mar. 2001.
- [10] V. Enescu and H. Sahli, "Recursive filtering approach to MS locating using quantized to a measurements," *3G Mobile Commun. Technol.*, pp. 206–210, Mar. 2001.
- [11] E. Dahlman, B. Gudmundson, M. Nilsson, and J. Sköld, "UMTS/IMT-2000 based on wideband CDMA," *IEEE Commun. Mag.*, vol. 36, pp. 70–80, Sept. 1998.
- [12] H. S. H. Gombachika and O. K. Tonguz, "Influence of multipath fading and mobile unit velocity on the performance of PN tracking in CDMA systems," in *IEEE Vehicular Technology Conf.*, May 1997, pp. 2206–2209.
- [13] R. D. J. Van Nee, "Spread-spectrum code and carrier synchronization errors caused by multipath and interference," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, pp. 1359–1365, Oct. 1993.
- [14] M. D. Yacoub, *Foundations of Mobile Radio Engineering*. Boca Raton, FL: CRC Press, 1993.
- [15] J. H. Reed, K. J. Krizman, B. D. Woerner, and T. S. Rappaport, "An overview of the challenges and progress in meeting the E-911 requirement for location service," *IEEE Commun. Mag.*, vol. 36, pp. 30–37, Apr. 1998.
- [16] P. Bernadin, M. F. Yee, and T. Ellis, "Cell radius inaccuracy: a new measure of coverage reliability," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 1215–1226, Nov. 1998.
- [17] M. McGuire, K. N. Plataniotis, and A. N. Venetsanopoulos, "Estimating position of mobile terminals from delay measurements with survey data," in *Can. Conf. Electrical and Computer Engineering*, Toronto, ON, Canada, May 2001, pp. 129–134.
- [18] —, "Location of mobile terminals using time measurements and survey points," *IEEE Trans. Veh. Technol.*, to be published.
- [19] G. A. Babich and O. I. Camps, "Weighted Parzen windows for pattern classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 567–570, May 1996.
- [20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Toronto, ON, Canada: Wiley, 2001.
- [21] M. A. Spirito, "On the accuracy of cellular mobile station location estimation," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 674–685, May 2001.
- [22] S. Takezono, H. Minamoto, and K. Tao, "Two-dimensional motion of four-wheel vehicles," *Vehicle Syst. Dyn.*, vol. 32, pp. 441–458, 1999.
- [23] C.-F. Lin, A. G. Ulsoy, and D. J. LeBlanc, "Vehicle dynamics and external disturbance estimation for vehicle path prediction," *IEEE Trans. Contr. Syst. Technol.*, vol. 8, pp. 508–518, May 2000.
- [24] E. Aleman-Llanes, D. Munoz-Rodriguez, and C. Molina, "PCS subscribers mobility modeling using fractional Brownian motion (FBM)," *Eur. Trans. Commun.*, vol. 11, no. 2, pp. 191–198, Mar./Apr. 2000.
- [25] *Traffic Engineering Handbook*, 5th ed., J. L. Pline, Ed., Institution of Transportation Engineers, Washington, D.C., 1999.
- [26] C. F. Daganzo, *Fundamentals of Transportation and Traffic Operations*. London, U.K.: Elsevier Science, 1997.
- [27] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. Don Mills, ON, Canada: Addison-Wesley, 1994.
- [28] B. Jabbari, Y. Zhou, and F. Hillier, "Simple random walk models for wireless terminal movements," in *Vehicular Technology Conf.*, Houston, Texas, May 1999, pp. 1784–1788.
- [29] K.-S. Kim, M.-H. Cho, and K.-R. Cho, "A simple analytic approach for the cell sojourn time in the Gaussian distributed mobile velocity," *IEICE Trans. Commun.*, vol. E83-B, no. 5, pp. 1148–1151, May 2000.
- [30] C. Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," *ACM Int. Workshop Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, pp. 19–27, July 2001.
- [31] H. Xie and D. J. Goodman, "Mobility models and biased sampling problem," in *Int. Conf. Universal Personal Communications*, Ottawa, ON, Canada, Oct. 1999, pp. 803–807.
- [32] M. M. Zonoozi and P. Dassanayake, "A novel method for tracing mobile users in a cellular mobile communication system," *Wireless Personal Commun.*, vol. 4, pp. 185–205, Mar. 1997.
- [33] ETSI, "Universal Mobile Telecommunications Systems (UMTS): Selection procedures for the choice of radio transmission technologies of the UMTS (umts 30.02 Version 3.2.0)," European Telecommunications Standards Institute, 1998.
- [34] E. Brookner, *Tracking and Kalman Filtering Made Easy*. Toronto, ON, Canada: Wiley, 1998.
- [35] A. Gelb, Ed., *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
- [36] T. A. Magill, "Optimal adaptive estimation of sampled stochastic processes," Stanford Electronic Laboratories, Tech. Rep. 6302-3, 1963.
- [37] R. L. Moose, H. F. VanLandingham, and D. H. McCabe, "Modeling and estimation for tracking maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, pp. 448–456, May 1979.
- [38] J. V. Candy, *Signal Processing: The Model-Based Approach*. New York: McGraw-Hill, 1986.
- [39] K. Ogata, *Discrete-Time Control Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1995.



**Michael McGuire** (S'96–M'97) received the B.Eng. degree in computer engineering and the M.A.Sc. degree from the University of Victoria, Victoria, BC, Canada, in 1995 and 1997, respectively. He has recently received the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.

His research interests are estimation and control algorithms for wireless cellular networks.



**Konstantinos N. Plataniotis** (S'90–M'92) received the B.Eng. degree in computer engineering from the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece, in 1988 and the M.S. and Ph.D. degrees in electrical engineering from the Florida Institute of Technology, Melbourne, in 1992 and 1994, respectively.

He was a Research Associate with the Computer Technology Institute, Patras, from 1989 to 1991. He was a Postdoctoral Fellow at the Digital Signal and Image Processing Laboratory, Department of Electrical and Computer Engineering, University of Toronto, from 1995 to 1997. From 1997 to 1999, he was an Assistant Professor with the School of Computer Science, Ryerson Polytechnic University. While there, he was a Lecturer in 12 courses for industry and continuing education programs. Since 1999, he has been with the University of Toronto as an Assistant Professor in the Department of Electrical and Computer Engineering, where he researches and teaches adaptive systems and multimedia signal processing. He is coauthor (with A. N. Venetsanopoulos) of *Color Image Processing and Applications* (Berlin, Germany: Springer Verlag, 2000). He is a contributor to three books and has published more than 100 papers in refereed journals and conference proceedings on the areas of adaptive systems, signal and image processing, and communication systems and stochastic estimation. His current research interests include adaptive systems statistical pattern recognition, multimedia data processing, statistical communication systems, and stochastic estimation and control. He was Technical Cochair of the Canadian Conference on Electrical and Computer Engineering (CCECE 2001, May 13–16, 2001, Toronto).

Dr. Plataniotis is a member of the IEEE Technical Committee on Neural Networks for Signal Processing.