Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks

Tong Liu, Paramvir Bahl, *Senior Member, IEEE,* and Imrich Chlamtac, *Fellow, IEEE*

***Abstract*— Wireless ATM networks require efficient mobility management to cope with frequent mobile handoff and rerouting of connections. Although much attention has been given in the literature to network architecture design to support wide-area mobility in public ATM networks, little has been done to the important issue of user mobility estimation and prediction to improve the connection reliability and bandwidth efficiency of the underlying system architecture. This paper treats the problem by developing a hierarchical user mobility model that closely represents the movement behavior of a mobile user, and that, when used with appropriate pattern matching and Kalman filtering techniques, yields an accurate location prediction algorithm, HLP, or *hierarchical location prediction,* which provides necessary information for advance resource reservation and advance optimal route establishment in wireless ATM networks.**

***Index Terms*—Location tracking, mobility modeling, trajectory prediction, wireless ATM networks.**

I. Introduction

THE extension of broadband ATM networks to the wireless domain presents a challenging set of problems for both network designers and managers. One of the key problems within this set is mobility management. Mobility management as defined in [1] entails both *connection management* and *location management*. Since ATM is a connection-oriented technology, it contains both a connection establishment phase prior to data exchange and a connection release phase after data exchange. In a wireless network, as terminals move, segments of connections have to be torn down and reestablished with a frequency that corresponds to the speed of the mobile. Meanwhile, data integrity in terms of cell sequence preservation, duplicate cell prevention, and cell loss avoidance has to be provided. Additionally, quality-of-service (QoS) guarantees have to be maintained regardless of the terminal's mobility.

At the most fundamental level, quality of service, a cornerstone of ATM networks, can only be provided if the system is able to maintain connectivity with the mobile terminal, even when the terminal frequently changes its physical location. It is possible to maintain connectivity and guarantee QoS to the mobile if the system knows, prior to the mobile's movement, the exact trajectory it will follow. With this information, the system can determine if there are enough resources available along the mobile's path for the lifetime of the connection. If such is the case, the system can plan in anticipation of the mobile's demands, and take appropriate steps such as setting up end-to-end routes from base stations in the mobile's path, reserving resource along these routes, and planning quick handoffs between the involved base stations. With these kinds of preparations, QoS can be guaranteed.

Recently, there has been some work on providing QoS to mobile terminals. In [2], Acampora and Naghshineh propose a *virtual connection tree* (VCT) scheme in which multiple connections are preestablished between a fixed (*root*) switch and a set of base stations with whom the mobile could potentially connect. While providing a good starting point, the VCT algorithm in its native form suffers from the lack of inclusion of accurate knowledge of the mobile's trajectory. There is a danger of underutilizing resources in base stations to which the mobile never connects, and a potential of overloading base stations when a large number of mobiles connect at the same time. Overloading can lead to congestion, which can result in ATM cells being either dropped or buffered or both. Buffering can cause temporary violation of delay and cell loss guarantees. As a possible solution to these kinds of problems, Levine *et al.* [3] have proposed the concept of a *shadow cluster*. A shadow cluster defines the area of influence of a mobile terminal (i.e., a set of base stations to which the mobile terminal is likely to attach in the near future). Like a shadow, this set moves along with the mobile, incorporating new base stations while leaving the old ones as they come under and out of the mobile's influence. Each base station in the shadow cluster anticipates the mobile's arrival and reserves resources for it. A close association exists between the mobile's arrival prediction and reservation of resources for it. The accuracy of the mobile's path prediction determines the number of base stations that reserve resources, and consequently determines the overall system efficiency.

Location management or location tracking incorporates the set of mechanisms with which the system can locate a particular mobile at any given time. Two strategies are possible: location updating and location prediction. Location updating is a passive strategy in which the system periodically records the current location of the mobile in some database that it maintains. Tracking efficiency is based on the frequency of these updates which, in most systems, are initiated by the mobile. Location prediction is a dynamic strategy in which the system proactively estimates the mobile's location based on a user movement model. Tracking capability depends on

Manuscript received September 26, 1997; revised March 12, 1998. This work was supported in part by the Army Research Office under Contract DAAH04-96-1-0308.

T. Liu is with Ascend Communications Inc., Westford, MA 01886 USA.

P. Bahl is with Microsoft Research, Redmond, WA 98052 USA.

I. Chlamtac is with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX 75083-0688 USA.

Publisher Item Identifier S 0733-8716(98)04800-8.

the accuracy of this model and the efficiency of the prediction algorithm. While most recent studies have focused on the update method [4]–[7], relatively little has been done on the prediction side [21]. As a consequence, location management or tracking is generally perceived as a purely database updating and querying procedure. If accurate movement prediction per mobile was possible, the task of locating mobiles given their last location would become substantially efficient in terms of both speed and system resources used.

Thus, to sum it up, the above discussions on connection and location management make a compelling case for placing emphasis on developing algorithms and techniques for mobile trajectory prediction. Being able to determine the mobile's future locations and access points as it moves inside the network while being connected can result in significant improvement in system efficiency and connection quality.

One way for the system to know the future direction of the mobile is to have a formal mechanism in place that allows the mobile to indicate to the system its intended destination and the duration of the connection. The system can combine this information with its knowledge of the geography of the terrain, and the location of the base stations within the terrain, to determine the path of the mobile. Unfortunately, this is not a conclusive solution since there can be multiple paths to the destination, and even the mobile cannot exactly know its precise future movement patterns. It is not unreasonable to have the mobile diverge without warning from its stated path in order to adjust to its dynamically changing environment. Without the system dynamically adapting to such unanticipated but reasonable trajectory changes, the amount of resources required to provide improved connectivity would be prohibitive, and consequently unattractive.

In this paper, we propose a novel *hierarchical location-prediction* (HLP) algorithm, which substantially increases the system's probability of providing uninterrupted service to the mobile user while consuming minimal resources from the network. Derived from some classical and well-established stochastic signal processing techniques, HLP is a two-tier scheme that combines location updating with location prediction to offer enhanced connection management functions. HLP raises the level of intelligence within the wireless ATM system so that the system aggressively and effectively maintains connectivity (essential for providing QoS features) with the mobile.

Some previous works in the area of mobility prediction includes Tabbane's [8] proposal, which suggests that the mobile's location may be determined based on its quasi-deterministic mobility behavior represented as a set of movement patterns stored in a user profile. This method was further pursued by Liu and Maquire [9], in which a user's moving behavior is modeled as repetitions of some elementary movement patterns. Based on these movement patterns, a pattern matching/recognition-based *mobile motion prediction* algorithm (MMP) is proposed which is used to estimate the future location of the mobile. The main drawback of the MMP algorithm is its high sensitivity to so called "random movements." Any movement that cannot be classified by the simple mobility patterns defined is classified as random movement. As reported in [9], prediction performance of MMP decreases linearly with the increase in the random factor γ. Other methods for predicting speed and trajectory have also been proposed in the literature [10]–[13], but these have generally been limited in scope as they consider rectilinear ("highway") movement patterns only.

In order to develop a prediction algorithm for mobiles with different mobility characteristics, we require a movement model that allows us to explore regularity and rationality in the seemingly random movement. We avoid using mobility models that incorporate deterministic mobility [13]–[15], but instead propose a novel pseudostochastic movement model which integrates deterministic behavior with randomness in an attempt to mimic actual human behavior.

To achieve this, we model the user's quasi-deterministic intercell movement by editing her deterministic movement patterns with *insertion*, *deletion*, and *changing* operations. Additionally, we model the mobile's micromovement as a nonstationary process with dynamic states that are nonlinearly related to a time-correlated Gaussian process whose mean value behaves as a semi-Markov process. Based on this model, we develop a set of low-complexity recursive mobility prediction algorithms that are trivial to implement in real-time systems. HLP is composed of an *approximate pattern-matching* algorithm that extracts any existing regular movement pattern to estimate the global intercell direction, and uses an *extended self-learning Kalman filter* that deals with "unclassifiable" random movements by tracking intracell trajectory and predicting the next-cell crossing. We study the performance of our prediction algorithms in the presence of path loss, shadow fading, and random user movements. Simulation results and performance analysis show that our algorithm is robust in the presence of noisy input, being able to predict the speed and direction of travel of the mobile with a high degree of accuracy. With good next-cell prediction, algorithms that improve handoffs, relieve congestion, provide advance resource reservations and advance optimal route establishment, and which improve the overall QoS in wireless ATM networks can be built easily.

The rest of the paper is organized as follows. In Section II, we describe the proposed stochastic movement model. In Section III, we introduce the *hierarchical location-prediction* (HLP) algorithm. In Sections IV and V, we present our simulation results, and provide prediction performance analysis. In Section VI, we present potential ways of incorporating HLP for connection and location management in wireless ATM systems. Finally, we conclude with some discussions in Section VII.

II. User Mobility Model

The mobility model we advocate in this paper attempts to mimic human (operator) movement behavior. Our model is built as a two-level hierarchy in which the top level is the *global mobility model* or GMM whose resolution is in terms of cells crossed by the mobile during the lifetime of the connection, and the bottom level is the *local mobility model* (LMM), whose resolution is in terms of a 3-tuple sample

space (speed, direction, position) that varies with time. Stated another way, GMM is a deterministic model that is used to create intercell movements, while LMM is a stochastic model with dynamically changing state variables to model intracell movement. The LMM model interacts with the GMM model to create a semirandom movement trajectory for the mobile. Our complete model (LMM + GMM) is based on the observation that the directional movement of mobiles is generally not ad hoc. The variations (or perturbation) from a particular path occur when the mobile dynamically adjusts trajectory to compensate for the changing environment (e.g., changing traffic conditions, unexpected roadblocks, weather conditions, etc.). These perturbations are recreated in our model by the intracell stochastic behavior which affects the intercell movement pattern with varying degrees of influence. Details follow.

A. Global Mobility Model

The global mobility model, as shown in Fig. 1(a), is motivated by the fact that most mobile users exhibit some regularity in their daily movement, and this regularity can best be characterized by a number of *user mobility patterns* (UMP's), recorded in a *profile* for each user and indexed by the occurrence time. The UMP's we proposed are similar to the movement patterns in [8] and [9], but are more robust in the sense that we decrease the UMP's sensitivity to small deviations from the *user's actual path* (UAP). We are able to do this while maintaining their effectiveness for estimating the mobile's intercell directional movement intention by modeling UAP as the edited version of a UMP, and employing an *approximate pattern-matching* technique to find the UMP that most resembles UAP as explained in Section III. Consequently, the number of UMP's needed to span the network is greatly reduced, which in turn substantially reduces the time needed for pattern classification. Specifically, if a UMP is described by a cell (a_i) sequence ($= a_1 a_2 \cdots a_{i-1} a_i a_{i+1} \cdots a_n$), then we model the regular movement of a mobile user as an *edited* UMP by allowing the following legal operations:

- *inserting* a cell c at position i of the UMP gives UAP: $a_1 a_2 \cdots a_{i-1} c a_i \cdots a_n$,
- *deleting* the cell a_i at position i of the UMP gives UAP: $a_1 a_. \cdots a_{i-1} a_{i+1} \cdots a_n$,
- *changing* a cell a_i to another cell c gives UAP: $a_1 a_2 \cdots a_{i-1} c a_{i+1} \cdots a_n$.

Fig. 1(b) demonstrates an example of a UMP $c_3 c_0 c_6 c_7$, and its edited version UAP $c_3 c_4 c_5 c_6 c_7$, which can be obtained by *changing* c_0 to c_4 and inserting c_5.

The degree of resemblance of a UAP with a UMP is measured by the *edit distance*, a well-known metric for finite string comparison [16]. The simplest way to find this distance is by determining the smallest number of *insertions, deletions*, and *changes* by which the two finite cell sequences can be made alike. In order to reflect the geometric relationship between UAP and UMP in the physical domain, we assign a nonnegative number to each edit operation as the spatial weight. As a result, the *edit distance* between a UAP and a UMP becomes the sum of the weights of the *editing* operations which are chosen not only to make the two cell sequences alike, but also to have the smallest possible total weight. If the *edit distance* is less than a matching threshold t, an approximately matched UMP is found, indicating the general moving intention of the user.

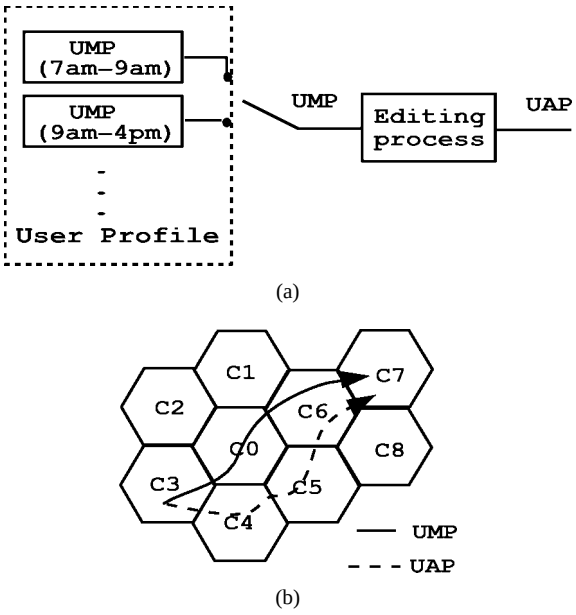


Fig. 1. Global mobility model.

For large systems or systems with complex network topologies, the calculation of the spatial weights can be quite involved. In this work, for the sake of clarity and without losing generality, we limit the temporary deviation of a mobile user within the neighboring cells of the edited UMP. Specifically, we define the weight as follows.

- The cost of *inserting* a cell c at position i:

$$W_{Ici} = \begin{cases} 1, & c \text{ is the adjacent cell of } a_i \\ & \text{for } i = 1, \cdots, n \\ \infty, & \text{otherwise.} \end{cases}$$

- The cost of *deleting* the cell a_i:

$$W_{Dai} = \begin{cases} 0, & a_1, \cdots, a_{i-1} \text{ have already} \\ & \text{been deleted} \\ 1, & \text{otherwise.} \end{cases}$$

- The cost of *changing* a cell to another cell c at position i:

$$W_{Cci} = \begin{cases} 1, & c \text{ is the adjacent cell of } a_i \\ & \text{for } i = 1, \cdots, n \\ \infty, & \text{otherwise.} \end{cases}$$

B. Local Mobility Model

Our motivation behind creating a local mobility model is based on the observation that the seemingly random choice of intercell movement is actually a logical function of the user's position, speed, direction, and cell geometry. The user mobility models found in the literature assume straight line movement or constant speed [10]–[13], which does not reflect reality.

In order to develop time-varying movement patterns, we model a moving user as a dynamic linear system driven by

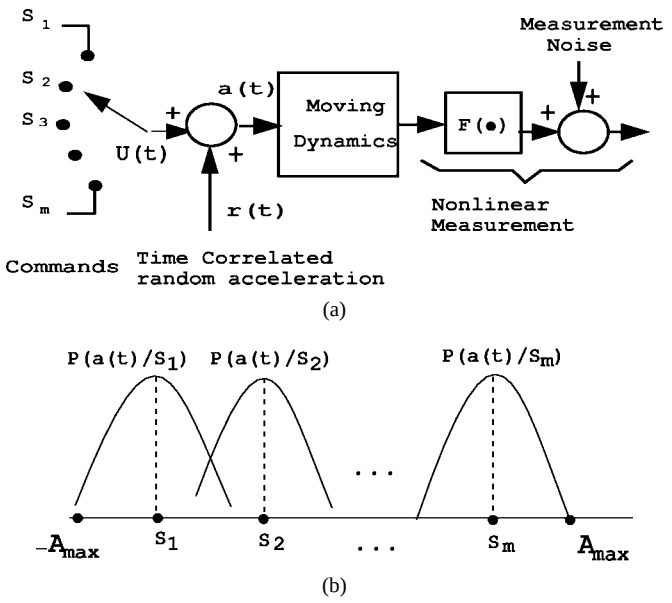


Fig. 2. Local mobility model.

deterministic command $U(t)$ and random acceleration $r(t)$ shown in Fig. 2(a). In real situations, the acceleration range for a moving user can be fairly wide. Furthermore, traffic lights and road turns can lead to abrupt changes in speed. In order to recreate such sudden and unexpected changes, while covering the wide acceleration range, $U(t)$ is modeled as a semi-Markov process with a finite number of "states" $S_1, S_2, \cdots, S_m$ as possible discrete levels of acceleration. A semi-Markov process implies the Markovian state transition probability and random duration of time in one state prior to switching to another state [22]. Random acceleration $r(t)$ is modeled as a zero-mean Gaussian random variable with a variance that is chosen to cover the "gap" between adjacent acceleration states. Fig. 2(b) shows the conditional distribution of $r(t)$ given a particular state S_i, for $i = 1, 2, \cdots, m$, within the acceleration range $[-A_{\max}, A_{\max}]$. This modeling idea has successfully been applied in tactical weapon systems for maneuvering target tracking by Singer [17] and Moose [18].

The following two subsections present the mathematical description of the local mobility model in terms of dynamical equations and measurement equations.

1) Dynamical Equations for a Moving User: Based on the model described above, the dynamical equations are derived for continuous-time movement, and are then expressed in discrete time according to the standard discretization procedure, thereby providing accurate statistical representation of the movement behavior.

It is known that in two-dimensional Cartesian coordinates, the movement can be described by a first-order vector differential equation [23] with the dynamic state vector $\boldsymbol{x}(t) = [x(t), \dot{x}(t), y(t), \dot{y}(t)]^T$, where $x(t)$ and $y(t)$ represent the position at time t, and their first-order derivatives of $\dot{x}(t)$ and $\dot{y}(t)$ represent the relative speed along the x and y directions. Furthermore, let $\boldsymbol{u}(t) = [u_x(t), u_y(t)]^T$ denote the two-dimensional driving command with $u_x(t)$ and $u_y(t)$ as independent semi-Markovian processes acting in the x and y directions, respectively, and let $\boldsymbol{r}(t) = [r_x(t), r_y(t)]^T$ denote the two-dimensional random acceleration vector; we have

$$\dot{\boldsymbol{x}}(t) = F\boldsymbol{x}(t) + E\boldsymbol{u}(t) + G\boldsymbol{r}(t), \qquad i = 1, \cdots, m \tag{1}$$

where

$$F = \begin{bmatrix} \Theta & 0 \\ 0 & \Theta \end{bmatrix} \quad E = G = \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}$$

with

$$\Theta = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \Phi = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Random acceleration $\boldsymbol{r}(t)$ is correlated in time, i.e., if a moving object is accelerating at time t, it is likely to continue accelerating at time $t + \tau$ for sufficiently small τ. A typical representative model of the correlation function is [17]

$$R_{\boldsymbol{r}}(\tau) = E[\boldsymbol{r}(t)\boldsymbol{r}(t+\tau)] = \sigma_m^2 e^{-\alpha|\tau|} I, \qquad \alpha \geq 0 \tag{2}$$

where I is a 2×2 identity matrix, σ_m^2 is the variance of the random acceleration of a single dimension, and α is the reciprocal of the random acceleration time constant. Such a random process can be obtained by passing the white Gaussian signals $\boldsymbol{w}(t) = [w_x(t), w_y(t)]^T$ through a one-pole shaping filter, where $w_x(t)$ and $w_y(t)$ are uncorrelated

$$\dot{\boldsymbol{r}}(t) = -\alpha\boldsymbol{r}(t) + \boldsymbol{w}(t), \qquad \text{with} \quad R_{\boldsymbol{w}}^2 = 2\alpha\sigma_m^2\delta(\tau)I. \tag{3}$$

Combining (1) and (3), and applying the state-space method, the discrete-time dynamic equation can be expressed as the following in terms of discrete time Gaussian white noise W_n and driving command U_n (for the derivation, see part A of the Appendix):

$$X_{n+1} = AX_n + BU_n + W_n \tag{4}$$

where

$$X_n = [x(n) \quad \dot{x}(n) \quad r_x(n) \quad y(n) \quad \dot{y}(n) \quad r_y(n)]^T$$
$$U_n = \begin{bmatrix} u_x(n) \\ u_y(n) \end{bmatrix} \quad W_n = \begin{bmatrix} w_x(n) \\ w_y(n) \end{bmatrix}.$$

The dimension of the dynamic state vector X_n is now extended to three states per Cartesian axis which represent position, velocity, and acceleration, respectively. A and B are the state and disturbance transition matrices relating the system at times $n+1$ and n.

2) Measurement Equations: In existing cellular systems, the distance between the mobile and a known base station is practically observable. Such information is inherent in the forward link RSSI (or received signal strength indication) of a reachable base station. Measured in decibels at the mobile station, RSSI can be modeled as the sum of two terms: one due to path loss, and another due to shadow fading. Fast fading is neglected assuming that a low-pass filter is used to attenuate Rayleigh or Rician fade. Therefore, the RSSI from a particular *cell*$_i$, p_i, can be formulated as [20]

$$p_i = p_{oi} - 10r \log d_i + \xi_i \tag{5}$$

where p_{oi} is a constant determined by transmitted power, wavelength, and antenna gain of *cell*$_i$. r is a slope index (typically, $r = 2$ for highways and $r = 4$ for microcells in a city),

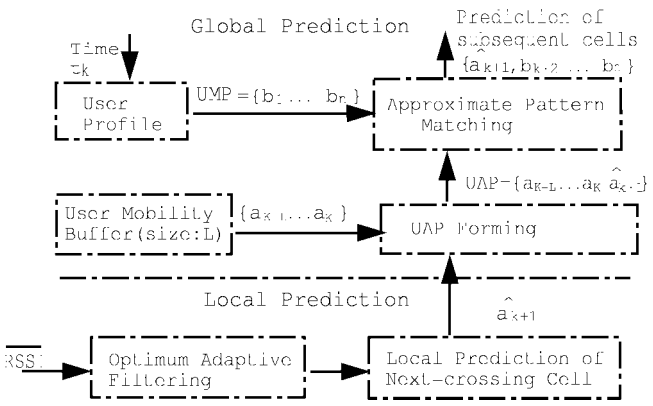


Fig. 3. Mobility prediction.

and ξ_i is the logarithm of the shadowing component, which is found to be a zero-mean Gaussian random variable with standard deviation 4–8 dB. d_i represents the distance between the mobile and base station of *cell*$_i$, which can be further expressed in terms of the mobile's position $(x(n), y(n))$ at time n and the location of base station (a_i, b_i)

$$d_i = \left[(x(n) - a_i)^2 + (y(n) - b_i)^2\right]^{1/2}. \tag{6}$$

To locate a moving user in the two-dimensional domain, at least three independent distance measurements are needed [23]. The necessary data are available in GSM systems where each 0.48s, the mobile samples the forward link signal levels of six neighboring cells [6]. For this problem, we select the three largest measurements to form the observation vector $Z_n = [p_1, p_2, p_3]^T$, which is nonlinearly related to dynamic states X_n:

$$Z_n = h(X_n) + \xi_n \tag{7}$$

where $h(X_n)$ can be derived from (5) and (6).

Applying the linearization method used in *extended Kalman filer* design [23], where the linearization takes place about the filter's estimated trajectory, the linearized measurement equation becomes

$$Z_n = HX_n + \xi_n, \qquad \text{with} \quad H = \left.\frac{\partial h}{\partial X}\right|_{X=\hat{X}_n} \tag{8}$$

where $\hat{X}_n$ is the optimal estimate of X_n.

III. The Hierarchical Location Prediction Algorithm

The algorithm proposed for mobility prediction is illustrated in Fig. 3. As shown, user mobility prediction is carried out at two levels—*local prediction* (LP) and *global prediction* (GP). LP provides a best estimate of the next cell to be crossed based on instantaneous trajectory tracking and cell geometry, while GP identifies the overall movement pattern of the mobile. With LP, a high degree of accuracy for next-cell prediction is achieved without any assumption of the user's mobility history. Furthermore, the UMP identification error is greatly reduced since GP uses the prediction data from LP to look ahead before making a decision on the best matched pattern. Details follow.

A. High-Level Global Prediction

For global prediction, we solve the following problem: given a number of UMP's for the current time interval, a UAP which reflects the current movement trend of the mobile, find the UMP that best describes the UAP.

A UAP is composed of recently crossed cells, stored in the mobile's user mobility buffer, and an LP for the next cell (if any). Both UAP and UMP are finite-length sequences, and we assume that the memory length of a UAP does not exceed the length of the largest stored UMP. Consequently, it is possible for some subset of the UMP to resemble the UAP in the sense that the edit distance is the minimum and less than the matching threshold t. If such an approximately matched sequence is found, the remaining sequence in UMP becomes the output of the global prediction.

The solution to this problem can be reduced to the following dynamic programming method [16]: suppose that the UAP $= a_1, \cdots, a_m$, and the UMP $= b_1, \cdots, b_n$; we can compute the edit distance $d(a_1, \cdots, a_m, b_1, \cdots, b_n)$ by constructing an $(m+1) \times (n+1)$ matrix $[d(a_1, \cdots, a_i, b_1, \cdots, b_j)]$ with (i, j) ranging from (0, 0) to (m, n), as follows: if $i = 0$ or $j = 0$, which means $a_1 \cdots a_i = 1$ or $b_1 \cdots b_j = 1$, where 1 represents the empty sequence, then the matrix values are

$$\begin{aligned} d(1, 1) &= 0 \\ d(a_1 \cdots a_i, 1) &= \begin{cases} 1, & i = 1 \\ \infty, & 1 < i \le m \end{cases} \\ d(1, b_1 \cdots b_j) &= 0 \end{aligned}$$

$$\begin{aligned} &d(a_1 \cdots a_i, b_1 \cdots b_j) \\ &\quad = \begin{cases} d(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1}), \\ \qquad \text{if } a_i = b_i \\ \min\{d(a_1 \cdots a_{i-1}, b_1 \cdots b_{j-1}) + W_{Ca_ib_j} \\ \quad d(a_1 \cdots a_{i-1}, b_1 \cdots b_j) + W_{Ia_ii} \\ \quad d(a_1 \cdots a_i, b_1 \cdots b_{j-1}) + W_{Db_j}\}, \\ \qquad \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

As explained in Section II-A, $W_{Ca_ib_j}$ is the spatial weight of changing b_j to a_i, W_{Ia_ii} is the weight of inserting cell a_i at position i, and W_{Db_j} is the weight of deleting b_j. Matrix $(d(a_i \cdots a_i, b_1 \cdots b_j))$ can be evaluated column-by-column in time $O(mn)$; whenever a $d(a_1 \cdots a_m, b_1 \cdots b_j)$ is found that has the property $d(a_1 \cdots a_m, b_i \cdots b_j) \le t$, an approximate occurrence of UAP ending at the jth cell b_j is announced. If a_m is the LP of next cell, the future cell sequence predicted by GP becomes $a_m, b_{j+1}, \cdots, b_n$. An example that clarifies the operation of this procedure is detailed in Section IV.

Additional matching criteria such as the elapsed time matching [9] can further be applied in case multiple UMP subsequences have a similar *edit distance* from UAP.

B. Low-Level Local Prediction

Local prediction can be achieved by two steps:

1) estimate the dynamic state of a moving user using subsequent RSSI measurements,

2) select the neighboring cell with maximum cell-crossing probability as LP output based on dynamic state estimation and cell geometry.

For the first step, an optimum adaptive filter is needed for real-time dynamic state estimation. As discussed previously, since the instantaneous sampled trajectory of the mobile is a nonstationary process, a modified Kalman filter seems to be the best candidate for achieving low-level local prediction. The conventional Kalman filter has to be modified since it is not designed for deterministic input U_n, which we represent as a semi-Markovian process with m possible (mobile acceleration) states. We overcome the problem of dealing with the randomness within the states by using a bank of m filters with each filter operating on a possible state. Fortunately, as discussed in [18], when certain practical assumptions are made, the filter bank can be reduced to a single Kalman filter augmented by a recursive technique of estimating U_n. The adaptive state estimator then becomes (see part B of the Appendix for derivation)

$$\hat{X}_{n+1} = A\hat{X}_n + B\hat{U}_n + K_{n+1}\Big(Z_{n+1} - HA\hat{X}_n - HB\hat{U}_n\Big). \tag{10}$$

Here, K_{n+1} is the standard Kalman gain matrix, and $\hat{U}_n$ is an estimate of U_n. The recursive technique for computing $\hat{U}_n$ has been developed in detail by Moose [18], the final results of which are given by the following equations:

$$\hat{U}_n = \sum_{i=1}^{n} U_n(S_i) P\big(S_{n+1}^i / Z_{n+1}\big) \tag{11}$$

$$P\big(S_{n+1}^i / Z_{n+1}\big) = (\text{const}) f(z_{n+1}/S_{n+1}^i, Z_n) \sum_{\alpha=1}^{m} \cdot \theta_{\alpha i} P(S_n^i / Z_n) \tag{12}$$

where the following are true.

1) Let Z_n denote all of the measurements up to and including time $n+1$, (i.e., $z_1, \cdots, z_n$), and let S_{n+1}^i denote the state S_i at time $n+1$. The probability density function $f(z_{n+1}/S_{n+1}^i, Z_n)$ has a Gaussian distribution with mean $H_{n+1}AX_n(S_i) + H_{n+1}BU_n(S_i)$ and variance $H_{n+1}[AM_{n/n}A^T + Q_n]H_{n+1}^T + R_n$; $M_{n/n}$ is the state estimation matrix; R_n is the measurement error covariance matrix, and Q_n is the Gaussian disturbance covariance matrix.
2) Probability $\theta_{\alpha i} = P(U_n = S_i | U_{n-1} = S_\alpha)$ is obtained from semi-Markov considerations. This parameter can be approximated by a value p near unity for $i = \alpha$ and $(1-p)/(m-1)$ for $i \neq \alpha$ for many tracking situations.
3) The constant (const) is evaluated from $\Sigma_{i=1}^m \; P(S_n^i/Z_n) = 1$.

Based on this result, we are able to complete the adaptive linear optimum (in the sense of minimum mean-square error) filter needed to estimate and predict the dynamic states from the RSSI measurements. The resulting algorithm turns out to be very simple and easily implementable in software.

Prediction:

$$X_{n+1/n} = AX_{n/n} + B\hat{U}_n. \tag{13}$$

Minimum prediction MSE matrix:

$$M_{n+1/n} = AK_{n/n}A^T + Q. \tag{14}$$

Kalman gain matrix:

$$K_{n+1} = M_{n+1/n}H_{n+1}^T\big[R_{n+1} + H_{n+1}M_{n+1/n}H_{n+1}\big]. \tag{15}$$

Correction:

$$\hat{X}_{n+1/n+1} = \hat{X}_{n+1/n} + K_{n+1}\Big[Z_{n+1} - h\Big(\hat{X}_{n+1/n}\Big)\Big]. \tag{16}$$

Deterministic input update:

$$\hat{U}_n = \sum_{i=1}^{n} U_n(S_i) P\big(S_{n+1}^i / Z_{n+1}\big). \tag{17}$$

Minimum MSE matrix update:

$$M_{n+1/n+1} = [I - K_{n+1}H_{n+1}]M_{n+1/n} \tag{18}$$

where

$$H_{n+1} = \left.\frac{\partial h}{\partial X_{n+1}}\right|_{X_{n+1}=\hat{X}_{n+1/n}}. \tag{19}$$

$P(S_{n+1}^i/Z_{n+1})$ for $i = 1, \cdots, m$ can be obtained using the recursive equation (12).

To improve the accuracy of local prediction of the next cell crossing, the second step is carried out when the mobile moves in an area that is closer to the cell boundary, and where the chances of it making a dramatic change in its direction and speed are reduced. This area is called the *correlation area* of the neighboring cell$_i$, denoted as Γ_i, for $i = 1, \cdots, 6$ in a hexagon cell environment, as shown in Fig. 4. The definition is as follows:

$$\Gamma_i = \{X_n | [x(n), y(n)] \epsilon \beta_i,\ d_0 \geq d_t \quad \text{and} \quad \theta \epsilon [\theta_i, \theta_{i+1}]\}$$

where β_i is the 60° sector adjacent to neighboring cell$_i$, d_0 is the distance between the mobile [with coordinate $(x(n), y(n))$] and serving base station [with coordinate (a_0, b_0)], d_t is the distance threshold that determines the confidence in the prediction result, θ is the moving direction, and θ_i is the bearing of cell vertex V_i, $i = 1, \cdots, 6$. Note that since fast-moving users are more likely to follow straight line constant velocity trajectories than the slower ones, d_t can be set dynamically to achieve early prediction for a given prediction confidence.

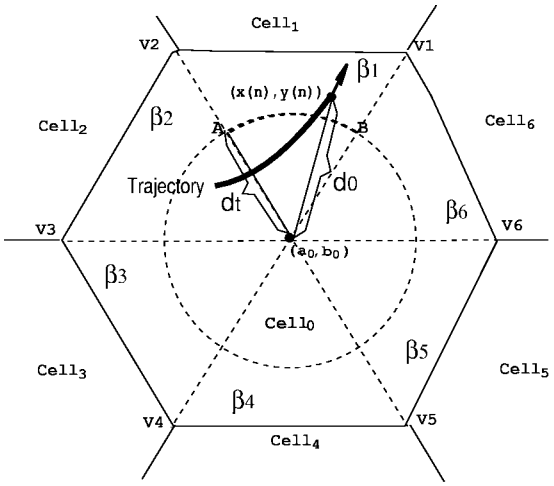
Once the moving user steps into the *correlation area*, the cell-crossing probability can be calculated based on the user's dynamic state, i.e.,

$$P(\text{cell}_i / X_n) = \int_{\theta_i}^{\theta_{i+1}} f(\theta / X_n)\, d\theta, \quad \text{with} \quad i = 1, 2, \cdots, 6 \quad \theta_7 = \theta_1$$

where $f(\theta/X_n)$ is the probability density function of the moving direction θ given the dynamic state X_n.

Applying the *minimum error prediction* criterion, the prediction result of the next-crossing cell is

$$\text{next cell}/X_n = \underset{i}{\text{argmax}}\{P(\text{cell}_i/X_n)\}, \qquad i = 1, \cdots, 6.$$



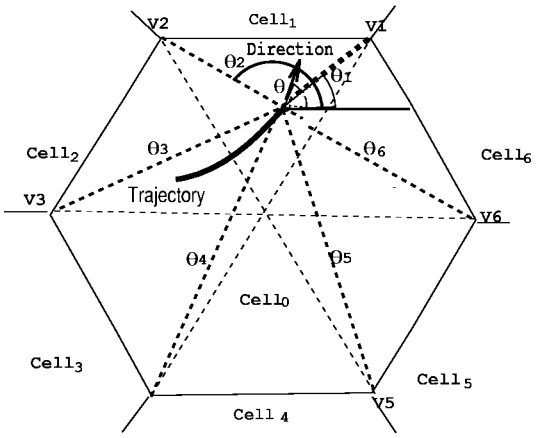


Fig. 4. Cell geometry.

Calculation of $P(\text{cell}_i/X_n)$ is not trivial in general since θ is nonlinearly related to the dynamic state X_n, i.e.,

$$\theta = g(V_n) = \tan^{-1}\frac{\dot{y}(n)}{\dot{x}(n)} \tag{20}$$

where $V_n = [\dot{x}(n), \dot{y}(n)]^T$ is the velocity vector. Since V_n is part of X_n, it also has a Gaussian distribution with mean $\mu_{V[n]}$ and variance Σ_V. If $\mu_{V[n]} \approx 0$, the case when the speed of the mobile station is very slow, $f(\theta/X_n)$ becomes a simple uniform distribution over $[0, 2\pi)$, and

$$P(\text{cell}_i/X_n) = \frac{\theta_{i+1} - \theta_i}{2\pi}, \quad \text{with} \quad i = 1, \cdots, 6, \quad \theta_7 = \theta_1.$$

For more general cases, numerical methods have to be used to calculate $f(\theta/X_n)$. However, if the variance of θ is small, $f(\theta/X_n)$ can be approximated by a Gaussian distribution. We achieve this by assuming that within the correlation area, there is only a small change in velocity, then (20) can be linearized as

$$\theta \approx g(\hat{V}_n) + G(\Delta V) \tag{21}$$

where

$$\begin{aligned} G &= \left.\frac{\partial g}{\partial V}\right|_{V=\hat{V}_n} \\ &= \left[\frac{-\dot{y}(n)}{\dot{x}^2(n)+\dot{y}^2(n)} \quad \frac{\dot{x}(n)}{\dot{x}^2(n)+\dot{y}^2(n)}\right] \\ \Delta V &= V_{n+s} - V_n. \end{aligned} \tag{22}$$

ΔV is the change of velocity between time n and $n+s$ with $s \geq 1$. Since ΔV has a Gaussian distribution with mean $\mu_{\Delta V}$ and covariance $\Sigma_{\Delta V}$, $f(\theta/X_n)$ becomes

$$f(\theta/X_n) \sim N(\mu_\theta, \Sigma_\theta), \qquad \theta \in [\mu_\theta - \pi, \mu_\theta + \pi] \tag{23}$$

with

$$\mu_\theta = g(V^*) + H\mu_{\Delta V}; \qquad \Sigma_\theta = H\Sigma_V H^T.$$

Notice that for small Σ_θ, we assume $f(\theta/X_n) \approx 0$; if $\theta \notin [\mu_\theta - \pi, \mu_\theta + \pi]$, then the cell-crossing probability can be represented by a Q function, i.e.,

$$P(\text{cell}_i/X_n) = Q\left[\frac{\theta_{i+1} - \mu_\theta}{\Sigma_\theta}\right] - Q\left[\frac{\theta_i - \mu_\theta}{\Sigma_\theta}\right], \quad \text{with } i = 1, \cdots, 6, \quad \theta_7 = \theta_1. \tag{24}$$

θ_i can be easily calculated from cell geometry. For example,

$$\theta_1 = \tan^{-1}\left(\frac{R\cos 30° - [y(n) - b_0]}{R/2 - [x(n) - a_0]}\right) \tag{25}$$

where R is the cell radius, $[x(n), y(n)]$ is the position of the mobile station, and (a_0, b_0) is the location of the base station of the current cell.

Note that, although the calculations shown above are for the classical hexagonal cell geometry case, they can easily be carried out for a variety of other shapes as well, which may be closer to reality than the idealized coverage span considered in our calculations.

The significance of trajectory tracking-based local prediction is that a kind of looking-ahead mode can be enabled in UMP identification. In this mode, the decision about the matched UMP is postponed until we look ahead at the prediction of a subsequent cell based on trajectory tracking. As a result, UMP identification error can be greatly reduced, especially in the cell where there are multiple possible leaving UMP's and the crossed cell sequence is not enough to identify the current UMP, for example, as shown in Fig. 5.

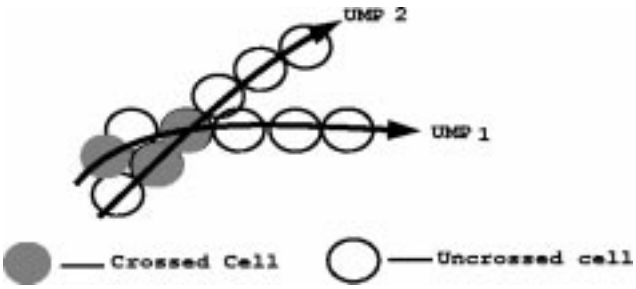


Fig. 5. A practical situation necessitates looking-ahead mode for UMP identification.

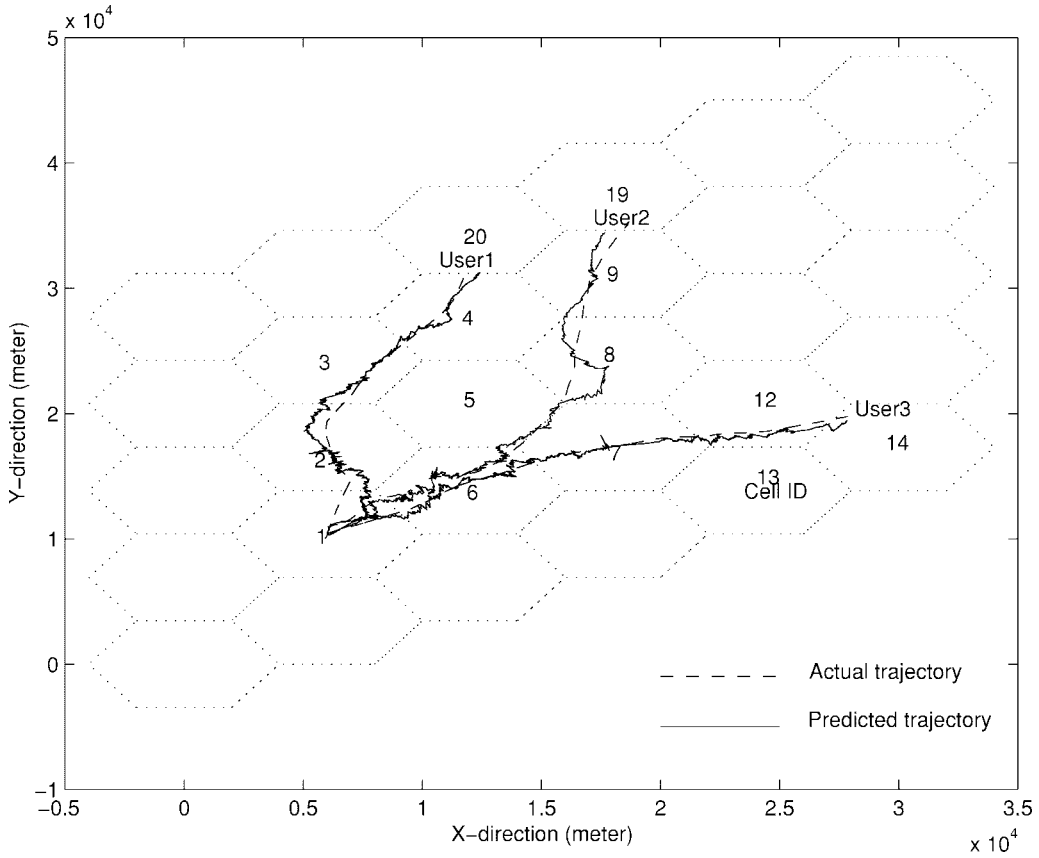


Fig. 6. Actual and predicted user trajectories.

IV. Simulation and Results

To examine the performance of HLP, a simple simulation is carried out for the conventional hexagon cell environment. The simulated service area contains 30 base stations with cell radius of 4 km, as shown in Figs. 6 and 8. No constitutional constraints are assumed, and the mobile is permitted to move to any cell in the network along random trajectories with nonconstant speed. The moving dynamics are based on the movement model discussed in Section II, except that the acceleration is preset so that a known trajectory can be achieved for the purpose of performance tests. On-line mobility-related information includes recently crossed cells stored in a user mobility buffer and subsequent forward link RSSI measurements of the current cell and two neighboring cells. Simulation parameters are summarized in the Table I. In order to cover the range of dynamic acceleration $[-10$ m/s^2, 10 m/s$^2]$, five levels (0, ± 2.5, ± 7.5) m/s^2 are selected as the states of the deterministic driving input.

TABLE I
Simulation Parameters

Parameters	Comments
$T = 0.5s$	Sampling interval
$\sigma_m^2 = 0.5m/s^2$	Variance of random acceleration
$\lvert A_{\max}\rvert = 10m^2/s$	Maximum acceleration
$V \in [30, 60]$ miles/hr	Speed range
$1/\alpha = 10s$	Random acceleration constant
$\sigma_\xi = 5dB$	Standard deviation of lognormal shadowing
$p_0 = 20w$	Base station transmission power
$g_b = 6dB$	Power gain of base station
$g_m = 1dB$	Power gain of mobile station
λ	Wavelength of RF signal

A. Results from Local Prediction

Fig. 6 shows the results from trajectory tracking of three moving users. The dashed curve depicts the actual trajectory (UAP), and the solid curve shows the predicted trajectory. Fig. 7 demonstrates the result of time-varying velocity prediction. The initial value of the dynamic state is estimated from the averaged RSSI value with a position error up to 1000 m and a speed error up to 5 m/s. Because of the strong "pull-in" power of the filter, it turns out that the adaptive filter is relatively insensitive to the initial conditions. The result of LP for the next-cell crossing is summarized in Table II with prediction ratios = 75, 80, 100% for users 1, 2, and 3, respectively. Here, the prediction ratio is defined as the ratio of the number of cells correctly predicted to the total number of cells need to be predicted in the path.

On analyzing the prediction results of the next-cell crossing, together with the mobile's trajectory tracking, we find that a high degree of prediction accuracy is achieved once the Kalman filter becomes stable. The error caused by the initial instability is limited to the prediction of the first cell crossed at the beginning of the journey. In the stable state, the prediction accuracy is related to the geometric relation of the mobile's

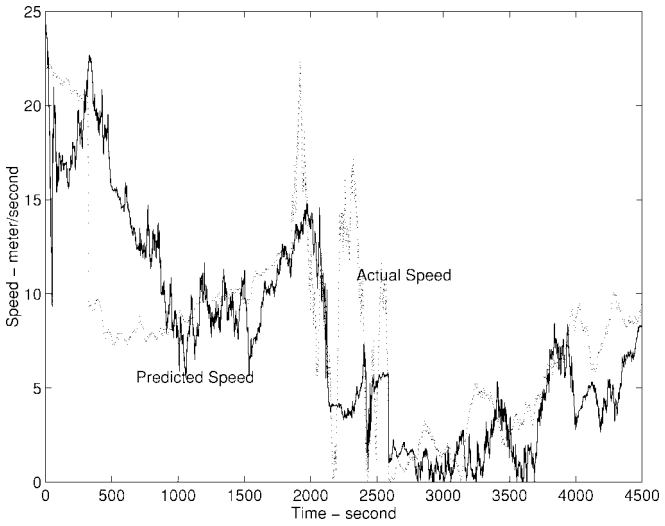


Fig. 7. Actual and predicted user speed.

TABLE II
PREDICTION RESULT OF NEXT-CELL CROSSING

User1	Current Cell	1	2	3	4	20	
	Predicted Cell	6	3	4	20		
User2	Current Cell	1	6	5	8	9	19
	Predicted Cell	6	5	7	9	19	
user3	Current Cell	1	6	7	12	14	
	Predicted Cell	6	7	12	14		

trajectory and cell boundary, which can be compensated by UMP matching.

B. Results from Global Prediction

Fig. 8 shows the results from global prediction, where a moving user is simulated with two possible UMP's, UMP_1 ($c_2c_5c_8c_{12}c_{14}$) and UMP_2 ($c_2c_5c_4c_9c_{19}c_{18}c_{17}c_{16}$). In order to identify the UMP from the tracked UAP, a recursive *approximate pattern-matching* algorithm is implemented with the matching threshold t as 3 and the mobile's user mobility buffer size as 4, which means that the GP output is available when the user reaches the fourth cell c_8. The GP output along the user moving path is given in Table III.

For cell c_8, the *edit distance*, for UMP_1 and UMP_2, is identical (i.e., $d(\text{UAP, UMP}_1) = d(\text{UAP, UMP}_2) = 2$). Since no pattern can be identified, HLP gives the prediction of the next cell crossed based on what is provided by LP. For cell c_9, UMP_2 is identified as the matched pattern, therefore, the GP output becomes the LP output (c_{10}) and the remaining cell sequence on UMP_2 ($c_{18}c_{17}c_{16}$). Fig. 9 shows the matrices used for *edit distance* calculation at cell c_9, which demonstrate the recursive procedure that was described in Section III-A. Line segments in the matrices mark the direct arc in the *dependency graph*, going straight down or down at an angle or to the right. Notice that the path with minimum spatial weight may not be unique, which may result in different predictions. For example, in the matrix for UMP_2, if $\hat{c}_{10}$ is viewed as the result of *insertion*, c_{19} will follow $\hat{c}_{10}$ as part of the global prediction; otherwise, the cell after $\hat{c}_{10}$ should be c_{18} since $\hat{c}_{10}$ is the result

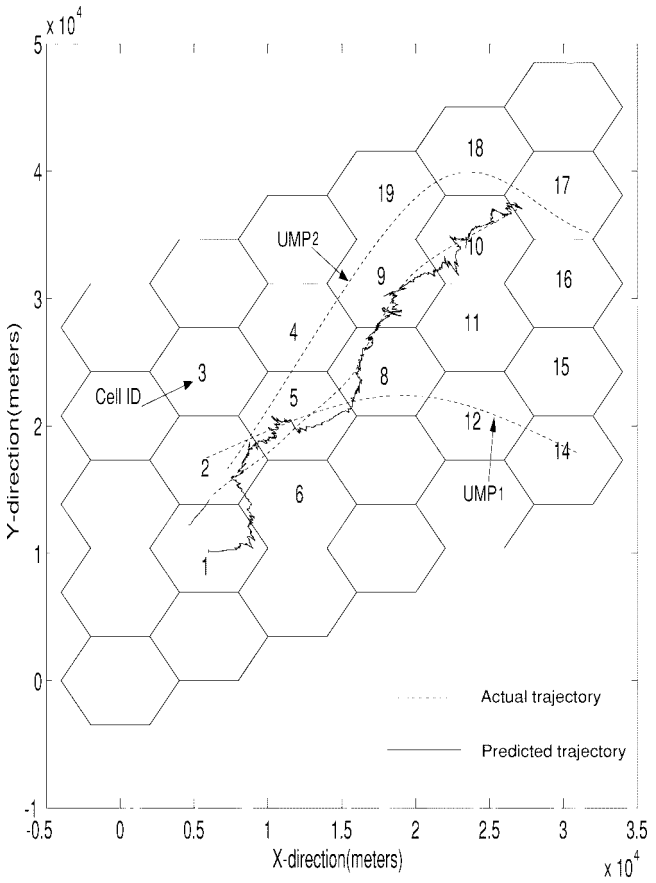


Fig. 8. Actual and predicted user trajectories with multiple potential UMP's.

TABLE III
GLOBAL PREDICTION RESULT

Current Cell	$d(UAP, UMP_1)$	$d(UAP, UMP_2)$	GP output
c_8	2	2	c_9
c_9	∞	2	$c_{10}c_{18}c_{17}c_{16}$
c_{10}	∞	3	$c_{17}c_{16}$

of *changing* c_{19}. In this case, we select the second possibility, which is reasonable since c_{18} is closer to the destination.

It is important to note that at cell c_9, both UMP_1 and UMP_2 have two cells that appear in UAP; if the *state-matching* method as described in [9] is used instead, UMP_2 would not be identified since the UMP_1 and UMP_2 will have the same degree of similarity as UAP.

V. PREDICTION PERFORMANCE

In this section, we discuss the evaluation results of the average prediction performance of the proposed HLP algorithm.

Let P_l represent the probability of a correct local next cell prediction, let $P(c_i)$, $i = 1, \cdots, 6$ denote the *a priori* probability of crossing a neighboring cell_i, and let $P(d_i/c_j)$ represent the conditional probability that we decide cell_i to be the next-crossing cell given that cell_j is the actual cell that the user will cross next. Then

$$P_l = \sum_{i=1}^{6} P(c_i)P(d_i/c_i). \tag{26}$$

UAP \ UMP_1	1	C_2	C_5	C_8	C_{12}	C_{14}
1	0	0	0	0	0	0
C_2	1	0	1	2	3	4
C_5	∞	1	0	1	2	3
C_8	∞	∞	1	0	1	2
C_9	∞	∞	∞	1	2	3
$\hat{C}_{10}$	∞	∞	∞	∞	∞	∞

UAP \ UMP_2	1	C_2	C_5	C_4	C_9	C_{19}	C_{18}	C_{17}	C_{16}
1	0	0	0	0	0	0	0	0	0
C_2	1	0	1	2	3	4	5	6	7
C_5	∞	1	0	1	2	3	4	5	6
C_8	∞	∞	1	1	2	3	4	5	6
C_9	∞	∞	∞	2	1	2	3	4	5
$\hat{C}_{10}$	∞	∞	∞	∞	2	2	3	4	5

Fig. 9. Matrices used for *approximate pattern matching* for cell c_9, with $\hat{c}_{10}$ as the local prediction of the next-crossing cell.

$P(d_i/c_i)$ can be obtained by integrating over set Ω which contains all possible values of dynamic state X

$$\begin{aligned} P(d_i/c_i) &= \int_{\Omega} P(d_i, X/c_i)\, dX \\ &= \int_{\Omega} P(d_i/X, c_i) P(X/c_i)\, dX \\ &= \int_{\Omega} P(d_i/X) P(X/c_i)\, dX. \end{aligned} \tag{27}$$

The last equation is obtained due to the fact that local prediction $P(d_i/X)$ is independent of the mobility history reflected in $P(c_i)$.

Assuming that the next-cell prediction is made when the user enters the *correlation area* Γ_i, and that, $P(X/c_i)$ has uniform distribution over Γ_i, we can further simplify the estimation by only considering a finite number of positions on the boundary of correlation area $\widehat{AB}$, as shown in Fig. 4. Since $\widehat{AB} \in \Gamma_i$, we have

$$\int_{\widehat{AB}} P(d_i, X/c_i) \leq \int_{\Gamma_i} P(d_i, X/c_i) \tag{28}$$

hence, the estimation results over $\widehat{AB}$ can serve as a lower bound. P_l is further calculated for a spectrum of values representing the standard deviation of log-normal shadowing σ_ξ and the residual time when prediction is made for a given acceleration range $[-A_{\max}, A_{\max}]$. Fig. 10 illustrates the parametric behavior of P_l for residual time between 10 and 50 s, and standard deviation of shadowing from 4 to 8 dB with $A_{\max} = 10$ m/s^2. Fig. 10 can be extremely useful for providing a quick, first-cut estimate of the prediction performance for the wireless ATM system.

For GP, if the mobile's movement is classifiable, i.e., a matched UMP has been found, the prediction performance is linearly related to the mobile's random factor γ (γ determines the degree of randomness in choosing the next cell based on a uniform distribution [9]). The probability of a correct GP is

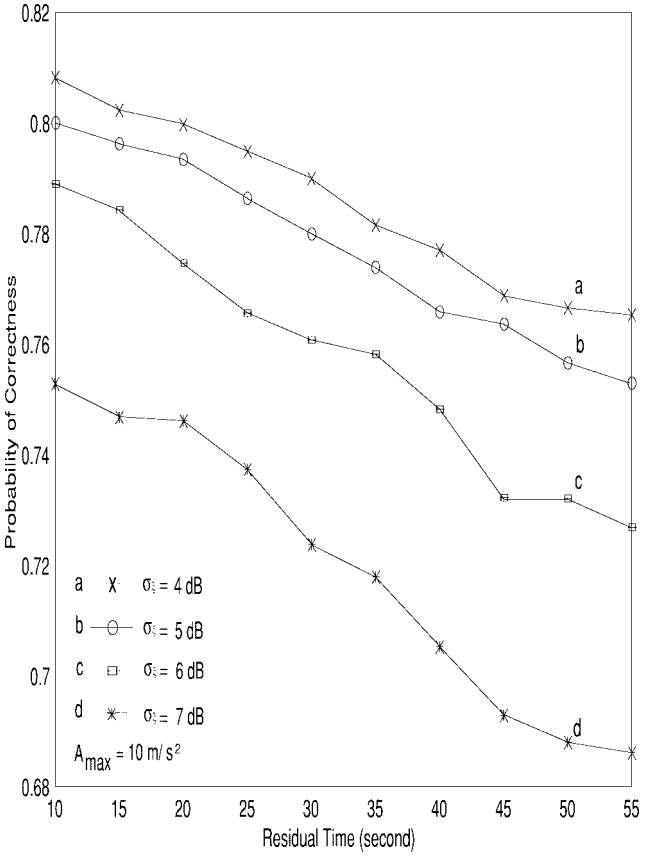


Fig. 10. Parametric behavior of probability of correctness for local prediction P_l.

$$P_g = 1 - \gamma \frac{n-1}{n} \tag{29}$$

where n is the number of neighboring cells from which the next cell for the mobile may be chosen. Obviously, GP is not reliable if γ is high (unclassifiable movement in [9] corresponds to the case when $\gamma = 1$). However, prediction of the next cell to be crossed can still be achieved in HLP with a high degree of accuracy since LP is completely independent of the long-term mobility patterns. Fig. 11 shows the performance of our HLP algorithm in comparison to the MMP algorithm proposed in [9]. The figure illustrates next-cell prediction versus change in the random factor γ used for UAP. With HLP, if γ is greater than a threshold, T_γ, trajectory-tracking-based LP is enabled for next-cell prediction. A logical choice of T_γ is the value of γ of the inflection point (0.3 in Fig. 11), i.e.,

$$T_\gamma = (1 - P_l) \frac{n}{n-1}. \tag{30}$$

In Fig. 11, the value of P_l is obtained for a residual time $= 30$ s, $A_{\max} = 10$ m/s^2, and $\sigma_\xi = 6$ dB. Comparing HLP with MMP (the dotted line obtained from pure pattern matching [9]), it is clear that the prediction accuracy of HLP shows a marked improvement over MMP *unclassifiable* intercell movement.

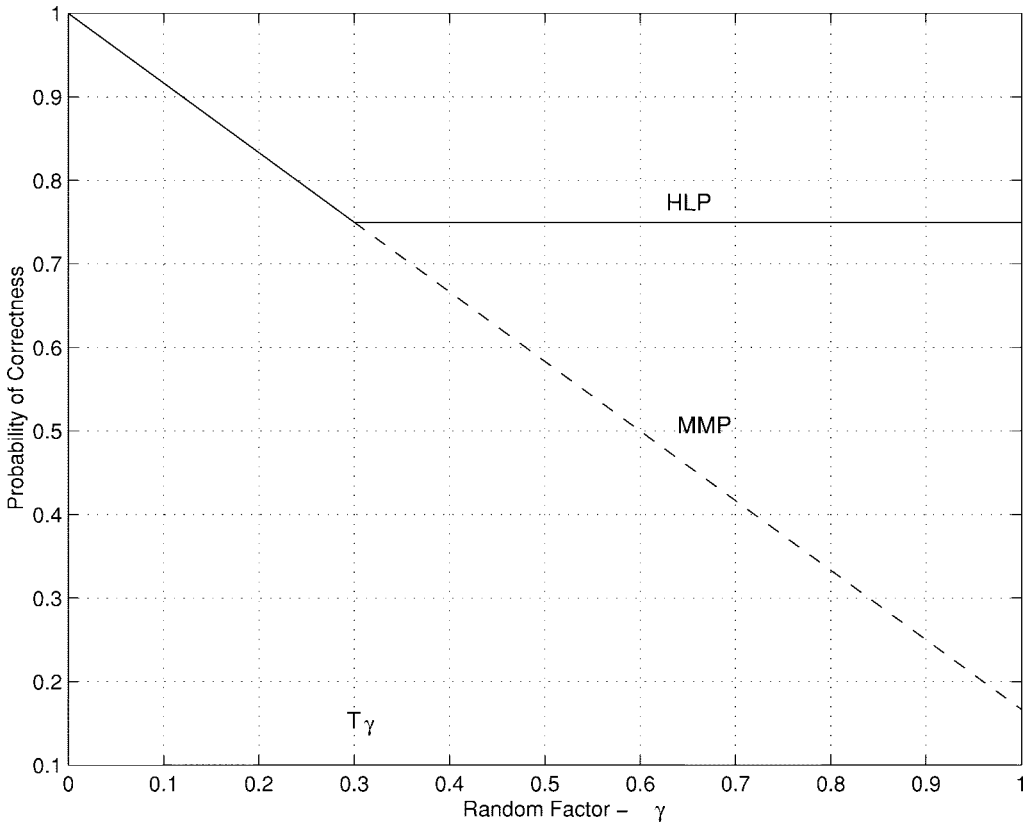


Fig. 11. Comparison of probability of correctness of the next-crossing cell using HLP and MMP [9].

VI. Systems Implementation

We now present two examples to illustrate how HLP can be applied within a wireless ATM system. The examples include a *prediction-based dynamic virtual connection tree* (PVCT) strategy for providing enhanced connection management functionality, and a *prediction-based dynamic location update* strategy.

For the sake of this explanation, a minimal system architecture is assumed. Specifically, we assume that the wireless ATM system consists of: 1) a wired backbone infrastructure that supports ATM packet transport; 2) a collection of radio base station transreceivers, each of which is connected to the fixed wired infrastructure, and each of which supports on-demand packet access to a shared radio channel (the base stations provide connectivity to all mobiles within their area of influence called a cell; and 3) a set of mobile users who are equipped with portable RF wireless terminals. These users (mobiles) are free to roam anywhere in the wireless network, relying on the system to maintain continuous connectivity even as they move from one cell to another.

To implement the HLP algorithm in wireless ATM systems, we notice that both Kalman filtering and the approximate pattern-matching algorithm have a recursive nature and modest use of memory storage, which make it feasible to be implemented as a low-power, embedded software process. The computation load of Kalman filtering can be further lowered by taking advantage of matrix sparseness or matrix symmetry which will significantly reduce the number of additions and multiplications needed to implement the matrix equations [23]. Furthermore, if the measurement availability and rate are high, as the sampling interval of 0.5 s used in the simulation, the error covariance and corresponding Kalman gain do not change much from one cycle to the next. Then, instead of updating these quantities every cycle, doing so at a lower rate will result in a substantial amount of computational savings.

To reduce signaling traffic, distribute computation load, and improve system security while still achieving location and speed prediction, we suggest implementing the HLP algorithm within the mobile hosts rather than in the base stations. By storing the historical long-term movement patterns (UMP's) and the current movement history (UAP) within the mobile, security against unwanted predictors and locators can be provided. Additional information needed by the HLP algorithm to function correctly, such as cell identification numbers and cell topology, can be obtained from the base station broadcasting on a protected control channel at the beacon frequency in each cell. The results from HLP are transmitted to the network infrastructure for predictive mobility management.

A. Prediction-Based Dynamic Virtual Connection Trees

As explained in Section I, the *virtual connection tree* (VCT) approach [2] maintains QoS by minimizing latency during

handoffs initiation and completion, by minimizing cell loss, and by reducing processing delay during handoff. The strength of the VCT is in the admission control and rerouting algorithms which are fairly simple to implement. The primary drawback of the VCT approach is its inefficient use of network resources, a potential for causing system overload, and a long setup processing time needed for assigning virtual connection (VC) numbers on a typically large area. System inefficiencies occur since the preassigned VC's do not accurately account for the mobile's current and projected movement patterns. Consequently, many preassigned VC's are wasted, and efficient resource reservation cannot be achieved.

While keeping the advantages of the VCT approach, we propose to reduce its disadvantages by dynamically allocating VC's based on location prediction. When a matched UMP is found, the mobile informs the current serving base station of the results of the HLP algorithm, which include identifiers for the cells in the path of the mobile's predicted trajectory and the mobile's predicted speed (cell-dwell time) in these cells. Using this information, the system sets up end-to-end connections with appropriate resources from the base stations in the predicted cells. A set of *virtual circuit identifier numbers* (VCN's) is associated with these connections and passed on to the mobile. Each of these connections is maintained by the system (VCN stays valid) for a time duration determined from the mobile's predicted velocity and predicted cell dwell time. In the case when no UMP is found to match the UAP and only information from LP is available, VCN's of the most likely neighboring cells are assigned to the mobile. By following the fairly accurate hints about the mobile's trajectory, provided by the HLP, unwanted cell overloading is reduced, admission control is faster, and system efficiency is superior since only a subset of the VCT footprint (cluster of base stations) is involved in resource reservation for each ATM connection. Thus, the PVCT approach trades off minimal QoS reliability for vastly improved system efficiency when compared to the native exhaustive VCT approach.

B. Prediction-Based Dynamic Location Update

HLP is based on the belief that, at a global level, a user's movement pattern is fairly regular, and can be approximated by a representative UMP. (Implicit in this statement is the fact that, with time, the user "learns" and stores the regular patterns of its movement.) Therefore, for the purpose of location tracking, the predefined zones, as proposed in the two-tier architecture in [4], can dynamically be mapped to the matched UMP obtained from the GP. As long as the mobile follows the assumed UMP, no location update is necessary since the current location server can preinform the rest of the predicted location servers through the wired network. Therefore, the mobile can continue to use the services without conventional registration and location update procedures when it arrives at the zone.

In the special case when no UMP is found to match UAP, the conventional location update method can be enabled to collect location information with a lower periodicity. As most mobile users are quite regular in their daily movement, signaling traffic due to location update can be significantly reduced.

VII. Conclusion

We have explored the fundamental problem of providing lifetime connectivity to ongoing sessions initiated by mobile users in a cell-based wireless ATM network. The motivation behind this research is derived from the recognition that the performance of the mobility management subsystem is key to any QoS-based wireless ATM network.

The approach we took was to develop a robust algorithm for predicting the future locations and speeds of the mobiles. In order to develop such an algorithm, we proposed a novel human-centric pseudostochastic mobility model that rejects the notion that all movement is ad hoc. With this refined observation of human mobility behavior, we developed a two-level hierarchical location prediction (HLP) algorithm for accurate prediction of the cells the mobile will cross during the lifetime of its connection. The HLP algorithm had two main components in it: 1) global prediction and 2) local prediction.

For global prediction, we proposed an *approximate pattern-matching* algorithm good for any finite sequence comparisons. We showed that this algorithm improves the prediction accuracy by abstracting the geometric similarity between two cell sequences which may otherwise seem dissimilar. Our method effectively enlarges the area covered by a UMP, making pattern classification flexible and robust, while reducing memory requirements significantly. Only a few UMP's have to be stored as they can represent a large number of mobility patterns, needed for accurate intercell prediction.

For local prediction, we applied classical stochastic signal processing techniques to extract user mobility information from noisy measurements. Analysis and simulation results proved that a *self-adaptive extended Kalman filter* provides a high degree of accuracy for next-cell location and instantaneous speed prediction. Local prediction in HLP was kept independent of global prediction (vice versa is not true) so that reasonably accurate short-distance prediction can be obtained even when the system has no knowledge of the user's historical mobility patterns.

Finally, HLP is independent of the architecture of the underlying wireless ATM system. Two strategies, *prediction-based dynamic virtual connection trees* (PVCT) and *prediction-based dynamic location update*, were presented as examples of deployable predictive mobility management in wireless ATM networks.

Appendix

A. Derivation of Discrete-Time Equations for a Mobile User

The continuous-time dynamic equation has the form

$$\dot{\boldsymbol{x}}(t) = A'\boldsymbol{x}(t) + B'\boldsymbol{u}(t) + C'\boldsymbol{w}(t), \qquad i = 1, \cdots, m \quad (31)$$

where

$$\begin{aligned}\boldsymbol{x}(t) &= [x(t) \quad \dot{x}(t) \quad r_x(t) \quad y(t) \quad \dot{y}(t) \quad r_y(t)]^T \\ \boldsymbol{u}(t) &= [u_x(t) \quad u_y(t)]^T, \\ \boldsymbol{w}(t) &= [w_x(t) \quad w_y(t)]^T, \qquad \text{with variance } 2\alpha\sigma_m^2\end{aligned}$$

$$A' = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\alpha \end{bmatrix}$$

$$B' = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C' = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Sampling the process every T s, the appropriate (discrete-time) dynamic equations are given by

$$X_{n+1} = A(T, \alpha)X_n + B(T)U_n + W_n. \tag{32}$$

Since

$$\begin{aligned}\boldsymbol{x}(t+T) = e^{A'T}\boldsymbol{x}(t) &+ \int_t^{t+T} e^{A'(t+T-\tau)}B'\,d\tau \\ &+ \int_t^{t+T} e^{A'(t+T-\tau)}C'\boldsymbol{w}(\tau)\,d\tau\end{aligned} \tag{33}$$

it follows that

$$\begin{aligned}A(T, \alpha) &= e^{A'T}, \\ B(T) &= \int_t^{t+T} e^{A'(t+T-\tau)}B'\,d\tau, \\ W_n &= \int_t^{t+T} e^{A'(t+T-\tau)}C'\boldsymbol{w}(\tau)\,d\tau.\end{aligned} \tag{34}$$

These terms can be calculated using eigenvalue analysis. The eigenvalues of A' satisfy

$$\det(\lambda - A) = \lambda^2(\lambda + \alpha) = 0 \quad \text{so that} \quad \lambda = 0, 0, -\alpha. \tag{35}$$

It can be verified that

$$A(T, \alpha) = \begin{bmatrix} 1 & T & p_1 & 0 & 0 & 0 \\ 0 & 1 & p_2 & 0 & 0 & 0 \\ 0 & 0 & p_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & p_1 \\ 0 & 0 & 0 & 0 & 1 & p_2 \\ 0 & 0 & 0 & 0 & 0 & p_3 \end{bmatrix} \quad B(T) = \begin{bmatrix} p_4 & 0 \\ p_5 & 0 \\ 0 & 0 \\ 0 & p_4 \\ 0 & p_5 \\ 0 & 0 \end{bmatrix}$$

$$W_n = \int_{nT}^{(n+1)T} A[(n+1)T - \tau]C'\boldsymbol{w}(\tau)\,d\tau \tag{36}$$

where

$$\begin{aligned}p_1 &= \frac{1}{\alpha^2}(-1 + \alpha T + e^{-\alpha T}) \\ p_2 &= \frac{1}{\alpha}(1 - e^{-\alpha T}) \\ p_3 &= e^{-\alpha T} \\ p_4 &= T - 1 + e^{-T} \\ p_5 &= 1 - e^{-T}.\end{aligned}$$

Since $w(t)$ is white noise, $E[W_nW_{n+i}] = 0$ for $i \neq 0$ so that W_n is a discrete-time white noise sequence. The covariance matrix Q of $W'[n]$ is given by

$$\begin{aligned}Q'[n] &= E\{W'[n](W'[n])^T\} \\ &= 2\alpha\delta_m^2 \begin{bmatrix} q_{11} & q_{12} & q_{13} & 0 & 0 & 0 \\ q_{21} & q_{22} & q_{23} & 0 & 0 & 0 \\ q_{31} & q_{32} & q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{11} & q_{12} & q_{13} \\ 0 & 0 & 0 & q_{21} & q_{22} & q_{23} \\ 0 & 0 & 0 & q_{31} & q_{32} & q_{33} \end{bmatrix}\end{aligned} \tag{37}$$

where

$$\begin{aligned}q_{11} &= \frac{1}{2\alpha^5}\Big(1 - e^{-2\alpha T} + 2\alpha T + \frac{2\alpha^3T^3}{3} - 2\alpha^2T^2 \\ &\qquad - 4\alpha T e^{-\alpha T}\Big) \\ q_{12} &= \frac{1}{2\alpha^4}\left(e^{-2\alpha T} + 1 - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T}\right. \\ &\qquad \left. -2\alpha T + \alpha^2T^2\right) \\ q_{13} &= \frac{1}{2\alpha^3}\left(1 - e^{-2\alpha T} - 2\alpha T e^{-\alpha T}\right) \\ q_{22} &= \frac{1}{2\alpha^3}\left(4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T\right) \\ q_{23} &= \frac{1}{2\alpha^2}\left(e^{-2\alpha T} + 1 - 2e^{-\alpha T}\right) \\ q_{33} &= \frac{1}{2\alpha}\left(1 - e^{-2\alpha T}\right).\end{aligned}$$

B. Derivation of the Adaptive State Estimator

The desired estimator is to be the conditional mean given by

$$\begin{aligned}\hat{X}_{n+1} &= E[X_{n+1}/Z_{n+1}] \\ &= \sum_{X_{n+1}} X_{n+1}P(X_{n+1}/Z_{n+1})\end{aligned} \tag{38}$$

where Z_{n+1} denotes all of the measurements up to and including time $n+1$ (i.e., $z_1, \cdots, z_{n+1}$). Define $P(S_{n+1}^i)$ as the probability that the input is in state S_i at time $n+1$ given the data sequence Z_{n+1}, $P(X_{n+1}/Z_{n+1})$ can be represented by

$$\begin{aligned}P(X_{n+1}/Z_{n+1}) = \sum_{i=1}^{n} &P\big(X_{n+1}/Z_{n+1}, S_{n+1}^i\big) \\ &\cdot P\big(S_{n+1}^i/Z_{n+1}\big).\end{aligned} \tag{39}$$

Combining (39) and (40)

$$\hat{X}_{n+1} = \sum_{i=1}^{n} \hat{X}_{n+1}(S_i)P\big(S_{n+1}^i\big)/Z_{n+1}\big) \tag{40}$$

where

$$\hat{X}_{n+1}(S_i) = \sum_{x_{n+1}} X_{n+1}P\big(X_{n+1}/Z_{n+1}, S_{n+1}^i\big). \tag{41}$$

Denoting $Z_{n+1} = (Z_n, z_{n+1})$, we have

$$P(S_{n+1}^i/Z_{n+1}) = \frac{f(z_{n+1}/S_{n+1}^i, Z_n)P(S_{n+1}^i/Z_n)}{f(Z_{n+1}/Z_n)}. \tag{42}$$

Assuming that the probability of state transition between any two adjacent time samples is small, the first term $f(z_{n+1}/S^i_{n+1}, Z_n)$ can be approximated by a Gaussian density, which can be established from the Kalman filtering algorithms conditioned on s_i, i.e.,

$$f(z_{n+1}/S^i_{n+1}, Z_n) \sim N\{[H_{n+1}AX_n(S_i) + H_{n+1}BU_n(S_i)]; \left(H_{n+1}M_{n+1/n}H^T_{n+1} + R_{n+1}\right)\} \tag{43}$$

where $M_{n+1/n}$ is the one-step prediction error matrix in the Kalman algorithm, and R_{n+1} is the measurement noise covariance matrix.

Expanding $P(S^i_{n+1}/Z_n)$

$$P\left(S^i_{n+1}/Z_n\right) = \sum_{\alpha=1}^{m} P\left(S^i_{n+1}/S^\alpha_n, Z_n\right)P(S^\alpha_n/Z_n) = \sum_{\alpha=1}^{m} P\left(S^i_{n+1}/S^\alpha_n\right)P(S^\alpha_n/Z_n). \tag{44}$$

The last equation is obtained by observing that Z_n and $P(S^\alpha_n)$ are strongly dependent; actually, Z_n aids in determining $P(S^\alpha_n)$; therefore, $P(S^i_{n+1}/S^\alpha_n, Z_n)$ can be expressed as $P(S^i_{n+1}/S^\alpha_n)$. It has been pointed out that for uniform sampling of a semi-Markovian process this transition probability only depends on the sampling interval. A good engineering approximation is [7]

$$P(S^i_{n+1}/S^\alpha_n) = \theta_{\alpha i} = \begin{cases} 0.95, & i = j \\ \dfrac{0.95}{m-1}, & i \neq j. \end{cases} \tag{45}$$

Combining (40) and (42), we finally get the recursive equation to compute $P(S^i_{n+1}/Z_{n+1})$

$$P\left(S^i_{n+1}/Z_{n+1}\right) = bf\left(z_{n+1}/S^i_{n+1}, Z_n\right)\sum_{\alpha=1}^{m}\theta_{\alpha i}P(S^\alpha_n/Z_n) \tag{46}$$

where b is a normalization constant.

If the covariance matrix of the random acceleration and measurement noise Q and R does not change with the input states S_i, it is possible to greatly simplify the Kalman filter bank structure to only one Kalman filter. In this case, the Kalman gain $K_{n+1}(S_i)$ becomes the same for each $i(i = 1, 2, \cdots, m)$. By adding the weighted estimates from the bank of m filters and again making the approximation that the weighting coefficients change very little from sample to sample, i.e., $P(S^i_{n+1}/Z_{n+1}) \approx P(S^i_n/Z_n)$, it follows that the adaptive estimator reduces to

$$\hat{X}_{n+1} = A\hat{X}_n + B\hat{U}_n + K_{n+1}\left(Z_{n+1} - HAX_n - HB\hat{U}_n\right) \tag{47}$$

where

$$\hat{U}_n = \sum_{i=1}^{n} U_n(S_i)P(S^i_{n+1}/Z_{n+1}) \tag{48}$$

with $P(S^i_{n+1}/Z_{n+1})$ given by (43) and K_{n+1} the standard Kalman gain.

REFERENCES

[1] S. Ramanathan and M. Steenstrup, "A survey of routing techniques," *ACM/Baltzer Mobile Networks Appl.*, pp. 89–103, 1996.

[2] A. Acampora and M. Naghshineh, "An architecture and methodology for mobile-executed hand-off in cellular ATM networks," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1365–1375, Oct. 1994.

[3] D. A. Levine, I. F. Akyildiz, and M. Naghshineh, "The shadow cluster concept for resource allocation and call admission in ATM-based wireless networks," in *Proc. 1st ACM MobiCom*, Nov. 1995, pp. 142–150; also, *IEEE/ACM Trans. Networking*, vol. 6, pp. 1–10, Feb. 1997.

[4] B. A. Akyol and D. C. Cox, "Signaling alternatives in a wireless ATM network," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 35–49, Jan. 1997.

[5] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update?," in *Proc. IEEE INFOCOM*, 1994.

[6] M. Mouly and M. B. Pautet, *The GSM System for Mobile Communications*. ISBN 2-9507190-0-7, 1992.

[7] Telecommunications Industry Association, Cellular Radio Telecommunications Intersystem Operation, TIA/EIA IS-41B, 1991.

[8] S. Tabbane, "An alternative strategy for location tracking," *IEEE J. Select. Areas Commun.*, vol. 13, June 1995.

[9] G. Y. Liu and G. Q. Maguire, Jr., "A predictive mobility management algorithm for wireless mobile computation and communication," in *Proc. IEEE Int. Conf. Universal Personal Commun.*, 1995.

[10] M. D. Austin and G. L. Stuber, "Velocity adaptive handoff algorithm for microcellular systems," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 549–561, Aug. 1994.

[11] ——, "Direction biased handoff algorithms for urban microcells," in *Proc. ICUPC*, 1994, pp. 101–104.

[12] R. Viayan and J. M. Holtzman, "A model for analyzing handoff algorithms," *IEEE Trans. Veh. Technol.*, vol. 42, Aug. 1993.

[13] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Veh. Technol.*, vol. 42, Aug. 1993.

[14] K. Ivanov and G. Spring, "Mobile speed sensitive handover in mixed cell environment," in *Proc. IEEE Veh. Technol. Conf.*, 1995, pp. 892–896.

[15] R. Guerin, "Queuing and traffic in cellular radio," Ph.D. dissertation, California Inst. Technol., Pasadena, May 1986.

[16] P. H. Sellers, "An algorithm for the distance between two finite sequences," *J. Algorithms 1*, pp. 359–373, 1980.

[17] R. A. Singer, "Estimating optical tracking filter performance for manned maneuvering targets, " *IEEE Trans. Aerosp. Electron. Syst.*, 1970.

[18] R. L. Moose and H. F. Vanlandingham, "Modeling and estimation for tracking maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 15, May 1979.

[19] G. P. Pollini, "Trends in handover design," *IEEE Commun. Mag.*, pp. 82–90, Mar. 1996.

[20] H. H. Xia, "An analytical model for predicting path loss in urban and suburban environments," in *Proc. PIRMC*, 1996.

[21] J. Ho, Y. B. Lin, and I. Akylidiz, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Trans. Networking*, vol. 4, pp. 629–639, Aug. 1996.

[22] R. A. Howard, "System analysis of semi-Markov processes," *IEEE Trans. Mil. Electron.*, vol. MIL-8, pp. 114–124, Apr. 1964.

[23] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed. New York: Wiley, 1997.

Tong Liu received the B.S. degree in electrical engineering from Tianjin University, Tianjin, China, in 1991, and the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1994.

She is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, Boston University. From 1997 to 1998, she worked at Tellabs Wireless Division, Burlington, MA, as a System Engineer. She joined Ascend Communications Inc., Westford, MA, in 1998 where she is carrying out research in performance analysis and modeling for broadband networks. Her current research interests include traffic mobility management, resource management, performance analysis, and network architecture design for broadband networks.

Paramvir Bahl (M'95–SM'97) received the Ph.D. degree in computer systems engineering from the University of Massachusetts, Amherst.

He is currently with Microsoft Corporation, where he is carrying out research in peripatetic computing, multihop multiservice ad hoc networks, mobility-aware self-configuring operating systems, and real-time audio–visual communications. Prior to being with Microsoft, he spent nine years at Digital Equipment Corporation. There, he initiated, led, and contributed to several seminal multimedia projects, including the industry's first desktop hardware adapters for audio/video compression, world's fastest software implementations of ISO and ITU audio/video codecs, and fast and scalable image and video rendering algorithms. He has authored more than two dozen publications, and is the author of one patent issued and several patent applications in the areas of digital signal processing, computer communications, and wireless networks.

In 1994, Dr. Bahl received Digital's Multiyear Engineering Fellowship Award for carrying out research in real-time visual communications over wireless networks. He is the Vice Chair of ACM SIGMOBILE, and the founding Editor-in-Chief of *ACM Mobile Computing and Communications Review*. He has served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the *ACM Journal on Mobile Networking and Applications*, and the IEEE COMMUNIATIONS MAGAZINE. He has served on a number of organizing and technical program committees of IEEE and ACM conferences, including MobiCom, INFOCOM, PIMRC, and MoMuC. He has organized sessions and panels on the topic of multimedia communications, wireless LAN's, and mobile computing. He is a member of ACM, and a Past President of the electrical engineering honor society Eta Kappa Nu (Zeta Pi chapter).

Imrich Chlamtac (SM'86–F'93) received B.Sc. and M.Sc. degrees in mathematics awarded with the highest distinction, and the Ph.D. degree in computer science from the University of Minnesota in 1979.

He currently holds the Distinguished Chair in Telecommunications at the University of Texas at Dallas, and is the President of Boston Communications Networks. He is the author of more than 200 papers in refereed journals and conferences, multiple books, and book chapters.

Dr. Chlamtac is the founding Editor-in-Chief of the *ACM-URSI-Baltzer Wireless Networks (WINET)* and the *ACM-Baltzer Mobile Networks and Applications* journals, and served on the Editorial Board of IEEE TRANSACTIONS ON COMMUNICATIONS and other leading journals. He served as the General Chair of several ACM and IEEE conferences and workshops, was a Fulbright Scholar, and an IEEE, Northern Telecom, and BNR Distinguished Lecturer. He is the founder of ACM/IEEE MobiCom and of the ACM Sigmobile, of which he is the current Chairman. He is a Fellow of the ACM, and is an Honorary Member of the Senate of the Technical University of Budapest.