# Bearing-Only Maneuvering Mobile Tracking With Nonlinear Filtering Algorithms in Wireless Sensor Networks

Dah-Chung Chang, Student Member, IEEE, and Meng-Wei Fang

Abstract-Mobile node localization is important to offer wireless services in vehicular communication applications. Some typical methods realize the mobile node tracking through data fusion from time of arrival (TOA) and received signal strength (RSS) measurements provided by sensor nodes or base stations (BSs). Although the TOA/RSS method is not expensive under a concern of cost, it is very sensitive to multipath signal propagation effects. As the technology of angle of arrival (AOA) antennas is showing a rapid progress, we turn to consider AOA estimation. In this paper, the nonlinear extended Kalman filter (EKF) and the particle filter (PF) along with a three-model interacting multiple model (IMM) algorithm are utilized and compared for maneuvering mobile station (MS) tracking with bearing-only measurements. A coordinated turn model is used to improve the tracking performance since the MS frequently turns in the streets. We also propose an efficient method for resampling particles to alleviate the degeneracy effect of particle propagation in the interacting multiple model particle filter (IMMPF) algorithm. Moreover, a BS sensor selection scheme is also exploited for the long-haul MS tracking case which often changes BSs in a wireless vehicular sensor network. Numerical simulations show that the three-model IMMPF algorithm outperforms the interacting multiple model extended Kalman filter algorithm and achieves a root-mean-square tracking performance which is quite close to the posterior Cramer-Rao lower bound.

Index Terms-Angle of arrival (AOA), interacting multiple model (IMM), Kalman filtering, mobile tracking, particle filtering, posterior Cramer-Rao lower bound (CRLB), resampling.

## I. INTRODUCTION

OBILE POSITIONING is a rapidly growing and crucial research topic in advanced vehicular technology [1]–[4]. Knowing the accurate location of the mobile station (MS) can provide diverse applications in wireless communication networks, such as the localization of metropolitan transportation vehicles, emergency care, public traffic control, etc. The mobile location can be measured by widely distributed base stations (BSs). In fact, those BSs can be viewed as the sensors in a wireless sensor network (WSN). The measured MS information,

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2013.2260641

including received signal strength (RSS) and arriving signal angle, can be transmitted to the data fusion center for signal postprocessing. Hence, some typical methods used to estimate the mobile position may utilize the angle of arrival (AOA) [5], time difference of arrival (TDOA), time of arrival (TOA), RSS [3], [6], GPS, etc.

Mobile location estimation mainly determines the location and velocity of an MS based on the measured signals from its neighboring BSs. TOA [4] is a technique to measure the round trip time of signal transmission from the BS to the MS and back to the BS, and the arrival time of the radio signals is usually measured from different wireless BSs. The TDOA technique measures the time difference between the radio signals, and the distance differences of the MS to at least three BSs are measured. The AOA technique is only conducted within the BS by observing the arriving angle of the signals coming from the MS. Each BS determines the direction of the received signal angle. The MS location is derived by the intersection of at least two lines of antenna arrays. To eliminate unsatisfactory effect, more than two BSs are usually utilized. The AOA method has the advantage over TOA that the BS does not need time synchronization and is robust against multipath signal propagation effects. However, the accuracy of mobile location estimation is still a very difficult problem and continues attracting a lot of works. Moreover, the data fusion techniques are also employed to improve accuracy.

Although some recent research can be found in developing mixed localization methods such as TOA/AOA [7], TDOA/ AOA [8], AOA/RSS [9], or TOA/RSS [10], in this paper, owing to the merit of being immune to multipath signal propagation effects, we focus on the AOA method using an advanced nonlinear filtering algorithm for tracking a maneuvering MS. It is well known that the Kalman filter has been widely used for parameter estimation in a state-space model since it is the optimally recursive linear filter in the Gaussian noise environment. Nevertheless, with the AOA method, the mobile location in the Cartesian coordinate is inherently a nonlinear relationship with the arriving angle measurements. Then, a nonlinear modification to the linear Kalman filter, the extended Kalman filter (EKF) [11], can be used. From the point of view of practical model on states, the EKF does not guarantee an optimal solution in a complicated situation of nonlinearity and non-Gaussian distribution. The particle filter (PF) is an alternative with competitive performance compared to the EKF [12], [13]. However, in a practical mobile localization problem, the MS cannot be described by a uniformly kinematic equation in all

Manuscript received July 17, 2012; revised November 3, 2012; accepted January 16, 2013. Date of publication June 26, 2013; date of current version February 5, 2014. This work was supported in part by the National Science Council of Taiwan under Contract NSC 101-2221-E-008-071.

D.-C. Chang is with the Department of Communication Engineering, National Central University, Taoyuan 32001, Taiwan (e-mail: dcchang@ ce.ncu.edu.tw).

M.-W. Fang was with the Department of Communication Engineering, National Central University, Taoyuan 32001, Taiwan.

<sup>1932-8184 © 2013</sup> IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications\_standards/publications/rights/index.html for more information.

motion situations. An effective multiple model method, called interacting multiple model (IMM) algorithm [14], is suited to improve the tracking performance, which can mix appropriate dynamic models based on a Bayesian probability metric.

In previous works, the combination of the IMM algorithm and EKF (or PF) for solving the TOA/RSS problem can be found in the literature [3]. However, its dynamic model for a maneuvering MS was quite simple, where two dynamic models, constant velocity motion and acceleration motion, were only considered. An MS frequently makes a turn in the street. As we take into account this situation, we add the third dynamic model describing the turning motion in this paper. With bearingonly BS data, we derive the nonlinear measurement equation and the turning motion dynamic equation for the use of EKF and PF. In numerical simulation, the performance of using the new dynamic model along with the interacting multiple model particle filter (IMMPF) and interacting multiple model extended Kalman filter (IMMEKF) algorithms is compared to show its effectiveness for the application of AOA maneuvering MS tracking. In this paper, only the line-of-sight situation for AOA MS localization is considered. The non-line-of-sight situation is more complicated [15], [16] and out of this scope, which will be addressed in another work.

The rest of this paper is organized as follows. Section II describes the system model in our framework and establishes the AOA measurement model and the state equations to be used in this paper. In Section III, we explain how the EKF and PF algorithms work with the proposed dynamic and measurement models in the IMM algorithm. The posterior Cramer–Rao lower bound (CRLB) is also analyzed for providing theoretical performance comparison. Moreover, a BS selection scheme is introduced for the long-haul tracking application. The setup of numerical simulations and the tracking performance are evaluated in Section IV. A brief conclusion about the proposed work is deferred in Section V.

#### **II. PROBLEM FORMULATION**

The mobile localization system to be explored in this paper is depicted in Fig. 1, where four BSs, numbered by BS1, BS2, BS3, and BS4, are assumed to be located at four corners, respectively, and the corresponding four arriving signal angles  $\theta_1(k)$ ,  $\theta_2(k)$ ,  $\theta_3(k)$ , and  $\theta_4(k)$  at time k can be measured for the MS target. Suppose that a Cartesian coordinate with the horizontal axis x and the vertical axis y is built in this configuration and the target position is at (x(k), y(k)).

# A. System Model

Let  $\mathbf{X}(k)$  denote the state vector,  $\mathbf{Z}(k)$  denote the measurement vector,  $f(\cdot)$  denote the state transition function, and  $h(\cdot)$  denote the measurement function. The state-space system model can be described by the following two equations [17], [18]:

$$\mathbf{X}(k+1) = f\left(\mathbf{X}(k), t(k), m(k)\right) + \mathbf{w}\left(k, m(k)\right)$$
(1)

$$\mathbf{Z}(k) = h\left(\mathbf{X}(k), t(k), m(k)\right) + \mathbf{v}(k)$$
(2)

where t(k) denotes the sampling time (here, we simply let  $t(k) = \Delta t$ ), m(k) denotes the different dynamic model,



Fig. 1. Configuration of mobile localization and BS location.

 $\mathbf{w}(k, m(k))$  is the process noise vector, and  $\mathbf{v}(k)$  is the measurement noise vector. For simplicity,  $\mathbf{w}(k, m(k))$  and  $\mathbf{v}(k)$  are modeled as two uncorrelated white Gaussian noise vectors, where  $\mathbf{w}(k, m(k)) \sim \mathcal{N}(\mathbf{0}, \widetilde{\mathbf{Q}}_{m(k)})$  with a process noise covariance  $\widetilde{\mathbf{Q}}_{m(k)}$  and  $\mathbf{v}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$  with a diagonal covariance matrix  $\mathbf{R}$ .

## B. State Dynamic Model

In this paper, the full state vector is defined as

$$\mathbf{X}(k) = [x(k), \dot{x}(k), \ddot{x}(k), y(k), \dot{y}(k), \ddot{y}(k)]^T$$
(3)

where x(k),  $\dot{x}(k)$ , and  $\ddot{x}(k)$  represent the target position, velocity, and acceleration, respectively, in the horizontal axis and y(k),  $\dot{y}(k)$ , and  $\ddot{y}(k)$  are the same physical elements in the vertical axis. For simplicity without losing generality, we will only write out the horizontal dynamic expression to be considered in this paper. The dynamic expression defined in the vertical axis can be applied in a similar manner as well.

For the problem of interest in the tracking of mobile motion in the street, we classify the dynamic model into three different equations: constant velocity (CV), constant acceleration (CA), and constant turning (CT). Denote that  $m(k) \in \{1, 2, 3\}$ , where m(k) = 1 represents the CV model, m(k) = 2 is the CA model, and m(k) = 3 is the CT model.

*CV Model* (m(k) = 1). In this model, the elements  $\ddot{x}(k)$  and  $\ddot{y}(k)$  are neglected because no acceleration is supposed, i.e., we can simplify that  $\mathbf{X}'(k) = [x(k), \dot{x}(k)]^T$  as only the horizontal model is written out for simplicity. By removing the estimate of acceleration, the process noise due to acceleration estimation will never be involved in order to improve estimation accuracy. We express the CV dynamic model as

$$f(\mathbf{X}'(k), \Delta t, 1) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{X}'(k)$$
(4)

$$\widetilde{\mathbf{Q}}_1 = \mathbf{I}_2 \otimes \begin{bmatrix} \frac{(\Delta t)^4}{4} & \frac{(\Delta t)^3}{2} \\ \frac{(\Delta t)^3}{2} & (\Delta t)^2 \end{bmatrix} \sigma_1^2 \qquad (5)$$

where  $I_m$  denotes the  $m \times m$  identity matrix,  $\otimes$  is the Kronecker product, and  $\sigma_1^2$  is a parameter to control the process noise variance of the CV model.

CA Model (m(k) = 2). In this model, all of the elements x(k),  $\dot{x}(k)$ , and  $\ddot{x}(k)$  are used in estimation. Let  $\mathbf{X}''(k) = [x(k), \dot{x}(k), \ddot{x}(k)]^T$ , and the CA dynamic model is expressed as

$$f(\mathbf{X}''(k), \Delta t, 2) = \begin{bmatrix} 1 & \Delta t & \frac{(\Delta t)^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}''(k)$$
(6)

$$\widetilde{\mathbf{Q}}_{2} = \mathbf{I}_{3} \otimes \begin{bmatrix} \frac{(\Delta t)^{4}}{4} & \frac{(\Delta t)^{3}}{2} & \frac{(\Delta t)^{2}}{2} \\ \frac{(\Delta t)^{3}}{2} & (\Delta t)^{2} & \Delta t \\ \frac{(\Delta t)^{2}}{2} & \Delta t & 1 \end{bmatrix} \sigma_{2}^{2} \quad (7)$$

where  $\sigma_2^2$  is a parameter to control the process noise variance of the CA model.

*CT Model* (m(k) = 3). In [19] and [20], the kinematics of a turn performed in a horizontal plane with known turning rate  $\omega(k)$  was proposed. We can rewrite the kinematic equations in this paper as

$$f(\mathbf{X}(k), \Delta t, 3) = \begin{bmatrix} x(k) + \Delta t \times \dot{x}(k) - \frac{1}{2}(\Delta t)^2 \dot{y}(k)\omega(k) \\ y(k) + \Delta t \times \dot{y}(k) + \frac{1}{2}(\Delta t)^2 \dot{x}(k)\omega(k) \\ (1 - \frac{1}{2}(\Delta t)^2 \omega^2(k)) \times \dot{x}(k) - \Delta t \times \dot{y}(k)\omega(k) \\ (1 - \frac{1}{2}(\Delta t)^2 \omega^2(k)) \times \dot{y}(k) + \Delta t \times \dot{x}(k)\omega(k) \end{bmatrix}.$$
(8)

The turning rate highly depends on the traffic status in nobile tracking applications, and preassigning some typical value may be not practical. Moreover, the clockwise or anticlockwise turn is also affected by the value to be positive or negative. Hence, we can view  $\omega(k)$  as a random variable to be estimated and augment the state vector as  $\mathbf{X}^{a}(k) = [x(k), \dot{x}(k), y(k), \dot{y}(k), \omega(k)]^{T}$  in this model. Then, we can rewrite the dynamic model for m(k) = 3 as

$$f(\mathbf{X}^{a}(k), \Delta t, 3) = \begin{bmatrix} x(k) + \Delta t \times \dot{x}(k) - \frac{1}{2}(\Delta t)^{2}\dot{y}(k)\omega(k) \\ y(k) + \Delta t \times y(k) + \frac{1}{2}(\Delta t)^{2}\dot{x}(k)\omega(k) \\ (1 - \frac{1}{2}(\Delta t)^{2}\omega^{2}(k)) \times \dot{x}(k) - \Delta t \times \dot{y}(k)\omega(k) \\ (1 - \frac{1}{2}(\Delta t)^{2}\omega^{2}(k)) \times \dot{y}(k) + \Delta t \times \dot{x}(k)\omega(k) \\ \omega(k) \end{bmatrix}$$
(9)  
$$\widetilde{\mathbf{Q}}_{3} = \begin{bmatrix} \widetilde{\mathbf{Q}}_{3}^{\prime} & \mathbf{0} \\ \mathbf{0}_{3} \end{bmatrix}$$
(10)

• where  $\sigma_w^2$  is the modeling error variance of  $\omega(k)$ 

$$\widetilde{\mathbf{Q}}_{3}' = \mathbf{I}_{2} \otimes \begin{bmatrix} \frac{(\Delta t)^{4}}{4} & \frac{(\Delta t)^{3}}{2} \\ \frac{(\Delta t)^{3}}{2} & (\Delta t)^{2} \end{bmatrix} \sigma_{3}^{2}$$
(11)

and  $\sigma_3^2$  is a parameter to control the process noise variance of the CT model. Note that the noise variance matrices  $\widetilde{\mathbf{Q}}_1$ ,  $\widetilde{\mathbf{Q}}_2$ , and  $\widetilde{\mathbf{Q}}'_3$  used in (5), (7), and (11), respectively, are the discretized forms obtained from their continuous dynamic models which are discussed in detail in [21], and therefore, the discussion is omitted in this paper. Measurement Model. The available measurement at time k is the angle from the BS platform to the MS target as depicted in Fig. 1. The angle measurement at BS#n is defined as

$$\theta_n(k) = \arctan\left(\frac{y(k) - y_n}{x(k) - x_n}\right), \quad n = 1, 2, 3, 4$$
(12)

where  $(x_n, y_n)$  is the location of BS#*n* in the Cartesian coordinate. Note that it is not necessary to define the angles as the same reference (e.g., clockwise positive) because we can easily modify it by adding or subtracting multiples of  $\pi$ from the same referenced angle to fit the definition in (12). The minimum number of BSs able to track the target is two. However, using more BSs can help the resulted accuracy. In our system, the data from four BSs are collected at the fusion center and modeled in the measurement equation. Let  $\mathbf{Z}(k)$  denote the measurement vector and  $\mathbf{Z}(k) = [\theta_1(k), \theta_2(k), \theta_3(k), \theta_4(k)]^T$ . The measurement equation can be written by

$$\mathbf{Z}(k) = \left[ \arctan\left(\frac{y(k) - y_1}{x(k) - x_1}\right), \cdots, \arctan\left(\frac{y(k) - y_4}{x(k) - x_4}\right) \right]^T + \mathbf{v}(k) \quad (13)$$

where  $\mathbf{v}(k)$  is the measurement noise vector with covariance matrix  $\mathbf{R} = \sigma_{\theta}^2 \mathbf{I}_4$  and  $\sigma_{\theta}^2$  is the angle measurement noise variance.

#### III. NONLINEAR FILTERING METHODS FOR MANEUVERING MOBILE TRACKING

In our formulation, the mobile motion is described by three possible models, CV, CA, and CT. In this complicated and nonlinear system, it is very difficult to estimate required parameters, such as mobile position and velocity, only by a single model. In practical wireless propagation environments, we cannot exactly switch the state-space filtering algorithm among these models. The combination of IMM and the filtering algorithm can provide parallel filtering results and mix the outputs based on their mixing probabilities. However, before employing the IMM algorithm, the transition probabilities defined in the Markovian transition matrix P need to be given. As depicted in Fig. 2,  $p_{ij}$  denotes the mode transition probability from model i to model j where  $1 \le i, j \le 3$  in this system, and P can be written by

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}.$$
 (14)

The model transition matrix is usually determined by an evaluation through traffic analysis about the MS in the practical situation. From an intuitive concept, the MS may remain the present mode in a long time such that  $p_{ii}$  can be set close to unity, for example, 0.96, while  $p_{ij}$ ,  $i \neq j$ , denotes the transition between two different modes such that the value can be close to zero, for example, 0.02. For a normalized probability,  $\sum_i p_{ij} = 1$ .

In general, each cycle in the IMM algorithm consists of four main stages as depicted in Fig. 3. First, the initial state vector estimate  $\widehat{\mathbf{X}}^{0j}(k)$  and the state vector estimation error



Fig. 2. Mode transition in IMM algorithm.



Fig. 3. Structure of the IMM algorithm.

covariance matrix  $\widehat{\mathbf{P}}^{0j}(k)$  are obtained by mixing the estimates  $\widehat{\mathbf{X}}^{j}(k)$  and  $\widehat{\mathbf{P}}^{j}(k)$  of all filters at the previous index k based on the mode probabilities  $\hat{\mu}_i(k)$  under the assumption that those filters are in effect at the current index k + 1, which is performed in the Interaction/mixing block. Then, followed by a regular filtering stage, the new state vector  $\widehat{\mathbf{X}}^{j}(k+1)$ and covariance matrix  $\mathbf{P}^{j}(k+1)$  are produced in parallel from the filters with different models. By calculating the likelihood functions  $L^{j}(k+1)$  of the measurement prediction error based on Gaussian approximation, new mode probabilities  $\hat{\mu}_i(k+1)$ can be updated, and the resultant estimates at the current index,  $\mathbf{X}(k+1)$  (or  $\mathbf{P}(k+1)$ ), are obtained by calculating the mixture of  $\widehat{\mathbf{X}}^{j}(k+1)$  (or  $\widehat{\mathbf{P}}(k+1)$ ) and  $\widehat{\mu}_{i}(k+1)$ . The mode probability plays a key role in the IMM algorithm for mixing the states and covariance matrices that resulted from different dynamic models. Next, we explain how to use the three dynamic models in the IMMEKF and IMMPF algorithms.

#### A. IMMEKF Algorithm

We use the following four stages to describe the IMMEKF algorithm used in this paper.

Stage 1: Interaction. Suppose that  $\hat{\mu}_j(k)$  is the mode probability obtained at the previous index. The mixing probability from mode *i* to mode *j* is calculated by

$$\mu_{i|j}(k) = \frac{1}{c_j} p_{ij} \mu_i(k)$$
ization factor such that
(15)

where  $c_i$  is a normalization factor such that

$$c_j = \sum_{i=1}^{3} p_{ij} \mu_i(k).$$
 (16)

As the estimate vector  $\widehat{\mathbf{X}}^{j}(k)$  and the covariance matrix  $\widehat{\mathbf{P}}^{j}(k)$  are known from the previous cycle, we can compute the initial state vector and the covariance matrix for the input of model *j* in this cycle by

$$\widehat{\mathbf{X}}^{0j}(k) = \sum_{i=1}^{3} \widehat{\mathbf{X}}^{i}(k) \mu_{i|j}(k)$$
(17)

$$\widehat{\mathbf{P}}^{0j}(k) = \sum_{i=1}^{3} \mu_{i|j}(k) \left\{ \widehat{\mathbf{P}}^{i}(k) + [\widehat{\mathbf{X}}^{i}(k) - \widehat{\mathbf{X}}^{0j}(k)][\cdot]^{T} \right\}$$
(18)

where  $[\cdot]^T$  denotes the pairwise same product terms for simplicity in this paper, and here, for example,  $[\cdot] = \widehat{\mathbf{X}}^i(k) - \widehat{\mathbf{X}}^{0j}(k)$ . Stage 2: Prediction and Filtering. Since we have three different dynamic models, the prediction and filtering processes are also different for the three models. The prediction process is directly derived by taking the expectation on both sides of (1), and we can express the predicted state vector for model m(k) = j as

$$\overline{\mathbf{X}}^{j}(k+1) = f\left(\widehat{\mathbf{X}}^{0j}(k), \Delta t, j\right).$$
(19)

The state vector prediction error covariance matrix for model j becomes

$$\overline{\mathbf{P}}^{j}(k+1) = \mathbf{F}^{j}(k)\widehat{\mathbf{P}}^{0j}(k)\mathbf{F}^{jT}(k) + \widetilde{\mathbf{Q}}_{j}(k)$$
(20)

where  $\mathbf{F}^{j}(k)$  is the Jacobian matrix of model j and

$$\mathbf{F}^{j}(k) = \left. \frac{\partial f(\mathbf{X}, \Delta t, j)}{\partial \mathbf{X}} \right|_{\mathbf{X} = \widehat{\mathbf{X}}^{0j}(k)}.$$
 (21)

From (4), (6), and (9), we have

$$\mathbf{F}^{1}(k) = \mathbf{I}_{2} \otimes \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$
(22)

$$\mathbf{F}^{2}(k) = \mathbf{I}_{2} \otimes \begin{bmatrix} 1 & \Delta t & \frac{(\Delta t)^{2}}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$
(23)

and  $\mathbf{F}^{3}(k)$  can be found as (24) shown at the bottom of the next page.

From the prediction equations, the state vector estimate and the state vector estimation error covariance matrix for model j can be updated by

$$\widehat{\mathbf{X}}^{j}(k+1) = \overline{\mathbf{X}}^{j}(k+1) + \mathbf{K}^{j}(k+1)$$

$$\times \left( \mathbf{Z}(k) - h\left(\overline{\mathbf{X}}^{j}(k+1)\right), \Delta t, j \right) \quad (25)$$

$$\widehat{\mathbf{P}}^{j}(k+1) = \left( \mathbf{I} - \mathbf{K}^{j}(k+1)\mathbf{H}^{j}(k+1) \right) \overline{\mathbf{P}}^{j}(k+1) \quad (26)$$

where  $\mathbf{K}^{j}(k+1)$  is called the Kalman gain with

$$\mathbf{K}^{j}(k+1)$$

$$= \overline{\mathbf{P}}^{j}(k+1)\mathbf{H}^{jT}(k+1)$$

$$\times \left[\mathbf{H}^{j}(k+1)\overline{\mathbf{P}}^{j}(k+1)\mathbf{H}^{jT}(k+1) + \mathbf{R}(k+1)\right]^{-1} (27)$$

$$\mathbf{H}^{j}(k+1)$$

$$= \left. \frac{\partial h(\mathbf{X}, \Delta t, j)}{\partial \mathbf{X}} \right|_{\mathbf{X} = \overline{\mathbf{X}}^{j}(k+1)}.$$
(28)

Note that

·ice

$$\frac{\partial \theta_n(k)}{\partial x(k)} = \frac{-[y_n - y(k)]}{[x_n - x(n)]^2 + [y_n - y(k)]^2}$$
(29)

$$\frac{\partial \theta_n(k)}{\partial y(k)} = \frac{[x_n - x(k)]}{[x_n - x(n)]^2 + [y_n - y(k)]^2}$$
(30)  
$$\frac{\partial \theta_n(k)}{\partial \dot{x}(k)} = \frac{\partial \theta_n(k)}{\partial \dot{y}(k)} = \frac{\partial \theta_n(k)}{\partial \ddot{x}(k)} = \frac{\partial \theta_n(k)}{\partial \ddot{y}(k)} = \frac{\partial \theta_n(k)}{\partial \omega(k)} = 0$$
(31)

then, (28) actually becomes (32) shown at the bottom of the page.

Stage 3: Mode Probability Update. As the estimates  $\widehat{\mathbf{X}}^{j}(k+1)$  and  $\widehat{\mathbf{P}}^{j}(k+1)$  are obtained, the likelihood functions to be used for mode probability update are computed based on the Gaussian approximation of the measurement prediction error as follows:

$$\widehat{L}^{j}(k+1) = \mathcal{N}\left(\mathbf{e}^{j}(k+1); \mathbf{0}, \mathbf{S}^{j}(k+1)\right)$$
(33)

where 
$$\mathcal{N}(\mathbf{p}; \mathbf{0}, \mathbf{G})$$
 denotes a Gaussian function of vector  $\mathbf{p}$  with mean  $\mathbf{0}$  and covariance matrix  $\mathbf{G}$ , and

$$\mathbf{e}^{j}(k+1) = \mathbf{Z}(k+1) - h\left(\overline{\mathbf{X}}^{j}(k+1), \Delta t, j\right)$$
(34)  
$$\mathbf{S}^{j}(k+1) = E\left\{\mathbf{e}^{j}(k+1)\mathbf{e}^{j}(k+1)^{T}\right\}$$
$$= \mathbf{H}^{j}(k+1)\overline{\mathbf{P}}^{j}(k+1)\mathbf{H}^{jT}(k+1) + \mathbf{R}(k+1).$$
(35)

The update mode probability for model j is

$$\mu_j(k+1) = \frac{1}{c} L^j(k+1)c_j$$
(36)

where c is a normalizing factor with

$$c = \sum_{j=1}^{3} L^{j}(k+1)c_{j}.$$
(37)

*Stage 4: Mixture.* The resultant state vector estimate and the estimation error covariance matrix are finally obtained as

$$\widehat{\mathbf{X}}(k+1) = \sum_{j=1}^{3} \widehat{\mathbf{X}}^{j}(k+1)\mu_{j}(k+1)$$
(38)

$$\widehat{\mathbf{P}}(k+1) = \sum_{j=1} \mu_j(k+1)$$

$$\times \left\{ \widehat{\mathbf{P}}^j(k+1) + \left[ \widehat{\mathbf{X}}^j(k+1) - \widehat{\mathbf{X}}(k+1) \right] [\cdot]^T \right\}.$$
(39)

#### B. IMMPF Algorithm

The Kalman filtering (KF) is derived based on the assumption of Gaussian distribution for involved random variables presenting in linear state equations. Exempting from the Gaussian assumption and linearity, the PF also relies on the Bayesian

$$\mathbf{F}^{3}(k) = \begin{bmatrix} 1 & \Delta t & 0 & \frac{-(\Delta t)^{2}\hat{\omega}(k)}{2} & \frac{-(\Delta t)^{2}\hat{y}(k)}{2} \\ 0 & \frac{(\Delta t)^{2}\hat{\omega}(k)}{2} & 1 & \Delta t & \frac{(\Delta t)^{2}\hat{x}(k)}{2} \\ 0 & \frac{2-(\Delta t)^{2}\hat{\omega}(k)}{2} & 0 & -\Delta t \times \hat{\omega}(k) & -\Delta t \left(\Delta t \times \hat{x}(k)\hat{\omega}(k) + \hat{y}(k)\right) \\ 0 & \Delta t \times \hat{\omega}(k) & 0 & \frac{2-(\Delta t)^{2}\hat{\omega}(k)}{2} & -\Delta t \left(\Delta t \times \hat{y}(k)\hat{\omega}(k) + \hat{x}(k)\right) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$
(24)

$$\mathbf{H}^{j}(k+1) = \begin{bmatrix} \frac{-[y_{1}-\bar{y}^{j}(k+1)]}{[x_{1}-\bar{x}^{j}(k+1)]^{2}+[y_{1}-\bar{y}^{j}(k+1)]^{2}} & \frac{[x_{1}-\bar{x}^{j}(k+1)]}{[x_{1}-\bar{x}^{j}(k+1)]^{2}+[y_{1}-\bar{y}^{j}(k+1)]^{2}} & 0 & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \frac{-[y_{4}-\bar{y}^{j}(k+1)]}{[x_{4}-\bar{x}^{j}(k+1)]^{2}+[y_{4}-\bar{y}^{j}(k+1)]^{2}} & \frac{[x_{4}-\bar{x}^{j}(k+1)]}{[x_{4}-\bar{x}^{j}(k+1)]^{2}+[y_{4}-\bar{y}^{j}(k+1)]^{2}} & 0 & 0 & 0 & 0 \end{bmatrix}$$
(32)

filtering concept for recursive state estimation. With a different approach, PF-based algorithms are the sequential Monte Carlo method that uses a finite and large set of particles to approximate a required probability density, which are useful for non-Gaussian and nonlinear systems.

As shown in Fig. 3, we divide the IMMPF algorithm into the following stages.

*Stage 1: Interaction.* The interaction stage is the same as those employing (15)–(18) in the IMMEKF algorithm.

*Stage 2: State Update with Importance Sampling.* The basic concept of a PF algorithm is to make the important sampling function recursive, i.e., sequential importance sampling (SIS). On the sequential reception of each measurement, support points and their associated importance weights are recursively propagated through the SIS algorithm.

Let us draw  $N_p$  particles (samples)  $\overline{\mathbf{X}}_n^j(k)$ , where  $n = 1, 2, \dots, N_p$ , from the initial statistics  $\widehat{\mathbf{X}}^{0j}(k)$  and  $\widehat{\mathbf{P}}^{0j}(k)$ ) based on the Gaussian distribution for model j by the following scheme:

$$\overline{\mathbf{X}}_{n}^{j}(k) \sim \mathcal{N}\left(\widehat{\mathbf{X}}^{0j}(k), \widehat{\mathbf{P}}^{0j}(k)\right).$$
(40)

As we have particle  $\overline{\mathbf{X}}_n^j(k)$ , the predicted state particle  $\overline{\mathbf{X}}_n^j(k+1)$  and the corresponding weight  $\overline{\mathbf{W}}_n^j(k+1)$  can be calculated by

$$\overline{\mathbf{X}}_{n}^{j}(k+1) = f\left(\overline{\mathbf{X}}_{n}^{j}(k), \Delta t, j\right)$$

$$\mathbf{W}_{n}^{j}(k+1) = \mathcal{N}\left(\left(\mathbf{Z}(k+1) - h\left(\overline{\mathbf{X}}_{n}^{j}(k+1)\right), \Delta t, j\right), \mathbf{R}(k+1)\right).$$

$$\mathbf{K}(k+1)\right).$$

$$(41)$$

Note that the weights obtained from (42) should be normalized as

$$\overline{\mathbf{W}}_{n}^{j}(k+1) = \frac{\mathbf{W}_{n}^{j}(k+1)}{\sum_{n=1}^{N_{p}} \mathbf{W}_{n}^{j}(k+1)}.$$
(43)

Stage 3: Mode Probability Update. Once the particle  $\overline{\mathbf{X}}_n^j(k+1)$  and its corresponding normalized weight  $\overline{\mathbf{W}}_n^j(k+1)$  are obtained, we can evaluate the mean of the predicted measurement vector for model  $j, \overline{\mathbf{Z}}^j(k+1)$ , by

$$\overline{\mathbf{Z}}^{j}(k+1) = \sum_{n=1}^{N_{p}} h\left(\overline{\mathbf{X}}_{n}^{j}(k+1), \Delta t, j\right).$$
(44)

The measurement prediction error vector (residual measurement vector) for model j,  $\mathbf{e}_n^j(k+1)$ , and the residual covariance matrix  $\mathbf{S}^j(k+1)$  are

$$\mathbf{e}_{n}^{j}(k+1) = \mathbf{Z}(k+1) - h\left(\overline{\mathbf{X}}_{n}^{j}(k+1), \Delta t, j\right)$$
(45)

$$\mathbf{S}^{j}(k+1) = \sum_{n=1}^{N_{p}} \left\{ \left[ h\left( \overline{\mathbf{X}}_{n}^{j}(k+1), \Delta t, j \right) - \overline{\mathbf{Z}}^{j}(k+1) \right] [\cdot]^{T} \right\}$$
(46)



Fig. 4. Concept of particle resampling.

respectively. The likelihood function for mode probability update is

$$L_{n}^{j}(k+1) = \mathcal{N}\left(\mathbf{e}_{n}^{j}(k+1); \mathbf{0}, \mathbf{S}^{j}(k+1)\right).$$
(47)

Stage 4: Resampling. It is shown in the literature, only with SIS, that only very few particles have nonzero importance weights after some iteration. To alleviate the adverse degeneracy effects of the SIS algorithm, some resampling scheme should be considered to eliminate particles with low importance weights and increase particles with high importance weights, with unchanged number of particles. Fig. 4 is depicted to illustrate the concept of resampling. Particles are represented by circles, and their weights are reflected by the corresponding diameters. The bottom row of circles are the particles with equal weights before resampling. As their weights at time index k are calculated, the particles are going to be propagated to time index k + 1. After resampling, the large particles (with high importance weights) are replicated, and the small particles are removed. Then, the number of new particles at time k+1remains the same as that at time k, and the new particles are assigned equal weights. From the principle of resampling, we propose a new scheme for implementing particle propagation in the PF algorithm.

Proposed Resampling Scheme:

1) Given the normalized weight  $\overline{\mathbf{W}}_{n}^{j}(k+1)$  of particle  $\overline{\mathbf{X}}_{n}^{j}(k+1)$  for  $n = 1, 2, \dots, N_{p}$  as obtained from (41) and (43), employ the sorting of  $\overline{\mathbf{W}}_{n}^{j}(k+1)$  in descending order as

$$\overline{\mathbf{W}}_{(1)}^{j}(k+1) \ge \overline{\mathbf{W}}_{(2)}^{j}(k+1) \ge \dots \ge \overline{\mathbf{W}}_{(N_{p})}^{j}(k+1) \quad (48)$$

and denote their corresponding particles by  $\overline{\mathbf{X}}_{(1)}^{j}(k+1)$ ,  $\overline{\mathbf{X}}_{(1)}^{j}(k+1)$ , respectively.

$$\mathbf{A}_{(2)}(k+1), \cdots, \mathbf{A}_{(N_p)}(k+1), \text{ respectively.}$$
2) Pick  $N_r$  particles

$$\left\{\overline{\mathbf{X}}_{(1)}^{j}(k+1), \overline{\mathbf{X}}_{(2)}^{j}(k+1), \cdots, \overline{\mathbf{X}}_{(N_{r})}^{j}(k+1)\right\}$$

with 
$$\mathcal{W}^{j} = \sum_{n=1}^{\gamma} \overline{\mathbf{W}}_{(n)}^{j}(k+1)$$
 such that  
 $N_{r} = \arg \max_{\gamma} \left\{ \mathcal{W}^{j} : \mathcal{W}^{j} \le \max \left\{ R^{j}, \overline{\mathbf{W}}_{(1)}^{j}(k+1) \right\} \right\}$ 
(49)

where  $R^j$  is a predetermined constant for model j that controls the number of particles propagated for resampling and  $R^j \leq 1$ . For example, if we let  $R^j = 0.9$ , only the larger particles whose total weight is up to 90% will remain after resampling, and the other particles are removed.

3) Let  $\langle x \rangle$  denote rounding x to the nearest integer and  $\widetilde{\mathbf{X}}_n^j(k+1)$  denote the new particle after resampling. Particles replication is implemented by the following recursions:

Initialization: i = 0. Loop: i = i + 1

$$N_{i} = \left\langle \frac{\overline{\mathbf{W}}_{(i)}^{j}(k+1)}{\mathcal{W}^{j}} \times N_{p} \right\rangle$$
$$\widetilde{\mathbf{X}}_{n}^{j}(k+1) = \overline{\mathbf{X}}_{(i)}^{j}(k+1) \text{ for}$$
$$n = N_{p} - N_{i} + 1, N_{p} - N_{i+1} + 2, \cdots, N_{p}$$
$$\mathcal{W}^{j} := \mathcal{W}^{j} - \overline{\mathbf{W}}_{(i)}^{j}(k+1)$$
$$N_{p} := N_{p} - N_{i}.$$

Go back to loop until  $i = N_r - 1$ .

 $\widetilde{\mathbf{X}}_{n}^{j}(k+1) = \overline{\mathbf{X}}_{(N_{r})}^{j}(k+1)$  for  $n = 1, 2, \dots, N_{r}$ . The weights for the new particles are set to be equal. That is

$$\widetilde{W}_{n}^{j}(k+1) = \frac{1}{N_{p}}, \qquad n = 1, 2, \cdots, N_{p}.$$
 (50)

Notice that, in (49), if some particle has a significant weight larger than  $R^j$ , we set  $\mathcal{W}^j = \overline{\mathbf{W}}_{(1)}^j(k+1)$ , and this leads to the assignment  $\widetilde{\mathbf{X}}_n^j(k+1) = \overline{\mathbf{X}}_{(1)}^j(k+1)$  for  $n = 1, 2, \dots, N_p$ .

Stage 5: Mixture. By averaging over the  $N_p$  particles, the state vector and the estimation error covariance matrix for model j can be found by

$$\widehat{\mathbf{X}}^{j}(k+1) = \frac{1}{N_{p}} \sum_{n=1}^{N_{p}} \widehat{\mathbf{X}}_{n}^{j}(k+1)$$
(51)

$$\widehat{\mathbf{P}}^{j}(k+1) = \frac{1}{N_{p}} \sum_{n=1}^{N_{p}} \left\{ \left[ \widetilde{\mathbf{X}}_{n}^{j}(k+1) - \widehat{\mathbf{X}}^{j}(k+1) \right] \left[ \cdot \right]^{T} \right\}$$
(52)

respectively. Taking into account the update mode probability for model j, the resultant state vector and the estimation error covariance matrix then use the same equations as (38) and (39).

#### C. Posterior CRLB

In time-invariant statistical models, a commonly used lower bound is the CRLB which is defined to be the inverse of the Fisher information matrix (FIM). In our maneuvering mobile tracking system, the estimated parameter vector is random since it corresponds to three differently and randomly driven dynamic models. Here, we consider the posterior CRLB provided as a powerful tool for assessing the performance of tracking filters.

The estimation error covariance for model j is bounded by the posterior CRLB [22] as

$$E\left[\left(\widehat{\mathbf{X}}^{j}(k) - \mathbf{X}(k)\right)\left(\widehat{\mathbf{X}}^{j}(k) - \mathbf{X}(k)\right)^{T}\right] \ge \mathbf{J}_{j}^{-1}(k) \quad (53)$$

where  $\mathbf{J}_{j}(k)$  is the corresponding FIM for model j and  $\mathbf{J}_{j}^{-1}(k)$  is the posterior CRLB. Let  $\mathbf{Z}^{k}$  denote the measurement vectors collected up to time k, i.e.,  $\mathbf{Z}^{k} = {\mathbf{Z}(1), \mathbf{Z}(2), \dots, \mathbf{Z}(k)}$ , and  $p(\mathbf{Z}^{k}, \mathbf{X}^{j}(k))$  is the joint probability density function for model j. From [22], the FIM is

$$\mathbf{J}_{j}(k) = E\left[\left(\frac{\partial \log p\left(\mathbf{Z}^{k}, \mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k)}\right) \left(\frac{\partial \log p\left(\mathbf{Z}^{k}, \mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k)}\right)^{T}\right].$$
(54)

The FIM  $J_j(k+1)$  is a general nonlinear filtering framework and can be given by the following recursion [22], [23]:

$$\mathbf{J}_{j}(k+1) = \mathbf{D}_{j}^{22}(k) - \mathbf{D}_{j}^{21}(k) \left(\mathbf{J}_{j}(k) + \mathbf{D}_{j}^{11}(k)\right)^{-1} \mathbf{D}_{j}^{12}(k)$$
(55)

where, under the case of Gaussian noise assumption, the matrices in (55) are found as (56)–(59) shown at the bottom of the page. The CRLB to be assessed in our simulation is based on a fixed trajectory. That is, taking expectation in (56)–(59) can be ignored. The initial value for FIM can be simply set as  $J_j(0) = \mathbf{P}^{-1}(0)$ . For the purpose of analysis, we suppose that the maneuvering sequence defining model j is known in the simulation.

$$\begin{aligned} \mathbf{D}_{j}^{11}(k) &= -E\left[\left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k)}\right) \\ &\times \left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k)}\right)^{T}\right] \\ &= E\left[\left(\mathbf{F}^{j}(k)\right)^{T} \mathbf{Q}_{j}^{-1}(k)\mathbf{F}^{j}(k)\right] \quad (56) \\ \mathbf{D}_{j}^{12}(k) &= -E\left[\left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k)}\right) \\ &\times \left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k+1)}\right)^{T}\right] \\ &= E\left[\left(\mathbf{F}^{j}(k)\right)^{T}\right] \mathbf{Q}_{j}^{-1}(k) \quad (57) \\ \mathbf{D}_{j}^{21}(k) &= -E\left[\left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k+1)}\right) \\ &\times \left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k)}\right)^{T}\right] \\ &= \left(\mathbf{D}_{i}^{12}(k)\right)^{T} \quad (58) \end{aligned}$$



Fig. 5. Concept of dynamic BS sensor selection.

$$\mathbf{D}_{j}^{22}(k) = -E\left[\left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k+1)}\right) \times \left(\frac{\partial \log p\left(\mathbf{X}^{j}(k+1)|\mathbf{X}^{j}(k)\right)}{\partial \mathbf{X}^{j}(k+1)}\right)^{T}\right] \\ = -E\left[\left(\frac{\partial \log p\left(\mathbf{Z}(k+1)|\mathbf{X}^{j}(k+1)\right)}{\partial \mathbf{X}^{j}(k+1)}\right) \times \left(\frac{\log p\left(\partial \mathbf{Z}(k+1)|\mathbf{X}^{j}(k+1)\right)}{\partial \mathbf{X}^{j}(k+1)}\right)^{T}\right] \\ = \mathbf{Q}_{j}^{-1}(k) \\ + E\left[\mathbf{H}^{jT}(k+1)\mathbf{R}^{-1}(k+1)\mathbf{H}^{j}(k+1)\right].$$
(59)

## D. Dynamic BS Sensor Selection

In this MS tracking system, the fusion center selects the active BS sensors and collects their measurement data to employ the tracking algorithm. Before the MS is going to move outside of the tracking region of the BSs, the fusion center needs to activate the next BSs by filling proper working data (such as those variables used in the recursion of the IMMPF algorithm) to maintain the long-haul MS tracking.

As depicted in Fig. 5, suppose that the estimated state  $\hat{\mathbf{X}}(k)$  is known to the fusion center at current instant k and we need to select BSs at the next time k + 1. Then, the predicted state  $\overline{\mathbf{X}}_n(k+1)$  for the IMMPF algorithm can be calculated from (41) by

$$\overline{\mathbf{X}}_n(k+1) = \frac{1}{N_p} \sum_{j=1}^3 \sum_{n=1}^{N_p} \mu_j(k) f\left(\overline{\mathbf{X}}_n^j(k), \Delta t, j\right)$$
(60)

where  $\overline{\mathbf{X}}_{n}^{j}(k)$  is obtained from (40).

As the predicted MS location is obtained, the search region based on a given radius with the circle center determined from  $\overline{\mathbf{X}}_n(k+1)$  is used to find the proper candidates from the most nearby BSs. However, if the previously active BSs are still within this search region, for example, BS sensors A and B in Fig. 5, they will remain in order to reduce the reset number of working data due to changing BSs. For the new selected BSs, the measurement equation  $h(\cdot)$  needs to be modified based on the new BS locations. Here, we suppose to use the same noise variance matrix  $\mathbf{R}(k+1)$  for the new BSs.

#### IV. NUMERICAL RESULTS

In our simulation, the IMMEKF and IMMPF algorithms along with two dynamic models and three dynamic models are compared with bearing-only measurements observed from the four BSs as depicted in Fig. 1. During the MS tracking period, the MS trajectory includes three motion situations, i.e., CV, CA, and CT.

First, we introduce the tracking trajectory used in our simulation. The distance between two successive BSs is 600 m. The sampling duration  $\Delta t$  is 0.2 s, and the overall tracking interval is 46 s. The initial position of the MS is at (0, 60), the initial velocity is (0, 11 m/s), and the initial acceleration is (0, 0). During the first 50 samples, the MS keeps the same constant velocity as the initial value. During the 51th to the 100th samples, a constant acceleration (0,  $-1 \text{ m/s}^2$ ) is boosted. During the 101th to the 120th samples, a constant turn with a turning rate of -0.25 is presented, and the acceleration ceases. During the 131th to the 100th samples, another constant acceleration (1 m/s<sup>2</sup>, 0) is boosted. From the 181th to the 230th samples, the acceleration ceases, and constant velocity remains.

Next, we give the parameters used in the tracking algorithms. The process noise parameters are  $\sigma_1^2 = 0.05$ ,  $\sigma_2^2 = 0.17$ ,  $\sigma_3^2 = 0.05$ , and  $\sigma_w^2 = 0.0625$ . The standard deviation of the measurement noise is  $\sigma_{\theta} = 0.5^{\circ}$ . The number of particles is  $N_p = 300$ , and the resampling control parameter  $R^j = 0.95$ . The coefficients of the Markovian transition matrix for the two-model IMM are  $p_{ii} = 0.98$  and  $p_{ij} = 0.02$  for  $i \neq j$  while those for the three-model IMM are  $p_{ii} = 0.96$  and  $p_{ij} = 0.02$  for  $i \neq j$ . The initial mode probabilities for the two-model IMM are  $\mu_1(0) = \mu_2(0) = 0.5$  while those for the three-model IMM are  $\mu_1(0) = 0.8$  and  $\mu_2(0) = \mu_3(0) = 0.1$ .

To evaluate the tracking performance for different algorithms, we calculate the root mean square (rms) metrics for rms position tracking error and rms velocity tracking error with 100 Monte Carlo simulations. The rms metric is defined as

$$\operatorname{rms}(k) = \sqrt{\frac{1}{L} \sum_{l=1}^{L} \left(\hat{\alpha}^{l}(k) - \alpha(k)\right)^{2} + \left(\hat{\beta}^{l}(k) - \beta(k)\right)^{2}}$$
(61)

where  $\alpha(k)$  and  $\beta(k)$  denote the true MS position or velocity at time k for the x- and y-axes, respectively, while  $\hat{\alpha}(k)$  and  $\hat{\beta}(k)$  denote the corresponding estimated results for the xand y-axes, respectively, and L = 100 for 100 Monte Carlo simulations. The posterior CRLB for the three-model IMM 12

10

Position RMSE (m)

0

0

Fig. 6. Comparison of rms position tracking error.

50

algorithm at time k can be referred to as  $\mathbf{J}^{-1}(k)$  obtained from (55). Let  $J_{i,j}^{-1}(k)$  denote the (i, j)th element in  $\mathbf{J}^{-1}(k)$ , then the rms metric of the posterior CRLB for position tracking error at time k is

100

Sample index

$$\operatorname{CRLB}_{p}(k) = \sqrt{J_{1,1}^{-1}(k) + J_{4,4}^{-1}(k)}$$
(62)

150

IMMPF(3 models)

IMMPF(2 models)

IMMEKF(2 models)

CRLB

IMMEKF(3 models)

200

and the rms metric of the posterior CRLB for velocity tracking error at time k is

$$\operatorname{CRLB}_{v}(k) = \sqrt{J_{2,2}^{-1}(k) + J_{5,5}^{-1}(k)}.$$
(63)

Fig. 6 shows the comparison of rms position tracking errors using the IMMEKF and IMMPF algorithms with two- and three-model IMMs. The IMMPF algorithm significantly outperforms the IMMEKF algorithm for the three-model IMM. For the two-model IMM, the position tracking accuracy is heavily deteriorated for both algorithms during a turning movement because the CT model is not considered in the two-model IMM. The posterior CRLB plotted in this figure is analyzed by taking into account the three dynamic models with known maneuvering sequences. The three-model IMMPF algorithm has the best performance in this comparison and much approaches the posterior CRLB. Fig. 7 shows the comparison of rms velocity tracking error for the same case as in Fig. 6. During the CV motion period, we can notice that the two-model IMM algorithms have slightly better rms error than the three-model IMM algorithms because the CA model somewhat introduces more • modeling error noise. However, the advantage of the threemodel algorithms in alleviating the tracking error during CT motion is more attractive.

Figs. 8 and 9 show the mode transition probabilities for the three-model IMMPF algorithm and IMMEKF algorithm, respectively. In these two figures, the correct model is identified as a higher mode probability. As a mode can be identified correctly as a particularly high probability such as the CT model in this simulation, the tracking error can be reduced significantly. Hence, we can see that the rms position and velocity tracking errors during CT motion are even smaller than those during



Fig. 7. Comparison of rms velocity tracking error.



Fig. 8. Mode transition probabilities for three models for the IMMPF algorithm.

CA motion if ignoring the tracking error due to the inevitable transient effect of mode interchange.

The bearing-only tracking method requires at least two BSs to determine the intersection of two lines of antenna arrays for the MS location. The more BSs the data fusion center collects, the better the tracking performance is. As shown in Fig. 10, the position rms errors of the three-model IMMPF algorithm are compared by using different numbers of BSs. Although the tracking performance can be improved with more BSs, the filtering complexity also increases at the fusion center. Here, we suppose to use the maximum of four BSs under the concern of complexity for dynamic MS trajectory tracking.

In Fig. 11, we apply the proposed dynamic BS sensor selection scheme for long-haul MS position tracking with the three-model IMMPF algorithm. To ease our simulation, the BS sensors marked as "+" are uniformly distributed with a distance of 600 m between two nearby BSs. The only different parameter here is that we assume that the measurement noise



Fig. 9. Mode transition probabilities for three models for the IMMEKF algorithm.



Fig. 10. Comparison of rms position tracking error versus different numbers of BSs.

standard deviation is 5° for a possibly larger measurement noise in the field. The initial location of the MS is at (-400, -200), and the MS experiences CV, CA, and CT motion during the tracking period. The positions for the fusion center to determine a new group of BS sensors are marked as "\*." The dashed circle indicates the region for the fusion center to determine the best suitable BSs. In the first and second circles, two BSs located at (-900, -300) and (-900, 300) are involved in the overlapped area. Although the BS (-1500, 900) can be used for the second \* point, we choose to maintain the two old BSs at first so as not to reset too many working data in order to keep the tracking algorithm as smooth as possible. However, as the MS moves with an high acceleration between the 5th and 6th  $\star$  points, all new groups of BS sensors are required without using any old BSs. The working data for the all new BSs in the tracking algorithm should be carefully managed.



Fig. 11. Maneuvering MS tracking trajectory and the selected BS sensors in a BS WSN.

#### V. CONCLUSION

In this paper, the nonlinear EKF and PF are integrated to the IMM algorithm for maneuvering MS location estimation with bearing-only measurements. In this system, the motion of the MS is a complicated and nonlinear kinematic model. We consider three general motion cases, CV, CA, and CT, in the EKF and PF dynamic models. The IMM algorithm yields the output by mixing the estimated results obtained from the different filtering models. The performance of using the threemodel IMM is better than that using only CV and CA when the MS makes a turn. The three-model IMM algorithm along with either the EKF or the PF also does not show a significant rms tracking performance loss due to the addition of the CT model when the MS moves without a turn. It is worth to notice the difference between the EKF and the PF. Although the EKF has tailored the linear KF to be suited to the nonlinear filtering problem, it computes Jacobian matrices for the first-order gradient approximation. Its performance cannot be guaranteed compared to some typical nonlinear algorithms such as the PF. In our results, the IMMPF is superior to the IMMEKF, but at the cost of 300 particles involved in the PF algorithm. However, the IMMPF does not need to compute matrix inversion for the Kalman gain, which is required in the IMMEKF algorithm.

In this paper, an efficient resampling method is proposed for particle propagation over different time instants in the IMMPF algorithm. The long-haul MS position tracking case is also considered as a localization problem in a WSN, where the BSs providing data for the fusion center are the active sensors and may need to be changed during the tracking. In our proposal, the predicted location from the IMM tracking algorithm is used to find the next active BS sensors when the time of changing BSs is up. In numerical analysis, the posterior CRLB is also exploited to evaluate the performance of the tracking algorithms in addition to the Monte Carlo simulation. The numerical results show that the three-model IMMPF algorithm outperforms the IMMEKF algorithm with a close performance to the posterior CRLB.

#### REFERENCES

- L. Mihaylova, D. Angelova, D. R. Bull, and N. Canagarajah, "Localization of mobile nodes in wireless networks with correlated in time measurement noise," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 44–53, Jan. 2011.
- [2] E. Kim and K. Kim, "Distance estimation with weighted least squares for mobile beacon-based localization in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 559–562, Jun. 2010.
- [3] B.-S. Chen, C.-Y. Yang, F.-K. Liao, and J.-F. Liao, "Mobile location estimator in a rough wireless environment using extended Kalman-based IMM and data fusion," *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1157–1169, Mar. 2009.
- [4] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1157–1169, Apr. 2004.
- [5] M. Malajner, P. Planinsic, and D. Gleich, "Angle of arrival estimation using RSSI and omnidirectional rotatable antennas," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1950–1957, Jun. 2012.
- [6] J. Yan, C. C. J. M. Tiberius, P. J. G. Teunissen, G. Bellusci, and G. J. M. Janssen, "A framework for low complexity least-squares localization with high accuracy," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4836–4847, Sep. 2010.
- [7] L. Taponecco, A. DAmico, and U. Mengali, "Joint TOA and AOA estimation for UWB localization applications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2207–2217, Jul. 2011.
- [8] L. Cong and W. Zhuang, "Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems," *IEEE Trans. Wireless Commun.*, vol. 1, no. 3, pp. 439–447, Jul. 2002.
- [9] X. Chen, D. Schonfeld, and A. A. Khokhar, "Localization and trajectory estimation of mobile objects using minimum samples," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4439–4446, Oct. 2009.
- [10] A. Catovic and Z. Sahinoglu, "The Cramer-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes," *IEEE Commun. Lett.*, vol. 8, no. 10, pp. 626–628, Oct. 2004.
- [11] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. Int. Symp. Aerosp./Defense Sens., Simul. Controls*, Apr. 1997, pp. 1–12.
- [12] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, Jul. 2000.
- [13] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.
- [14] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*. Hoboken, NJ, USA: Wiley, 2001.
- [15] C. K. Seow and S. Y. Tan, "Non-line-of-sight localization in multipath environments," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 647–660, May 2008.
- [16] J. M. Huerta, J. Vidal, A. Giremus, and J.-Y. Tourneret, "Joint particle filter and UKF position tracking in severe non-line-of-sight situations," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 874–888, Oct. 2009.
- [17] Y. Boers and J. Driessen, "Interacting multiple model particle filter," *Proc. Inst. Elect. Eng.—Radar, Sonar Navig.*, vol. 150, no. 5, pp. 344–349, Oct. 2003.

- [18] R. Guo, Z. Qin, X. Li, and J. Chen, "Interacting multiple model particletype filtering approaches to ground target tracking," *J. Comput.*, vol. 3, no. 7, pp. 23–30, Jul. 2008.
- [19] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA, USA: Artech House, 1999.
- [20] G. A. Watson and W. D. Blair, "IMM algorithm for tracking targets that maneuver through coordinated turns," in *Proc. SPIE, Signal Data Process. Small Target*, 1992, vol. 1698, pp. 236–247.
- [21] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Orlando, FL, USA: Academic, 1988, ch. 2.
- [22] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, May 1998.
- [23] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearings-only tracking of manoeuvring targets using particle filters," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 15, pp. 2351–2365, Jan. 2004.
- [24] J. V. Candy, Bayesian Signal Processing: Classical, Modern Particle Filtering Methods. Hoboken, NJ, USA. Wiley. 2009.
- [25] S. Arulampalam and B. Ristic, "Comparison of the particle filter with range-parameterized and modified polar EKFs for angle-only tracking," in *Proc. SPIE, Signal Data Process. Small Target*, 2000, vol. 4048, pp. 288–299.



**Dah-Chung Chang** (S'98) received the B.S. degree in electronic engineering from Fu Jen Catholic University, Taipei, Taiwan, in 1991 and M.S. and Ph.D. degrees in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993 and 1998, respectively.

In 1998, he joined Computer and Communications Research Laboratories at Industrial Technology Research Institute, Hsinchu. Since 2003, he has been a faculty member in the Department of Communication Engineering, National Central University,

Taoyuan, Taiwan. His recent research interests include multirate and adaptive digital signal processing, orthogonal frequency division multiplexing, and digital transceiver algorithm and architecture design.



Meng-Wei Fang was born in Taoyuan, Taiwan, on May 30, 1988. He received the B.S. degree in communication engineering from the Feng Chia University, Taichung, Taiwan, in 2010 and the M.S. degree in communication engineering from the National Central University, Taoyuan, in 2012.

His research interests lie in the area of target tracking for communications.